

TUFLOW FV User Manual

Build 2019.01

www.tuflow.com
[TUFLOW FV Wiki](#)
[TUFLOW Tutorial Model](#)
support@tuflow.com

[How to Use This Manual](#)
[Chapters](#)
[Table of Contents](#)
[List of Figures](#)
[List of Tables](#)
[Appendices](#)

[Simulation Commands](#)

[Material Commands](#)

[Boundary Commands](#)

[Structure Commands](#)

[Output Commands](#)

[AD Module Commands](#)

Manual Overview

This document is the User Manual for the TUFLOW FV hydrodynamic computational engine 2019-01 release. The table below provides a quick reference to TUFLOW FV's control file and commands. Each chapter heavily references Appendix A, Appendix B and Appendix C which contain the command syntax for TUFLOW FV.

Command Category	Description	Chapter
Introduction	Overview of flexible mesh modelling, the TUFLOW FV solution scheme and benchmark studies used to verify TUFLOW FV's performance.	Chapter 1
Running TUFLOW FV	TUFLOW FV software structure, its installation process, licensing and running simulations.	Chapter 2
The Modelling Process	A high-level overview of the modelling process. If you are new to modelling it is recommended to read through this chapter first prior to proceeding with other sections of the manual.	Chapter 3
Model Setup and Control Files	Recommended layout and syntax used in the TUFLOW FV control file.	Chapter 4
GIS Setup	GIS format, spatial reference assignment, empty file creation.	Chapter 5
Simulation Configuration	Solution include statements, spatial order selection, units selection, coordinate type, bed roughness model.	Chapter 6
Time and Timestep	Time format and reference time, model start and end times, CFL limits, timestep limits.	Chapter 6
Model Parameters	Wetting / drying, turbulence related commands, stability limits, reference values.	Chapter 7
Geometry	Mesh file, topographic updates, external nodestring locations.	Chapter 8
Materials	Material block properties (roughness, mixing parameters, etc.), spatial definition of materials (GIS Layers).	Chapter 8
3D Geometry	<u>3D</u> geometry definitions, layer types and discretisation.	Chapter 9
Initial Conditions	Initial model state (initial parameters, restart files).	Chapter 10
Boundary Conditions	Global (winds, waves, rainfall, etc.), nodestring (external boundaries, water levels, flows, etc.), cell (source) and node (point source).	Chapter 10
Hydraulic Structures	Weirs, culverts, bridges, porous and wall structures, operational structures and controls.	Chapter 11
Model Results and Output	Output directory, map output formats and data types check files, time series and profile outputs, logging and diagnostics.	Chapter 12

How to Use This Manual

This manual is designed for both hardcopy and digital usage. It is provided in both its native Microsoft Word 2013 version and as a pdf. The pdf version is faster to navigate around and, for the first time, with all links preserved (previously the manual was only issued as a Word 2010 document as not all links were preserved in the pdf conversion).

Section, Table and Figure references are [styled like this](#) and linked. In Word, to go to the link hold down the control (Ctrl) key and click on the Section, Table or Figure number in the text to move to the relevant page. In pdf, the links should be directly accessible.

Similarly, and most importantly, script or control file commands are hyperlinked and are easily accessed through the lists at the end of the manual. To quickly go to the end of the manual press Ctrl End. There are also command hyperlinks within the text (normally blue and underlined). Command text can be copied and pasted into the text files to ensure correct spelling.

Web and email links are [styled like this](#). An increasing amount of content now resides on the TUFLOW Wiki with web links provided in the document to these pages. Other useful keys are Alt Left / Right arrow to link backwards / forwards to the last locations. Ctrl Home returns to the front page, which also contains useful links.

A secondary window can be opened in Word by selecting View, New Window or in Adobe Acrobat by selecting Window, New Window, allowing you to view different sections of the document in different windows. For example, the TUFLOW command lists could be viewed in one window and the section describing the functionality in another window.

Constructive suggestions are always welcome (please email support@tuflow.com).

If simulating models using prior TUFLOW FV builds, reference may need to be made to the release notes and documentation relevant for that particular build. These can be downloaded from the [All TUFLOW Downloads page](#) or requested from support@tuflow.com.

Chapters

1	Introduction	1-1
2	Running TUFLOW FV	2-1
3	The Modelling Process	3-1
4	Model Setup and Control Files	4-1
5	GIS Setup	5-1
6	Simulation Configuration	6-1
7	Model Parameters	7-1
8	Model Geometry	8-1
9	Modelling in 3D	9-1
10	Boundaries and Initial Conditions	10-1
11	Hydraulic Structures	11-1
12	Model Outputs	12-1

Appendix A .fvc File Commands	A-1
Appendix B Advection Dispersion fvc Commands	B-1
Appendix C Water Quality fvc Commands	C-1
Appendix D TUFLOW FV NetCDF Output Format	D-2
Appendix E Common NetCDF Input File Example Format	E-1

Table of Contents

1	Introduction	1-1
1.1.1	Introduction	1-2
1.2	TUFLOW FV	1-4
1.2.1	Flexible Mesh Modelling	1-5
1.3	Solution Scheme	1-6
1.3.1	Non-Linear Shallow Water Equations	1-6
1.3.2	Scalar Conservation Equations	1-7
1.3.3	Conserved vs. Scalar Variables	1-8
1.3.3.1	<i>Conserved Variables</i>	1-8
1.3.3.2	<i>Scalar Variables</i>	1-8
1.4	Schematisation	1-9
1.5	Multi-Core Processing	1-10
1.6	UK Benchmarking Study	1-11
2	Running TUFLOW FV	2-1
2.1	Software Structure	2-2
2.2	Installing and Running TUFLOW FV	2-5
2.2.1	TUFLOW FV Downloads and Installation	2-5
2.2.2	Linux Installation	2-5
2.2.3	USB Locks (Dongles) and Licencing	2-5
2.2.4	Dongle Types and Setup	2-5
2.3	Performing Simulations	2-7
2.3.1	Batch File Example and Run Options (Switches)	2-7
2.3.2	From a Text Editor	2-8
2.3.3	From the SMS Interface	2-8
2.3.4	Directly from GIS	2-8
2.3.5	Change Priority, Pause, Restart or Cancel a Simulation	2-8
2.4	Licence Free Simulations	2-10
2.4.1	Tutorial Models	2-10
2.5	Tips and Tricks	2-11
3	The Modelling Process	3-1
3.1	Introduction	3-2
3.2	Model Schematisation	3-3
3.2.1	Problem Definition	3-3
3.2.2	Model Limits (Space and Time)	3-4

3.2.3	Base Data Preparation	3-5
	<i>Bathymetry and Topography</i>	3-5
3.3	Mesh Construction	3-7
3.4	Assignment of Boundary Conditions	3-9
3.4.1	Open Boundaries	3-9
3.4.2	Water Surface Boundary	3-9
3.5	Initial Conditions	3-11
3.6	Simulation Times and Computational Timestep	3-12
3.7	Model Calibration and Validation	3-13
4	Model Setup and Control Files	4-1
4.1	Introduction	4-2
4.2	Folders and File Types	4-3
4.2.1	Suggested Folder Structure	4-3
4.2.2	File Types and Naming Conventions	4-5
4.3	Simulation Control File	4-8
4.3.1	Control File Rules and Notation	4-8
4.3.2	Absolute and Relative File Paths	4-10
4.3.3	Include Files	4-10
4.4	Control File Layout	4-12
5	GIS Setup	5-1
5.1	Introduction	5-2
5.2	“GIS” Commands	5-3
5.2.1	Naming Conventions	5-4
5.3	GIS Initialisation	5-6
5.3.1	Define the Project Projection	5-6
5.3.2	Setting the GIS Output Format	5-6
5.3.3	Creating ‘Empty’ GIS Template Files	5-7
5.3.4	TUFLOW Viewer QGIS Plugin	5-8
5.4	TUFLOW FV GIS Integration Tutorials	5-9
5.5	Layering of GIS Datasets	5-10
6	Simulation Configuration	6-1
6.1	Introduction	6-2
6.2	Solution Include Statements	6-3
6.3	Spatial Order and Gradient Limiters	6-5
6.4	Units – Metric or US Customary/English	6-6

6.5	Spherical or Cartesian	6-7
6.6	Bottom Drag Model	6-8
6.7	Wind Stress Model	6-9
6.8	Model Timestep	6-10
6.8.1	Adaptive Timestepping and CFL	6-10
6.8.2	Mode Splitting	6-11
6.9	Time Commands	6-12
7	Model Parameters	7-1
7.1	Introduction	7-2
7.2	Turbulence	7-3
7.2.1	Eddy Viscosity	7-3
7.2.1.1	<i>Horizontal</i>	7-3
7.2.1.2	<i>Vertical</i>	7-3
7.2.2	Scalar Diffusivity	7-4
7.2.2.1	<i>Horizontal</i>	7-4
7.2.2.2	<i>Vertical</i>	7-4
7.3	Reference Values	7-5
7.4	Wetting and Drying Thresholds	7-7
7.5	Stability Limits	7-8
8	Model Geometry	8-1
8.1	Mesh Generation	8-2
8.1.1	Choosing a Mesh Generator	8-3
8.1.2	Mesh File Format (.2dm)	8-3
8.2	External Nodestrings	8-7
8.3	Elevations	8-8
8.3.1	Direct Reading from .2dm Mesh	8-8
8.3.2	Updating Elevations from a Grid	8-8
8.3.3	Breakline Layers	8-9
8.3.4	Legacy Elevation Update Options	8-10
8.3.4.1	<i>Cell Elevation Polyline File</i>	8-10
8.3.4.2	<i>Cell Elevation Polygon File</i>	8-11
8.4	Materials	8-12
8.5	Material Block	8-14
8.5.1	Bed Resistance	8-14
8.5.2	Active/Inactive Areas	8-14
8.5.3	Horizontal Eddy Viscosity	8-15

8.5.4	Horizontal and Vertical Eddy Viscosity Limits	8-15
8.5.5	Horizontal Scalar Diffusivity	8-16
8.5.6	Horizontal and Vertical Scalar Diffusivity Limits	8-16
8.5.7	Bed Elevation Limits	8-17
8.5.8	Spatial Reconstruction	8-17
9	Modelling in 3D	9-1
9.1	Introduction	9-2
9.2	Overview	9-3
9.3	3D Layer Z Coordinate Options	9-4
	<i>Sigma Coordinates</i>	9-6
	<i>Z Coordinates</i>	9-7
	<i>Hybrid Z-Sigma Coordinates</i>	9-7
9.4	Density Coupling	9-9
9.5	Reviewing 3D Results	9-10
9.6	3D Boundary and Initial Conditions	9-11
9.7	3D Structure Conditions	9-12
10	Boundaries and Initial Conditions	10-1
10.1	Introduction	10-2
10.2	Boundary Control Block	10-3
10.3	Assigning Boundary Locations	10-4
10.4	Input Data for Boundary Conditions	10-5
10.4.1	Time Interpolation	10-5
10.4.2	Spatial Location	10-6
10.4.3	BC Headers	10-6
10.4.4	BC Scale and Offsets	10-8
10.4.5	BC Default	10-8
10.5	Boundary Condition Types	10-10
10.5.1	Cell specified boundaries	10-10
10.5.2	Nodestring specified boundaries	10-11
10.5.3	Mesh specified (global, spatially/temporally varying)	10-11
10.5.3.1	<i>Grid Definition Block</i>	10-12
10.5.3.2	<i>Parametric Tropical Cyclone Wind and Pressure Field</i>	10-14
10.5.4	Moving boundaries	10-14
10.5.5	3D Boundary Conditions	10-15
10.5.5.1	<i>Modifying Basic Boundary Conditions</i>	10-15
10.5.5.2	<i>'Curtain' Boundary Conditions</i>	10-15
10.5.5.3	<i>Profile Boundary Conditions</i>	10-15

10.5.5.4 OBC 3D Gridded Boundary Conditions	10-16
10.5.6 Advanced 3D Gridded Boundary Conditions	10-17
10.6 Boundary Types	10-18
10.6.1 Boundary Sub-Types	10-21
10.7 Initial Conditions	10-23
10.7.1 Globally Specified	10-23
10.7.2 Spatially Varying	10-24
10.7.3 Restart Files	10-25
10.8 Transport File	10-26
11 Hydraulic Structures	11-1
11.1 Introduction	11-2
11.2 Structure Types	11-3
11.2.1 Nodestring and Linked Nodestrings	11-4
11.2.2 Autowein	11-5
11.2.3 Linked Zones	11-5
11.2.4 Cell or Zone	11-6
11.2.5 Defining a Nodestring, Linked Nodestring or Linked Zones Structure Block	11-7
11.2.6 Defining a Cell or Zone Structure Type	11-8
11.2.7 Structure Block Example	11-8
11.3 Flux Functions	11-10
11.3.1 Non-Linear Shallow Water Equations	11-10
11.3.2 Timeseries	11-11
11.3.3 Matrix	11-12
11.3.4 Weir Types	11-14
11.3.4.1 Fixed Weirs	11-14
11.3.4.2 Variable Height Weirs	11-15
11.3.5 Wall	11-16
11.3.6 Porous	11-16
11.3.7 Culvert	11-17
11.4 Energy Losses	11-21
11.4.1 Energy Loss Functions	11-21
11.4.2 Energy Loss Considerations	11-21
11.4.3 Form Loss Coefficient	11-22
11.4.4 Energy Table	11-22
11.5 Scalar Functions	11-24
11.6 Structure Controls	11-26
11.6.1 Control Block	11-26
11.6.2 Control Types	11-26

11.6.3	Control Parameters	11-26
11.6.4	Structure Logging File	11-27
11.6.5	Structure Control Examples	11-27
12	Model Outputs	12-1
12.1	Introduction	12-2
12.2	Output Formats	12-3
12.3	Point Output	12-4
12.4	Profile Output	12-5
12.5	Flux/Mass Output	12-6
12.6	Map Output	12-8
12.6.1	Overview	12-8
12.6.2	2D Only Map Output Formats	12-8
12.6.3	3D Map Outputs	12-9
	<i>3D Vertical Averaging Options</i>	12-9
12.6.4	2D and 3D NETCDF Format	12-10
12.6.5	Map Output Statistics	12-15
12.7	Map Output Parameters	12-16
12.8	Structure Outputs	12-23
12.8.1	Structure Flux	12-23
12.8.2	Structure Check	12-23
12.9	Check Files and GIS Layers	12-25
12.9.1	GIS Workspaces (.wor and .qgs files)	12-25
12.10	GIS Check Files	12-26
SIMULATION COMMANDS		A-2
MATERIAL BLOCK COMMANDS		A-1
BOUNDARY CONDITION BLOCK COMMANDS		A-5
STRUCTURE BLOCK COMMANDS		A-1
OUTPUT BLOCK COMMANDS		A-1
AD SIMULATION CONFIGURATION COMMANDS		B-2
WQ SIMULATION CONFIGURATION COMMANDS		C-2

Appendices

Appendix A .fvc File Commands	A-1
Appendix B Advection Dispersion fvc Commands	B-1
Appendix C Water Quality fvc Commands	C-1
Appendix D TUFLOW FV NetCDF Output Format	D-2
Appendix E Common NetCDF Input File Example Format	E-1

List of Figures

Figure 1-1	Schematisation of Common TUFLOW Flexible Mesh Solver Features	1-3
Figure 1-2	Schematisation of Common TUFLOW Fixed Grid Solver Features	1-3
Figure 1-3	Cell Schematisation	1-9
Figure 1-4	Example decrease in runtime using TUFLOW FV multi-thread processing	1-10
Figure 3-1	Digital Elevation Model of Port Curtis, Queensland, Australia	3-6
Figure 3-2	TUFLOW FV Mesh of Port Curtis Estuary, Queensland, Australia	3-7
Figure 3-3	Example Tidal Water Level Timeseries Calibration Plot	3-14
Figure 3-4	Example Current Speed and Current Direction Timeseries Calibration Plots for a Tidal Estuary	3-14
Figure 3-5	Example Flow Timeseries Calibration Plots for a Tidal Estuary	3-15
Figure 4-1	Example TUFLOW FV Sub-Folder Structure	4-3
Figure 4-3	Example TUFLOW FV Control File using Include command	4-11
Figure 4-4	Contents of Materials.fvc called via Include command	4-11
Figure 4-5	Example TUFLOW FV Simulation Control File	4-13
Figure 6-1	1 st (left) and 2 nd (right) Order Velocity Results Dambreak Flow	6-5
Figure 6-2	TUFLOW FV Mode Split Timestepping	6-11
Figure 8-1	Mesh Development Example	8-3
Figure 8-2	Example 2dm File	8-4
Figure 8-3	Example Node Definition	8-5
Figure 8-4	Example Quadrilateral Element Definition	8-5
Figure 8-5	Example Triangular Element Definition	8-5
Figure 8-6	Example Nodestring Definition	8-6
Figure 8-7	Cell Centre Elevations Assigned via Read GRID Zpts	8-8
Figure 8-8	Cell Elevation Polyline Example	8-11
Figure 8-9	Cell Elevation Polygon Example	8-11
Figure 8-9	Materials Specification using 2d_mat GIS Layers	8-12
Figure 9-1	3D Model Vertical Discretisation Options	9-4
Figure 9-2	Hybrid Z-Sigma example setup (left) with z layer face file (right)	9-8
Figure 9-3	Salinity long section in coastal ‘salt wedge’ estuary.	9-9
Figure 9-4	Counter current velocity at depth. Considerations for result depth averaging.	9-10
Figure 10-1	Boundary Condition Block Examples	10-3
Figure 11-6	Auto Weir Example	11-5
Figure 11-1	Structure Block Example	11-9
Figure 11-1	hQh Structure Example	11-12
Figure 11-2	hQh Calculation Design Options	11-13

Figure 11-3	Weir Flow	11-14
Figure 11-4	Fixed Elevation Weir Example	11-15
Figure 11-5	Variable Elevation Weir Example	11-15
Figure 11-7	1D Outlet Control Culvert Flow Regimes	11-18
Figure 11-8	1D Inlet Control Culvert Flow Regimes	11-19
Figure 11-9	1D Culvert Example	11-19
Figure 11-11	Trigger Control Example	11-27
Figure 11-12	Timeseries Control Example	11-28
Figure 11-13	Sample Rule Control Example	11-28
Figure 11-14	Sample Rule Control Example	11-28
Figure 12-1	Points Output Commands and 'Output_points.CSV' Example	12-4
Figure 12-2	TUFLOW FV 2D Points Output File Example	12-4
Figure 12-3	Profile Output Example Commands	12-5
Figure 12-4	Example profile output	12-5
Figure 12-5	SMS Mapped Output Commands Example	12-8
Figure 12-6	TUFLOW FV Mapped Current Velocity Output in the SMS Generic Mesh Module Environment	12-9
Figure 12-7	TUFLOW FV Sheet Plot with Zoom Example: Velocity Magnitude Top 50% Water Column (top); Velocity Magnitude Bottom 50% Water Column (bottom)	12-12
Figure 12-8	TUFLOW FV Salinity Vertical Distribution: Model Mesh and Curtain Polyline (top); Salinity Curtain Plotted with Polyline Chainage; Salinity Curtain Plotted with Polyline Coordinates (bottom)	12-13
Figure 12-9	TUFLOW FV Velocity Vertical Distribution: River Bend Flood Flow and Cross-Section Locations (top); Total Velocity Magnitude (contours) with Radial Flow Vectors Cross-Sections (bottom three)	12-14
Figure 12-10	Statistical Output Example Commands	12-15

List of Tables

Table 1-1	Table Summary of United Kingdom Environment Agency Benchmark Tests	1-12
Table 2-1	Suggested Supporting Software	2-3
Table 2-1	TUFLOW FV Tutorial/Demo Models	2-10
Table 4-1	Recommended TUFLOW FV Sub-Folder Structure Description	4-4
Table 4-2	TUFLOW FV File Formats	4-5
Table 4-4	Reserved Characters – Text Files	4-9
Table 4-4	Notation Used in Command Documentation – Text Files	4-10
Table 4-5	TUFLOW FV Simulation Control File Layout	4-12
Table 5-1	TUFLOW Interpretation of GIS Objects	5-3
Table 5-2	GIS Input Data Layers and Recommended Prefixes	5-5
Table 6-1	Solution Include Terms	6-3
Table 6-2	Model Units	6-6
Table 6-3	Wind Stress Model and Drag Coefficients	6-9
Table 8-1	GIS Nodestring (2d_ns) Attributes	8-7
Table 8-2	2D Z (2d_z_) Attribute Description	8-9
Table 8-3	Cell Elevation File Examples (the example on the left references X,Y co-ordinates and the one on the right uses Cell ID)	8-10
Table 8-4	2D Material (2d_mat) Attribute Description	8-13
Table 9-1	3D Geometry Layer Options	9-6
Table 10-1	Grid Definition Block Commands	10-12
Table 10-2	Boundary Condition Types	10-18
Table 10-3	Boundary Condition Types (Advanced)	10-19
Table 10-4	Boundary Sub-types	10-22
Table 11-1	Flux Function Types	11-10
Table 11-2	1D Culvert Flow Regimes	11-17
Table 11-3	Culvert File Inputs	11-20
Table 12-1	Output Data Formats	12-3
Table 12-2	Flux and Mass Outputs	12-6
Table 12-3	Common Map Output Types	12-16
Table 12-4	Advanced Map Output Types	12-16
Table 12-5	Sediment Transport Map Output Types	12-20
Table 12-5	GIS Check Files	12-26

GLOSSARY & NOTATION

ArcMap/ArcGIS	ArcMap, also referred to as ArcGIS, is distributed by ESRI (http://www.esri.com). ArcMap is a GIS software package that can be used to develop the georeferenced input data for TUFLOW. TUFLOW writes result and check files which are compatible with ArcMap also.
Attribute	Data associated with / or attached to a GIS object. For example, an elevation is attached to a point using a column of data named “Height”. The “Height” of the point is an attribute of the point.
Build	The TUFLOW Build number is in the format of year-month-xx where xx is two letters starting at AA then AB, AC, etc. for each new build for that month. The Build number is written to the first line in the .tlf log files so that it is clear what version of the software was used to simulate the model. The first Build was 2001-03-AA. Prior to that, no unique version numbering was used.
Cell	Quadrilateral or triangular shaped computational element in a domain.
Centroid	The centre of a region, polygon or cell
Command	Instruction in a control file.
control file	Text file containing a series of commands (instructions) that control how a simulation proceeds or a 1D or 2D domain is built.
.dbf	Industry standard database file format used by ESRI GIS .shp layers to store attribute data.
DTM	Digital Terrain or Elevation Model
GIS	Geographic Information System, for use in TUFLOW modelling it will need to be able to import/export files in .mif or .shp format.
Read GIS	The TUFLOW command “ <code>Read GIS <data type> ==</code> ” is used to input spatial data into TUFLOW. For example, <code>Read GIS Nodestring == 2d_ns_myboundaries.mif</code> Would be used to read nodestring location data.
Invert	The elevation of the base (bottom) of a culvert or other structure.
IWL	Initial Water Level
land cell	A land cell is one that will never wet, ie. an inactive cell.
Layer	A GIS data layer (referred to as a “table” in MapInfo).

Line	A GIS object defining a straight line defined by two points. See also, polyline (Pline).
MapInfo	MapInfo is distributed by Pitney Bowes (http://www.mapinfo.com/). MapInfo is a GIS software package that can be used to develop the georeferenced input data for TUFLOW. TUFLOW writes result and check files which are compatible with MapInfo also.
MAT	Material type.
Material	Term used to describe a bed resistance category or land-use. Examples of different materials are: river, river bank, mangroves, roads, grazing land, sugar cane, parks, etc.
QGIS	QGIS is freeware software (www.qgis.org/). QGIS is a GIS software package that can be used to develop the georeferenced input data for TUFLOW. TUFLOW writes result and check files which are compatible with QGIS also.
.mid	MapInfo Industry standard GIS import/export file that contains the attribute data of geographic objects in a .mif file. The .mif and .mid files are a pair, and can't be opened in GIS unless both files are present.
.mif	MapInfo Industry standard GIS import/export file that contains the attribute data types and the geographic coordinates of objects. The attribute data of the objects is stored in the .mid file by the same filename.
Node	Water level computation point in a 1D domain. Node in a model mesh used for viewing 2D results in SMS. The nodes are located at the cell corners. Node is also used by MapInfo to refer to vertices along a polyline or a region (polygon).
null cell	A null cell is an inactive 2D cell used for defining the inactive side of an external boundary.
Obvert	The elevation of the underside (soffit) of a culvert or other structure.
Point	GIS object representing a point on the earth's surface. A point has no length or area.
Polygon	See region.
polyline (or Pline)	A GIS object representing one or more lines connected together. A polyline has a length but no area.
polyline segment	One of the line segments that make up a polyline.

Region	A GIS object representing an enclosed area (ie. a polygon). A region has a centroid, perimeter and area. Polygons can have internal holes.
.shp	ESRI GIS layer file containing the geographic coordinates of objects. This is referred to as a Shapefile, however, a GIS dataset in Shapefile format will also contain a number of additional files (.dbf, .shx, .prj).
SMS	Surface Water Modelling Software distributed by Aquaveo (formerly EMS-I) (www.aquaveo.com). SMS can be used as an interface for TUFLOW, allowing the user to view results and also to create a TUFLOW model within the SMS interface. This is commercially available software and a licence is required.
Snap	When geographic objects are connected exactly at a point or along a side. ArcMap, QGIS and MapInfo all have a “snap” feature, which ensures the features have the same coordinates. The snap tolerance can be changed in TUFLOW using the Snap Tolerance command.
Soffit	The elevation of the underside of a bridge deck or the inner top of a culvert. Same as obvert. Note this manual uses the term obvert.
Vertices	Plural of vertex.
Vertex	Digitised point on a line, polyline or region (polygon).

1 Introduction

Chapter Contents

1	Introduction	1-1
1.1.1	Introduction	1-2
1.2	TUFLOW FV	1-4
1.2.1	Flexible Mesh Modelling	1-5
1.3	Solution Scheme	1-6
1.3.1	Non-Linear Shallow Water Equations	1-6
1.3.2	Scalar Conservation Equations	1-7
1.3.3	Conserved vs. Scalar Variables	1-8
1.3.3.1	<i>Conserved Variables</i>	1-8
1.3.3.2	<i>Scalar Variables</i>	1-8
1.4	Schematisation	1-9
1.5	Multi-Core Processing	1-10
1.6	UK Benchmarking Study	1-11

1.1.1 Introduction

TUFLOW Products are a suite of world leading urban drainage, catchment flood and coastal simulation software. It is developed through collaboration with universities and our users. We deliver software of high scientific standard, that is rigorously benchmarked, practical, and workflow efficient.

The TUFLOW suite includes two separate product ranges:

- 1) TUFLOW's Flexible Mesh Solver, TUFLOW FV, is a numerical hydrodynamic model for the one-dimensional (1D), two-dimensional (2D) and three-dimensional (3D) Non-Linear Shallow Water Equations (NLSWE). The model is suitable for solving a wide range of hydrodynamic systems ranging in scale from the open channels and floodplains, through estuaries to coasts and oceans. TUFLOW FV also includes optional modules that provide advection dispersion, sediment transport and morphology, particle tracking, water quality and three-dimensional modelling capabilities. Figure 1-1 shows common TUFLOW FV solver features.

This manual provides user guidance for TUFLOW's Flexible Mesh solver, TUFLOW FV.

- 2) TUFLOW's Fixed Grid Solvers, using a matrix (grid) of square cells as the computation structure. It includes two 2D engine options and a coupled 1D engine:

- TUFLOW Classic: A 2D implicit Finite Difference solver;
- TUFLOW HPC (Heavily Parallelised Compute): A 2D explicit Finite Volume solver; and
- TUFLOW 1D (formally ESTRY): TUFLOW's native 1D open channel and underground pipe network engine.

TUFLOW's fixed grid solver includes world leading 1D/2D and 2D/2D dynamic linking and is compatible for CPU and GPU hardware. These features are shown in Figure 1-2.

The fixed grid solvers are well suited to simulating integrated urban drainage situations (above and below ground), distributed hydrology direct rainfall scenarios, catchment flooding, tides and storm tide hydraulics.

This manual DOES NOT provide user guidance for our fixed grid solver TUFLOW Classic and TUFLOW HPC. This documentation is available for download from the [TUFLOW website](#).

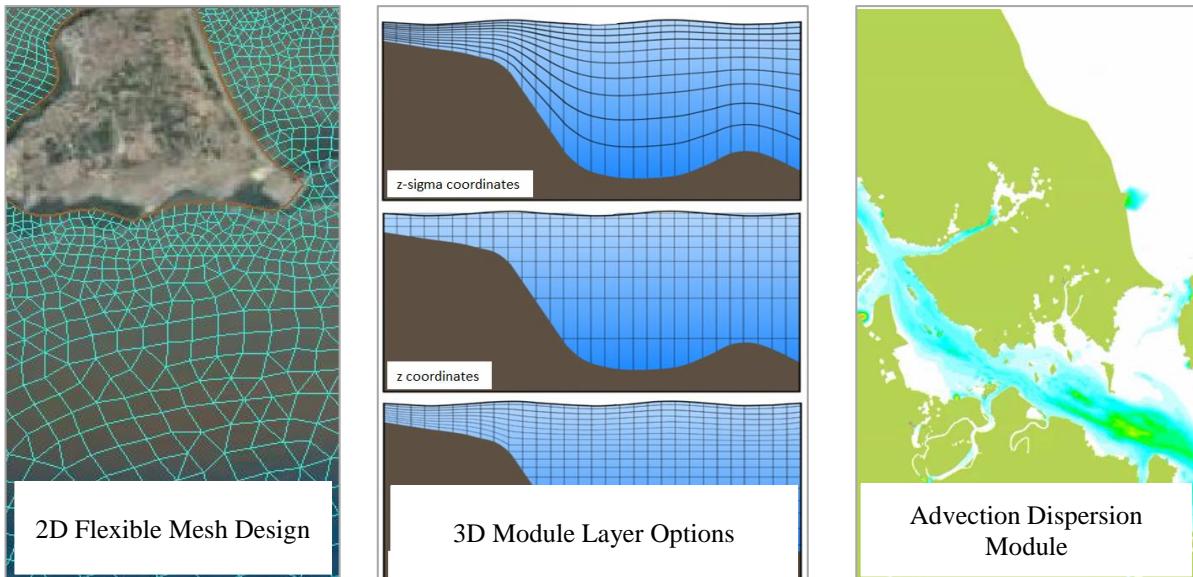


Figure 1-1 Schematisation of Common TUFLOW Flexible Mesh Solver Features

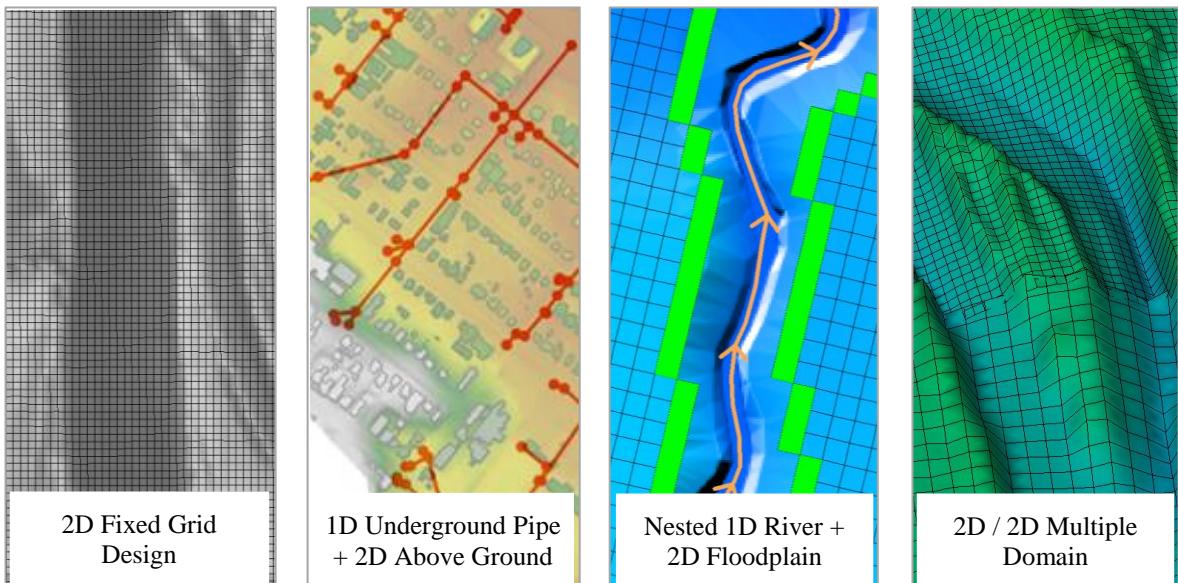


Figure 1-2 Schematisation of Common TUFLOW Fixed Grid Solver Features

For any enquiries about the fixed grid or flexible mesh solvers, please email sales@tuflow.com or support@tuflow.com.

1.2 TUFLOW FV

The Finite Volume (FV) numerical scheme employed by TUFLOW FV is capable of solving the NLSWE on both structured rectilinear grids and unstructured meshes comprised of triangular and quadrilateral elements. The flexible mesh allows for seamless boundary fitting along complex coastlines or open channels as well as accurately and efficiently representing complex bathymetries with a minimum number of computational elements. The flexible mesh capability is particularly efficient at resolving a range of scales in a single model without requiring multiple domain nesting. The governing equations are updated using an appropriate timestep that obeys the Courant-Friedrich-Lowy (CFL) constraints imposed by the flow characteristics. Further details regarding the numerical scheme employed by TUFLOW FV are provided in the [TUFLOW FV Science Manual](#).

Unstructured mesh geometries can be created using a suitable mesh generation tool such as Aquaveo's SMS or Rising Water Software's GIS Mesher. Both Cartesian and Spherical mesh geometries can be used as the basis for TUFLOW FV simulations. Model elements such as topography, boundary locations and structures can be specified interactively using GIS files in industry standard GIS packages including ArcGIS, Mapinfo and QGIS.

Three-dimensional simulations can be performed within TUFLOW FV using either sigma-coordinate or a hybrid z-coordinate vertical mesh. Three-dimensional simulations can optionally use a mode-splitting approach to efficiently solve the external (free-surface) mode in 2D at a timestep constrained by the surface wave speed while the internal 3D mode is updated less frequently. TUFLOW FV provides various options to vertically average 3D output and thereby simplify post-processing tasks. Both MATLAB and Python visualisation libraries are available for download on the TUFLOW website to assist with review and presentation of 3D results.

Advection-Diffusion (AD) of multiple water-borne constituents can be solved within TUFLOW FV, either coupled with a hydrodynamic simulation, or alternatively in transport mode using a pre-calculated transport file. Simple constituent decay and settling can be accommodated in the AD solutions, or alternatively more complex sediment transport algorithms can be applied through the Sediment Transport Module.

Both cohesive and non-cohesive Sediment Transport (ST) routines can be accessed through in-built TUFLOW FV modules which handle both bed and suspended load mechanisms. Dynamic morphology updating can be optionally activated to reflect changes upon the underlying topography.

Baroclinic pressure-gradient terms can be optionally activated to allow the hydrodynamic solution to respond to temperature, salinity and sediment induced density gradients. Atmospheric heat exchange can also be calculated from given standard meteorological parameter inputs by an integrated module.

TUFLOW FV has a variety of options for simulating horizontal turbulent mixing, including the Smagorinsky scheme. Simple parametric models for vertical mixing are incorporated within TUFLOW FV and for more complicated turbulence model algorithms an interface for linking with various external turbulence models has been implemented.

The water quality module couples TUFLOW FV with the [Aquatic EcoDynamics \(AED\)](#) model to simulate aquatic biogeochemical and ecological dynamics. AED has been developed in response to recognised deficiencies in existing water quality approaches used around the world, and it supports a common library of biogeochemical and ecological ‘components’. AED includes the ability to simulate interactions between biogeochemical variables including oxygen, carbon, nutrients (organic and inorganic), sediment, light, temperature and algal species.

TUFLOW FV provides a multitude of options for specifying model boundary conditions, including:

- Various open boundary conditions
- Point source inflows
- Moving point source inflows
- Spatially and temporally varied forcing e.g. windfields, short-wave forcing.

Model output files are primarily map output in SMS DAT or XMDF formats (2D or vertically averaged 3D), map output in NetCDF format (2D, vertically averaged 3D or full 3D) and time-series output in comma-delimited format (2D or vertically averaged 3D). The NetCDF output files can be viewed using any numerical analysis package with a NetCDF library interface, including MATLAB, R, GNU Octave or Python NumPy. The TUFLOW FV NetCDF output file structure is described in Appendix C.

1.2.1 Flexible Mesh Modelling

TUFLOW FV is designed for solving the NLSWE on unstructured geometries and is commonly referred to as a ***flexible mesh model***. Compared to structured rectilinear grids (i.e. fixed grids) the design of the flexible mesh tends to have a greater influence on model performance. Therefore, more time and effort should be spent preparing the model mesh geometry. Over the life cycle of a modelling project, a well assembled mesh will save time (both the modellers and the computers).

The flexible mesh consists of a network of irregular triangular and quadrilateral elements. This has inherent advantages, including:

- Mesh resolution can be adjusted according to the needs of the study (i.e. fine resolution within the area of interest and coarser resolution in the regional extents). Therefore, a range of spatial scales can be modelled without resorting to nesting.
- Mesh alignment can neatly fit bathymetric contours and boundary extents, optimising mesh resolution. This is particularly relevant in regions with complex bathymetric or topographic features.
- Specific features, the geometry such as hydraulic structures and coastal barriers can be included explicitly in the model mesh.

To exploit these advantages, the mesh needs to be designed carefully and appropriately for the specific model application. There are several mesh generators available to construct a model mesh discussed further in Section 8.1.

1.3 Solution Scheme

1.3.1 Non-Linear Shallow Water Equations

TUFLOW FV solves the NLSWE, including viscous flux terms and various source terms on a flexible mesh comprised of triangular and quadrilateral elements. A summary of the equation set is provided in the following sections. For full detail please refer to the [TUFLOW FV Science Manual](#).

The NLSWE is a system of equations describing the conservation of fluid mass/volume and momentum in an incompressible fluid, under the hydrostatic pressure and Boussinesq assumptions. The standard form of the NLSWE, which relates the time-derivative of the conserved variables to flux-gradient and source terms, is given below.

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}(\mathbf{U}) \quad (1)$$

The finite-volume schemes are derived from the conservative integral form of the NLSWE, which are obtained by integrating the standard conservation equation over a control volume, Ω .

$$\int_{\Omega} \frac{\partial \mathbf{U}}{\partial t} d\Omega + \int_{\Omega} \nabla \cdot \mathbf{F}(\mathbf{U}) d\Omega = \int_{\Omega} \mathbf{S}(\mathbf{U}) d\Omega \quad (2)$$

Gauss' theorem is used to convert the flux-gradient volume integral into a boundary-integral:

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{U} d\Omega + \oint_{\partial\Omega} (\mathbf{F} \cdot \mathbf{n}) ds = \int_{\Omega} \mathbf{S}(\mathbf{U}) d\Omega \quad (3)$$

where $\int_{\Omega} d\Omega$ represent volume integrals and $\oint_{\partial\Omega} ds$ represents a boundary integral and \mathbf{n} is the boundary unit-normal vector.

The NLSWE conserved variables are volume (depth), x-momentum and y-momentum:

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix} \quad (4)$$

where h is depth, u is x-velocity and v is y-velocity.

The x , y and z components of the inviscid flux (\mathbf{F}^I) and viscous flux (\mathbf{F}^V) terms in the NLSWE are given below.

$$\begin{aligned} \mathbf{F}_x^I &= \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \quad \mathbf{F}_x^V \approx \begin{bmatrix} 0 \\ -hK_v \frac{\partial u}{\partial x} \\ -hK_v \frac{\partial v}{\partial x} \end{bmatrix} \\ \mathbf{F}_y^I &= \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}, \quad \mathbf{F}_y^V \approx \begin{bmatrix} 0 \\ -hK_v \frac{\partial u}{\partial y} \\ -hK_v \frac{\partial v}{\partial y} \end{bmatrix} \end{aligned} \quad (5)$$

$$\mathbf{F}_z^I = \begin{bmatrix} hw \\ hwu \\ hwv \end{bmatrix}, \quad \mathbf{F}_z^V \approx \begin{bmatrix} 0 \\ -\nu_t \frac{\partial u}{\partial z} \\ -\nu_t \frac{\partial v}{\partial z} \end{bmatrix}$$

where K_v and ν_t are the horizontal and vertical eddy-viscosity terms.

Some of the various source terms to the NLSWE are provided below:

$$\mathbf{S} = \begin{bmatrix} 0 \\ gh \frac{\partial z_b}{\partial x} + fvh - \frac{h}{\rho_0} \frac{\partial p_a}{\partial x} - \frac{hg}{\rho_0} \int_z^\eta \frac{\partial \rho}{\partial x} dz - \frac{1}{\rho_0} \left(\frac{\partial s_{xx}}{\partial x} + \frac{\partial s_{xy}}{\partial y} \right) + \frac{\tau_{sx}}{\rho_0} - \frac{\tau_{bx}}{\rho_0} \\ gh \frac{\partial z_b}{\partial y} - fuh - \frac{h}{\rho_0} \frac{\partial p_a}{\partial y} - \frac{hg}{\rho_0} \int_z^\eta \frac{\partial \rho}{\partial y} dz - \frac{1}{\rho_0} \left(\frac{\partial s_{yx}}{\partial x} + \frac{\partial s_{yy}}{\partial y} \right) + \frac{\tau_{sy}}{\rho_0} - \frac{\tau_{by}}{\rho_0} \end{bmatrix} \quad (6)$$

where,

- $\frac{\partial z_b}{\partial x}, \frac{\partial z_b}{\partial y}$ are the x- and y-components of bed slope;
- f is the coriolis coefficient;
- ρ is the local fluid density, ρ_0 is the reference density and p_a is the mean sea level pressure;
- s_{ij} is the short-wave radiation stress tensor; and
- τ_s and τ_b are respectively the surface and bottom shear stress terms (where applicable).

Other source terms not included above include inflow/outflow to/from the water column.

For full detail on the above source terms please refer to the [TUFLOW FV Science Manual](#).

1.3.2 Scalar Conservation Equations

Analogous conservation equations are solved for the transport of scalar constituents in the water column.

$$U = [hC] \quad (7)$$

where C is the constituent concentration. The flux components of the scalar conservation equation are:

$$\begin{aligned} F_x^I &= [huC], \quad F_x^V \approx \left[-h \left(D_{xx} \frac{\partial C}{\partial x} + D_{xy} \frac{\partial C}{\partial y} \right) \right] \\ F_y^I &= [hvC], \quad F_y^V \approx \left[-h \left(D_{yx} \frac{\partial C}{\partial x} + D_{yy} \frac{\partial C}{\partial y} \right) \right] \\ F_z^I &= [hwC], \quad F_z^V \approx \left[-h\nu'_t \frac{\partial C}{\partial z} \right] \end{aligned} \quad (8)$$

The source components may include scalar decay and settling:

$$S = [-K_d hC - w_s C] \quad (9)$$

where K_d is a scalar decay-rate coefficient and w_s is a scalar settling velocity.

1.3.3 Conserved vs. Scalar Variables

Throughout the manual variables used or output by the model are referred to as conserved or scalar variables. This section provides an overview of each and the differences between the two.

1.3.3.1 *Conserved Variables*

Conserved variables are the physical quantities that are tracked by TUFLOW FV and are conserved within the numerics of the model, other than being changed by source/sink boundary conditions and structures. When referring to them in TUFLOW FV, they follow a prescribed order as follows:

- Water volume per cell area (equivalent to the depth of water in a cell): For incompressible fluids, this is the water mass per cell area in the cell.
- X-component of momentum: The x-component of velocity multiplied by the water volume.
- Y-component of momentum: The y-component of velocity multiplied by the water volume.
- Salt mass (if salinity is included): The salinity concentration multiplied by the water volume.
- Heat mass (if temperature included): The temperature concentration multiplied by the water volume.
- Sediment mass (for each sediment that is modelled): The sediment concentration multiplied by the water volume.
- Tracer mass (for each tracer that is modelled): The tracer concentration multiplied by the water volume.
- Water Quality scalar mass (for each tracer that is modelled): The water quality parameter concentration multiplied by the water volume. The order of this is dependent on external water quality module, see the relevant external water quality manual for information.

1.3.3.2 *Scalar Variables*

Scalar variables are physical quantities that are tracked within TUFLOW FV, are conserved and can be represented by one value per cell. Examples of these are temperatures and concentrations such as the salinity, tracers, sediments and water quality parameters, as well as secondary variables originating from these such as the density. They do not include variables that require multiple pieces of information to define (like vectors) such as velocity, momentum, flow or shear stress.

1.4 Schematisation

The mesh cells (or elements) are the computational blocks of the finite volume approach used by TUFLOW FV. TUFLOW FV uses a cell centred scheme with a single bed elevation value assigned to each cell in its calculations, and then produces output that is applicable for each cell (cell velocities are derived from the values across each cell face as shown in Figure 1-3).

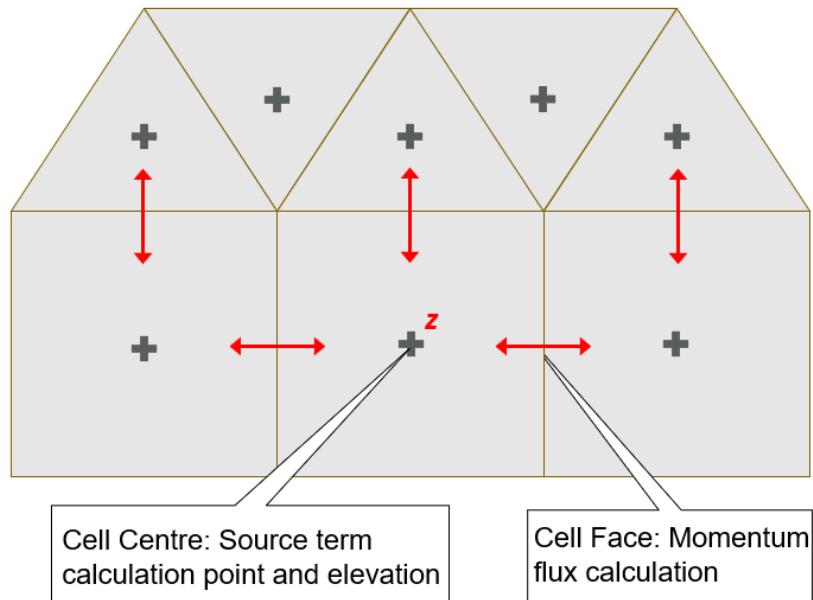


Figure 1-3 Cell Schematisation

1.5 Multi-Core Processing

TUFLOW FV is parallelised for multi-processor machines using the OpenMP implementation of shared memory parallelism. This means that a TUFLOW FV model simulation will run faster if there is more than one processor (or thread) on a single computer. The increase in computational speed (or decrease in runtime) is not quite linear with the number of threads, as demonstrated in Figure 1-4.

Unless the user decides otherwise, TUFLOW FV will run using the maximum number of threads available to it, only limited by the software licence or computer hardware.

Upcoming releases of TUFLOW FV are planned to include GPU acceleration and domain decomposition to further improve computational performance.

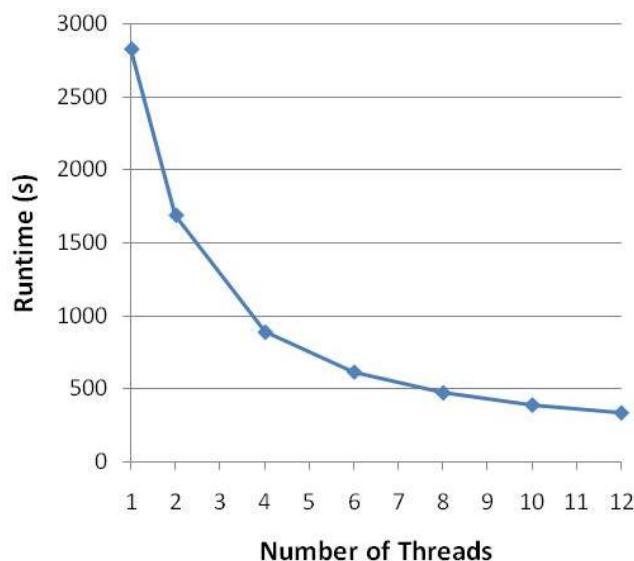


Figure 1-4 Example decrease in runtime using TUFLOW FV multi-thread processing

1.6 UK Benchmarking Study

The “Desktop Review of 2D Hydraulic Modelling Packages” report (Néelz and Pender 2009) released in the United Kingdom by the Environment Agency highlighted the rapidly growing number of hydraulic modelling packages available for flood inundation estimation. A vast amount of documentation exists on appropriate applications of each modelling package however little of it discusses the influence that the choice of modelling package may have. The conclusions of the report recommended a series of benchmarking test cases to provide guidance on choosing the appropriate modelling package for future applications.

In response to the recommendations, a series of 10 2D flood inundation modelling benchmarking tests were conducted in 2010 using a variety of modelling packages. Additional testing was undertaken in 2012 and published in 2013 due to the availability of new modelling packages and the further development undertaken on existing modelling packages. 15 software development organisations tested a total of 19 modelling packages. The results are documented in the report, [“Benchmarking the Latest Generation of 2D Hydraulic Modelling Packages” \(Néelz, S. and Pender, G. 2013\)](#).

The TUFLOW suite was submitted for the initial phase of testing in 2010 with all three 2D schemes, TUFLOW’s implicit and explicit GPU solvers, and TUFLOW FV’s explicit solvers undergoing the more recent phase of testing in 2012. The results demonstrated consistency between each of the three TUFLOW engines and with other fully dynamic schemes. All three TUFLOW engines were found to be suitable for the following applications:

- Prediction of inundation extent;
- Prediction of maximum depth;
- Prediction of maximum velocity;
- Prediction of temporal variation in inundation extent;
- Prediction of temporal variation in depth; and
- Prediction of temporal variation in velocity.

The 10 tests are outlined in Table 1-1. For further information refer to [“Benchmarking the Latest Generation of 2D Hydraulic Modelling Packages” \(Néelz, S. and Pender, G. 2013\)](#).

All tests were completed using TUFLOW FV with the exception of Tests 7 and 8B. These two tests require 1D/2D linking which is currently in development.

Table 1-1 Table Summary of United Kingdom Environment Agency Benchmark Tests

Test Number	Description	Purpose
1	Flooding a disconnected water body	Assess basic capability to simulate flooding of disconnected water bodies on floodplains or coastal areas.
2	Filling of floodplain depressions	Tests capability to predict inundation extent and final flood depth for low momentum flow over complex topographies.
3	Momentum conservation over a small (0.25m) obstruction	Tests capability to simulate flow at relatively low depths over an obstruction with an adverse slope.
4	Speed of flood propagation over an extended floodplain	Tests simulation of speed of propagation of flood wave and the prediction of velocities at the leading edge of the advancing flood.
5	Valley flooding	Tests simulation of major flood inundation at the valley scale.
6A and 6B	Dam break	Tests simulation of shocks and wake zones close to a failing dam.
7	River to floodplain linking	Evaluates capability to simulate flood volume transfer between rivers and floodplains using 1D to 2D model linking. This test has not yet been undertaken for TUFLOW FV but will be included following 1D/2D integration.
8A and 8B	Rainfall and sewer surcharge flood in urban areas	Tests capability to simulate shallow flows in urban areas with inputs from rainfall (8A) and sewer surcharge (8B). Test 8B has not yet been undertaken for TUFLOW FV but will be included following 1D/2D integration.

2 Running TUFLOW FV

Chapter Contents

2.1	Software Structure	2-2
2.2	Installing and Running TUFLOW FV	2-5
2.2.1	TUFLOW FV Downloads and Installation	2-5
2.2.2	Linux Installation	2-5
2.2.3	USB Locks (Dongles) and Licencing	2-5
2.2.4	Dongle Types and Setup	2-5
2.3	Performing Simulations	2-7
2.3.1	Batch File Example and Run Options (Switches)	2-7
2.3.2	From a Text Editor	2-8
2.3.3	From the SMS Interface	2-8
2.3.4	Directly from GIS	2-8
2.3.5	Change Priority, Pause, Restart or Cancel a Simulation	2-8
2.4	Licence Free Simulations	2-10
2.4.1	Tutorial Models	2-10
2.5	Tips and Tricks	2-11

2.1 Software Structure

TUFLOW FV offers a powerful and efficient approach to modelling compared with other hydrodynamic modelling software. TUFLOW FV scripts (control files) allow modellers to readily and easily setup, modify and run numerous simulations, whether it be different calibration events, a batch of design events or various what-if scenarios investigating flood mitigation options. Combined with the power of GIS, the TUFLOW FV concept offers substantial benefits in terms of efficient workflow, modifications to models, quality control and user satisfaction, especially on investigations involving the examination of a multitude of events and “what-if” scenarios.

TUFLOW FV requires the generation of a mesh which forms the basis for model topography/bathymetry, resolution and material coverage. Mesh generators are discussed in detail in Section 8.1. A good mesh generator tool used skilfully is fundamental when developing a model in TUFLOW FV. For those familiar with the ‘TUFLOW Classic’ workflow where the model mesh is automatically generated at runtime, the need for a manually and carefully developed user-defined mesh is a significant departure from the methodology you are used to.

Text files are used for controlling simulations and simulation parameters, whilst the bulk of data input is in GIS, comma delimited text files (.CSV) in combination with the model mesh. An approach that combines mesh generation and GIS offers several benefits including:

- The unparalleled power of GIS as a “work environment”;
- The many GIS data management, manipulation and presentation tools;
- Input data such as materials, topography and structures can be geographically referenced to provide independency from the underlying model mesh;
- Efficiency in producing high quality GIS based mapping for reports, brochures, plans and displays;
- Flexibility in choice of GIS package;
- Seamless handover of model inputs and results to clients requiring data in GIS format; and
- Improved data quality control.

For those modellers preferring to work predominantly within a Graphical User Interface, there are also several third-party options as discussed later in this chapter. For experienced modellers, the ability to be able to essentially script up a model, and use GIS in combination with these GUIs, offers a highly proficient arrangement by utilising the optimal software for different modelling tasks.

The fundamental software necessary for building and viewing TUFLOW FV models are:

- A text editor.
- A mesh generation software such as [Aquaveo SMS](#) or the [Rising Water Software GIS Mesher](#).
- Spreadsheet software.
- GIS software that can import/export .mif files or .shp files.

- 3D surface modelling software for the creation and interrogation of a DTM, and for importing 3D surfaces of water levels, depths, hazard, etc. This functionality is available in most GIS packages, sometimes as an add-on.
- GIS, Aquaveo SMS and/or a GUI for viewing results.

The above combination offers a very powerful, workflow efficient and economical system for hydraulic modelling, driven by the unparalleled range of features and functionality available in TUFLOW FV, and the efficient modelling workflow that is generated using TUFLOW FV scripts or control files.

Suggested software packages include (but are not limited to) those listed in Table 2-1. A typical model setup workflow comprises the following elements:

- The GIS system is used to set up, modify, thematically map and manage all geographic data.
- The mesh generator to provide the underlying computation domain.
- Time-series and other non-geographically located data tabulated using spreadsheet software. Often scripting and programming software such as MATLAB or Python may be used to develop or modify advanced model inputs.
- The text editor is used to create and edit TUFLOW FV simulation control files. The control files list all the simulation commands and file path references to the above mentioned GIS and tabular datasets.
- Visualisation of TUFLOW outputs results in text editor, spreadsheet, GIS or via/in combination with MATLAB or Python visualisation libraries.

Table 2-1 Suggested Supporting Software

Software Type	Suggested Software
Text Editor	UltraEdit / TUFLOW Wiki UltraEdit Tips Notepad++ / TUFLOW Wiki Notepad++ Tips Textpad / TUFLOW Wiki Textpad Tips Other: Any text editor can be used for creating TUFLOW FV control files, including the Microsoft Windows default, Notepad. However, the above listed editors are recommended. They allow for advanced options, such as colour highlighting of TUFLOW FV control files and launching TUFLOW FV simulations from the editor – see downloads on this TUFLOW FV page .
Spreadsheet Software	Microsoft Excel / TUFLOW Wiki Excel Tips Libre Office
Mesh Generator	Aquaveo SMS Rising Water Software GIS Mesher
Scripting Packages	MATLAB and the accompanying TUFLOW FV MATLAB Tools Python and the accompanying TUFLOW FV Python Tools

Software Type	Suggested Software
GIS Platforms	ArcGIS with Spatial Analyst MapInfo Professional with Vertical Mapper or equivalent QGIS with the TUFLOW Viewer Plugin TUFLOW Wiki QGIS Tips
GUI Platforms (Additional to or as an alternative to using GIS)	Aquaveo SMS FEWS SMS / TUFLOW Wiki SMS Tips

2.2 Installing and Running TUFLOW FV

2.2.1 TUFLOW FV Downloads and Installation

TUFLOW FV does not have an automatic installation, but instead requires the user to copy or unzip the downloaded files into a folder. Whilst this may seem “old-fashioned”, this approach allows the modeller to have as many releases or versions of TUFLOW FV available as required on their computer, as there is often a need to run or re-run legacy models using older TUFLOW FV versions. All TUFLOW FV versions, manuals and release notes officially released are available from the [All Downloads Page](#).

The TUFLOW FV software may even be located in a folder under the parent folder for the model, so that the version of TUFLOW FV used for that particular model is associated and archived with the model.

2.2.2 Linux Installation

TUFLOW FV is available for both Windows and Linux Operating Systems. For Linux installation instructions please contact support@tuflow.com.

2.2.3 USB Locks (Dongles) and Licencing

A TUFLOW FV licence (eg. USB lock or dongle) is required to run TUFLOW FV, but is not required when using third party software such as a GIS, text editor or GUI. TUFLOW can be used licence free for the [TUFLOW FV Tutorial Models](#). A Free or Demo Mode allows license free simulation of models under a certain size and runtime (available soon).

For third-party USB locks that have TUFLOW FV licences please refer to the vendor’s documentation for configuring the licence.

2.2.4 Dongle Types and Setup

TUFLOW FV is licensed using either a WIBU Codemeter (metal) dongle or software licence. These dongles and software licenses were introduced in 2010 and 2018 respectively. They offer the following features:

- Support for 64-bit on both Windows and selected Linux¹ platforms.
- Network licence manager running as a service (ie. the computer with the network dongle needs to be on, but no one needs to be logged in).
- Multiple WIBU dongles (local and/or network licensed) are accessible together (ie. if all licences from one dongle are taken, licences from other dongles are automatically checked and taken).
- No requirement for TUFLOW to control limiting of local licences (TUFLOW’s run key is not used if the number of CPUs/cores exceeds the local licence limit).

Two types of licences are available:

¹ Please contact support@tuflow.com if interested in running TUFLOW FV on Linux.

- Local or Standalone Licence – Allows up to a specified limit a number of TUFLOW simulations to be run from the one computer. For example a Local 1 licence will allow a single simulation to be performed on the machine. A local 4 licence allows up to 4 concurrent simulations and requires a quad core CPU to fully utilise. Local 1 to Local 16 licences can be configured.
- Network Licence – allows for multiple TUFLOW processes running at any one time across the organisation's LAN (Local Area Network) up to a specified limit. For example a Network 5 licence allows up to 5 concurrent simulations to be performed, this could be 5 simulations on a single computer, or a single simulation on 5 different computers or anything in between.

Refer to the [installation instructions](#) on the TUFLOW Wiki for further information on WIBU dongle setup.

2.3 Performing Simulations

TUFLOW FV simulations are started by running the TUFLOW FV executable and passing the input TUFLOW FV control file. There are a number of ways of initiating simulations:

- Running a batch file. Batch files can be set up to loop through events and scenarios to run a multitude of simulations or to push simulations to different processors.
- Directly from the text editor – ideal for one off simulations especially whilst constructing a model to test inputs.
- Via a GUI, such as the SMS TUFLOW FV Interface.
- Directly from a GIS.
- From a Command (Console) Window.

Details of each method are provided in the following sections and also are available on the [TUFLOW Wiki](#).

2.3.1 Batch File Example and Run Options (Switches)

One or many simulations can be specified within a batch file. The simplest format is to specify each simulation in series. The following shows the contents of a 4-line batch file (which could be named “TUFLOW FV Simulations.bat”):

```
"C:\TUFLOWFV\2019.01.009\TUFLOWFV.exe" run01.fvc  
"C:\TUFLOWFV\2019.01.009\TUFLOWFV.exe" run02.fvc  
"C:\TUFLOWFV\2019.01.009\TUFLOWFV.exe" run03.fvc  
Pause
```

The .bat file is run or opened by double clicking on it in Explorer. This opens a Console Window and then executes each line of the .bat file. The pause at the end stops the Console window from closing automatically after completion of the last simulation.

Note that the full path and executable is within double quotes; this is needed when there is a space in the command.

TUFLOW FV is a multi-threaded program based on the OpenMP shared-memory model. It will automatically spawn multi-threaded simulations. If not explicitly specified, the number of threads will equal either the limit of the computer’s capacity or the number of licences available. A user can control the number of threads used during a simulation using the batchfile command “OMP_NUM_THREADS”.

For example:

```
C:\>set OMP_NUM_THREADS=4  
"C:\TUFLOWFV\2019.01.009\TUFLOWFV.exe" run01.fvc  
"C:\TUFLOWFV\2019.01.009\TUFLOWFV.exe" run02.fvc  
"C:\TUFLOWFV\2019.01.009\TUFLOWFV.exe" run03.fvc  
Pause
```

2.3.2 From a Text Editor

The benefits of running TUFLOW FV from a text editor is that it provides a common environment where the control files can be edited, simulations started, and text file output be viewed. There is no need to close the .fvc file (or other control and output files) to run TUFLOW FV.

Setting up TUFLOW FV to run from UltraEdit and Notepad ++ is described in the TUFLOW FV wiki: http://fvwiki.tuflow.com/index.php?title=Running_TUFLOW_FV.

2.3.3 From the SMS Interface

Should a complete GUI that allows the user to create, manage and view models and model output within the one interface be desired, an interface for TUFLOW FV within SMS has been developed. At present the interface does not allow access to all the features of TUFLOW FV but can allow for a more efficient initial model development.

The TUFLOW FV wiki outlines the steps required to install the SMS interface:

http://fvwiki.tuflow.com/index.php?title=SMS_Tips#TUFLOW_FV_SMS_Interface

Tutorial Module 2 on the TUFLOW FV wiki steps through the process of developing and running a model using the interface:

http://fvwiki.tuflow.com/index.php?title=Tutorial_Module02

2.3.4 Directly from GIS

TUFLOW FV can be run directly from within QGIS via the [TUFLOW Viewer Plugin](#).

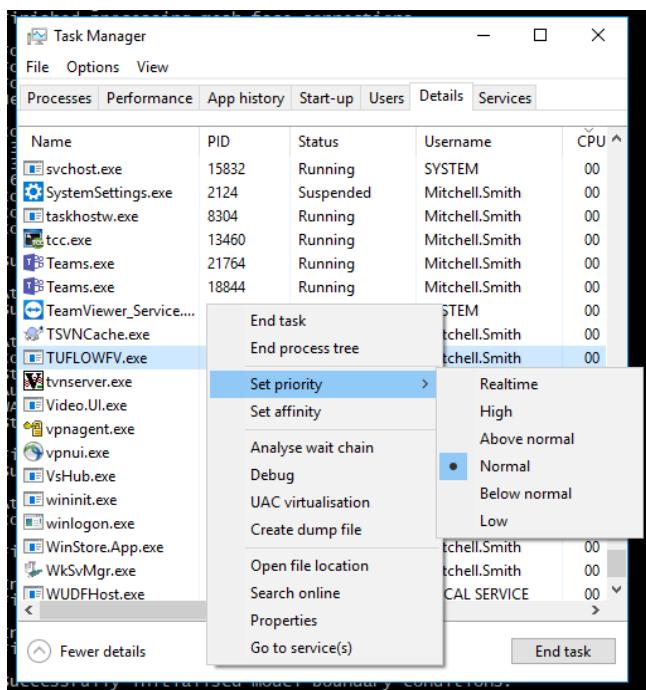
2.3.5 Change Priority, Pause, Restart or Cancel a Simulation

Windows can assign a process a different priority level using the Task Manager. This is very useful for running simulations in the “background” without slowing down other computational tasks being conducted .

To change the priority level of simulation manually:

- Open Task Manager (see your System Administrator if you’re not sure how to do this)
- Click on the details tab
- Find the TUFLOWFV.exe process you wish to change
- Right click, choose Set Priority, then the priority desired as shown in the image below

Note, don’t choose High or Realtime as this will cause TUFLOW FV to take over your CPU and you may not able to do much until the simulation is finished.



Simulation priority can also be specified when using a batchfile. Refer to the TUFLOW FV wiki for more details:http://fvwiki.tuflow.com/index.php?title=Running_TUFLOW_FV.

To pause a model simulation, highlight the console window and press “Ctrl-S”. This will temporarily halt the model simulation.

To continue again, press “Ctrl-Q”.

To cancel a simulation, “Ctrl-C”. This will include all result outputs up until the time of cancellation.

2.4 Licence Free Simulations

2.4.1 Tutorial Models

Tutorial models are available for download from www.tuflow.com. The purpose of the tutorial is to provide guidance on a number of the most commonly used TUFLOW FV features. The [tutorial documentation](#) is provided within the [TUFLOW FV Wiki](#). Documentation has been provided for developing models within SMS, with this to be expanded to include GIS-based environments.

There is no need to have a TUFLOW FV licence to simulate the tutorial model. Changes to the model's boundaries and other inputs are allowed so that the user can test and evaluate TUFLOW FV's various features. The command `Tutorial Model == ON` must occur within the .fvc file to simulate the model without needing a licence. An error will be generated if this command is included in any model other than the TUFLOW FV tutorial model available from the TUFLOW website.

The tutorial models are fully supported via the [TUFLOW support services](#), therefore, should you have any queries please don't hesitate to contact support@tuflow.com.

New users are recommended to complete these tutorial models to learn how to create a model mesh and setup a TUFLOW FV model.

Table 2-2 TUFLOW FV Tutorial/Demo Models

Tutorial Module	Description
Tutorial Module 1	Simple Trapezoidal channel. Build a mesh in your choice of SMS or GIS Mesher. Integrate basic GIS functionality and review results.
Tutorial Module 2	Meandering river. Build the mesh in your choice of SMS or GIS Mesher. Try the TUFLOW FV SMS interface.
Tutorial Module 3	Floodplain and estuarine example. Include structures and advection dispersion. This example provides you with a pre-constructed mesh.
Tutorial Module 4	Coastal example. Look at the application of an astronomical tide and model a tropical cyclone induced storm tide.
Tutorial Module 5	3D estuary example. Begin with a 2D model and then increase the complexity to a stratified 3D model. Optionally add water quality and assess the impact of a wastewater treatment outfall. Get an introduction to the TUFLOW FV MATLAB and TUFLOW FV Python 2D and 3D visualisation tools.
Mesh Generation Tips	A set of tips to consider when building and updating your project mesh.

2.5 Tips and Tricks

Useful tips and tricks for a variety of third party software used in the development and visualisation of TUFLOW models have been collated on the [TUFLOW FV Wiki](#). The guidance highlights how in-built features or user-developed functions can be used to enhance and streamline the modelling process. The Wiki is updated regularly. Please email any feedback to support@tuflow.com.

3 The Modelling Process

Chapter Contents

3.1	Introduction	3-2
3.2	Model Schematisation	3-3
3.2.1	Problem Definition	3-3
3.2.2	Model Limits (Space and Time)	3-4
3.2.3	Base Data Preparation <i>Bathymetry and Topography</i>	3-5
3.3	Mesh Construction	3-7
3.4	Assignment of Boundary Conditions	3-9
3.4.1	Open Boundaries	3-9
3.4.2	Water Surface Boundary	3-9
3.5	Initial Conditions	3-11
3.6	Simulation Times and Computational Timestep	3-12
3.7	Model Calibration and Validation	3-13

3.1 Introduction

This chapter discusses key considerations that should be addressed during the design phase of a hydraulic modelling study, prior to the commencement of model build. Consideration of how the model is to be schematised and selection of key model parameters may have a significant influence on the accuracy, stability and simulation time of the model, and may avoid redundant work later in the study.

3.2 Model Schematisation

3.2.1 Problem Definition

Define the problem(s) that the numerical modelling exercise will seek to solve and explain.

Defining a modelling exercise often starts with a preferred, highly rigorous and scientifically thorough approach that strives to replicate the physical system as accurately as possible. This is followed by a series of compromises and simplifications due to practical constraints. The final problem definition strikes a balance, providing a fit-for-purpose outcome. Key considerations include:

- **What is the model expected to deliver?**
 - The purpose of the modelling exercise should be clearly defined.
- **What are the key physical processes?**
 - A clear understanding of what processes need to be investigated will inform the type of model, what parameters and modules will be used, the extents and degree of accuracies required and, importantly, whether modelling is required at all!
 - An understanding of scale is important in this regard:
 - time scales (hours, months, years, decades, etc.)
 - spatial scales (global, regional, local, sub-grid, etc.)
- **What data is available?**
 - Successful application of a specific modelling approach can only be achieved if suitable data is available.
- **What are the time, economic and logistic constraints?**
 - Sophisticated and rigorous modelling studies can take up significant time and resources. Timing, economic and/or logistical constraints can limit the modelling exercise.
 - Computer power is a common constraint that can limit the temporal and spatial extent, resolution and accuracy of a modelling exercise.

3.2.2 Model Limits (Space and Time)

Define a model domain that best fits the key physical processes to be represented and achieves the required spatial and temporal scales within the constraints of available computational power.

The computational effort required to run a model simulation is a function of:

- The spatial extent of the model domain (i.e. the area to be modelled). This is typically guided by:
 - the spatial extent of the problem to be solved
 - the availability and locality of data with which to define boundary conditions
 - the spatial extent of the key physical processes to be represented
- The specified start and end time of the simulation which is typically guided by the temporal extent of the key physical processes to be represented. Examples include:
 - a flood assessment requires simulation of individual flood events with durations in the hours
 - a coastal or estuarine assessment, where tidal forces dominate, which may require the simulation of several tidal cycles over weeks, months or years
 - a morphological assessment which may require simulation periods of years or decades
- The model mesh geometry and the number of active, wet elements (or cells) in the model domain. For coastal, estuarine and flood assessments the number of wet elements may vary with time.
- The timestep, which varies throughout a simulation and is selected by taking into account physical and numerical convergence and stability considerations. The appropriate timestep is calculated by TUFLOW FV such that CFL constraints imposed by the flow characteristics are obeyed.
- The complexity of the processes being simulated. A simulation that includes scalar transport calculations such as tracers, sediments or water quality constituents will require additional computational effort compared to a hydrodynamics only simulation.

3.2.3 Base Data Preparation

Consolidate and prepare base data. This typically includes bathymetry / topography and also boundary condition information.

The base data required to develop a TUFLOW FV model will typically comprise of:

- Spatial datasets that define elevations (bathymetric and topographic) throughout the model domain.
- Timeseries datasets that define the open boundary conditions, such as a temporally varying water level (tidal) or inflow condition.

This information is normally easy to prepare, especially with pre-processing tools such as spreadsheets, SMS, GIS/3D surface modelling software and other numerical analysis packages (e.g. MATLAB) and scripting (e.g. Python). Quality checking of input data is a crucial component of any modelling exercise (yes, the often quoted “garbage in, garbage out” phrase cannot be left out of any modelling manual). Being proactive in quality assuring the input data can significantly reduce the possibilities of further issues.

Bathymetry and Topography

A good representation of the underlying bathymetry (elevations below the water surface in open channels, rivers, seas or oceans) is crucial for all hydrodynamic modelling exercises. For overland flooding assessments or for locations with a significant intertidal area (such as an estuary), an accurate representation of the topography is also required.

Bathymetric data is typically obtained via hydrographic surveys and/or nautical charts. These sources of data are generally restricted to areas of ship movements and recreational boating. In some instances, a hydrographic survey specific to the project may be available. In the absence of reliable hydrographic survey or nautical chart information, bathymetry estimated from satellite data may be available.

For flooding or coastal inundation, a description of the land topography is also required. This information is typically obtained via satellite radar or plane-mounted Laser Detection and Ranging (LIDAR or LADS) instruments although is increasingly being obtained from terrestrial instruments

In most modelling exercises an early step will be to develop a Digital Elevation Model (DEM) or Digital Terrain Model (DTM) of the study area using the available sources of bathymetry/topography data and GIS/3D surface mapping software. A DEM/DTM is a regular structured grid of elevation values. An example DEM/DTM constructed using MapInfo and Vertical Mapper software from a combination of hydrographic survey, LIDAR and digitised nautical chart data sources is shown in Figure 3-1.

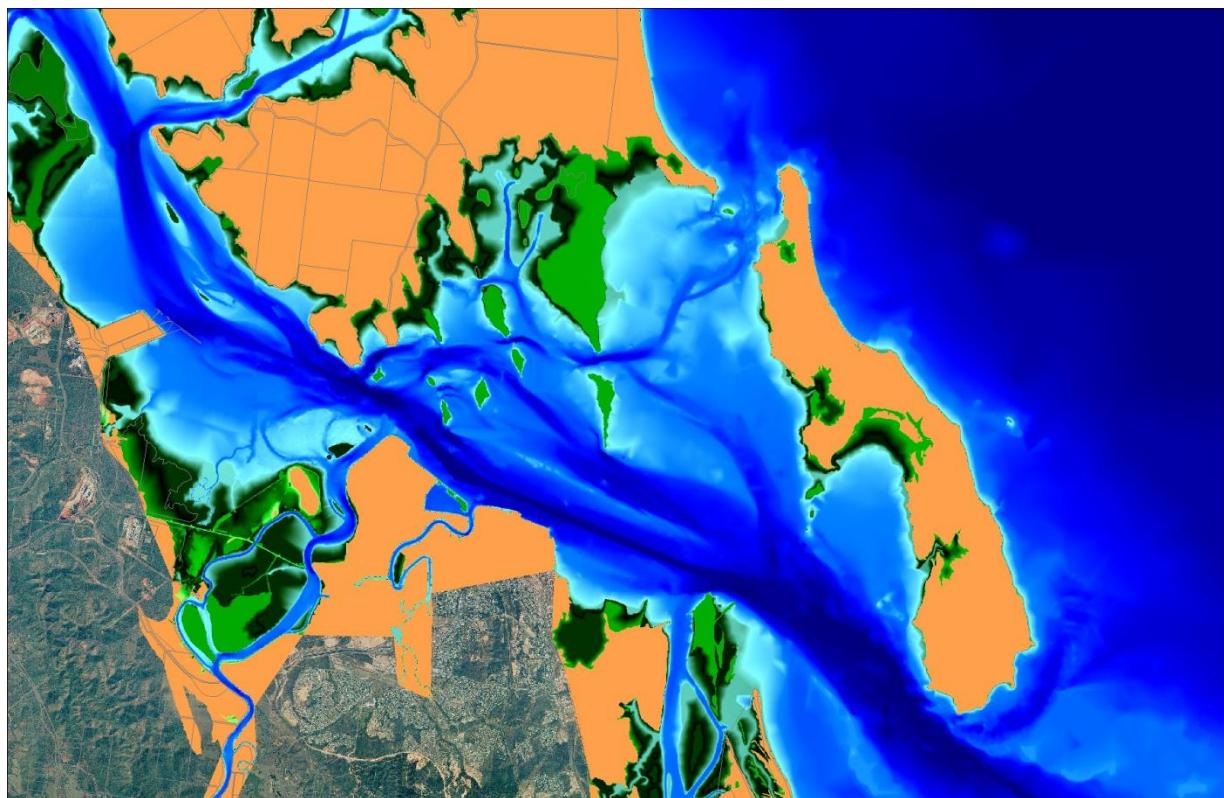


Figure 3-1 Digital Elevation Model of Port Curtis, Queensland, Australia

3.3 Mesh Construction

Using mesh generation software, create a suitable model mesh over which the hydrodynamic calculations are undertaken. Design a mesh that takes full advantage of the flexible mesh approach and also avoids common pitfalls and disadvantages.

TUFLOW FV solves the NLSWE on regular structured grids or unstructured meshes. Most TUFLOW FV users take advantage of the flexible mesh capability, with the model mesh comprising of triangular and quadrilateral elements. The flexible mesh approach allows for seamless boundary fitting along complex coastlines or open channels as well as accurately and efficiently representing complex bathymetries with a minimum number of computational elements. The flexible mesh capability is particularly efficient at resolving a range of resolutions within a single model without requiring multiple domain nesting.

Figure 3-2 shows a TUFLOW FV mesh and DEM of Port Curtis on the east coast of Australia (the DEM without the mesh is shown in Figure 3-1). This mesh was primarily developed to assess the impacts of a proposed shipping navigation channel expansion. Consequently, the mesh was constructed to neatly resolve the existing and proposed shipping channel geometry. Smaller mesh elements (higher mesh resolution) were necessary to resolve the complex tidal flows in the vicinity of the smaller islands and the harbour constriction. Larger mesh elements (lower mesh resolution) were used in regions located away from the areas of interest and/or where the flow varied more gradually, such as the shallow mud flats represented by the dark green areas in Figure 3-2.

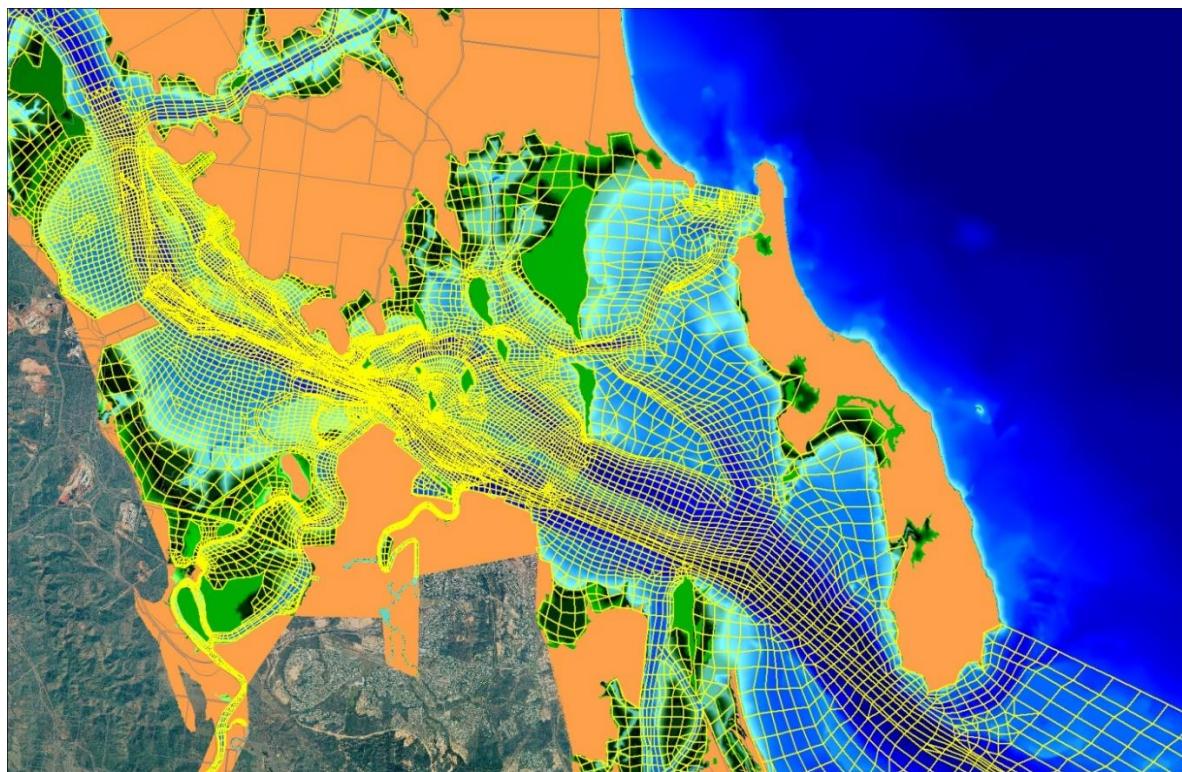


Figure 3-2 TUFLOW FV Mesh of Port Curtis Estuary, Queensland, Australia

Unstructured or flexible mesh geometries can be created using any suitable mesh generation tool. TUFLOW FV users generally use the [Aquaveo SMS Generic Mesh Module](#) or [Rising Water Software GIS Mesher](#) for building meshes as well as undertaking a range of model pre-processing and post-processing tasks. Both Cartesian and Spherical mesh geometries can be used as the basis for TUFLOW FV simulations. Mesh building/editing tutorials are available from the following sources:

- Included with a SMS installation
- Via the Aquaveo SMS website: <http://www.aquaveo.com/software/sms-learning-tutorials>
- Via the Tutorial models on the TUFLOW FV Wiki: <http://fvwiki.tuflow.com> and https://fvwiki.tuflow.com/index.php?title=Mesh_Generation_Tips

In addition to these web-based resources, Section 8.1 provides guidance on mesh generation and is particularly useful for new flexible mesh modellers. Section 8.1.1 describes the contents and required format of a TUFLOW FV mesh geometry file which follows the SMS Generic Mesh Module format. Mesh geometry files generated using an alternative mesh generation tool need to follow the format described in Section 8.1.1. This may require manual manipulation of the mesh geometry file contents.

3.4 Assignment of Boundary Conditions

The effects at the boundaries of a TUFLOW FV model determine the resulting fluid motion and hydrodynamic prediction. Understanding what is happening at the edges of the model domain is therefore critical. There are different types of boundaries to be considered when developing a TUFLOW FV model:

- The open boundaries at the “wet” edges of the model domain
- The closed boundaries at the seabed, open channel bed and water surface
- The boundary at the coastline, river bank or other wet/dry interface
- The initial condition at the start of the simulation

3.4.1 Open Boundaries

Open boundaries to the TUFLOW FV model domain should be located where there is some knowledge of the behaviour at that location. For a given period, this information may come from a tide station or other instrument deployed to continuously measure the variation in water level, a gauging station that provides a river discharge measurement, or may be output from larger-scale model. Some coastal models require additional forcing from ocean circulation models (e.g. HYCOM, <http://hycom.org/>) to accurately resolve density-driven flows.

Descriptions of the various open boundary conditions, their commands and associated inputs are provided in Chapter 10. TUFLOW support can be contacted for further information on applying boundary conditions derived from ocean circulation models: support@tuflow.com.

3.4.2 Water Surface Boundary

Boundary conditions can be applied to the water surface. These typically include wind, ambient pressure and/or wave fields. In many locations, or for particular events (such as a storm), these forcing mechanisms can have a significant influence on local hydrodynamics or scalar transport and may extend through the water column. Wind, atmospheric and wave boundary conditions are typically defined by measurements and/or output from other models. These conditions may be applied globally (i.e. constant throughout the model domain) or allowed to vary spatially for a given timestep.

Simulations that require spatially and temporally varied forcing typically rely on input data arranged on regular structured grids and stored in NetCDF format. For example, often SWAN (Simulating WAves Nearshore) wave model output or hindcast meteorological data from global models (e.g. NCEP/NCAR <http://www.ncep.noaa.gov/>) are utilised as inputs to TUFLOW FV simulations.

Spatially and temporally varying data accessed from online sources or generated using other models can be very large datasets (up to gigabytes in file size) and generally require some degree of processing prior to being used as input to a TUFLOW FV simulation. MATLAB or Python are typically used by BMT for preparing input data; however, other numerical analysis software packages with GRIB (a binary

format commonly used to store meteorological data) and NetCDF libraries could also be used. Examples of common TUFLOW FV NetCDF input files and the associated commands are provided in Appendix E.

3.5 Initial Conditions

All TUFLOW FV simulations start with an “initial condition”. Many hydrodynamic simulations will start with quiescent (still water) conditions and simply “warm-up” based on the boundary condition forcing. Under this scenario, the warm-up period should be long enough to allow any transients generated at the start of the simulation to propagate out of the model. Alternatively, the simulation initial condition can be defined manually by the modeller (and read from a .CSV file or a NetCDF) based on observed conditions.

In some situations the modeller may wish to set the initial condition using a TUFLOW FV restart file which contains the spatially varying conserved variables (at an instant in time) generated by a previous simulation. Further information on the initial condition and restart file options can be found in Section 10.7.

3.6 Simulation Times and Computational Timestep

Once the required input files have been prepared, model simulation performance should be tested.

The range of model performance tests undertaken by the modeller will depend on the type of TUFLOW FV application; nevertheless, there are a number of key checks to be completed prior to an initial simulation with a TUFLOW FV model:

- Once a mesh has been generated, the modeller needs to check for inconsistent element shapes or sizes that may unnecessarily constrain the model timestep. The accidental creation of a very small model cell (often a thin triangle) is a common issue and these “bad” cells must be removed from the mesh in order to achieve maximum model efficiency.
 - The Aquaevo XMS Wiki provides several mesh construction “rules” that will assist the user to create a clean mesh: http://www.xmswiki.com/xms/SMS:Mesh_Quality
- The mesh should accurately represent the bathymetry and topography, this can be checked by:
 - Specifying ‘ZB’ as a mapped output variable (either SMS Data File or NetCDF format, see Section 12.6) and running the TUFLOW FV model for short period. The ZB results represent the model bathymetry and should be compared to the base bathymetry and topography dataset.
 - Using the [Write Check Files](#) command will output a GIS layer of the mesh that includes core information such as cell ZB, material and element ID. This can be quickly read into GIS platforms for review. For more information on GIS Check Files please refer to Section 12.9.
 - By default, TUFLOW FV automatically outputs a number of model geometry files to the log directory (*_geo.nc and a series of .CSV files). The .CSV files provide a log of the model geometry and structure inputs and are an important check for flood applications. Using 3D surface modelling software, the user can create a TIN of the dataset and compare it against the base bathymetry/topography dataset and structure locations. *Note: This method is deprecated. It is now recommended to integrate TUFLOW FV models with GIS and use the Write Check Files command.*

The TUFLOW FV Wiki lists a number of ways to review a simulation timestep and also various ways to increase the efficiency of a model:

http://fvwiki.tuflow.com/index.php?title=A_Model_Runs_Slow

For more information on setting the and reviewing computational timestep commands please refer Section 6.8.1

3.7 Model Calibration and Validation

Calibrate the model to available observed data.

Verify the model to another set of independent observed data, preferably from a different location and/or a different time (with correspondingly different physical conditions).

Where knowledge or data is lacking, perform sensitivity tests on model parameters to quantify the uncertainty of model results.

Calibration is the process where the parameters of a model are adjusted, within reasonable bounds, so that simulation results match observed measurements. Validation is the process where a calibrated model is compared to observed measurements from a different period with different physical conditions. In combination, calibration and validation provide confidence that the model can replicate the physical processes and is a useful tool.

Choice of measurement periods for calibration depends upon the physical processes that need to be captured in the model. Typically, timeseries of response (for example river discharge, stage or tidal variations and current speed/direction) are more valuable for calibration purposes compared to instantaneous spot readings, however all relevant, reliable data should be absorbed into a calibration exercise.

As a minimum requirement for calibration and validation of a hydrodynamic tidal model, the following measurements are recommended:

- Calibration: A timeseries of water level, current speed and current direction at two or more locations, performed over a period that captures the tidal variation (e.g. over a spring-neap period)
- Validation: A timeseries of water level, current speed and current direction at two or more locations, over an independent period.

If seasonal variations are important, this exercise could be repeated at different times of year.

Example tidal calibration timeseries plots are shown in Figure 3-3 and Figure 3-4. The water level data was obtained from a permanent tide recording location. The current data was recorded using a fixed-location, bottom-mounted Acoustic Doppler Current Profiler (ADCP) instrument. TUFLOW FV point output has been obtained from these locations for direct comparison with the recorded data.

Flow (discharge) measurement across the entrance to an estuary or harbour is also a valuable model calibration dataset as shown in Figure 3-5. These measurements are typically obtained using a boat-mounted ADCP and performing continuous transects across the entrance or channel over a tide cycle. The model output corresponds to the flow through the location where the ADCP transect data was recorded. This type of output is obtained using the TUFLOW FV flux command.

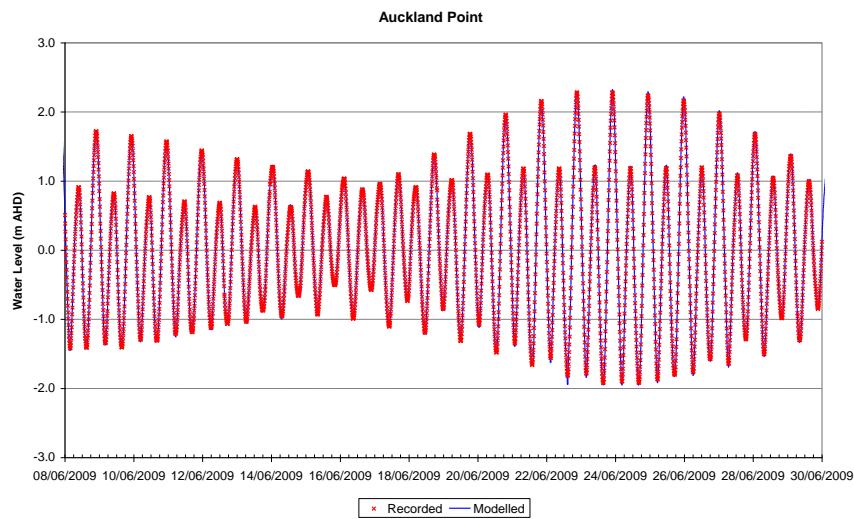


Figure 3-3 Example Tidal Water Level Timeseries Calibration Plot

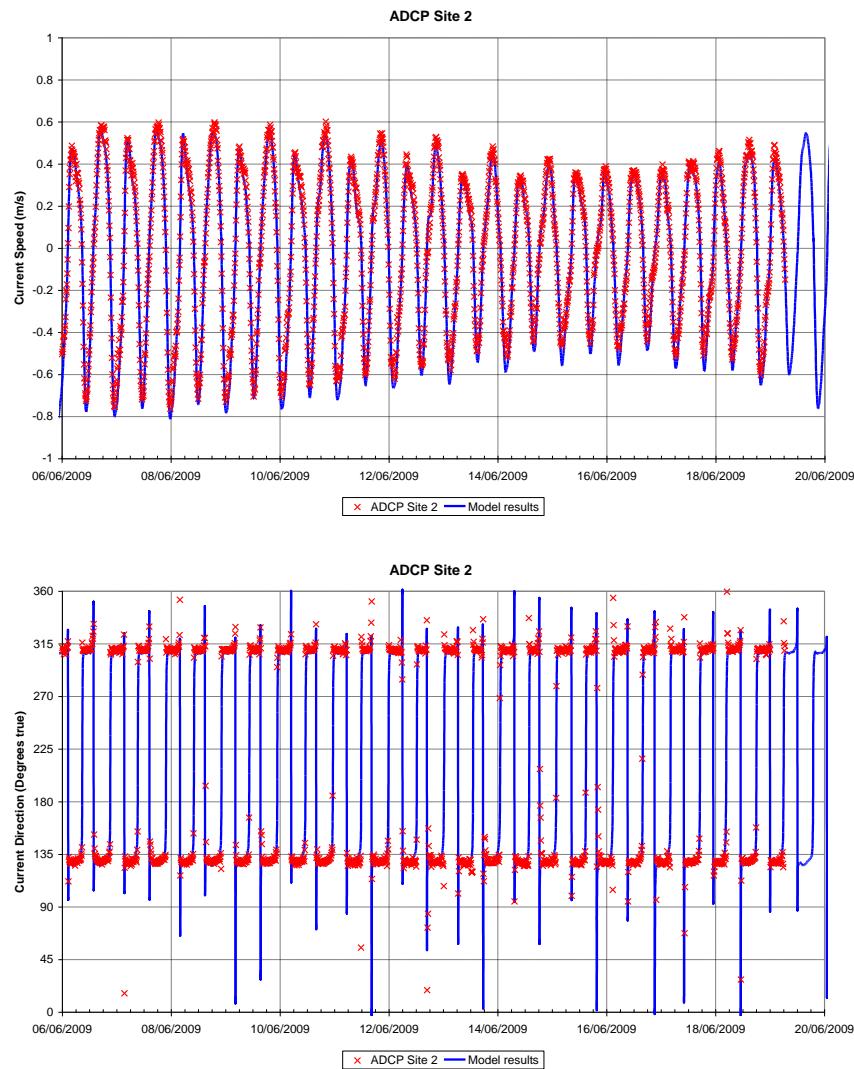


Figure 3-4 Example Current Speed and Current Direction Timeseries Calibration Plots for a Tidal Estuary

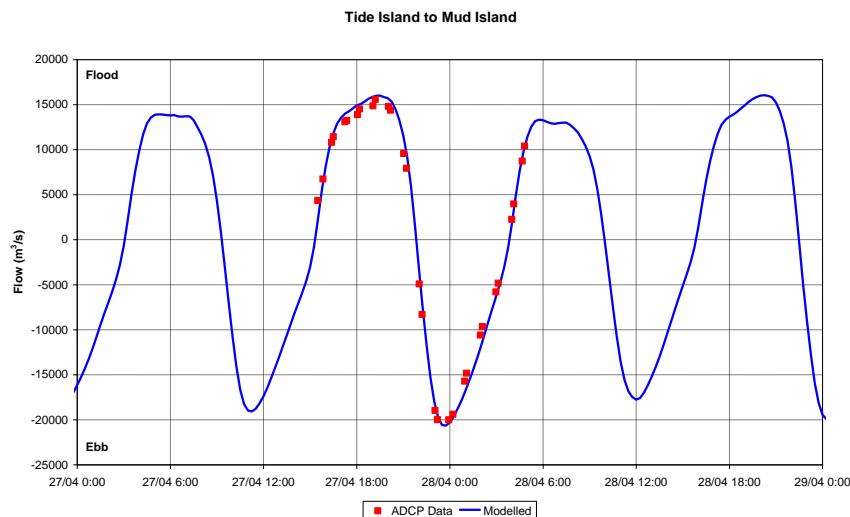


Figure 3-5 Example Flow Timeseries Calibration Plots for a Tidal Estuary

Overland flow calibration is less dependent upon instantaneous measurements performed at the time of the modelling study and more dependent upon historical records of floods. In these circumstances, all available information should be sought, quality checked and analysed, and used in the calibration exercise.

If a model cannot be calibrated due to a lack of data, don't despair; application of an uncalibrated model can still be useful. Be cautious with the model; interpret the results as indicators of specific trends and processes which, when combined with available data and experience, can provide worthwhile information.

4 Model Setup and Control Files

Chapter Contents

4.1	Introduction	4-2
4.2	Folders and File Types	4-3
4.2.1	Suggested Folder Structure	4-3
4.2.2	File Types and Naming Conventions	4-5
4.3	Simulation Control File	4-8
4.3.1	Control File Rules and Notation	4-8
4.3.2	Absolute and Relative File Paths	4-10
4.3.3	Include Files	4-10
4.4	Control File Layout	4-12

4.1 Introduction

This chapter of the Manual discusses the TUFLOW FV recommended folder structure and the differing types of model input and output files used by the model. Focus is provided on the TUFLOW FV Control File (.fvc extension) which is the master run file input to TUFLOW FV on execution. Syntax rules, naming conventions and suggestions for control file layout are explored in this chapter.

Section 4.3.3 provides a useful reference to guide you through the .fvc file structure and links to specific chapters of the Manual for further guidance.

4.2 Folders and File Types

4.2.1 Suggested Folder Structure

TUFLOW FV can be set up with a simple folder structure containing the relevant TUFLOW FV control files, the model geometry data and boundary data. For larger modelling jobs with many scenarios and simulations, a robust and complex folder structure may be warranted, but should be based on that below. Other sub-folders can of course be added by the modeller. For example, a “matlab”, “python” or “analysis” sub-folder to store project related pre and post-processing scripts may be desired.

The new folder names and structure differs slightly from that recommended in the previous manual and used in model builds pre-2019. This new configuration is designed to align with the folder structure used by TUFLOW FV’s sister products: TUFLOW Classic and TUFLOW HPC/QPC. It also allows for integration with the QGIS Plugin and GIS input files. You can find a copy of the recommended folder structure in the data download package of [Tutorial Model 01](#).

If running an older model there’s no requirement for you to modify your existing folder structure and we recommend leaving it ‘as is’. For new models we suggest that you adopt the scheme in Figure 4-1 and Table 4-1 as this is the new standard for TUFLOW FV model development. Where applicable the description column of Table 4-1 provides pre-2019 folder names.

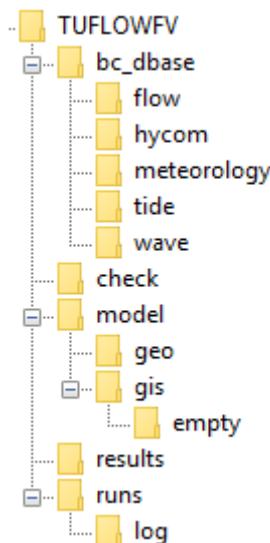


Figure 4-1 Example TUFLOW FV Sub-Folder Structure

Table 4-1 Recommended TUFLOW FV Sub-Folder Structure Description

Sub-Folder	Description
Locate folders below on the system network under a folder named “TUFLOWFV” in the project folder (e.g. J:\Project12345\tuflowfv) These folders should be backed up regularly	
bc_dbase	Boundary conditions, often with additional sub-folders for specific boundary condition types (e.g. tide, flow, meteorology, etc.). This folder was typically called ‘bc’ for model builds prior to 2019.
check	If the Write Check Files == command is included in the .fvc check files will be output from TUFLOW FV. These are a series of output files in both GIS (MapInfo or Shapefile format) and tabular data in .CSV format. These check files contain information on the data processed by TUFLOW FV.
exe	Optional sub-folder, placing the tuflowfv.exe (and associated dlls) within the TUFLOW FV folder structure may be desired. Alternatively, the tuflowfv.exe is located elsewhere on the network or local computer.
model	Include files and the geo and gis sub-folders
model\geo	Model mesh development often with additional sub-folders or links to locations where mesh development data is located if required. The ‘geo’ folder is as per model builds prior to 2019 but now resides as a sub-folder of ‘model’.
model\gis	GIS layers that are inputs to the 2D model domain.
model\gis\empty	Empty template GIS files.
runs	TUFLOW FV simulation control files. Batch files are also stored here when performing multiple simulations in a series. This folder was typically called ‘input’ for model builds prior to 2019.
runs\log	Location for automatically generated simulation log and model performance files.
results	The directory where specified model output is written. Often placed on a local drive rather than a network drive. This folder was typically called ‘output’ for model builds prior to 2019.

4.2.2 File Types and Naming Conventions

Files are generally classified as:

- Control Files
- Data Input Files
- Model Output Files
- Check Files

Control files are used for directing inputs to the simulation and setting parameters. The style of input is very simple, free form text commands, like writing down a series of instructions. This offers the most flexible and efficient system for experienced modellers. It is also easy for inexperienced users to learn.

Data input files are primarily comma-delimited files prepared using spreadsheet software. Simulations that require spatially and temporally varied forcing (e.g. wind fields or short-wave forcing) typically rely on NetCDF format data input. Common examples of NetCDF input file formats used by TUFLOW FV are provided in Appendix E. In some instances, the model initial condition may be defined by spatially and vertically (for 3D or layered sediments) varying outputs from a previous simulation, referred to as a TUFLOW FV restart file.

Data output files are primarily map output in SMS format (2D or vertically averaged 3D), map output in NetCDF format (2D, vertically averaged 3D or full 3D) and time-series output in comma-delimited format. The TUFLOW FV NetCDF output file structure is summarised in Appendix C. In addition to the above, a range of check files are produced in text and comma-delimited formats to carry out quality control and model efficiency checks.

The most common file types and their extensions are listed in Table 4-2.

Table 4-2 TUFLOW FV File Formats

File	Extension	Description	Format
Control Files			
TUFLOW FV Simulation Control File	.fvc	Controls the data input and output for a 2D or 3D simulation. Mandatory for 2D and 3D. Also used to include other files.	Text
TUFLOW FV Sediment Control File	.fvsed	Control sediment fraction numbers, sediment layers, erosion, deposition and bed load models. For more information on the sediment transport module please contact support@tuflow.com.	Text
TUFLOW FV Particle Tracking Control File	.fvptm	Control particle fractions, bed layering, deposition, motility, and erosion characteristics. For more information on the particle tracking module please contact support@tuflow.com.	Text
Data Input			

File	Extension	Description	Format
Mesh Geometry File	.2dm	A file containing the 2D geometry of the model mesh and elevations. It also contains information on the material types used to define areas with a specified bed roughness and the location of open boundaries. The structure of the .2dm file follows the SMS Generic Mesh Module structure. Mandatory for 2D and 3D.	Text
3D Model Vertical Mesh File	.CSV	A file containing the z-coordinates of the vertical mesh. Mandatory for 3D using z-coordinate or z-sigma coordinate discretisation.	Text
GIS Files	.shp or .mif	Mapinfo MIF or ESRI Shape Files containing vector point, polyline or polygon data. Used to assign the location and attributes of a range of model inputs including nodestring, materials and topographic/bathymetry datasets.	GIS
GIS Grid Files	.asc or .flt	ESRI ASCII grid (.asc) or binary grid (.flt) formats. Typically used to assign bathymetry or topography and can be read by the Read GRID Zpts command.	GIS
Comma Delimited Files	.CSV	These files are used for temporally varying boundary condition input, such as a tidally varying water level or inflow condition. They can be opened and saved using a text editor or spreadsheet software such as Microsoft Excel.	Text
NetCDF File	.nc	These files are typically used to store data inputs that vary spatially and temporally. These inputs are often derived from outputs from other models and may include wind fields, atmospheric conditions, short-wave forcing or ocean current forcing.	NetCDF
Restart File	.rst	These files are generated by TUFLOW FV and contain the spatially varying conserved variables at an instant in time. Restart files are optionally used to define the initial condition of a TUFLOW FV simulation.	Binary
Model Output			
Comma Delimited Files	.CSV	These files are used for time-series data output (2D and vertically averaged 3D). They are typically opened and viewed using numerical analysis software (e.g. spreadsheet software such as Microsoft Excel).	Text

File	Extension	Description	Format
SMS Data File	.dat, .xmdf, .sup	SMS generic formatted simulation output file. TUFLOW FV map output can be written in the SMS .dat or .xmdf format (2D and vertically averaged 3D). The .sup file is used to combine the input mesh and output results and is read by SMS or the TUFLOW Viewer Plugin for QGIS.	Binary
NetCDF File	.nc	NetCDF formatted simulation output file. TUFLOW FMTUFLOW FV map output can be written in the NetCDF .nc format (2D, vertically averaged 3D and full 3D).	NetCDF
Restart File	.rst	Spatially varying conserved variables at an instant in time for restarting TUFLOW FV simulations.	Binary
GIS Check Files	.mif, .shp	Point, polyline or region GIS check files that reproduce information on the location and attributes of model input datasets. Very useful for model debugging.	Text (.mif), Binary (.shp)

4.3 Simulation Control File

4.3.1 Control File Rules and Notation

The TUFLOW FV simulation control file (.fvc) is simply a command or keyword driven text file. The commands are entered free form based on the rules described below. Comments may be entered at any line or after a command. The typical simulation control file structure is shown in Figure 4-4 and the common data inputs are described in this Chapter. Several simulation control file examples can be found on the TUFLOW FV Wiki: <http://fvwiki.tuflow.com/>.

Control File Rules

1. Only one command can occur on a single line.
2. A “==” following a command indicates the start of the parameter(s) for the command. For example: `Geometry 2D == ..\model\MyMesh.2dm`
3. “#” or “!” represent comment syntax. All text after the comment command from that point onward will be ignored. This is useful for “commenting-out” unwanted commands, and for including modelling documentation within the control file. Comments may be entered at any line or after a command. For example:

```
! This is the model mesh.  
  
Geometry 2D == ..\model\MyMesh.2dm
```

OR:

```
Geometry 2D == ..\model\MyMesh.2dm ! This is the model mesh.
```

Are both common commenting methods in TUFLOW FV.

Control files are command or keyword driven text files. The commands are entered free form, based on the rules described below. Comments may be entered at any line or after a command. The commands are listed in the index in [Appendix A](#) and [Appendix B](#)

Note: TUFLOW FV control file are NOT case sensitive on Windows Operating Systems. On Linux commands to the left of the == are not case sensitive, however file paths and names to the right of == are case sensitive, consistent with Linux file and folder naming convention.

An example of a command is:

```
End Time == 10. ! Simulation ends at 10:00am on 2/9/1962
```

This command sets the simulation end time to 10 hours. The text to the right of the “!” is treated as a comment and not used by TUFLOW when interpreting the line.

Automatic colour coding of files for easy viewing is available for the following text editors: Ultra Edit, Notepad++ and TextPad. Instructions outlining how this software can be configured can be found on the [TUFLOW Wiki](#).

Commands can be repeated as often as needed. This offers significant flexibility and effectiveness when modelling, particularly in building 2D model topography. Note that a repeat occurrence of a command may overwrite the effect of previous occurrences of the same command.

The style of input is flexible bar a few rules. The rules are:

- A few characters are reserved for special purposes as described in Table 4-3;
- Command syntax is not case sensitive;
- Only one command can occur on a single line;
- A few commands rely on another command being previously specified. These are documented where appropriate.

Unlike TUFLOW Classic or TUFLOW HPC/QPC additional text cannot be placed before and/or after a command. For example, a line containing the command Start Time to set the start time of a simulation to 10 hours can only be written as:

```
Start Time == 10    not      Start Time (h) --- 10
```

To indicate the units is it best to make use of comments i.e. `Start Time == 10 ! hours` would be acceptable, noting the use of the comment delimiter “!”.

Blank lines are ignored. Spaces or indentations can occur at the start of the line. This is recommended when using the logic control or material, structure, boundary or output blocks. The second and third lines in the following example is not required to be indented, however, it is recommended for ease of reviewing the model build:

```
output == NetCDF
  output parameters == h,v,d, temp, sal, Rhow
  output interval == 3600.
End If
```

The notation used to document commands and valid parameter values in [Appendix A](#) to [Appendix B](#) are presented in Table 4-4.

Table 4-3 Reserved Characters – Text Files

Reserved Character(s)	Description
“#” or “!”	A “#” or “!” causes the rest of the line from that point on to be ignored. Useful for “commenting-out” unwanted commands, and for all that modelling documentation.
“==”	A “==” following a command indicates the start of the parameter(s) for the command. Where there is more than one parameter, the parameter values are read as free-field formatted, i.e. are space or comma delimited.

Table 4-4 Notation Used in Command Documentation – Text Files

Notation	Description
< ... >	Greater than and less than symbols are used to indicate a variable parameter. For example, the commonly used <file> example is described below.
<file>	A filename (can include an absolute or relative path, or a UNC path). See Section 4.3 for a more detailed description.
spaces	Spaces can occur in commands and parameter options. If a space occurs in a command, it is only one (1) space, not two or more spaces in succession. Spaces can occur in file and path names, however, third party software may not allow this and as such is not recommended. If using spaces in filenames, batch files will require that the filename is enclosed in quotes.

4.3.2 Absolute and Relative File Paths

TUFLOW FV control files reference additional files, for example GIS files, which provide the model geometry or boundary data. The three methods that can be used are absolute file path, relative file path and UNC file path referencing. A model can use any or all of these methods. However, relative file paths are typically preferred as then the model can be provided to another modeller without changing path names. Take an example of reading a GIS layer containing initial water levels, the command is Read GIS Mat == <filepath>. The following are all valid:

Absolute file path: `Read GIS Mat == L:\Job\Job1234\TUFLOWFV\model\gis\2d_mat_001_R.shp`

Relative file path: `Read GIS Mat == ..\model\gis\2d_mat_001_R.shp`

UNC file path: `Read GIS Mat == \\server1\Job1234\TUFLOWFV\model\gis\2d_mat_001_R.shp`

For the relative file path in TUFLOW FV the path is relative to the location of the fvc file. In the case above, if the command occurred in the .fvc file which is located in the TUFLOWFV\runs\ folder. The ..\ indicates to go up one level (from TUFLOWFV\runs\ to TUFLOWFV\)\ the model\ navigates into the TUFLOWFV\model\ folder, gis\ navigates from TUFLOWFV\model\ into TUFLOWFV\model\gis\). To go up more than one level simply use ..\ multiple times (e.g. ..\..\ would navigate up two folders). If the file sits under the same folder then the filename can be specified.

It is recommended where possible to use relative files paths. This makes it easy to move and package models e.g. if client provision is required or you need flexibility to move models from one computer to another.

4.3.3 Include Files

To assist in keeping the .fvc succinct and readable it is often desirable (but optional) to use TUFLOW FV's '[Include](#)' file functionality. The [Include](#) file allows you to use multiple input .fvc files to minimise repetitive specification of commands common to a group of files. For example, material specifications, boundary conditions or model outputs. An example of using the [Include](#) command is provided in Figure 4-2. Here two include files are used, materials.fvc to specify material specific commands and bc.fvc which contains boundary condition information. Figure 4-3 shows the contents of materials.fvc.

```
!-----  
! GEOMETRY  
geometry 2d == ..\model\geo\Mesh_Simplify_Roads_001.2dm  
Read GRID Zpts == ..\model\grid\DEM_SI_Unit_01.asc  
  
!-----  
! MATERIAL PROPERTIES  
include == materials.fvc  
  
!-----  
! INITIAL CONDITIONS  
initial water level == 40.0  
  
!-----  
! BOUNDARY CONDITIONS  
include == ..\bc_dbase\bc.fvc
```

Figure 4-2 Example TUFLOW FV Control File using Include command

```
!-----  
! MATERIAL PROPERTIES INCLUDE FILE  
Set Mat == 11  
Read GIS MAT == ..\model\gis\2d_mat_M01_002_R.shp  
Read GIS MAT == ..\model\gis\2d_mat_M01_003_R.shp  
  
material == 1  
    bottom roughness == 0.04  
end material  
  
material == 2  
    bottom roughness == 0.02  
end material  
  
material == 3  
    bottom roughness == 0.1  
end material  
!  
material == 4  
    bottom roughness == 0.03  
end material  
  
material == 10  
    bottom roughness == 0.08  
end material  
  
material == 11  
    bottom roughness == 0.04  
end material
```

Figure 4-3 Contents of Materials.fvc called via Include command

4.4 Control File Layout

A TUFLOW FV simulation control file (.fvc) is the text file run by TUFLOW FV and defines how a simulation will be executed. The format that you use to create this file is flexible (although some rules do exist as outlined in Section 4.3.1) and may vary depending on your personal or organisations preferences. As you get more experienced with TUFLOW FV it's unlikely that you will start the control file from scratch and will typically use a template to begin your model development. Some of these templates are provided in our [Tutorial Model datasets](#). A suggested order for command categories is summarised in Table 4-5 which also links you to the relevant Chapters that are specific to each command category.

Table 4-5 TUFLOW FV Simulation Control File Layout

Command Category	Description	Chapter
GIS Setup	GIS format, spatial reference assignment, empty file creation	Chapter 5
Simulation Configuration	Solution include statements, spatial order selection, units selection, coordinate type, bed roughness model.	Chapter 6
Time and Timestep	Time format and reference time, model start and end times, CFL limits, timestep limits.	Chapter 6
Model Parameters	Wetting / drying, turbulence related commands, stability limits, reference values.	Chapter 7
Geometry	Mesh file, topographic updates, external nodestring locations.	Chapter 8
Materials	Material block properties (roughness, mixing parameters, etc.), spatial definition of materials (GIS Layers)	Chapter 8
3D Geometry	3D geometry definitions, layer types and discretisation	Chapter 9
Initial Conditions	Initial model state (initial parameters, restart files)	Chapter 10
Boundary Conditions	Global (winds, waves, rainfall, etc.), nodestring (external boundaries, water levels, flows, etc.), cell (source) and node (point source)	Chapter 10
Hydraulic Structures	Weirs, culverts, bridges, porous and wall structures, operational structures and controls	Chapter 11
Model Results and Output	Output directory, map output formats and data types check files, time series and profile outputs, logging and diagnostics.	Chapter 12

An example fvc file is shown in Figure 4-4. New users wishing to learn how to create a simulation control file are advised to complete the TUFLOW FV tutorial models which are available via the TUFLOW FV Wiki (<http://fvwiki.tuflow.com>) and are described further in Section 2.4.1.

```
! Example FVC Layout
!
! GIS INITIALISATION
GIS FORMAT == SHP
SHP Projection == ..\model\gis\projection.prj
!Write Empty GIS Files == ..\model\gis\empty
!
! TIME COMMANDS
Start time == 0.0
End Time == 6.0
cfl == 1.0
Timestep Limits == 0.0001,100
!
! MODEL PARAMETERS
stability limits == 10.0,100.0
Momentum mixing model == Smagorinsky
Global horizontal eddy viscosity == 0.2
!
! GEOMETRY
geometry 2d == ..\model\geo\trap_steady_05.2dm
Read GIS Nodestring == ..\model\gis\2d_ns_T01_Boundary_001_L.shp
!
! MATERIAL PROPERTIES
material == 1
    bottom roughness == 0.018
end material
!
! INITIAL CONDITIONS
initial water level == -3.5
!
! BOUNDARY CONDITIONS
bc == Q, 1, ..\bc_dbbase\steadyQ_01.csv
    bc header == time,flow
    Sub-type == 4
end bc
bc == WL, 2, ..\bc_dbbase\steadyWL_01.csv
    bc header == time,WL
end bc
!
! OUTPUT COMMANDS
Output Dir == ..\results\
output == xmdf
    output parameters == h,v,d
    output interval == 600.
end output

Output == Flux
    Output interval == 600
End Output

Write Check Files == ..\check
```

Figure 4-4 Example TUFLOW FV Simulation Control File

5 GIS Setup

Chapter Contents

5.1	Introduction	5-2
5.2	“GIS” Commands	5-3
5.2.1	Naming Conventions	5-4
5.3	GIS Initialisation	5-6
5.3.1	Define the Project Projection	5-6
5.3.2	Setting the GIS Output Format	5-6
5.3.3	Creating ‘Empty’ GIS Template Files	5-7
5.3.4	TUFLOW Viewer QGIS Plugin	5-8
5.4	TUFLOW FV GIS Integration Tutorials	5-9

5.1 Introduction

As of TUFLOW FV Release 2019.01 you can use GIS layers to assist with model development. GIS data layers are transferred into and out of TUFLOW using the MapInfo data exchange .mif or ArcGIS .shp format. These formats are industry standards and publicly available. The .mif format is in text (ASCII) form, making it particularly easy to work with. Most mainstream CAD/GIS platforms recognise these formats.

The format of input layers is solely controlled by the file extension (i.e. .mif for the MIF format and .shp for the SHP format). TUFLOW FV requires that all GIS layers imported or exported by TUFLOW FV must be in the **same geographic projection**. The model projection is initialised using the [MI Projection](#) and/or [SHP Projection](#) commands. If a model has a mixture of .mif and .shp files as input, then both [MI Projection](#) and [SHP Projection](#) should be specified.

The default output format for GIS check layers and GIS outputs is the .mif format. To produce check and output GIS layers as .shp files, specify `GIS Format == SHP` in the .fvc file (refer [GIS Format](#)).

Please note that the use of GIS Integration is optional and not a requirement to run TUFLOW FV. If building new models (or modifying old ones) we do encourage you to use these features, particularly if using the pre-2019 CSV file formats for model geometry assignment such as the Nodestring Polyline File, Cell Elevation File, Cell Elevation Polyline File etc.

5.2 “GIS” Commands

Commands containing “GIS” read and/or write a GIS layer. GIS read both the .mif and .mid files (if a .mif extension is specified), and the .shp and .dbf files (if .shp is specified). The format of the GIS layer is based on the filename’s extension (.mif or .shp).

To appreciate how TUFLOW FV interprets GIS data it is important to understand the following.

- .mif or .shp files contain the geometrical (map) data about the objects.
- .mid or .dbf files contain the attribute data of the objects.

The geographic location of objects for GIS commands **is important**. The geographic position of the object controls which part of the TUFLOW model they affect. By default TUFLOW, checks that the projection of each .mif file matches that specified by the [MI Projection](#) command and the projection of each shapefile matches that specified by the [SHP Projection](#) command. A MapInfo GIS layer must have a projection specified, whilst a Shapefile layer does not need to have a projection defined. Regardless of which GIS software used, it is strongly recommended that each file has a defined projection. Table 5-1 defines the different data objects supported. If an unsupported GIS object is identified on input TUFLOW FV will throw the following warning to the console and log file: *WARNING 3023 Ignored Object in GIS Layer*.

Table 5-1 TUFLOW Interpretation of GIS Objects

Object Type	TUFLOW Interpretation
Used Objects	
Point	Refers to the 2D cell that the point falls within or a 1D node. Points snapped to the sides or corners of a 2D cell may give uncertain outcomes as to which cell the point refers to.
Line (straight line)	Continuous line of 2D cells, 1D channel, connects other objects, alignment of a 3D breakline and other applications. Note, the algorithm for selecting the 2D cells has varied with different TUFLOW builds. See Line Cell Selection .
Pline (line with one or more segments)	As for Line above.
Region (polygon)	For 2D cells either: <ul style="list-style-type: none"> • Modifies any 2D cell or cell mid-side/corner (e.g. Zpt) that falls within the region. If the command is modifying a whole 2D cell, it uses the cell’s centre to determine whether the cell falls inside or outside of the region. If the cell’s centre, mid-side or corner lies exactly on the region perimeter, uncertain outcomes may occur. Holes within a region are accepted except for polygon objects in shape layers used for TIN boundaries.

Object Type	TUFLOW Interpretation
	<ul style="list-style-type: none"> • Or, just uses the region's centroid. Examples are the original flow constriction layers (2d_fc) and time-series output locations (2d_po). <p>For 1D:</p> <p>1D nodes within the region are selected. If the 1D node falls exactly on the region perimeter uncertain outcomes may occur.</p>
Multiple (Combined) Objects	In later versions of TUFLOW, multiple point, polyline and region objects are generally accepted (ERROR or WARNING messages are given if not the case).
Unused (Ignored) Objects	
Arc	Ignored (do not use).
Collections	Not supported. Collections are groups of objects of differing type.
Ellipse	Ignored (do not use).
none	These objects are ignored and most commonly occur when a line of attribute data is added that is not associated with an object. In MapInfo, this occurs when a line of data is added directly to a Browser Window (i.e. no object was digitised).
Roundrect (Rounded Rectangle)	Ignored (do not use).
Rect (Rectangle)	Ignored (do not use).
Text	Ignored.

Object snapping is often used to relate point data with line and region data, for example with the [Read GIS Z Line](#) command. TUFLOW FV supports point, end and vertex snapping. It does not support edge snapping.

5.2.1 Naming Conventions

As the bulk of the data input is via GIS and text file data layers, efficient management of these datasets is essential. For detailed modelling investigations, the number of data layers has been known to reach over a hundred for large complex models, although the majority of models would utilise five to twenty layers. Good data management also caters for the many other GIS layers (aerial photos, cadastre, etc.) being used.

Different TUFLOW FV GIS input files require different attributes, for example the 2d_mat material input file only requires a single attribute (with the attribute – Mat), whereas a topographic breakline has a number of attributes. Each of these file types is described in Table 5-2. **It is strongly recommended** that the prefixes described in Table 5-2 be adhered to for all 2D GIS layers. This greatly enhances the data management efficiency and, importantly, makes it much easier for another modeller or reviewer to quickly interpret the model. The .fvc command [Write Empty GIS Files](#) can be used to automate the creation of template files which use the recommended GIS data naming convention.

Data input is structured so that there is no limit on the number of data sources. Commands are repeated indefinitely in the text files to build a model from a variety of sources. For example, a model's materials may be built from more than one source. A DTM may be used to define the general topography, while several 3D elevation lines (breaklines) define the crests of levees or roads. The sequential approach to reading datasets offers unlimited flexibility and increased efficiency particularly when representing different scenarios. This layered approach also offers good traceability and quality control.

Table 5-2 GIS Input Data Layers and Recommended Prefixes

GIS Data Type	Suggested File Prefix	Description	Refer to Section
2D Land-Use (Materials) Categories	2d_mat_	Layers to define or change the land-use (material) types on a cell-by-cell basis.	8.4
Elevation Lines (Breaklines) (Ridges and Gullies)	2d_zln_	Optional 3D breaklines defining the crest of ridges (e.g. levees, embankments, sandbanks) or thalweg of channels (eg. drains, creeks, small estuaries). These files are typically created by renaming the 2d_z_empty file.	8.3.3
Nodestrings	2d_ns_	Optionally input open boundary or internal flux lines, or structure links using these GIS polyline layers.	8.2

5.3 GIS Initialisation

5.3.1 Define the Project Projection

When using GIS integration TUFLOW FV requires that all input and output files are in the same spatial reference system. This can be either a cartesian system such as a UTM projection when using `Spherical == 0` (the default) or a longitude latitude system if using `Spherical == 1` (refer [Spherical](#)).

The .fvc commands [MI Projection](#) and [SHP Projection](#) are used for model inputs and outputs that are specified in MapInfo MIF layers or ESRI Shapefile format respectively. If a model has a mixture of .mif and .shp files as input, then both [MI Projection](#) and [SHP Projection](#) are required.

TUFLOW FV can extract the projection information from any mif or shp file layer. Typically, at project start up an empty ‘dummy’ mif or shp GIS layer is created that contains the projection information. The projection file can be created manually or can be assisted through use of the [TUFLOW Viewer Plugin for QGIS](#).

Examples:

```
MI Projection == ..\model\mi\Model_Projection.mif  
SHP Projection == ..\model\shp\Projection.prj
```

As an alternative to specifying a .mif or .prj file, the coordinate reference string can be copied out of the *.mif or *.prj file and read directly from the .fvc. For more information please consult the [MI Projection](#) and [SHP Projection](#) Appendix sections.

Note: All MID/MIF/SHP GIS layers read by TUFLOW FV MUST USE this projection or the following error message is produced:

ERROR 0305 - Projection of .shp file is different to that specified by the SHP Projection == command.

If you are confident that the projection you are using are in fact the same, then you can change this error to a warning via the command: `GIS Projection Check == WARNING`.

5.3.2 Setting the GIS Output Format

The [GIS Format](#) command sets the format of GIS outputs from TUFLOW FV. If not specified, the default of using .mif files is adopted.

```
GIS Format == SHP      ! Recommended for ArcMap or QGIS Users  
GIS Format == MIF      ! Recommended for MapInfo Users
```

Note: that the GIS format of an input layer is solely controlled by the file extension (i.e. .mif for the MIF format and .shp for the SHP format). GIS file extensions should be used in filenames and TUFLOW FV control files.

5.3.3 Creating ‘Empty’ GIS Template Files

Different TUFLOW FV input files require specific input layer dependant GIS attributes. For example, material input files only require a single attribute (with the attribute – Mat) whereas a boundary condition may have several attributes (nodestring ID, type, sub-type etc.). **It is strongly recommended** that the prefixes described in [the table below](#) be adhered to for all GIS layers. This greatly enhances the data management efficiency and, importantly, makes it much easier for another modeller or reviewer to quickly interpret the model.

GIS Data Type	Suggested File Prefix	Description	Relevant Section
2D Land-Use (Materials) Categories	2d_mat_	Optional layers to define or change the land-use (material) types on a cell-by-cell basis.	8.4
Elevation Lines (Breaklines) (Ridges and Gullies)	2d_zln_	Optional 2D or 3D breaklines defining the crest of ridges (e.g. levees, embankments) or thalweg of gullies (eg. drains, creeks). These files are typically created by renaming the 2d_z_empty file.	8.3.3
Nodestrings	2d_ns_	Optional polyline layer that can be used to define the location, direction and nodestring type of external nodestrings.	8.2

The .fvc command [Write Empty GIS Files](#) can be used to automate the creation of template (empty) files which use the recommended GIS data naming convention and contain the correct GIS attributes. The command requires a location where you would like empty files written. The TUFLOW FV convention is the folder *model\gis\empty* as shown below.

```
Write Empty GIS Files == ..\model\gis\empty
```

You only need to write the empty files once during model setup. The TUFLOW FV simulation will stop after writing the empty files which is the same behaviour implemented in TUFLOW Classic. Once they are created you can comment the command out (or delete) and continue to develop your .fvc file.

```
!Write Empty GIS Files == ..\model\gis\empty
```

The [TUFLOW Viewer Plugin for QGIS](#) can be used to assist with creating these empty files during model setup.

5.3.4 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:

- Creation of projection file.
- Automated methods to create the TUFLOW FV folder directory, generate empty files and set the GIS Format.
- 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.

The TUFLOW Viewer Plugin is available as a free download via the QGIS Plugin Manager .

5.4 TUFLOW FV GIS Integration Tutorials

For a step by step guide on the GIS initialisation process please first consult [Tutorial Model 1](#) and then more advanced GIS integration features are provided in [Tutorial Model 3](#) on the [TUFLOW FV Wiki](#).

5.5 Layering of GIS Datasets

The .2dm mesh forms the base layer for model development in TUFLOW FV. However, rather than contain all the bathymetric information in this one file, TUFLOW FV has a series of commands that assist with **building** the model. The commands are applied in sequential order; therefore, it is possible to override previous information with new data to modify the model in selected areas. This is very useful where a base dataset exists, over which areas need to be modified to represent other scenarios such as a proposed development. This eliminates or minimises data duplication and provides better traceability of input layers / datasets.

The commands can occur in any order (if it is a logical one). If an unrecognisable command occurs, TUFLOW FV stops and displays the unrecognisable text.

Notes & Tips:

- 1 Elevations and materials are always read from the .2dm file first, regardless of the order of where the `Read Geometry 2d ==` command is in the .fvc. There can currently only be one mesh file (.2dm) per model.
- 2 Commands can be repeated any number of times.
- 3 Commands are executed in the order they occur. If the data for a cell is supplied more than once, the last data that is read in the fvc is used (i.e. the latter data for a cell overrides any previous data for that cell).
- 4 The new GIS commands whilst recommended over legacy commands such as `Cell Elevation File ==`, `Elevation Polyline File ==`, `Cell Elevation Polygon ==` etc. will still be read in sequential order when combined with these command types. For example, cells intersected by the CSV-derived elevation polylines in the file ‘Shipping_Channel.CSV’ below will be read into the model following the `Read GRID Zpts ==` command. The command `Read GIS Z Line ==` will overwrite any ‘Shipping_Channel.CSV’ or grid elevations with those provided in my_road_001_L.shp and my_road_001_P.shp at cells where the datasets overlap.

```
! Geometry Specification with Updates
Geometry 2D == ..\model\geo\Bosos_Port_Development_001.2dm
Read GRID Zpts == ..\model\geo\grid\Existing_Config.txt
Nodestring Polyline File == ..\model\geo\Shipping_Channel.csv
Read GIS Z Line == ..\model\geo\gis\my_road_001_L.shp | ..\model\geo\gis\my_road_001_P.shp
```

- 5 Use the `Write Check Files ==` command (refer Section 12.10) to cross-check and carry out quality control checks on the final mesh ZB and Material inputs.

6 Simulation Configuration

Chapter Contents

6.1	Introduction	6-2
6.2	Solution Include Statements	6-3
6.3	Spatial Order and Gradient Limiters	6-5
6.4	Units – Metric or US Customary/English	6-6
6.5	Spherical or Cartesian	6-7
6.6	Bottom Drag Model	6-8
6.7	Wind Stress Model	6-9
6.8	Model Timestep	6-10
6.8.1	Adaptive Timestepping and CFL	6-10
6.8.2	Mode Splitting	6-11
6.9	Time Commands	6-12

6.1 Introduction

This chapter outlines the core simulation configuration commands including the solution ‘include’ statements, spatial order selection, units selection, coordinate type and the selection of bed roughness model.

TUFLOW FV’s adaptive time stepping, CFL control numbers, time stepping limits and mode splitting techniques are explored as well as the model time format, reference time, model start and end time definition.

6.2 Solution Include Statements

By default, TUFLOW FV is run in 2D hydrodynamic only mode. From the default terms in the non-linear shallow water equations the user has the option to activate or deactivate several source terms if they so choose. There are also ‘include’ statements that can be optionally specified to enable additional processes including TUFLOW FV’s advection dispersion, sediment transport, water quality or particle tracking modules. These commands are typically added near the top of your FVC. An example from a 3D estuary model is provided below which shows the inclusion of salinity, temperature, sediment and atmospheric heat exchange. The list of available include commands is provided in Table 6-1.

```
! SIMULATION CONFIGURATION
!
include salinity == 1,1
include temperature == 1,1
include sediment == 1,0
include heat == 1
spherical == 1
spatial order == 1,2
```

Please note these ‘include’ statements differ in purpose and should not be confused with the [Include](#) files functionality described in Section 4.3.3

Table 6-1 Solution Include Terms

Command	Details	Notes
Include Salinity	Optionally enable salinity modelling functionality with or without density coupling.	Accessed via TUFLOW FV’s advection dispersion module.
Include Temperature	Optionally enable temperature modelling functionality with or without density coupling	Accessed via TUFLOW FV’s advection dispersion module.
Include Heat	Optionally enable heat flux modelling between the water surface and atmosphere. Important for stratified environments or modelling with temperature enabled.	Accessed via TUFLOW FV’s advection dispersion module.
Include Sediment	Optionally enable the modelling of cohesive or non-cohesive suspended sediments (with or without hydrodynamic density coupling) or bed load sediments. Sediment modelling is set via a separate TUFLOW	Accessed via TUFLOW FV’s sediment transport module.

	FV Sediment Control File (.fvsed) ²	
Include Coriolis	Optional command used to switch off the Coriolis force source term from the momentum conservation equations.	Coriolis forcing can often be omitted/neglected for small study areas or where frictional forces dominate in shallow flow areas. i.e. where Coriolis deflection is limited. Will require either a spherical coordinates simulation, or specification of model latitude via a separate command.
Include bed friction	Optional command used to switch off bed friction.	Recommended to include bed friction through the use of the default value of 1. This option has been used previously for model testing and academic/training purposes.
Include wind	An optional command used to remove the wind stress terms from the momentum equations (only relevant if wind is a specified input using the BC command):	Turn off to remove wind effects from simulation e.g. for sensitivity testing.

² Please contact support@tuflow.com for further documentation and information on the sediment transport module.

6.3 Spatial Order and Gradient Limiters

In TUFLOW FV the horizontal and vertical spatial scheme (if running in 3D) can be specified as 1st or 2nd second order using the [Spatial Order](#) command.

Generally, initial model development is typically undertaken in 1st order schemes, with 2nd order spatial schemes tested during the latter stages of development. If a significant difference is observed between 1st and 2nd then the a 2nd order solution is probably necessary, or alternatively further mesh refinement is required.

For flow conditions with rapidly varying flows such as tsunamis, dam breaks or flood flows in deep, fast flowing channels, a 2nd order solution is usually necessary and recommended. Likewise, this is also often the case when modelling high gradients in density, concentration, temperature or other conserved variables. 2nd order spatial accuracy will typically be required in the vertical direction when trying to resolve sharp stratification.

A demonstration of the potential ‘smoothing’ effect of numerical diffusion is provided in the mapped velocity plots in Figure 6-1. The results shown are velocity contours from a laboratory-scale dam break analysis where both models have been run with identical setups apart from the horizontal spatial order specified. The left panel of Figure 6-1 uses a first order solution and shows the smoothing of shock waves when compared to the strong gradients shown in the 2nd order example.

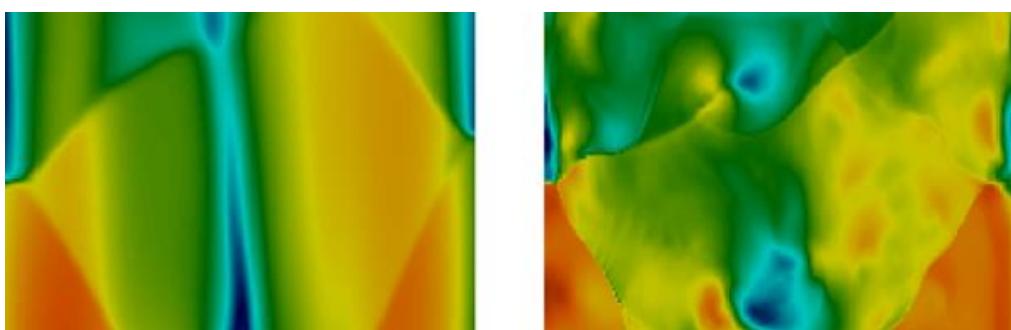


Figure 6-1 1st (left) and 2nd (right) Order Velocity Results Dambreak Flow

Spatial order can also be adjusted for selected material types using the [Spatial Reconstruction](#) command within a [Material Block](#).

When running the second order solution, the horizontal ([Horizontal AlphaR](#)) and vertical ([Vertical AlphaR](#)) gradient reduction factor commands may be of use for regions of high spatial gradients or to assist with improving model stability.

6.4 Units – Metric or US Customary/English

The default [Units](#) settings for all inputs in a TUFLOW FV model are metric, however, it is possible to create models using US Customary units (also known as Imperial or English units) by specifying the .tcf command `Units == US Customary` (`Units == Imperial` or `Units == English` are also accepted and treated identically). The equivalent input and output units are listed in Table 6-2.

Currently, Imperial Units are currently supported only for 2D flood assessments. They are yet to be fully implemented for use with the Advection Dispersion, 3D, Heat, Water Quality, Sediment Transport or Particle Tracking Modules. There has been improved error/warning messaging added to the 2019 Release if you do try to use: `Units == Imperial | US Customary | English` with unsupported modules similar to the following:

Imperial/US Customary/English Units are not yet supported for ice module. Please use Units == Metric

This manual uses the metric term for documentation purposes. If your model uses US Customary Units the metric term must be substituted for the equivalent US Customary unit as per [Table 6-2](#) when reading this manual, unless otherwise stated.

Table 6-2 Model Units

Parameter	Metric Units	US Customary Units
Length	m	ft
Velocity	m/s	ft/s
Constant eddy viscosity value	m ² /s	ft ² /s
Bed Shear Stress (TAUB)	N/m ²	lbf/ft ²

6.5 Spherical or Cartesian

By default, TUFLOW FV will adopt a cartesian coordinate system in metres or feet depending on the selection of [Units](#). The user can optionally set the model to use a longitude/latitude system using the [Spherical](#) command.

Note: Please use caution if using Cartesian coordinates in combination with the [Include Coriolis](#) command as you will need to manually specify the [Latitude](#) for Coriolis deflection (the default is at the equator). If using Spherical coordinates, the latitude is read directly from the model mesh and Coriolis deflection is spatially variable along the north-south axis of the model mesh.

6.6 Bottom Drag Model

For hydrodynamic simulations the bed boundary resistance is described using a [BottomDragModel](#). The default model is that attributed to Manning, in which case a Manning's "n" coefficient should be specified. An alternative bottom drag model assumes a log-law velocity profile and requires specification of a surface roughness length-scale, "ks". A single bed surface roughness can be set globally, or the modeller can assign different roughness values to particular mesh cells within the model domain to spatially distribute the roughness. The so-called "material type" definitions are stored in the TUFLOW FV mesh geometry file.

6.7 Wind Stress Model

If running with wind input boundaries (CYC_HOLLAND, W10 and W10_GRID, refer Table 10-3) there are three options to define how wind stress is calculated using the [Wind Stress Model](#) command. The specification of wind drag coefficient/s differs based on the selection of the [Wind Stress Model](#) and is applied using the [Wind Stress Parameters](#) command for the Wu (default) and Kondo models or using the [Bulk Momentum Transfer Coefficient](#) command if using a Constant wind stress (refer Table 6-3).

Table 6-3 Wind Stress Model and Drag Coefficients

Wind Stress Model	Specification of Wind Drag Coefficient
1 (default)	<p>If using the Wu Wind Stress Model the wind stress parameterisation is scaled based on wind speed using the Wind Stress Parameters command.</p> <pre>Wind Stress Parameters == <Wa (m/s), Ca (-), Wb (m/s), Cb (-)> (Optional); Default == <0.0, 0.8e-03, 50.0, 4.05e-03></pre>
2	<p>If using a Constant Wind Stress Model the wind stress parameterisation stays the same as pre-2019 builds and is applied using the Bulk Momentum Transfer Coefficient:</p> <pre>Bulk momentum transfer coefficient == <CDN> (Optional, Default == 0.0013)</pre>
3	<p>If using Kondo Wind Stress Model it is now possible to apply a scaling factor using a single term via the Wind Stress Parameters command.</p> <pre>Wind Stress Parameters == <Scale_factor> (Optional, Default == 1.0)</pre>

6.8 Model Timestep

6.8.1 Adaptive Timestepping and CFL

For solving the NLSWE, TUFLOW FV uses an adaptive timestep where a stable time step must be bounded by the Courant-Friedrich-Lowy (CFL) criterion for the wave propagation and advective terms and by the Peclet criterion for the diffusive terms (Murillo *et al.*, 2005). The CFL number is a function of the cell size and shape, water depth, flow velocity and the model timestep as given by:

$$\frac{|\mathbf{u} \cdot \mathbf{n} \pm \sqrt{gh}| \Delta t}{L^*} = \text{CFL}$$

where CFL is required to be ≤ 1 ,

Δt is the integration timestep,

h is the water depth,

g is the gravitational acceleration constant (9.81 m/s)

$\mathbf{u} \cdot \mathbf{n}$ is the flow velocity normal vector from a given cell face.

L^* is a cell-size dependent length scale $L^* = \frac{\min(A_i, A_j)}{L_k}$ where A_i, A_j are the adjacent cell-areas and L_k is the face length.

The internal mode CFL criterion is given by:

$$\frac{\max(|\mathbf{u} \cdot \mathbf{n}|, c_{baro}) \Delta t}{L^*} \leq 1$$

where c_{baro} is the baroclinic (internal) wave speed.

The Peclet criterion is given by:

$$\frac{|\mathbf{D} \cdot \mathbf{n}| \Delta t}{L^{*2}} \leq 1$$

A variable time step scheme is implemented to ensure that the [CFL](#) is maintained below the user specified [CFL](#) control number and Peclet criterion are satisfied at all points in the model with the largest possible time step. The minimum and maximum timestep limits are provided by the user via the [Timestep Limits](#) command. Outputs providing information relating to performance of the model with respect to the CFL criterion are provided to enable informed refinement of the model mesh in accordance with the constraints of computational time.

While the theoretic upper limit for the [CFL](#) is one, in practice this value is commonly lowered to provide additional stability for problems that exhibit large gradients in flow, density or constituent concentrations such as the assessment of Tsunami, dam break or lakes with strong vertical stratification. The maximum CFL used by TUFLOW FV can be modified globally using the [CFL](#) command or

specifically for the external and internal time stepping calculations (when mode splitting is enabled) using the [CFL External](#) and [CFL Internal](#) commands.

6.8.2 Mode Splitting

Efficient solving of the NLSWE is achieved through a mode splitting scheme, whereby “internal” and “external” components of the governing equations are updated using different timesteps selected by considering physical and numerical convergence and stability considerations.

A reduced set of equations comprising all terms other than the barotropic (free-surface) pressure-gradients is initially partially solved. As part of this solution, an appropriate “internal mode” timestep is calculated that obeys both:

- Courant-Friedrich-Lowy (CFL) constraints imposed by the advective current speeds
- CFL constraints imposed by an estimated internal wave speed.
- Peclet number (Pe) constraints imposed by the diffusion terms

An example of the calculated internal and external timesteps with mode-splitting enabled is shown in the TUFLOW FV Console Window (refer Figure 6-2). The internal timestep is highlighted by the yellow box, the external by the red. This internal mode timestep is used to solve internal waves, internal constituents with wave speeds typically an order of magnitude lower than the free surface shallow water wave speed and thus can be run at a much larger timestep than the “external mode”.

```
Entering timestep loop
t = 0.000000 hrs. dt = 0.000 / 0.000 s. elapsed time = 0.0 s.
      writing XMDF output. t = 0.000000 hrs.
t = 0.083436 hrs. dt = 2.822 / 0.282 s. elapsed time = 0.2 s.
t = 0.167015 hrs. dt = 2.808 / 0.281 s. elapsed time = 0.5 s.
t = 0.250220 hrs. dt = 2.782 / 0.278 s. elapsed time = 0.7 s.
```

Figure 6-2 TUFLOW FV Mode Split Timestepping

Prior to updating the solution an “external mode” loop is entered, in which a 2D depth-averaged reduction of the 3D NLSWE is solved multiple times, using a timestep that obeys the barotropic Courant-Friedrich-Lowy (CFL) constraint imposed by the shallow water wave speed $\bar{u} \pm \sqrt{gh}$ (where \bar{u} is the depth-averaged current speed). The external mode loop is repeated until the cumulative timestep is approximately equal to the internal mode timestep.

Mode splitting can be disabled for 2D simulations using the [Mode split](#) command and this configuration can be more computationally efficient for fast, shallow flow scenarios where the “internal mode” and “external mode” timesteps are similarly restrictive. Currently, 3D simulations are only supported with mode splitting enabled.

Unless there are compelling reasons to turn mode splitting off it is recommended to run TUFLOW FV with mode splitting enabled. If unsure you can always run a sensitivity/benchmark tests and compare your results and runtimes to check if you gain any performance benefit.

6.9 Time Commands

TUFLOW FV allows two time formats, ISODATE or hours (the default) which can be set using the [time format](#) command. The time format adopted must be consistent within all control files and model input time series. When using ISODATE, data and time inputs are provided in the format *dd/mm/yyyy HH:MM:SS* and when using hours time units are in decimal hours.

The [start time](#) and [end time](#) commands reside in the fvc (or an [include](#) or [read file](#)) and define the model simulation period. If using a [restart file](#), the [start time](#) may be overwritten depending on the configuration of the [use restart file time](#) command (for more information on using restart files please refer to Section 10.7.3).

If using `time format == hours`, the [Reference time](#) default value is 0hrs. For `time format == ISODATE`, the [Reference time](#) is set to 01/01/1990 00:00:00 by default. This is important to note when visualising and processing results and input data from programs such as MATLAB. For example, TUFLOW FV will output times to NetCDF output in decimal hours from 01/01/1990 00:00:00 whereas the convention in MATLAB is to output serial times in decimal days since 01/01/0000 00:00:00 and if overlaying datasets some time conversions will be likely required.

If reading in NetCDF boundaries with differing reference times, the [bc reference time](#) and [bc time units](#) commands may be of use (these are detailed further in Chapter 10. Examples of NetCDF input and output files are provided in Appendix D and Appendix E for reference.

7 Model Parameters

Chapter Contents

7.1	Introduction	7-2
7.2	Turbulence	7-3
7.2.1	Eddy Viscosity	7-3
7.2.1.1	<i>Horizontal</i>	7-3
7.2.1.2	<i>Vertical</i>	7-3
7.2.2	Scalar Diffusivity	7-4
7.2.2.1	<i>Horizontal</i>	7-4
7.2.2.2	<i>Vertical</i>	7-4
7.3	Reference Values	7-5
7.4	Wetting and Drying Thresholds	7-7
7.5	Stability Limits	7-8

7.1 Introduction

This chapter describes the horizontal and vertical turbulence related commands, stability limits, model reference values, and wetting and drying specification.

7.2 Turbulence

TUFLOW FV has a variety of options for simulating horizontal and vertical viscous fluxes. The horizontal eddy-viscosity can be specified directly or can be calculated using the Smagorinsky scheme. Simple parametric models for vertical mixing are also incorporated within the TUFLOW FV engine.

TUFLOW FV allows the horizontal scalar diffusivity to be specified as a constant value or calculated from a Smagorinsky or Elder model. The vertical scalar diffusivity may also be directly specified or calculated using parametric formulations which vary depending on the scalar type.

For more complicated turbulence model algorithms an interface for linking with various external turbulence models has been implemented (e.g. GOTM, <http://www.gotm.net/>). TUFLOW Support can be contacted for further information on coupling TUFLOW FV with external turbulence models: support@tuflow.com.

The horizontal and vertical-mixing options are set in the TUFLOW FV Simulation Control File (.fvc).

7.2.1 Eddy Viscosity

7.2.1.1 *Horizontal*

The horizontal-mixing eddy-viscosity can be defined as a constant value or can be calculated using the Smagorinsky model via the [Momentum mixing model](#) command. The Smagorinsky model sets the diffusivity proportional to the local strain rate. Viscosity values and limits can be applied globally using the [Global horizontal eddy viscosity](#) and [Global horizontal eddy viscosity limits](#) commands or spatially varied via a Material Block (refer Section 8.5).

7.2.1.2 *Vertical*

Vertical-mixing eddy-viscosity can be defined as a constant value or calculated using a parametric model or external scheme set by the [Vertical mixing model](#) command. Eddy-viscosity values for the constant and parametric models can be set globally via the [Vertical mixing parameters](#) command (if using an external scheme such as GOTM, the external model calculates the eddy viscosity (at a time increment defined by the [turbulence update dt](#)) and returns it to TUFLOW FV). The parametric model is based on a parabolic eddy-viscosity profile in the lower-half of the water column transitioning to a constant eddy-viscosity in the upper-half and applies the Munk & Anderson limiters in the case of stable stratification. For systems with strong vertical gradients the use of an external vertical turbulence mixing model is recommended. For parametric and external schemes eddy viscosity limits can be applied using the [Global vertical eddy viscosity limits](#) command. The global commands can be overwritten on a material basis using the [vertical eddy viscosity limits](#) command within a Material Block (refer Section 8.5).

7.2.2 Scalar Diffusivity

7.2.2.1 *Horizontal*

The [Scalar mixing model](#) command sets the horizontal-mixing scalar diffusivity as either a constant value or calculated using the Smagorinsky or Elder models. The respective coefficients for each approach are input via the [Global horizontal scalar diffusivity](#) command and can have minimum and maximum limits applied using the [Global horizontal scalar diffusivity limits](#) command. Spatially variable [horizontal scalar diffusivity](#) and [horizontal scalar diffusivity limits](#) can be applied within a Material Block (refer Section 8.5).

7.2.2.2 *Vertical*

The vertical-mixing scalar diffusivity can be defined as a constant value or can be calculated using a parametric or external model (see [Vertical mixing model](#)). Coefficients for each model are applied using the [Vertical mixing parameters](#) command. If using an external model, diffusivity is calculated on a user-defined time interval [turbulence update dt](#). The parametric model is based on a parabolic eddy-viscosity profile in the lower-half of the water column transitioning to a constant eddy-viscosity in the upper-half and applies the Munk & Anderson limiters in the case of stable stratification. Upper and lower bound values can be specified globally using the [Global vertical scalar diffusivity limits](#) command and also on a material basis (refer [vertical scalar diffusivity limits](#), and Section 8.5) which allows spatial variation.

7.3 Reference Values

The following reference parameter values are used in calculating terms within the NLSWE. In all cases default values are assumed unless specified in the control file.

Kinematic viscosity

Sets the kinematic viscosity of water. <Default = 1.05e-06 m²/s>. This value is used as a lower limit in the parametric vertical eddy viscosity and is also used in various formulae within sediment transport and particle tracking modules.

Latitude

Globally sets the model latitude. <Default = 0 degrees>. Positive value for northern hemisphere and negative for southern hemisphere. Used to calculate Coriolis source term for cartesian grid models (i.e. UTM coordinate system). Note that spherical coordinate system is preferred where Coriolis effects may be significant.

Density air

Globally sets the density of air <Default = 1.20 kg/m³>. This value is used to calculate wind stress and atmospheric heat exchange source terms. May be over-ridden by values specified in Holland (parametric Tropical Cyclone model, refer Section 10.5.3.2) boundary condition file or will be calculated internally when the Kondo [wind stress model](#) is specified.

Reference MSLP

Globally sets the mean sea level pressure <Default = 1013.25 hPa>. This value is used in the atmosphere module when a mean sea level pressure boundary condition has not been specified. The global MSLP value is used in various atmosphere module heat exchange routines. A globally uniform MSLP value will not drive a direct hydrodynamic response as it does not result in any pressure gradients.

Reference density

Sets the reference density (of water) <Default = 1000 kg/m³>. For baroclinic simulations (where density may vary locally based on salinity, temperature or suspended sediment concentration) this value is used as the baseline from which density gradients are calculated. Numerical accuracy of baroclinic simulations can be marginally improved by setting this value close to a typical value expected in the model (e.g. 1,027 kg/m³ for ocean simulations). For non-baroclinic simulations this command will globally set the density which is used in various source term and sediment transport formulae.

Reference salinity

Sets the reference salinity <Default = 0 psu>. Numerical accuracy of baroclinic terms in NLSWE can be marginally improved by setting this value close to a typical value expected in the model (e.g. 35.0

psu for ocean simulations). For non-baroclinic simulations this command will globally set the salinity. May also be used to globally initialise salinity in the absence of other relevant initial conditions.

[Reference temperature](#)

Sets the reference temperature (of water) <Default = 20 degrees celsius>. Numerical accuracy of baroclinic terms can be marginally improved by setting this value close to a typical value expected in the model. For non-baroclinic simulations this command will globally set the temperature. May also be used to globally initialise temperature in the absence of other relevant initial conditions.

[Specific heat air](#)

Sets the specific heat capacity of air <Default = 1005.0 J/Kg/ $^{\circ}$ C>. Used to calculate atmospheric heat exchange source terms. The specific heat capacity value will be calculated internally when the Kondo wind stress model is specified.

[Specific heat water](#)

Set the specific heat capacity of water <Default = 4181.3 J/Kg/ $^{\circ}$ C>. Used to calculate atmospheric heat exchange source terms. The specific heat capacity value will be calculated internally when the Kondo wind stress model is specified.

7.4 Wetting and Drying Thresholds

TUFLOW FV simulates the wetting and drying of areas within the model domain, such as that observed on a gently sloping beach over a tidal cycle or over land during a flood, storm surge or tsunami event. Dry/wet depths defined by the user will often depend on the scale of the simulation. For full-scale or “real world” simulations, dry/wet depths are typically in the order of centimetres. For some laboratory-scale simulations, for example a dam break or wave run-up experiment, the user defined wet/dry depths may be in the order of millimetres.

In terms of the TUFLOW FV computations, the drying value corresponds to a minimum depth below which the cell is dropped from computations (subject to the status of surrounding cells). The wet value corresponds to a minimum depth below which cell momentum is set to zero, in order to avoid unphysical velocities at very low depths. The default wetting and drying values can be modified using the [Cell wet/dry depth](#) command in the TUFLOW FV Control File (.fvc).

7.5 Stability Limits

Major instabilities in models will usually manifest in shocks within the model that can result in unrealistically high water levels or velocities. However, as models differ in the range of expected values that they are trying to capture, there is no ‘rule’ for what water level or current velocity might indicate an instability. Therefore, users can specify upper limits that are relevant to their study and will cause the model to exit computations with an error if these are exceeded using the [Stability limits](#) command.

Modellers are advised to set these limits to be as low as practicable to ensure that any real instabilities are ‘caught’ as soon as possible, but any small anomalous and acceptable spikes in water level or velocity do not cause a model exit. For typical ocean modelling in most parts of the world it might be appropriate to set these limits to a 10 m water elevation and a 5 m/s maximum current. These limits will capture most of the highest expected water levels due to tide and storm surge effects as well as set an appropriate limit for tidal currents in an open area. If modellers are trying to capture extreme water levels or high currents through narrow choke points, then it is possible that these limits should be increased.

Difficulties arise when investigating catchment models. In these situations, the topography may contain elevations above the maximum water level (if using a mean-sea-level datum for example). If any of these high cell elevations become wet due to rainfall inputs or upstream inflow boundary conditions, then this will instantly trigger a model exit. For these models it is important to set the maximum water level limit to be higher than the highest expected cell elevation that will get wet, plus any additional water depth expected in that cell.

8 Model Geometry

Chapter Contents

8.1	Mesh Generation	8-2
8.1.1	Choosing a Mesh Generator	8-3
8.1.2	Mesh File Format (.2dm)	8-3
8.2	External Nodestrings	8-7
8.3	Elevations	8-8
8.3.1	Direct Reading from .2dm Mesh	8-8
8.3.2	Updating Elevations from a Grid	8-8
8.3.3	Breakline Layers	8-9
8.3.4	Legacy Elevation Update Options	8-10
8.3.4.1	<i>Cell Elevation Polyline File</i>	8-10
8.3.4.2	<i>Cell Elevation Polygon File</i>	8-11
8.4	Materials	8-12
8.5	Material Block	8-14
8.5.1	Bed Resistance	8-14
8.5.2	Active/Inactive Areas	8-14
8.5.3	Horizontal Eddy Viscosity	8-15
8.5.4	Horizontal and Vertical Eddy Viscosity Limits	8-15
8.5.5	Horizontal Scalar Diffusivity	8-16
8.5.6	Horizontal and Vertical Scalar Diffusivity Limits	8-16
8.5.7	Bed Elevation Limits	8-17
8.5.8	Spatial Reconstruction	8-17

8.1 Mesh Generation

The primary goal when designing a flexible mesh is to **describe the key bathymetric and hydrodynamic features using the least, largest mesh element sizes possible that still provide adequate resolution for the problem you're trying to solve**. This is a key benefit of flexible meshes ; to optimise computational efficiency whilst achieving desired model accuracy.

Creating a mesh is a combination of manual and automated steps. Maintaining a reasonable amount of manual intervention into the design of the mesh will ultimately produce a far more efficient mesh which will be more accurate and computationally efficient.

Using a mesh generator, a TUFLOW FV mesh (SMS .2dm format) is constructed using nodes (points), arcs and vertices (polylines). These “mesh controls” are generally positioned manually by the modeller using their preferred mesh generation tool. Important features of an area to be modelled may include islands, rivers and inlets, deep channels or man-made infrastructure. A good mesh is constructed using the mesh controls (nodes, vertices and arcs) to neatly resolve the important features within the model domain.

Figure 8-1 provides an example of the mesh controls and the resulting mesh for a section along a river bend. The left panel shows the mesh controls, namely:

- Nodes (red circles)
- Arcs (lines between two nodes)
- Vertices (small black squares along an arc)

The positions of the mesh controls have been defined by the modeller and in this case are located to resolve the river banks and the main channel. The vertices have been distributed evenly along each arc and control the number of mesh cells that can occur along the arc. The right panel shows the resulting mesh that is generated by the mesh software.

You can reproduce the mesh shown in Figure 8-1 by following along the steps in [Tutorial Model 02](#).

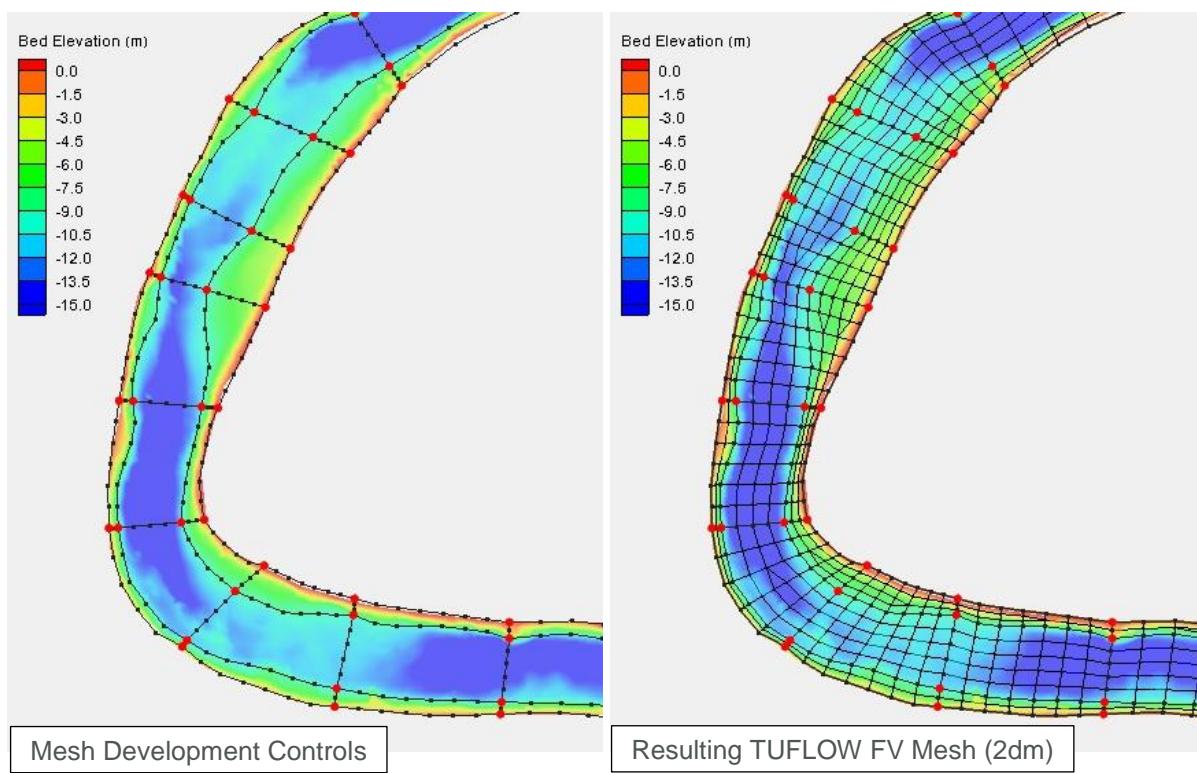


Figure 8-1 Mesh Development Example

8.1.1 Choosing a Mesh Generator

Unstructured mesh geometries can be created using any suitable mesh generation tool. Historically BMT has used the Aquaveo SMS Generic Mesh Module for building meshes. As a result, the TUFLOW FV mesh file format is the SMS mesh file format. As of 2019, there is also the Rising Water Software GIS-Mesher option for generating (.2dm) meshes. As TUFLOW FV does not generate its own flexible mesh, one of these packages is required for your model development. For more information on both meshers please refer to the following website links:

- [Aquaveo SMS](#)
- [Rising Water Software GIS Mesher](#)

8.1.2 Mesh File Format (.2dm)

The .2dm file format is used to define the TUFLOW FV mesh. It is an ASCII format from the SMS Generic Mesh Module. This section provides a reference to the various components of the mesh file and provides further insight into how TUFLOW FV uses it.

Setting up and running a TUFLOW FV model simulation does not necessarily require a detailed line by line inspection of the mesh file; SMS or GIS provides a graphical interface to do this instead. Nevertheless, a modeller may find it necessary at times to interrogate the 2dm file in detail.

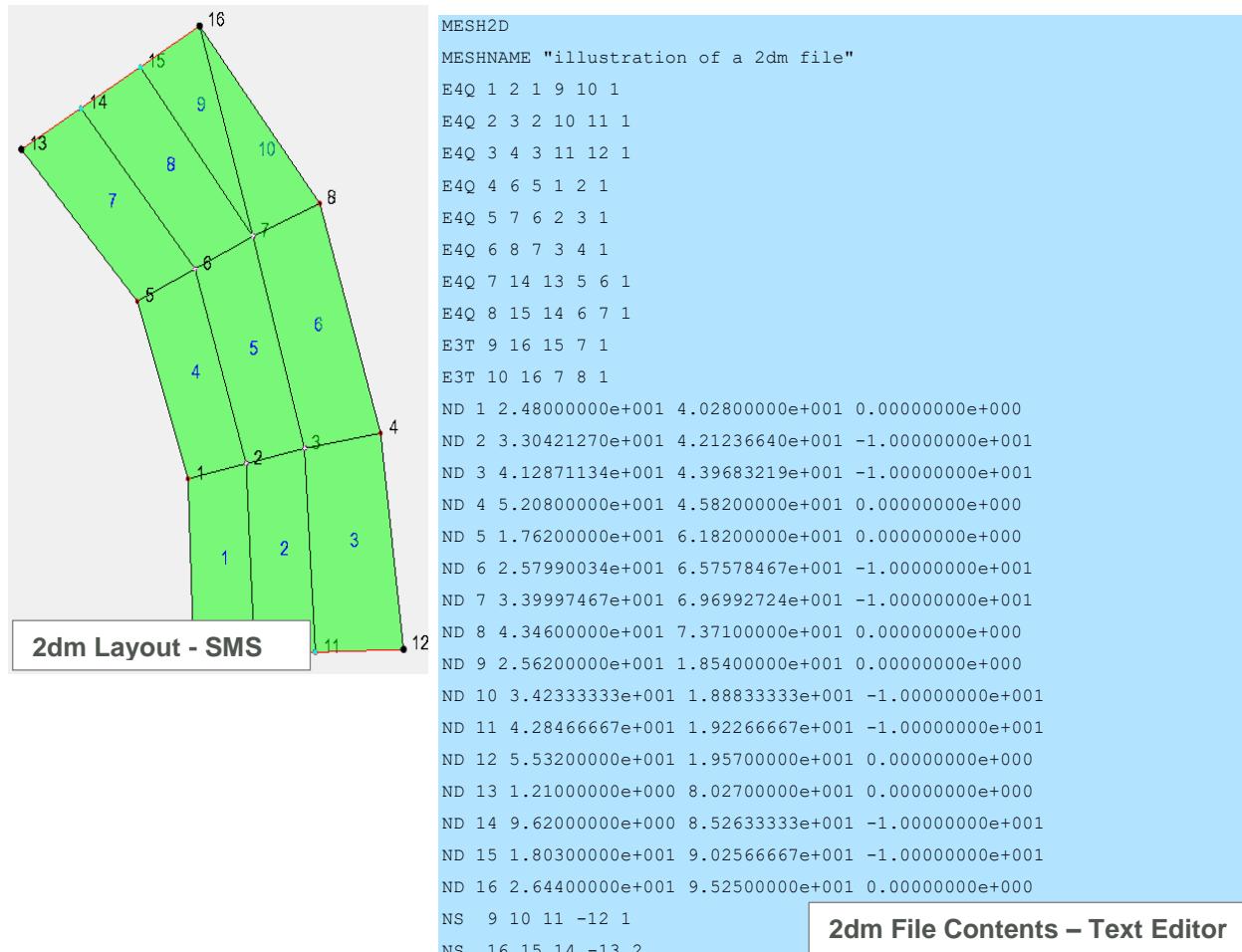
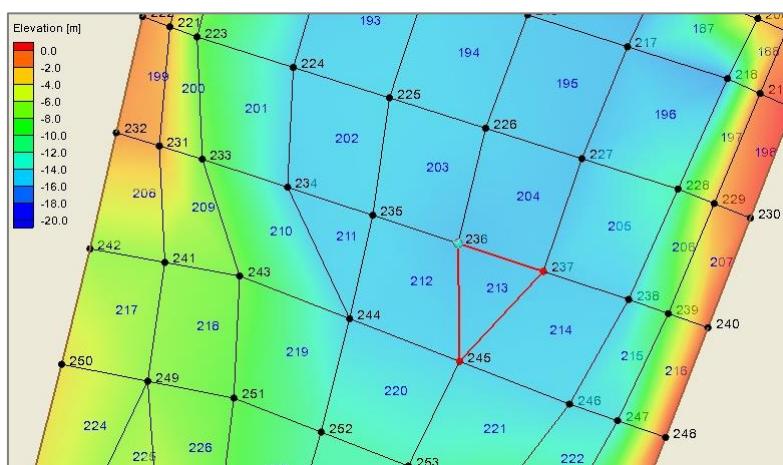


Figure 8-2 Example 2dm File

The contents of the .2dm file required for TUFLOW FV simulations are:

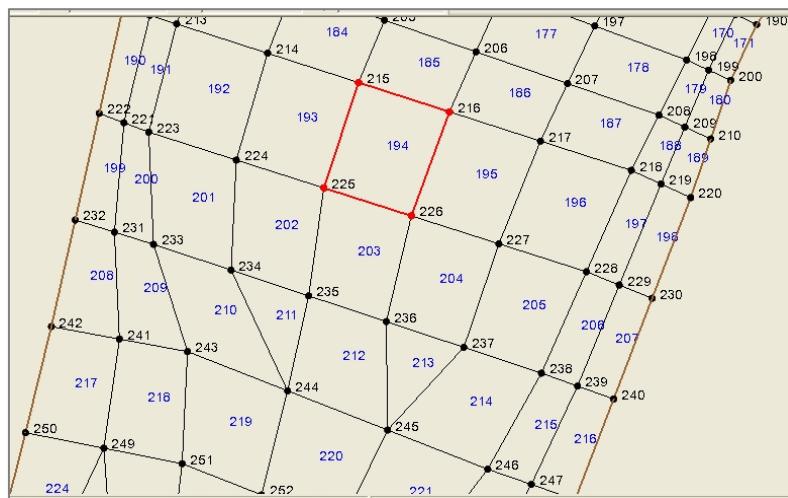
Nodes: Lines that commence with a “ND” are nodes, or the points that define the edges of the elements. Each ND line describes the node ID and its x, y and z (i.e. bed level) coordinate. The screen shot in Figure 8-3 shows node 236 selected and its corresponding position and elevation displayed in the X,Y,Z dialog boxes.



9731 ND 235 5.15399210e+005 6.97065840e+006 -1.59007000e+001
9732 ND 236 5.15491480e+005 6.97062790e+006 -1.65624000e+001
9733 ND 237 5.15583760e+005 6.97059739e+006 -1.63117000e+001

Figure 8-3 Example Node Definition

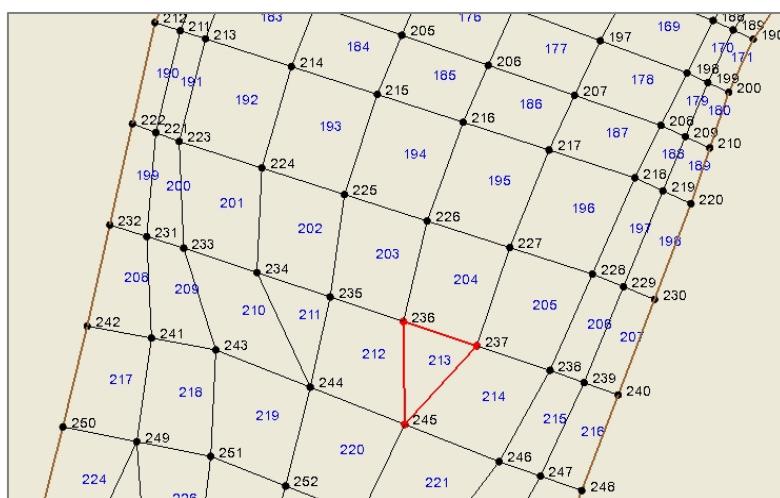
Elements: Lines that commence with an “E4Q” are quadrilateral (4 sided) elements. Each E4Q line describes the element ID, the four nodes that define its connectivity and spatial extent (in a counter-clockwise direction) and the material type.



195 E4Q 193 224 225 215 214 1
196 E4Q 194 225 226 216 215 1
197 E4Q 195 226 227 217 216 1

Figure 8-4 Example Quadrilateral Element Definition

Similar to E4Q, the “E3T” lines are triangular elements. Each E3T line describes the element ID, the three nodes that define its connectivity and spatial extent (in a counter-clockwise direction) and the material type.



214 E4Q 212 244 245 236 235 1
215 E3T 213 245 237 236 1
216 E4Q 214 245 246 238 237 1

Figure 8-5 Example Triangular Element Definition

Nodestrings: Lines that commence with a “NS” are nodestrings, which are used to define boundary conditions locations. Each NS line defines the series of nodes that form the string, the last node number is assigned as negative. The number following the negative number is the nodestring ID.

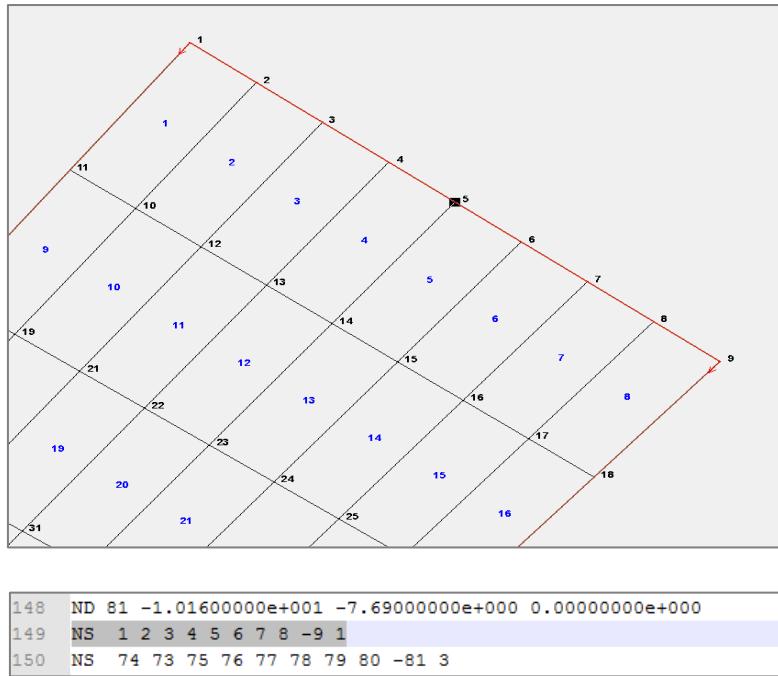


Figure 8-6 Example Nodestring Definition

Once the mesh is developed and saved, the .2dm file is read using the .fvc command [Geometry 2D](#) and can be used in conjunction and layered with other elevation commands, e.g. Read GIS Z Line and Cell Elevation File. These methods are described further in Section 8.3.3 and 8.3.4.

8.2 External Nodestrings

Nodestrings can be assigned directly during mesh creation (refer Figure 8-6) or via external nodestrings set in the .fvc file and processed on model initialisation. For external nodestrings, the [Read GIS Nodestring](#) command provides an alternative to the previous .CSV file method [Nodestring Polyline File](#). In the example below, both open water level and flow boundary locations are specified in the GIS layer “2d_ns_Open_Boundaries_001_L.shp”. This layer is being used to apply open boundary nodestrings. The layer 2d_ns_Flux_Lines_001_L.shp has been separated for data management purposes and is reporting fluxes within the model.

Nodestrings input by the [Read GIS Nodestring ==](#) command are additive to those existing in the .2dm or already specified in previous [Read GIS Nodestring](#) or [Nodestring Polyline File](#) commands.

The [Read GIS Nodestring ==](#) command requires that all nodestring ID's in the model are unique. This includes any that have been specified in the .2dm or nodestring polyline file. If nodestrings with identical IDs are input then TUFLOW FV will generate an error and you will need to renumber your inputs. Future versions of the software will include better handling of duplicate nodestring id's such as automatic renumbering. The required attributes when using a 2d_ns input layer is provided in Table 8-1.

Table 8-1 GIS Nodestring (2d_ns) Attributes

No.	Default GIS Attribute Name	Description	Type
1	ID	Unique integer identifier for each nodestring 1:n.	Integer
2	Type	<p>Either set to 'BD' or left blank.</p> <p>If set to 'BD' this will use TUFLOW FV's boundary algorithms to snap the nodestring along the outer edge of the model, as required for open boundary conditions. The 'BD' type should not be used for internal nodestrings such as those used to monitor flow or to define hydraulic structure locations.</p> <p>Leaving the field blank will snap the nodestring to the closest cell faces at cells that the polyline intersects.</p>	Char

8.3 Elevations

8.3.1 Direct Reading from .2dm Mesh

The .2dm file defines the base elevations within the model using the [Geometry 2d](#) command. The .2dm format supports elevations only at the nodes or corners of each cell. When TUFLOW FV reads the mesh geometry file cell centred bed elevations are interpolated from the .2dm node locations. In many instances, this interpolation is not an issue. However, there may be instances where this is not preferred and the [Read GRID Zpts](#) command can be used to specify exact cell centred elevation values.

8.3.2 Updating Elevations from a Grid

The [Read GRID Zpts](#) command interpolates an ESRI ASCII (.asc) or binary (.flt) grid to set the cell centroid elevations. The use of this command reduces the time required to assign cell centre elevations when compared to the previous method of manually carrying out a cell point inspection and assigning CSV files with the [Cell Elevation File](#) command. Like other topographic update commands, [Read Grid Zpts ==](#) may be specified more than once. If a raster grid does not contain data for a cell centroid (i.e. DEM is null) then the elevation point is not updated in the model.

Figure 8-7 shows an example study area where the grid DEM_5m.asc is read via the following command. This grid is interpolated to the mesh cell centres. The output _mesh_check file can be reviewed and coloured to show the interpolated ZB values from the grid DEM_5m.asc.

Example: [Read Grid Zpts == ..\model\geo\grid\DEM_5m.asc](#)

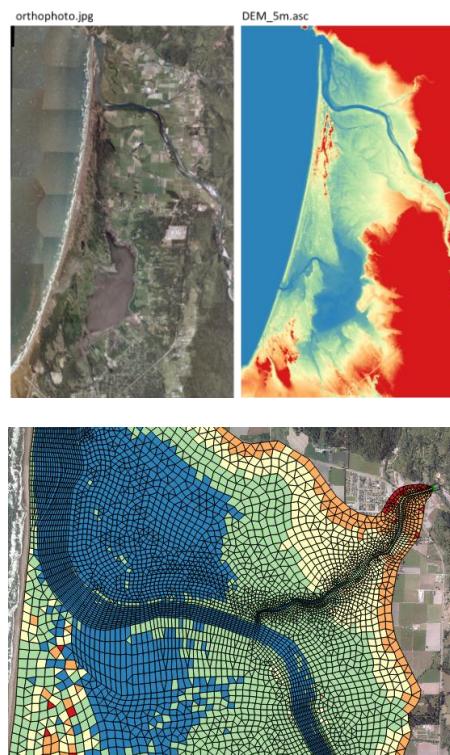


Figure 8-7 Cell Centre Elevations Assigned via Read GRID Zpts

8.3.3 Breakline Layers

A combination of GIS polyline layers and points can be used to define breaklines features which act as critical hydraulic controls within the study area. This is a useful feature for defining the crest elevation of levees or raised roads which traverse a floodplain, or alternatively the bed elevation of tributary creeks.

GIS layers used for [Read GIS Z Line](#) can be split into more than one layer to better manage the variety of data these commands sometimes require.

For example, one layer may contain the elevation points and another breaklines. This is useful in terms of managing the data, and especially when interrogating and/or viewing the data in GIS. It is a requirement of the shapefile format that the different geometries (points, lines and regions) are in separate shapefiles. The TUFLOW empty template files include the following filename suffixes to differentiate which files are suitable for point, line or region features.

- _P for point features (e.g. 2d_zln_M03_002_P.shp)
- _L for line features (e.g. 2d_zln_M03_002_L.shp)
- _R for region or polygon features (e.g. 2d_zln_M03_002_R.shp)

This is optional for MapInfo users; the different geometries can occur in the same MapInfo file or can be separated if preferred.

A maximum of nine (9) layers per command line is allowed, and each layer is separated by a vertical bar (“|”). For example, to read a Z Shape layer which has both line and points, the command may be:

```
Read GIS Z Line == gis\2d_zln_M03_002_L.shp | gis\2d_zln_M03_002_P.shp
```

Multiple points layers can be specified. The points layer can be referenced in any location except for the first layer within the command line entry.

Incorrect:

```
Read GIS Z Line == gis\2d_zln_M03_002_P.mif | gis\2d_zln_M03_002_L.mif
```

Correct:

```
Read GIS Z Line == gis\2d_zln_M03_002_L.mif | gis\2d_zln_M03_002_P.mif
```

Table 8-2 2D Z (2d_z_) Attribute Description

No.	Default GIS Attribute Name	Description	Type
1	Z	Elevation (or change in elevation for ADD option) of the point.	Float

8.3.4 Legacy Elevation Update Options

The [Cell Elevation File](#) or [Cell Elevation Points](#) commands can be used to update cell elevations by either referencing a cell ID or specifying an x,y coordinate which falls within a cell. These commands are useful when a user wishes to define an exact elevation value to a single cell, or multiple cells within a model (instead of applying interpolated values from the cell corners/nodes). Two example cell elevation files (.CSV) are shown below.

Table 8-3 Cell Elevation File Examples (the example on the left references X,Y co-ordinates and the one on the right uses Cell ID)

X	Y	Z	ID	Z
136.881	-11.0571	-58.7456	1	-58.7456
136.873	-11.0302	-58.7456	2	-58.7456
136.824	-11.1601	-55.6548	3	-55.6548
136.972	-10.7822	-58.9239	4	-58.9239
136.963	-10.7534	-60.1127	5	-60.1127
136.987	-10.7353	-59.9938	6	-59.9938
136.981	-10.7053	-59.9344	7	-59.9344
137.005	-10.6874	-59.9344	8	-59.9344

OR

More than one “cell elevation” command line can be defined with a simulation control file (fvc), and/or more than one point per cell can be entered. Depending upon input preference, each z value will overwrite the preceding z value entry, or an average of all points within each cell will be assigned. If multiple cell elevation files are listed, where inputs from one entry fall with the cell of a preceding entry, the later entry in the .fvc (lower) entry supersedes the previous (higher). You can check that elevation updates have been assigned correctly by using the Echo Geometry CSV to produce a set of text files (to the log directory) that identify cells that have been modified from the base .2dm as follows:

```
Echo geometry CSV == 1
```

The cell elevation file option does not interpolate between successive points. If using the cell elevation file for continuous linear features (such as a road or levee), ensure that the point resolution is sufficiently fine to accurately represent the elevations along the feature, or use the [Read GIS Z Line](#) command further described in Section 8.3.3.

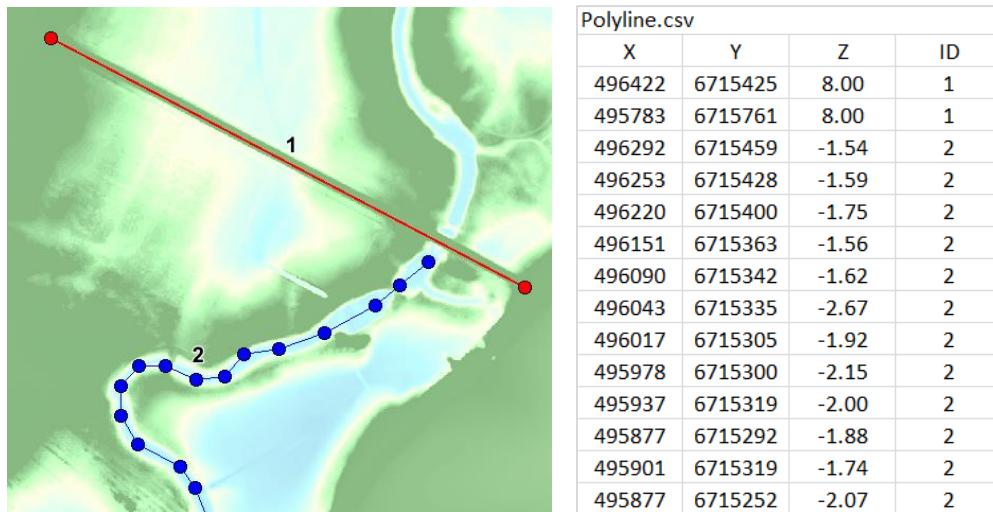
Note: A list of all cell ID and corresponding X,Y co-ordinates can be obtained from the geometry log files.

Although still supported the GIS Integrated [Read GIS Z Line](#) is now preferred over the [Cell Elevation Polyline File](#) or [Cell Elevation Polygon File](#) Commands.

8.3.4.1 Cell Elevation Polyline File

Input data is defined using a CSV file containing X,Y, Z and ID, data. Points within the CSV file define the vertices along the polyline. Intermediate cell elevations between vertices are interpolated.

Multiple polylines can be defined within a single CSV file. The ID attribute is used to differentiate between the different polylines. A unique command line input for each polyline is required within the simulation control file (fvc), as shown below.



```
cell elevation polyline file == ..\model\geo\Polyline.csv, 1 !Levee
cell elevation polyline file == ..\model\geo\Polyline.csv, 2 !Creek
```

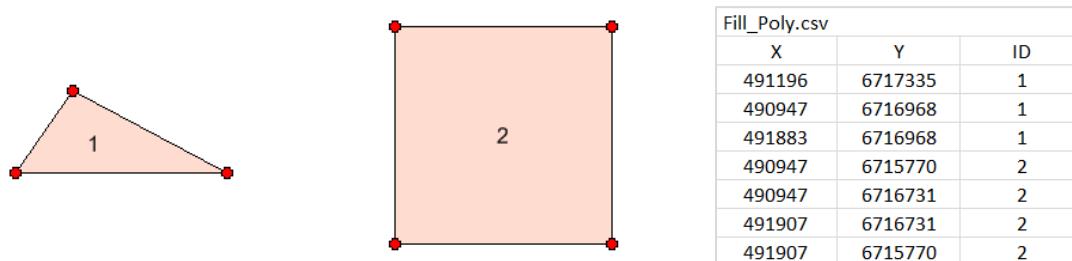
Figure 8-8 Cell Elevation Polyline Example

8.3.4.2 Cell Elevation Polygon File

The ‘Cell Elevation Polygon’ command applies a single elevation over a polygon. An example of its application is to assign a specific elevation to cells representing a reclamation or infill of a proposed development.

Input data is entered using a CSV file containing X,Y, and optionally ID, data. Points within the CSV file define the perimeter of the polygon. The definition of points needs to be consecutively listed and can be either clockwise or counter-clockwise.

As per the [Cell Elevation Polyline File](#) example, multiple polygons can be defined within a single CSV file. The ID attribute is used to differentiate between the different polygons. A unique command line input is required for each polygon within the simulation control file (fvc), as shown below.



```
cell elevation polygon file == ..\model\geo\Fill_Poly.csv, 6.0, 1 !Triangular area raised to 6m
cell elevation polygon file == ..\model\geo\Fill_Poly.csv, 6.0, 2 !Square region raised to 6m
```

Figure 8-9 Cell Elevation Polygon Example

8.4 Materials

Material layers are used to differentiate sections of the model domain. Materials layers can be used to spatially vary a range of bed properties including:

- [Bed resistance](#) (bottom roughness)
- [Active/inactive areas](#)
- [Horizontal and vertical eddy viscosity](#)
- [Bed elevation limits](#)
- [Spatial reconstruction](#)

The recommended approach is to assign a unique material to represent a different landuse or bed characteristic. GIS layers of land-use or vegetation often make excellent material layers. Examples of different materials are main river channel, river banks, floodplain, sea grass meadows, buildings, sand banks, mangroves, etc.

Materials can either be assigned during mesh development using your preferred mesh generator ([Aquaveo SMS](#) or the [GIS Mesher](#)) or assigned/modified following mesh development directly by TUFLOW FV using the [Set Mat](#) and [Read GIS Mat](#) commands.

Prior to the 2019.01.008 TUFLOW FV Release the only way to assign materials was directly via the .2dm file. While this method is still available, the preferred approach is to digitise one or more 2d_mat materials layers (see Table 8-4) and assign materials within a Material Block. This approach allows the easy adjustment of surface characteristics, for example modifying bed resistance during model calibration or sensitivity testing.



Figure 8-10 Materials Specification using 2d_mat GIS Layers

In creating the base 2d_mat layer, it is good practice to not digitise the most common or the most difficult to digitise material and use the following data layering of commands in the .fvc:

- Use [Set Mat](#) to set the most common material to all cells in the domain.
- Use [Read GIS Mat](#) to allocate the remaining material values.

The [Read GIS Mat](#) command may be used as many times as required to further modify the materials in parts of a 2D domain. Each subsequent dataset will overwrite the preceding assigned material value, as described. i.e. layers called lower down the input file will be read in preference to those above them where two layers overlap.

Table 8-4 2D Material (2d_mat) Attribute Description

No.	Default GIS Attribute Name	Description	Type
1	Material	The material ID value referenced within a Materials Block (refer Section 8.5).	Integer

8.5 Material Block

The properties associated with material can be changed in a material block. Material blocks refer to the material ID as specified in the mesh (.2dm) or alternatively in a material GIS file using [Read GIS Mat](#). A separate material block is required for each material for which the properties differ from the global values.

The following example shows a global bottom roughness, then a series of material blocks changing this for three of the material IDs.

```
bottom drag model == ks
global bottom roughness == 0.10

material == 10
    bottom roughness == 0.05
end material

material == 2
    bottom roughness == 0.15
end material

material == 8
    bottom roughness == 0.05
end material
```

8.5.1 Bed Resistance

Bed resistance formulation can be set to Manning's n or a log-law velocity profile using the [Bottom Drag Model](#) command in the .fvc file. The bottom roughness command within a material block will apply the bottom roughness to that material. The bottom drag model cannot be changed between materials.

The example below presents a global bottom roughness that is modified for three of the materials.

```
bottom drag model == ks
global bottom roughness == 0.10

material == 10
    bottom roughness == 0.05
end material

material == 2
    bottom roughness == 0.15
end material

material == 8
    bottom roughness == 0.05
end material
```

8.5.2 Active/Inactive Areas

The inactive flag allows users to switch the cells with a given material ID off for the purposes of the hydrodynamic calculations. The cells are still tracked within the model and are output, though are set to

be permanently dry. This is commonly used for impact assessments when switching off certain cells to simulate an area of development or can be used to switch off large sections of a model to conduct further testing on a model subset.

The below example shows how to switch a single material to inactive, the default status for all cells is active.

```
material == 10
  inactive == 1
end material
```

8.5.3 Horizontal Eddy Viscosity

The [horizontal eddy viscosity](#) command can be used to adjust the material specific eddy viscosity parameters. The specifics of this command are dependent on the adopted [momentum mixing model](#). This command can be used to increase or decrease the momentum mixing in a particular area.

This is usually done to smooth instabilities in regions of the model that are not near the area of interest (such as the corners of offshore boundaries). Users should take care with using this command without a suitable understanding of the physical relevance of adjusting the momentum mixing spatially.

The below example shows a [global horizontal eddy viscosity](#) being set for a Smagorinsky [momentum mixing model](#), with the Smagorinsky parameter being altered for two different materials.

```
momentum mixing model == Smagorinsky
global horizontal eddy viscosity == 0.2

material == 10
  horizontal eddy viscosity == 0.19
end material

material == 11
  horizontal eddy viscosity == 0.25
end material
```

8.5.4 Horizontal and Vertical Eddy Viscosity Limits

The [horizontal eddy viscosity limits](#) command allows the user to specify a minimum and maximum eddy viscosity that changes by region. This can be used to set a limit in a particular area to control the momentum mixing. This can be used in conjunction with or instead of the material [horizontal eddy viscosity](#) to increase or decrease the viscosity to smooth potential instabilities. The [vertical eddy viscosity limits](#) command is used to control the vertical turbulence i.e. to increase/decrease the effective vertical turbulence if there is insufficient or too much vertical momentum mixing.

The below example below shows the turbulence being set to a minimum level of mixing in the cells with the material ID of 10.

```

momentum mixing model == Smagorinsky
global horizontal eddy viscosity == 0.2
global horizontal eddy viscosity limits == 0.0, 99999.
global vertical eddy viscosity limits == 1.0e-04, 99999.

material == 10
  horizontal eddy viscosity limits == 100, 99999.
  vertical eddy viscosity limits == 100, 99999.
end material

```

8.5.5 Horizontal Scalar Diffusivity

The [horizontal scalar diffusivity](#) command can be used to adjust the material specific scalar diffusivity parameters. The specifics of this command vary for different [scalar mixing models](#). This command can be used to increase or decrease the scalar diffusion in a particular area.

This is usually done to smooth instabilities in regions of the model that are not near the area of interest (such as the corners of offshore boundaries). Users should take care with using this command without a suitable understanding of the physical relevance of adjusting the momentum mixing spatially.

The following example shows a [global horizontal scalar diffusivity](#) being set for a Smagorinsky [scalar mixing model](#), with the Smagorinsky parameter being altered for two different materials.

```

scalar mixing model == Smagorinsky
global horizontal scalar diffusivity == 0.2

material == 10
  horizontal scalar diffusivity == 25
end material

```

8.5.6 Horizontal and Vertical Scalar Diffusivity Limits

The [horizontal scalar diffusivity limits](#) or [Vertical Scalar Diffusivity Limits](#) command allows the user to specify a minimum and maximum scalar diffusivity that changes by region. This can be used to set a limit in a particular area to control the scalar diffusion. This can be used in conjunction with or instead of the material [horizontal scalar diffusivity](#) command to increase or decrease the mixing of scalar constituents. [Vertical Scalar Diffusivity Limits](#) can be used to control the vertical diffusion i.e. to increase/decrease the effective vertical mixing if there is insufficient or too much vertical diffusion occurring in sections of your model.

The example below shows the diffusivity being set to a minimum level of mixing in the cells with the material ID of 10.

```

scalar mixing model == Smagorinsky
global horizontal scalar diffusivity == 0.2
global horizontal scalar diffusivity limits == 1.0, 99999.
global vertical scalar diffusivity limits == 1.0e-06, 99999.

material == 10
  horizontal scalar diffusivity limits == 100, 99999.
  vertical scalar diffusivity limits == 100, 99999.
end material

```

8.5.7 Bed Elevation Limits

The [bed elevation limits](#) allow the user to truncate the bathymetry in a given region to a minimum or maximum level. This is commonly used to sensitivity test limits, or to enforce wet elements where coarse mesh resolution has resulted in spurious wetting/drying areas.

This example shows three materials set to different limits. Material 1 is trimmed so that any elevations below datum level -10m are set to -10 m, without limiting higher levels. Material 2 is entirely set to +5m. Material 3 is trimmed to the same minimum level as material 1, but with an upper limit of -2m which might be below the tidal minimum, ensuring wet cells.

```
material == 1
  bed elevation limits == -10,9999
end material

material == 2
  bed elevation limits == 5,5
end material

material == 3
  bed elevation limits == -10,-2
end material
```

8.5.8 Spatial Reconstruction

The [spatial reconstruction](#) command can be used to revert an area to first-order calculations where the model is otherwise a second-order model. If a first-order model is being used, then this flag will have no effect. Second order models can be prescribed using the [spatial order](#) command.

This is commonly used near offshore boundary conditions to reduce the computational overhead, or to reduce instabilities if second order computations are not otherwise used there.

The following example shows three materials, with the first one (material ID 10) being set back to first-order.

```
bottom drag model == ks
global bottom roughness == 0.10
spatial order == 2,2

material == 10
  bottom roughness == 0.05
  spatial reconstruction == 0
end material

material == 2
  bottom roughness == 0.15
end material

material == 8
  bottom roughness == 0.05
end material
```

9 Modelling in 3D

Chapter Contents

9.1	Introduction	9-2
9.2	Overview	9-3
9.3	3D Layer Z Coordinate Options	9-4
	<i>Sigma Coordinates</i>	9-6
	<i>Z Coordinates</i>	9-7
	<i>Hybrid Z-Sigma Coordinates</i>	9-7
9.4	Density Coupling	9-9
9.5	Reviewing 3D Results	9-10
9.6	3D Boundary and Initial Conditions	9-11
9.7	3D Structure Conditions	9-12

9.1 Introduction

This chapter describes commands specific to the 3D module of TUFLOW FV. It provides general guidance on the 3D modelling process including sections on the additional data and visualisation challenges that 3D modelling requires. The chapter outlines 3D model geometry commands and provides a quick reference to navigate the relevant 3D sections concerning boundary, intial conditions and model outputs.

9.2 Overview

2D depth-averaged simulations can be an excellent approximation for many applications and are often used for the simulation of flooding, tsunamis and storm surge amongst others. However, there are cases where 3D modelling is required to sufficiently describe the observed flow characteristics, for example, in areas where there is significant variation in magnitude and direction of velocity with depth (counter currents, helicoidal flows) or where the effects of density become important (stratified environments or dense plume outfalls for instance).

When modelling in 3D it is recommended to commence the modelling in 2D to optimise and validate the model setup prior to moving to a 3D schematisation. Whilst the commands to enable 3D modelling in TUFLOW FV are relatively straightforward (refer to Section 9.3), when moving from 2D to 3D the modeller must carefully consider the additional data requirements, problem conceptualisation and visualisation challenges that an extra dimension often demands.

For example, to represent a thermally stratified system well, we need to choose a suitable vertical discretisation of the water column to adequately capture thermal gradients and the location of the thermocline using the [Vertical Mesh Type](#) and a suitable vertical layering (refer to Section 9.3). To reproduce these gradients, the modelled vertical mixing needs to be realistic, (i.e. not too much or too little) and model performance can be sensitive to the selection of the [vertical mixing model](#) and vertical [spatial order](#). To allow model calibration, we need accurate information on inflows and the temperature profile of the system as a function of depth, thus additional monitoring project costs may be required. To ensure surface temperatures are well represented energy flux between the water surface and atmosphere TUFLOW FV's atmospheric heat flux module may be required (using the [include heat](#) command) driven by site specific meteorological data including: incoming shortwave energy, longwave energy, surface wind speeds and relative humidity. Similar modelling considerations are required if modelling an estuarine or offshore environment where both temperature and salinity play a role.

Assignment of 3D initial and boundary conditions are also required to allow the specification of currents, salinities, temperatures and other scalar variables as both a function of time, space and depth. Section 10.5.5 provides specific guidance on the development and application of these boundary condition inputs.

To become familiar with the setup, simulation, analyses and visualisation of an example 3D model we encourage you to try out our [free 3D estuary tutorial, Tutorial Module 5](#) on the [TUFLOW FV Wiki](#).

9.3 3D Layer Z Coordinate Options

Designing your vertical layer discretisation and the coordinate scheme for your problem can often be one of the most important factors when undertaking 3D modelling. 3D simulations can be performed within TUFLOW FV using either sigma-coordinate or z-coordinate via the [Vertical Mesh Type](#) command. A third hybrid option, using a combination of sigma and z-coordinates is also available using a combination of the [Vertical Mesh Type](#) and [Surface Sigma Layers](#) commands. Figure 9-1 provides an example of each discretisation type and Table 9-1 provides an overview of the compatible commands for each coordinate type. Notably, each coordinate scheme has its strengths and weaknesses depending on the problem you are trying to solve. The sections below provide a brief overview and guidance on the use of each scheme.

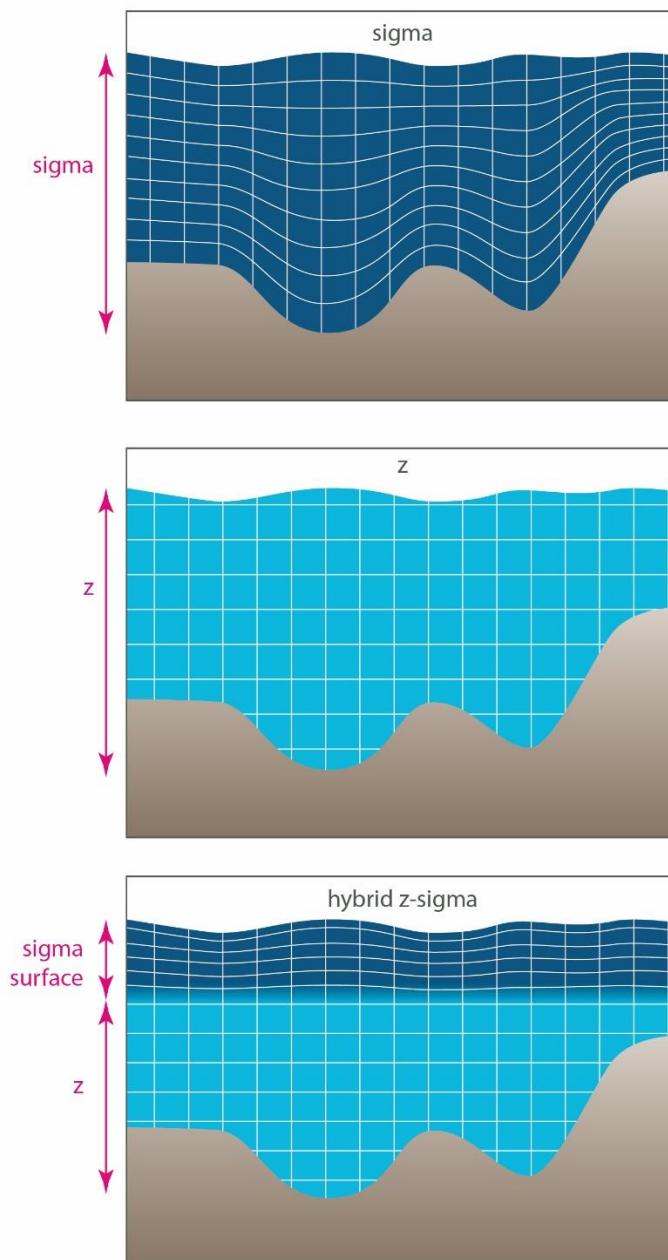


Figure 9-1 3D Model Vertical Discretisation Options

Table 9-1 3D Geometry Layer Options

Z Coordinate Type	Relevant Commands
Sigma	<p><u>Vertical Mesh Type</u> == Sigma</p> <p><u>Layer Face File</u> == <layer interface levels (.CSV)></p> <p><u>Cell 3D Depth</u> == <threshold depth (m)></p>
Z	<p><u>Vertical Mesh Type</u> == Z</p> <p><u>Layer Face File</u> == <layer interface levels (.CSV)></p> <p><u>Cell 3D Depth</u> == <threshold depth (m)></p> <p><u>Min Bottom Layer Thickness</u> == <dzmin></p>
Hybrid Z-Sigma	<p><u>Vertical Mesh Type</u> == Z</p> <p><u>Layer Face File</u> == < layer interface levels (.CSV)></p> <p><u>Surface Sigma Layers</u> == < Nsigma></p> <p><u>Cell 3D Depth</u> == < threshold depth (m)></p> <p><u>Min Bottom Layer Thickness</u> == < dzmin (m)></p>

Note: The assignment of z coordinates can be checked via the IDX3 attribute of the _mesh_check_R file (refer Table 12-5).

Sigma Coordinates

Using sigma coordinates, layers follow the bed profile with the spacing specified as a percentage of the water depth via the [Layer Face File](#). This layering is completed consistently for each cell throughout the model domain, regardless of the water depth. For example, a model with five equal sigma layers in 100m of depth will have 5 layers, each 20m thick. The same model in 1m of depth will have 5 layers, 0.2 m thick. Although this example assumes an even distribution of sigma layers, the user is free to distribute layers as they see fit. Optionally, to improve stability in areas of shallow flow or wetting and drying that are likely to be well mixed, a [Cell 3D Depth](#) threshold value can be applied, such that vertical fluxes between layers are switched off below these depths and the solution scheme effectively becomes 2D (depth-averaged). Compatible commands when using a sigma coordinate approach are provided in Table 9-1

Sigma layering is useful at providing high resolution layering typically near surface or near bed to capture boundary layer problems with a low computational overhead. i.e. bed following layers can be stacked tightly near the bed and water surface but left relatively sparse through mid-depths to avoid

unnecessary 3D cells. If using bed morphology coupling in 3D, then it is a software requirement to use sigma layering.

Limitations associated with the sigma coordinate scheme can occur in regions of highly sloping bathymetry. In these regions, sigma schemes can be prone to horizontal pressure gradient errors if horizontal grid cell resolutions are not well resolved. Additionally, in regions where thermal stratification is prevalent near the surface, i.e. lakes, a z coordinate or z-sigma hybrid may be more accurate at resolving a stratified water column.

Z Coordinates

Z coordinate layers are set at a series of fixed elevations using the [Layer Face File](#) in combination with the [Vertical Mesh Type](#) command. TUFLOW FV does not allow a z-layer to be switched off due to the water level dropping below its lower limit. Therefore, when using a Z coordinates scheme, it is important that the highest elevation within the [Layer Face File](#) remains wet, i.e. it's not at an elevation subject to wetting and drying.

The z coordinate approach can provide fine consistent resolution throughout the domain independent of water depth, which can be particularly useful if trying to model strong stratification in the upper layers of the water column. On the flipside, if the study area contains sloping bathymetry it can be challenging to efficiently provide high resolution near the bed (for example if near bed sediment processes are important) without including redundant high resolution in other parts of the model.

In areas of sloping bathymetry, z coordinates can result in thin layers close to the bed, that in turn can potentially lead to numerical instabilities. To avoid these issues TUFLOW FV offers the option to set a [Min Bottom Layer Thickness](#) that prevents a thin bottom layer from being generated when using the for z coordinate or z-sigma hybrid schemes.

Compatible commands when using a z coordinate approach are provided in Table 9-1

Hybrid Z-Sigma Coordinates

The hybrid z-sigma layering method defines a series of [Surface Sigma Layers](#) overlayed on a z scheme at depth. [Surface Sigma Layers](#) are evenly distributed sigma layers that reside between the water level surface and the topmost Z elevation specified in the [Layer Face File](#). The z-coordinate component of the hybrid scheme is discussed in the Z coordinates section above.

This method provides flexibility and stability in areas of wetting and drying not available via a z only coordinate system whilst providing the potential for high resolution near-surface exchange and gradient capturing. Compatible commands when using a z-sigma coordinate approach are provided in Table 9-1

An example excerpt using the hybrid z-sigma coordinate approach is provided in Figure 9-2. Using this example, a z-coordinate system is used below -3m (m in the project vertical datum). To avoid instabilities here the user has assumed that at all times, and at all cells, the model does not dry out below -3m. Below -3m there are 9 vertical layers between -3m and -16m and a single vertical layer below -16m terminating at the bed. At each cell location, if any layer faces are below the bathymetry they are ignored during the simulation. To avoid ‘skinny’ vertical layers truncated by the bathymetry (if the ZB at a cell is between two layer faces) then the user has used a [Min Bottom Layer Thickness](#) cell threshold 0.5 m. This means that we should not have any bottom layers in our 3D solution at the bed less than 0.5m.

Four equally distributed sigma layers are specified between -3m and the water surface, noting that these layers may stretch and contract in response to changes in water level as the simulation progresses. In regions of wetting and drying a [Cell 3D Depth](#) threshold is applied to disable 3D calculation in well mixed regions of the coastal zone, in our model case tidal mudflats adjacent to the main estuary channel.

!	
!3D Geometry Commands	
Vertical Mesh Type == z	
Layer Faces == ..\model\csv\3D_Z_Layers_003.csv	
Surface Sigma Layers == 4	
Min Bottom Layer Thickness == 0.5	
Cell 3D Depth == 1.0	
	1 z
	2 -3
	3 -4
	4 -5
	5 -6
	6 -7
	7 -8
	8 -10
	9 -12
	10 -14
	11 -16

Figure 9-2 Hybrid Z-Sigma example setup (left) with z layer face file (right)

9.4 Density Coupling

For 3D simulations, you can choose to run a barotropic configuration (the default) or with the baroclinic pressure-gradient terms optionally activated to allow the hydrodynamic solution to respond to temperature, salinity and sediment induced density gradients using the [Include Salinity](#), [Include Temperature](#) and/or [Include Sediment](#)³ commands. These commands can be used together or in isolation. For example, setting `Include Temperature == 1,1` and `Include Salinity == 1,1` will result in both temperature and salinity being used for fluid density and baroclinic pressure gradient calculations, a common combination for modelling in the ocean. The influence of these terms may be significant in estuarine environments for example, where fluvial and marine waters meet, and stratification is known to occur.

Figure 9-2 below provides an example model output from a coastal estuarine model with both salinity and temperature density coupling enabled in addition to atmospheric heat exchange. The model uses a hybrid z-sigma layering system. Fresh river water enters the systems at chainage zero at the left of the plot. At the right the ocean tide is applied. The density effects at the mixing interface of river and ocean is clearly visible, with relatively buoyant river water moving over the salt water “wedge” below.

If running TUFLOW FV in baroclinic mode, it is recommended that the user complete sensitivity testing on the a range of vertical layering discretisations, experiment with the vertical [spatial order](#) and vertical turbulence and diffusivity options (refer Section 7.2) early in the modelling process to allow you to quickly optimise your setup and runtime for the problem at hand. Further guidance on 3D baroclinic modelling is also in the overview section of this chapter (Section 9.2).

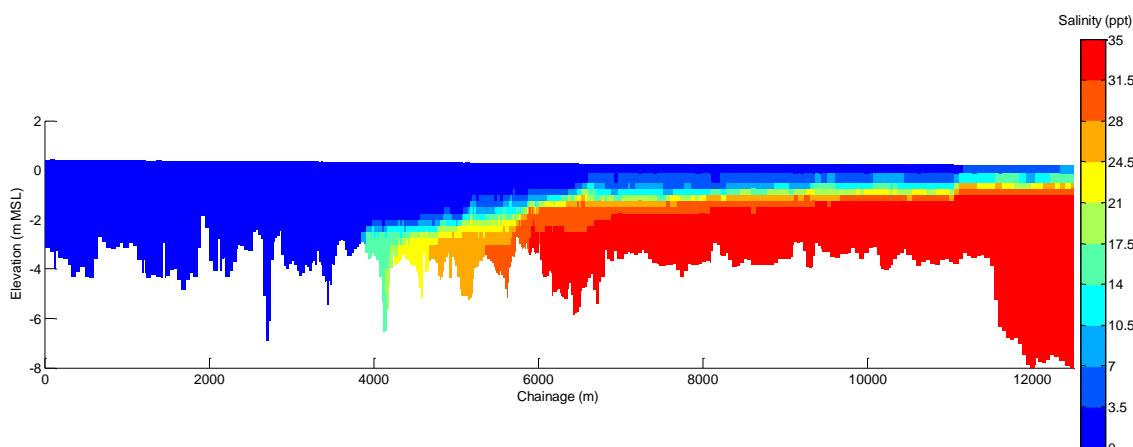


Figure 9-3 Salinity long section in coastal ‘salt wedge’ estuary.

³ Please contact support@tuflow.com for further information on TUFLOW FV’s sediment transport module.

9.5 Reviewing 3D Results

This section provides key points to consider when visualisation 3D results. For more detailed instruction on using the various 3D model outputs please refer to Chapter 12.

A commonly underestimated implication of 3D modelling is the need to post-process results in a way that is appropriate to the scope and objectives of the modelling project. This can be for the final publication of modelling outputs but is also key to resolving potential issues and instabilities within a model, and for usual quality assurance purposes.

While all basic outputs can be provided in 2D by depth-averaging the results, modellers should take care that this is an appropriate way to visualise and investigate your study area. Consider the following simple problem (refer Figure 9-4): a tidal river with a complex vertical profile of currents, with near-surface (fresh-water) currents flow downstream and near-bed (salt-water) currents flowing upstream. If these two overall flow magnitudes are similar, then the depth-averaged currents will be zero!

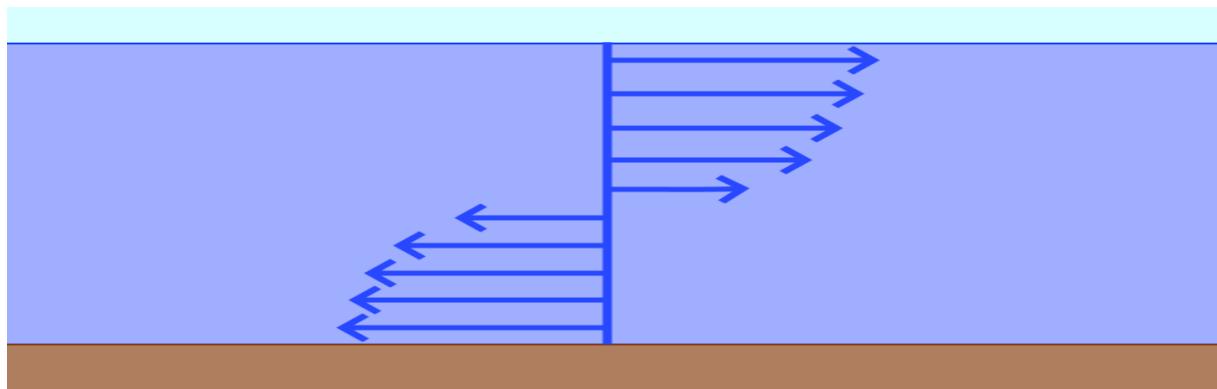


Figure 9-4 Counter current velocity at depth. Considerations for result depth averaging.

As there are many ways to visualise data, and a myriad combination of different features to be looking for, it is important that any tools used are as flexible as possible. For the visualisation of 3D results the Unidata NetCDF format is commonly used within TUFLOW FV due to NetCDF's ability to efficiently handle multi-dimensional data structures. To assist with the visualisation of 3D NetCDF results, we offer [MATLAB](#) and/or [Python toolboxes](#) with example scripts to help you get started. It is recommended that modellers investigate the usage these tools as they allow for the most customised requirements to be developed by the modeller in order to investigate specific aspects of each model.

TUFLOW FV also offers a range of depth averaging approaches that allow 2D visualisation of 3D results via the familiar DATV or XMDF outputs that can be readily opened in SMS and QGIS. For more detailed instruction on using the various model outputs please refer to Chapter 12.

9.6 3D Boundary and Initial Conditions

When transitioning to 3D, users need to consider the implications for any boundary conditions. Many simple boundary conditions make assumptions about how they are applied in the vertical dimension, with the user having the option to modify these. Additionally, the most complicated boundary conditions that can be applied to TUFLOW FV are the fully 3D boundaries, which are all specified using NetCDF files. These can either be a 3D gridded NetCDF file applied across the model domain, or a chainage vs. depth ‘curtain’ NetCDF that are used for open boundary conditions. The key differentiating and complicating factor is the addition of the boundary condition varying in the vertical dimension and the ability to visualise and check their correct application.

When applying 3D boundary conditions, users need to ensure that the depth values in the boundary condition file make sense relative to the bathymetry of the model. This is particularly noticeable with gridded 3D boundaries, or curtain boundaries where dummy values are used in the NetCDF file below valid depths. If the coarseness of the boundary condition means that these ‘invalid’ depths are actually valid in the model bathymetry then the dummy values will be applied to the model. Users should aim to ‘pad down’ the last appropriate real value, so that any variation between the boundary condition bathymetry and the model bathymetry does not cause problems.

The 3D boundary and initial conditions available within TUFLOW FV are provided in Section 10.5.5 on the Boundary and Intitial Condition Chapter (Chapter 10).

9.7 3D Structure Conditions

When modelling in 3D, the user can specify the type of depth averaging that occurs at a structure via the [vertical coordinate type](#) and [vertical distribution file](#) commands. Most structures in TUFLOW FV provide this option to allow the structure to act only within a specified depth range of the upstream and downstream sides of the structure. This can be particularly important when using the advection dispersion module for controlling how conserved tracers move through a system. For example, a pump intake at the bottom of a poorly mixed, deep dam is likely to have very different water quality characteristics to the surface.

When implementing 3D models with hydraulic structures, modellers need to exercise care to understand the assumptions and approximations required to transfer mass and momentum fluxes at the structure. For example, the flow distribution may be based on, depth, cell width or area upstream and downstream of the structure. These approximations are typically based on simple vertical profiles of velocity and constant vertical profiles of scalar terms. However no fully-3D structure controls have been implemented that guarantee that the vertical profile of the water will be maintained through the structure.

Section 11.2 provides an overview and several examples of using 3D commands within a [structure](#) block.

10 Boundaries and Initial Conditions

Chapter Contents

10.1	Introduction	10-2
10.2	Boundary Control Block	10-3
10.3	Assigning Boundary Locations	10-4
10.4	Input Data for Boundary Conditions	10-5
10.4.1	Time Interpolation	10-5
10.4.2	Spatial Location	10-6
10.4.3	BC Headers	10-6
10.4.4	BC Scale and Offsets	10-8
10.4.5	BC Default	10-8
10.5	Boundary Condition Types	10-10
10.5.1	Cell specified boundaries	10-10
10.5.2	Nodestring specified boundaries	10-11
10.5.3	Mesh specified (global, spatially/temporally varying)	10-11
10.5.3.1	<i>Grid Definition Block</i>	10-12
10.5.3.2	<i>Parametric Tropical Cyclone Wind and Pressure Field</i>	10-14
10.5.4	Moving boundaries	10-14
10.5.5	3D Boundary Conditions	10-15
10.5.5.1	<i>Modifying Basic Boundary Conditions</i>	10-15
10.5.5.2	<i>'Curtain' Boundary Conditions</i>	10-15
10.5.5.3	<i>Profile Boundary Conditions</i>	10-15
10.5.5.4	<i>OBC 3D Gridded Boundary Conditions</i>	10-16
10.5.6	Advanced 3D Gridded Boundary Conditions	10-17
10.6	Boundary Types	10-18
10.6.1	Boundary Sub-Types	10-21
10.7	Initial Conditions	10-23
10.7.1	Globally Specified	10-23
10.7.2	Spatially Varying	10-24
10.7.3	Restart Files	10-25
10.8	Transport File	10-26

10.1 Introduction

This chapter describes the assignment of model boundary and initial conditions including restart file setup and the use of hydrodynamic transport files which allows de-coupling of the hydrodynamics and any advection-dispersion, water quality, sediment and particle tracking modelling for improved efficiency when modelling multiple scenarios.

10.2 Boundary Control Block

Boundary conditions are defined in the simulation control file within a [boundary condition](#) block, requiring a unique block definition for each input to the model (type and location). TUFLOW FV offers a wide variety of boundary condition types. Basic hydrodynamic boundary condition types are summarised in Table 10-2. Advanced boundary conditions types used during advection dispersion, 3D, sediment transport/morphology and water quality modelling are listed in Table 10-3. The tables list the boundary condition type ID, its description, method of application to the model, boundary condition data file format and the default column headers for the input file.

When defining a boundary condition block the first line is used to specify the input type, the location and reference file that contains the relevant boundary condition data. Where boundary conditions inputs vary from default values these details are specified within the boundary condition block.

When developing complex models with many boundary conditions, [Include](#) files are recommended to manage/categorise data inputs. Some example boundary condition blocks are shown below. Worked examples are available via the TUFLOW FV tutorial models on the TUFLOW FV Wiki: <http://fvwiki.tuflow.com>

```
!
!BOUNDARY CONDITIONS

BC == QC, 6837, 15228, ..\bc\Trib_Q.csv      !Flow boundary condition, point location (x,y), bc file
  BC header == Date, QC                         !Headers of columns to read from file
End BC                                         !End bc block

BC == WL, 1, ..\bc\Tide.csv                    !WL boundary, location (nodestring), bc file (using default headers)
End BC                                         !End bc block

Grid definition file == ..\bc\Wind_Grid.nc    !File containing information of grid for later BC specifications
  grid definition variables == lon, lat        !Variable names for the grid (x-variable, y-variable)
  grid definition label == grid_01            !A name to refer to this grid (user specified)
End Grid                                       !End grid block

BC == W10_GRID, grid_01, ..\bc\Wind_Grid.nc   !Gridded Wind Boundary, grid label, bc file (netcdf with wind grid)
  BC header == Time, u, v                      !Names of variables to read from file (Time, W10_x, W10_y).
End BC                                         !End bc block

BC == MSLP_GRID, grid_01, ..\bc\MSLP_Grid.nc !Gridded Air Pressure Boundary, grid label, bc file
  BC header == Time, mslp                     !Names of variables to read
  BC scale == 0.01                            !Scale factor for mslp variable (converts Pa to hPa here)
End BC                                         !End bc block
```

Figure 10-1 Boundary Condition Block Examples

10.3 Assigning Boundary Locations

Four methods are available for applying model boundary condition inputs:

- 1) Cell: Applying the input boundary condition at a single cell location.
- 2) External Nodestring: Applied along a series of cell faces, defined by a nodestring.
 - a. Nodestrings can be defined within a model either when the model mesh is created (refer to Figure 8-6 or by using the [Nodestring Polyline File](#) or [Read GIS Nodestring](#) command).
- 3) Polygon: Applied to a series of cells with centroids within a polygon.
- 4) Gridded: Gridded NetCDF format data can be used to define temporally and spatially varying inputs over an entire computational mesh or a subregion. If using gridded data, a [grid definition file](#) and [grid definition variables](#) are first required to define grid coordinates and variable names within the input file. These commands need to be specified prior to the boundary condition block
- 5) Global: Common values can be specified across the entire model using CSV file inputs.

Not all of the above methods are available to all boundary condition types (water level boundaries can only be applied to nodestrings for example). For those that offer multiple options (for example an inflow boundary), there are different BC type commands to get these different behaviours (for example Q for inflow to a nodestring, QC for inflow to a cell and QC_POLY for inflow to a group of cells within a polygon).

10.4 Input Data for Boundary Conditions

Most boundary conditions require an input from a separate file with the conditions required. In most simple cases these are timeseries of the required input (such as water level or flow) within a CSV file. Occasionally this is a more complicated NetCDF file that allows conditions to be specified as spatially and temporally varying. The following sections provide guidance on the input of timeseries boundary data.

10.4.1 Time Interpolation

For any time series (whether spatially constant in a CSV file, or gridded in a NetCDF file) TUFLOW FV interpolates to the current timestep using linear interpolation. Values before or after the specified timeseries will use a constant value extrapolation, assuming the first or last timestep respectively as the constant value.

Additionally, a [BC update dt](#) command can be used to control how often this interpolation is done. All timesteps after one boundary condition update, but before the next one will use a constant value for that boundary condition. If no boundary condition update dt is specified, TUFLOW FV will interpolate the specified values at every computational timestep of the model.

Users must take care that this behaviour is understood and recognised in their model. Common errors that can arise when ignoring this behaviour are as follows:

- For an inflow release that only covers part of the modelled period not having a ‘zero’ value at the start and end can lead to the boundary condition continuing unintentionally. The below figures show a case where the desired behaviour is a flow release of 10 m³/s for 7 days. The left figure will continue the flow rate before the first timestep and after the last one. The example on the right has additional timesteps just before and just after the release to return the flow rate back to zero.

Time	Flow
01/01/2019 00:00	0
01/01/2019 00:01	10
07/01/2019 23:59	10
08/01/2019 00:00	0

- Occasionally (due to rounding errors or varied inputs) a boundary condition file will have a timeseries where some timesteps are repeated or are not in monotonically increasing order. As the interpolation method in TUFLOW FV does not account for the removal of any timesteps, and the model will cease with the error message ‘*TIME column must be monotonically increasing*’.
- If the boundary update timestep is too coarse (or the main timestep of the model), it is possible that not all inputs from a boundary condition will be used within the model. Users must take into consideration that the model is not reading a timeseries exactly, but is interpolating it onto

a different timestep, which may change the overall effect. For example, a flow input with large peaks that are not aligned with the timestep of the model will not have the same peak flow rate, or total inflow volume as the raw input data.

10.4.2 Spatial Location

TUFLOW FV will attempt to map most boundary condition types onto the centroids of cells within the model. In the case of nodestring boundary conditions, the values are applied to so called ‘ghost cells’ that exist just outside the model domain and are only connected to their nearest real cell within the model domain. Ghost cells are internal to the engine and are not accessible or visible to the modeller.

If a boundary condition is outside of the model domain, it will be ignored and no mapping of this to the nearest cell is available. Warnings will be thrown within the log file when this occurs, but may generally happen for the following reasons:

- Model reschematisation: If users change the shape of their mesh without adjusting the x and y coordinates of the inflows then they may occur outside the desired location.
- Coordinate projection: If a mesh uses geographic (lat/lon) coordinates but an inflow is specified at a UTM (eastings/northings) location then it will be seen as outside the model. Note: If running the model with GIS integration then this issue should be caught by *ERROR 0305 - Projection of .mif/shp file is different to that specified by the MI/SHP Projection == command will occur.*
- Moving BCs interpolating in space: Moving boundary conditions will interpolate their position from the specified file with respect to the current timestep. In concave mesh sections, this can cause an intermediate coordinate to fall outside of the model domain.

The above problems can also affect gridded NetCDF boundary conditions. Users must take care that the boundary conditions are being applied to the model in the way they expect. Usually this is best achieved by first running simplistic model cases, to QA the boundary conditions that are being applied.

10.4.3 BC Headers

TUFLOW FV allows users a great deal of flexibility in how they specify their input files. The requisite variables in a CSV or NetCDF file can be ‘called’ (by the header line in a CSV or the variable name in a NetCDF) whatever the user wishes. However, if these variables differ from the default variable names, then users must tell TUFLOW FV which headers to look for.

This is powerful as it allows users to have multiple variables in files that may have names that are more convenient to the project than a list of default inputs. The examples below show a set of desired input names and the corresponding default input header names for an inflow boundary (QC) with both salinity and two sediments modelled.

The way TUFLOW FV interprets the headers is with a [bc header](#) command in the relevant [BC](#) block. The order of these headers is what TUFLOW FV uses to map the header name to a tracked variable

within the model. Users must take care that these are specified in the correct way, for example TUFLOW FV does not throw a warning if you accidentally name the temperature variable ‘Salinity’.

Desired Input:

Time	Flow	Salinity_surface	Silt_concentration	Sand_concentration
01/01/2019 00:00	0	0	0	0
01/01/2019 00:01	10	0	200	10
07/01/2019 23:59	10	0	200	10
08/01/2019 00:00	0	0	0	0

Default Input

Time	Q	SAL	SED_1	SED_2
01/01/2019 00:00	0	0	0	0
01/01/2019 00:01	10	0	200	10
07/01/2019 23:59	10	0	200	10
08/01/2019 00:00	0	0	0	0

The expected default order of the variables in all files follows a standard order that differs for different boundary conditions (OBC conditions specify water level and currents for example, whereas Q conditions just specify flow). The main variables are specified in Table 10-2 and Table 10-3 followed by the tracers in a standard scalar order:

Salinity (if modelled), Temperature (if modelled), Sediment_1, ..., Sediment_n (If Sediment modelled), Tracer_1, ..., Tracer_n (If tracers modelled), WQ_1, ..., WQ_n (If WQ modelled).

The above are only included if they are being modelled. The previous input examples are modelling salinity and two sediments, so the BC headers would look as follows:

```
bc header == Time, Flow, Salinity_surface, Silt_concentration,
Sand_concentration
```

If the user decided to switch off tracking salinity (include salinity == 0,0) then they would also need to update the bc header line as follows:

```
bc header == Time, Flow, Silt_concentration, Sand_concentration
```

If they did not update this line, the model would interpret the third header as the input for SED_1, which would have been ‘salinity_surface’. All other variables after that would also be reading the incorrect columns.

The TUFLOW FV log file prints out the default expected names for each variable, followed by the headers that the user has specified. This can be used to ensure that the headers have been specified correctly.

10.4.4 BC Scale and Offsets

[BC scale](#) and [BC offset](#) allow modellers to modify the raw boundary condition data as it is read into TUFLOW FV. The scale is applied first, multiplying all data that it corresponds to by the relevant scale factor. The offset is applied afterwards with the specified value added onto the raw data within the file.

These inputs can be used to quickly sensitivity test boundary condition forcing if need be. Examples are scaling up a flow boundary condition by 10% to represent climate change or elevating an offshore tide boundary by a future sea-level-rise offset without needing to create a new copy of any of the boundary condition files. Note that each header within a boundary condition ([excluding time](#)) gets a separate scale and offset as specified with comma separated values.

```
bc == Q, 2, ..\bc\upstreamFlow_000.csv
  bc header == Time, US_Q, SAL, TEMP
  bc scale == 1.1, 1.0, 1.0 ! Scale flow by +10% but not salinity or temperature
end bc

bc == WL, 1, ..\bc\offshoreTide_000.csv
  bc header == Time, WL
  bc offset == 0.92      ! 100 year sea-level offset to water level
end bc
```

Another example is for splitting a boundary condition into separate inputs spatially or vertically without needing to create different boundary condition inputs. The following example shows distributing a cell flow boundary with 50% of the flow in the top 1m and the remainder in the bottom 1m.

```
bc == QC, 120.9962, 14.0092, ..\bc\catchmentInflow_000.csv
  bc header == Time, flowrate, SAL, TEMP
  bc scale == 0.5, 1.0, 1.0
  vertical coordinate type == depth
  vertical distribution file == ..\bc\InflowDistribution.csv
end bc

bc == QC, 120.9962, 14.0092, ..\bc\catchmentInflow_000.csv
  bc header == Time, flowrate, SAL, TEMP
  bc scale == 0.5, 1.0, 1.0
  vertical coordinate type == height
  vertical distribution file == ..\bc\InflowDistribution.csv
end bc
```

..\bc\InflowDistribution.CSV

HEIGHT	DEPTH	WEIGHT
0	0	1
1	1	1
1.01	1.01	0

10.4.5 BC Default

[BC default](#) can be used to control what value gets applied when TUFLOW FV reads a null-value from a boundary condition input, or it cannot find the header that the modeller specified.

It is useful when the modeller recognises that a boundary condition needs a header in place but doesn't have a particular value to assign to it. An example is for a model that is reading a source term of sediment into a model that is also modelling salinity and temperature. The simple source term of sediment in this case is not adding any salt or heat to the model, but still requires an input of the header. It is recommended that these headers are set to a name that is unlikely to be in the boundary condition file by coincidence – this example uses the header 'Dummy'. The header name 'Dummy' does not exist in the boundary condition CSV file and the default values will be applied. This saves the modeller from needing to add additional columns in the boundary condition file for variables that do not change.

```
bc == FC, 120.9962, ..\bc\SedimentSource_000.csv
bc header == Time, Dummy, Dummy, SED_1
bc default == 35.0, 25.0, NaN
end bc
```

This can be a useful feature but is also often a source of errors that can be hard to catch at times. Modellers should always check the log file output to ensure that the correct headers have been found and that the appropriate defaults have been applied if need be.

10.5 Boundary Condition Types

Many different boundary conditions are available within TUFLOW FV and can be used to force different locations of the model with increasing complexity as required. Boundary conditions range from the simplest boundaries that specify a constant timeseries at a point location, along a nodestring, or constant globally, up to the most complicated boundaries that allow the user to specify conditions (such as inflows) that differ over the horizontal domain, throughout the water column vertically and change with time within one gridded three-dimensional boundary condition file. Simpler boundary conditions can all be easily specified with CSV files, whereas more complicated boundary conditions (all gridded boundary conditions and most that vary in the vertical) require a NetCDF file for input. Specifications of the NetCDF input requirements can be found in Appendix E. Users unfamiliar with NetCDF files are advised to contact support@tuflow.com for further information on generating these.

Whether a boundary condition type can be applied as a cell, nodestring, grid or globally is described in the ‘BC Method’ column of Table 10-2 and Table 10-3.

10.5.1 Cell specified boundaries

For cell specified boundary conditions, users can specify a coordinate (x,y) in the model domain and the boundary condition is applied to the cell that this coordinate falls within. These boundary conditions are commonly used for point inflow source terms, such as the outfall from a drain. For more advanced models, other source/sink terms can be applied to individual cells, such as influxes of salt, heat or sediment (without an inflow of water), or even additional turbulence or shear stress.

```
!Point Inflow Location
bc == QC, 293569, 6177874, M01_002.csv
  bc header == time_hr,FC04
end bc
```

All inflows from cell boundary conditions are applied as a source volume/mass input only and have no direction or momentum added to them. By default, all cell boundaries are averaged over the full depth of the water column, though this can be changed by specifying a [vertical averaging type](#).

For advanced 3D models, some profile type boundaries are available, which are NetCDF files that can specify the cell source/sink timeseries as changing with depth. These require a vertical coordinate type and an additional variable within the NetCDF file of this vertical dimension.

Lastly, a special cell boundary condition, ‘CP boundaries’, exists. These specify a vertical profile of scalar concentrations in a cell, but do not change with time (i.e. these are fixed constants for that cell). These are commonly used for warming up a model using known profiles at specific points. By specifying CP boundaries at multiple locations within a model and setting diffusion parameters to high (non-physical) levels the internal diffusion parameterisation can then effectively interpolate between the fixed CP points and be used to create a restart file.

10.5.2 Nodestring specified boundaries

Nodestring specified boundaries allow users to specify the conditions along an edge of the model mesh as defined by a nodestring (refer Figure 8-6). The most common uses for these are to apply a boundary water level condition or a boundary inflow. For more complicated coastal and ocean models, nodestring boundaries can be used to specify the ocean boundary conditions including the water level, ocean currents, salinity and temperature, changing with depth as required.

```
!Nodestring Specified Boundary
bc == WL, 2, M01_002.csv
  bc header == time_hr,DS
end bc
```

Much like the cell-specified boundary conditions, simpler nodestring BCs can be specified using CSV files while more complicated conditions are available using NetCDF files. CSVs are available for conditions that either don't vary along the length of the nodestring or vary constantly (by specifying conditions at each end, with TUFLOW FV linearly interpolating between). When boundaries need to vary non-linearly along the nodestring, or vary with depth, NetCDF files can be used.

Common pitfalls when specifying such boundaries are ensuring that the conditions are specified as the user expects. Many nodestring boundary conditions have different [sub-types](#) available that affect the distribution of flow both across the nodestring and with depth. [Sub-type](#) selection can also control whether flow boundaries bring on the momentum or are treated more like a volume/mass source term. Users must also make sure that the nodestring boundaries do not over-specify the conditions. This occurs when any slight variances between the model internal conditions and the boundary conditions conflict near the boundary and can cause instabilities or reflecting of energy from the boundary, affecting the results. For ocean boundary conditions, [sub-types](#) are available to 'relax' some of the conditions to minimise these issues.

10.5.3 Mesh specified (global, spatially/temporally varying)

Some boundary conditions are required to be specified over the whole model domain. The simplest of these can be specified with a timeseries and are applied to every cell in the model. An example of this is the QG (global flow) boundary condition, which specifies a timeseries of flow per unit area (m^2) that is applied across the whole model. For example, a QG boundary can be used to specify the effects of rain and/or evaporation.

Large grid NetCDF boundary conditions often need to cover the whole model domain. In these cases, the user must keep in mind the extent of the relevant boundary condition grid (as specified by the [grid definition block](#) and also refer Section 10.5.3.1). If the grid does not cover the whole domain, then the areas outside of the grid's extent will not be extrapolated but will be set to the default relevant value. This can potentially cause issues for the model. In an MSLP_GRID boundary condition for example, the cells outside of the grid will be set to the reference pressure. This may cause an air-pressure gradient where the boundary condition grid ends, and consequently affect the water levels within the model.

10.5.3.1 Grid Definition Block

Grid definitions are used to describe the grid on which gridded boundary conditions types are applied. This grid definition creates an interpolation mapping from the grid to the cell-centres, or the boundary nodestrings as required. Please note the grid definition file method is the only option available for 3D grid definitions. 2D grids are also supported using this command.

Grid definitions are within a grid definition block which are typically used for NetCDF inputs as follows:

```
Grid definition file == <FILE_NAME>
Grid definition label == <LABEL>
Grid definition variables == <X_VARIABLE>, <Y_VARIABLE>, <Z_VARIABLE>
Vertical Coordinate Type == < {elevation} | Depth | Sigma | Height>
Cell Gridmap == < {1} | 0 >
Boundary Gridmap == < 1 | {0} >
Suppress Coverage Warnings == < 1 | {0} >
End Grid
```

Table 10-1 Grid Definition Block Commands

Command	Options	Description
Grid definition file	FILE_NAME	The NetCDF file containing coordinates used to define the grid to which gridded boundary conditions types are applied.
grid definition label	LABEL	Name to apply to this grid and can be referred to by multiple boundary conditions later within the model build process.
Grid definition variables	X_VARIABLE, Y_VARIABLE, Z_VARIABLE.	This command specifies which variables should be read from the NetCDF to create the grid map. If the inputs are only required on a 2D grid (e.g. wind) then the Z_VARIABLE can be omitted.
Vertical Coordinate Type	{ Elevation } Depth Sigma Height Default == Elevation	Vertical coordinate convention. For more information on the respective coordinate types, please refer to: https://fvwiki.tuflow.com/index.php?title=Depth_Averaging_Results
cell gridmap	{1} 0	If set to 0, TUFLOW FV will not calculate the interpolation weightings of the cells in the model for this gridded boundary condition (may be more efficient for grids that are only going to be applied to the boundaries and not internal model domain).
boundary gridmap	1 {0}	Set to 1 to calculate interpolation weightings from the grid onto the boundary nodestrings (this is required for OBC_grid boundary conditions that are to be later applied to nodestrings).

Command	Options	Description
<u>suppress coverage warnings</u>	1 or 0 Default == 0	If set to 1 suppresses warnings if the grid does not cover the entire TUFLOW FV domain. It may be more efficient in situations that the grid only needs to cover a small number of the cells in the model.
End Grid	NA	'Complete' the grid block opened by the Grid Definition File command.

The following .fvc example shows a typical grid definition file setup, in this example the NCEP Climate Forecast System Reanalyses (CFSR).

```
!Specifies a netCDF location and filename that defines grid coordinates
grid definition file == ./NPD_NCEP_CFSR_dlwr_Jan2011toDec2014.nc
  grid definition variables == lon, lat
  grid definition label == ncep
end grid
```

This grid definition is then later used to assign gridded boundary conditions such as wind using a bc block:

```
!Wind velocity at 10m (m s^-1)
bc == W10_grid, ncep, ./NPD_NCEP_CFSR_wind_Jan2011toDec2014.nc
  bc header == time,u,v
  bc update dt == 900.
  bc time units == hours
  bc reference time == 01/01/1990 00:00
end bc
```

The following .fvc example shows a typical 3D grid definition file setup, in this example the HYCOM Global Circulation Model.

```
! HYCOM exp 90.9
grid definition file == HYCOM_Jan2013_Mar2013.nc
  grid definition variables == x, y, z
  grid definition label == hycom
  boundary gridmap == 1
end grid
```

This grid definition is then later used to assign gridded boundary conditions such as wind using a bc block:

```
! Gridded BC for Input Global Ocean Model HYCOM
bc == OBC_GRID, hycom, HYCOM_Jan2013_Mar2013.nc
  sub-type == 6
  bc header == time,ssh,u,v,salinity,temperature
  bc nodestrings == 1
  bc update dt == 3600.
  bc time units == hours
  bc reference time == 01/01/1990 00:00
  bc offset == -0.33, 0.0, 0.0, 0.0, 0.0 ! ssh, u, v, salinity, temperature
  vertical coordinate type == depth
end bc
```

10.5.3.2 Parametric Tropical Cyclone Wind and Pressure Field

The parametric wind and pressure field model can be applied globally to the TUFLOW FV meshing using the CYC_HOLLAND boundary condition type (Refer Table 10-3). This boundary type is input via CSV files with the following headers:

- Time (hours or ISODATE)
- Cyclone position [X]
- Cyclone position [Y]
- Central pressure [P0]
- Ambient pressure [PA]
- Radius to maximum winds [RMAX]
- Holland Peakedness [B]
- Air density [RHOA]
- Km (mean near surface wind speed / gradient level wind speed) [KM]
- Line of maximum wind [THETMAX]
- Asymmetry factor [DELTAFM]
- Latitude [Latitude]
- Background wind X component [WBGX]
- Background wind Y component [WBGY]

The boundary is specified as follows in the .fvc file. The input file Example_Holland_Cyclone.CSV is also provided below.

```
bc == CYC_HOLLAND, ..\bc\Example_Holland_Cyclone.csv
end bc
```

Time	X	Y	P0	PA	RMAX	B	RHOA	KM	THETMAX	DELTAFM	Latitude	WBGX	WBGY
26/03/2010 16:30	137.756	-8.847	950	1010	50000	1.5	1.17	0.7	160	1	-8.847	0	0
1/04/2010 0:00	140.131	-14.741	950	1010	50000	1.5	1.17	0.7	160	1	-14.741	0	0

[Tutorial Module 4](#) on the [TUFLOW FV Wiki](#) provides an example model using the Holland tropical cyclone model boundary input.

10.5.4 Moving boundaries

Moving boundary conditions are available for the basic [cell specified boundary conditions](#) (ones that can be specified using a CSV file). These require an additional set of columns corresponding to the coordinates (x,y) of the source/sink term and can change with time. The cell that the source is applied to will be based on the cell that the interpolated coordinate falls within. This is most useful when simulating a moving source such as the sediment released by a dredger, which may move slowly with time.

When using moving boundary conditions, users must take care with a few things:

- Currently the depth at which sources are released must be kept constant. This is based on the [vertical averaging](#) commands in the boundary condition block (default over the whole water

column). If it is important to have boundaries moving in the vertical, users can use multiple boundaries to achieve this. For more complicated cases, please contact support@tuflow.com.

- Interpolation between the specified timesteps can cause problems, these are described further in Section 10.4.2

10.5.5 3D Boundary Conditions

10.5.5.1 *Modifying Basic Boundary Conditions*

By default, boundary conditions for inflows assume a depth-averaged approach. However, this can be modified by using ‘depth averaging limits’ to effectively input over custom vertical range using a combination of the [vertical coordinate type](#) and [vertical distribution file bc](#) block commands. An example of how this is completed is provided in the example below.

```
bc == QC, 145.4324, -19.1345, ..\bc\simple_inflow.csv           ! Basic cell inflow BC
  bc header == time, Flow                                         ! Read in headers
  vertical coordinate type == height                            ! Apply vertical coordinates from the bed
  vertical distribution file == ..\bc\flow_distribution.csv    ! Read weightings from this file
end bc
```

flow_distribution.csv	
height	weight
1	0
1.001	1
5	1
5.001	0

A similar approach can be adopted when using the QC, FC, QCM, FCM, QC_POLY, FC_POLY, QC_GRID, FC_GRID boundary inputs.

10.5.5.2 *‘Curtain’ Boundary Conditions*

Curtain boundary conditions allow users to specify a boundary condition that varies along a nodestring (chainage) and with depth (if required), within a NetCDF file (Appendix E, Item 3 outlines the required format of input curtain files). These are useful for specifying detailed Ocean Boundary Conditions offshore, but without the need for a full 3D gridded boundary file (see Section 10.5.5.4). They require some additional geometry terms in the main headers that are read in, that correspond to the chainage (in meters along the nodestring) and the vertical coordinate, which corresponds to the vertical coordinate type. In this way, they are ideal for nesting a smaller model within the domain of a larger 3D model and applying the relevant conditions from this at a nodestring.

The available curtain (_CURT) input boundaries are provided in Table 10-3.

10.5.5.3 *Profile Boundary Conditions*

3D profile boundary types are similar to curtain boundaries, however they have do not require/use a chainage dimension but instead boundary conditions are applied uniformly along the boundary nodestring (refer to Appendix E, Item 4 outlining the required format of depth varying input curtain files). They are used where spatially uniform boundary conditions are appropriate and at the same time both temporal variation and/or depth variation are required. If no depth variation is required then

boundary condition specification using a CSV file and compatible boundary condition type (for example, Q, QC, WL should suffice).

The available profile (_PROF) input boundaries are provided in Table 10-3.

10.5.5.4 OBC 3D Gridded Boundary Conditions

3D gridded boundary conditions are commonly used to nest a TUFLOW FV model within a coarser scale ocean model, using a combination of 3D open boundaries and initial conditions derived from the parent model. 3D boundaries such as these require NetCDF boundary condition files.

For 3D gridded NetCDF files, users must specify the grid geometry prior to specifying the boundary condition. This is so that multiple boundary conditions on the same grid can use the same set of interpolation calculations (see Section 10.5.3 for more information on gridded boundary conditions). Typically, 3D gridded boundary files specify the full ocean boundary conditions (water level, current velocity components and any requisite scalars such as salinity and temperature). An example is shown below for a grid definition and a corresponding Ocean Boundary Condition (OBC) grid applied onto two different nodestrings:

```
grid definition file == grid_file.nc          ! File with grid Geometry
  grid definition variables == x, y, z       ! Variable names in file
  grid definition label == gridlabel         ! Label for this grid
  boundary grid map == 1                     ! Create mapping to all nodestrings
end grid

bc == OBC_GRID, gridlabel, grid_file.nc      ! Gridded boundary using the previous grid and the grid file
  sub-type == 6                            ! Customise how forcing is applied
  bc header == time, ssh, u, v, sal, temp   ! Read these variables from netCDF file
  bc nodestrings == 1, 2                   ! Apply this BC to 2 different nodestrings
  bc update dt == 900.                      ! Update this BC every 900 seconds
  bc time units == hours                  ! Time is in hours
  bc reference time == 01/01/1990 00:00     ! Time is hours since this date
  vertical coordinate type == depth        ! Z coordinates are in meters below the water's surface
end bc
```

Furthermore, an option exists within TUFLOW FV to use the *OBC_GRID* boundary conditions to specify initial conditions ([initial condition OGCM](#), see initial conditions Section 10.7.2). Subsequent initial conditions can be used to overwrite this for certain variables, such as for setting the initial currents to zero via the [initial condition quiescent](#) to avoid initial shocks. The following example shows the usage of this:

```
initial condition ogcm           ! Use OBC_GRID to set initial conditions
initial condition quiescent     ! Set initial currents to zero
initial water level == -0.2      ! Set initial water level
```

Note: if separately applying water levels boundaries via a 2d *WL_CURT* for use with the *OBC_GRID*, for example to assign an astronomical tide boundary, then it's common to use [sub-type == 5](#) to apply tidal heights as incremental to the existing water level anomaly applied via the OBC_GRID (refer Section 10.6.1 for further information on sub-types).

```

! Tides
bc == WL_CURT, 1, GBR_Tide_AEST.nc
sub-type == 5
bc header == TIME,NS1_Chainage,NS1_WL
bc update dt == 60.
bc time units == hours
bc reference time == 01/01/1990 00:00
end bc

```

10.5.6 Advanced 3D Gridded Boundary Conditions

In addition to the *OBC_GRID* there are three advanced gridded 3D boundary conditions:

1. **SCALAR_GRID** boundaries, that are similar to OBC_GRID boundaries but without the water level and current velocity terms. These are used to specify just the scalars such as salinity, temperature, sediments, etc.
2. **QC_GRID** boundaries, that specify an inflow of water (with associated concentration of scalars) at each point of a 3D grid. These values are interpolated to the model's cells that fall within the grid domain. These can be used to specify complex flow inputs that may be the result of a nearfield model (such as for taking the outputs of a nearfield diffuser model and applying these to a far-field TUFLOW FV model).
3. **FC_GRID** boundaries, which are similar to QC_GRID boundaries, but only specify an inflow of scalar mass, without an associated inflow of water. These are useful to specify complex source terms of salinity, temperature, sediments, tracers or water quality constituents, also potentially from nearfield models.

If you are interested in using these boundary conditions, please contact support@tuflow.com and we can assist with examples and setup.

The available gridded input boundaries are provided in Table 10-3.

10.6 Boundary Types

Table 10-2 Boundary Condition Types

BC Type	BC Description	BC Method	BC file format	Default Columns Header ⁴
HQ	qH relationship (user specified)	External nodestring	CSV	H, Q, SCALARS ⁵
Q	Nodestring flow	External nodestring	CSV	TIME, Q, SCALARS
QC	Cell inflow (m ³ /s) - uses internal concentration during outflow.	Cell	CSV	TIME, Q, SCALARS
QG	Global Cell Inflow (m/s) – uses internal concentration during outflow.	Global	CSV	TIME, Q/A, SCALARS
QH	qH relationship (user specified)	External nodestring	CSV	H, Q, SCALARS
QN	Normal flow condition (automatic qH BC)	External nodestring	N/A	No CSV file required; second entry on BC line is friction slope. For example, the following lines specify a QN boundary along nodestring 2: <code>bc == QN, 2, 0.001 end bc</code>
WL	Water level (m or ft, see units)	External nodestring	CSV	TIME, WL, SCALARS
WLS	Sloping Water Level (m or ft, see units) WL_A = level at start of nodestring. WL_B = level at end of nodestring.	External nodestring ²	CSV	Time, WL_A, WL_B, SCALARS (A and B for each scalar).

⁴ Note that the header names listed here are defaults; if a “bc header ==” line is not included in the fvc file then these column header titles are required. If however a “bc header ==” line is included in the fvc then the header descriptions then match the column header in the CSV file. The complete list of default headers for the model configuration will also be printed in the log file.

² The direction of the nodestring determines which is point A and which is point B. The order is the order of the nodes in the nodestring (the direction that it was ‘clicked’ in sms, the order of the ‘external nodestring’ points or the order of nodes as listed at the end of the .2dm file).

⁵ Please refer to Section 1.3.3 for discussion on the ordering of scalar variables.

Table 10-3 Boundary Condition Types (Advanced)

BC Type	BC Description	BC Method	BC file format	Default Columns Header ⁶
AIR_TEMP	Temperature input (°C)	Global	CSV	TIME, AIR_TEMP
AIR_TEMP_GRID	Temperature input (°C)	Grid	NETCDF	TIME, AIR_TEMP
CLOUD	Cloud cover (fraction) - 0 = clear, 1 = full cloud cover.	Global	CSV	TIME, CLOUD
CLOUD_GRID	Cloud cover (fraction) - 0 = clear, 1 = full cloud cover.	Grid	NETCDF	TIME, CLOUD
CP	Cell concentration profile	Cell	CSV	DEPTH, SCALARS
CYC_HOLLAND	Parametric cyclone wind and pressure field	Global	CSV	[X], [Y],[P0],[PA],[RMAX],[B], [RHOA], [KM], [THETMAX], [DELTAFM],[WBGX], [WBGX] Refer Section 10.5.3.2.
FB	Sediment bed flux	Cell	CSV	TIME, FLUX_SED_1
FBM	Moving bed sediment flux	Cell	CSV	TIME, X, Y, FLUX_SED_1
FC	Cell scalar flux	Cell	CSV	TIME, SCALARS
FCM	Moving scalar flux	Cell	CSV	TIME, X, Y, SCALARS
LW_NET	Net longwave radiation (Wm ⁻²)	Global	CSV	TIME, LW_NET
LW_NET_GRID	Net longwave radiation (Wm ⁻²)	Grid	NETCDF	TIME, LW_NET
LW_RAD	Downward longwave radiation (Wm ⁻²)	Global	CSV	TIME, LW_RAD
LW_RAD_GRID	Downward longwave radiation (Wm ⁻²)	Grid	NETCDF	TIME, LW_RAD
MSLP_Grid	Mean sea level pressure field (hPa)	Grid	NETCDF	TIME, MSLP
OBC	Fully specified boundary condition ⁷⁸	External nodestring	NETCDF	TIME, WL, U, V, SCALARS

⁶ Note that the header names listed here are defaults; if a “bc header ==” line is not included in the fvc file then these column header titles are required. If however a “bc header ==” line is included in the fvc then the header descriptions then match the column header in the CSV file.

⁷ Note: this boundary application can be used to specify a supercritical flow condition.

⁸ Note: u,v are cartesian velocity vectors (components in x and y directions).

BC Type	BC Description	BC Method	BC file format	Default Columns Header ⁶
OBC_CURT	Fully specified open boundary condition curtain	External nodestring	NETCDF	TIME, WL, U, V, SCALARS
OBC_GRID	Fully specified boundary condition ^{9¹⁰}	Grid	NETCDF	TIME, WL, U, V, SCALARS
PRECIP	Precipitation (m/day)	Global	CSV	TIME, PRECIP
PRECIP_GRID	Precipitation grid (m/day)	Grid	NETCDF	TIME, PRECIP
REL_HUM	Relative humidity (%)	Global	CSV	TIME, REL_HUM
REL_HUM_GRID	Relative humidity (%)	Grid	NETCDF	TIME, REL_HUM
RS	Reflective, free slip	External nodestring	N/A	N/A
RNS	Reflective, no Slip	External nodestring	N/A	N/A
SCALAR	Scalar concentration specification	External nodestring	CSV	TIME, SCALARS
SCALAR_PROF	Scalar concentration profile	External nodestring	NETCDF	DEPTH, SCALARS
SCALAR_CURT	Scalar concentration curtain	External nodestring	CSV	TIME, SCALARS
SURF_TEMP	Water Surface Temperature input (°C)	Global	CSV	TIME, SURF_TEMP
SURF_TEMP_GRID	Water Surface Temperature input (°C)	Grid	NETCDF	TIME, SURF_TEMP
SW_RAD	Downward shortwave radiation (Wm ⁻²)	Global	CSV	TIME, LW_RAD
SW_RAD_GRID	Downward shortwave radiation (Wm ⁻²)	Grid	NETCDF	TIME, LW_RAD
TRANSPORT	Transport file written from previous TUFLOW FV simulation (refer Output == Transport)	Conserved variables stored on TUFLOW FV mesh	NETCDF	Generated by TUFLOW FV (refer to Section 10.8 for further information)

⁹ Note: this boundary application can be used to specify a supercritical flow condition.¹⁰ Note: u,v are Cartesian velocity vectors (components in x and y directions).

BC Type	BC Description	BC Method	BC file format	Default Columns Header ⁶
W10	Wind velocity at 10m (ms^{-1}) ¹¹	Global	CSV	TIME, W10_X, W10_Y
W10_GRID	Wind velocity at 10m (ms^{-1})	Grid	NETCDF	TIME, W10_X, W10_Y
WL_CURT	Water level curtain boundary condition	External nodestring	NETCDF	TIME, NS_1
ZG	Zero gradient boundary	External nodestring	N/A	N/A

10.6.1 Boundary Sub-Types

The default method that boundary conditions are applied to the model can be modified using the boundary [sub-type](#) command. The [sub-type](#) options are provided are provided in Table 10-4.

¹¹ Note: x,y are Cartesian velocity vectors (components in x and y directions).

Table 10-4 Boundary Sub-types

BC Type	Sub-type	Details
Q, HQ:	1	Specified flow rate boundary with momentum terms. Uniform distribution.
	2	Treated as a cell source term with a wall at the nodestring. Uniform distribution. This sub-type may be specified to constrain scalar mass inflow into 3D models, where sub-type 1/3 may allow simultaneous inflow and outflow at different depths.
	3	Specified flow rate boundary with momentum terms. $H^{1.5}$ flow distribution
	4	Treated as a cell source term with a wall at the nodestring. $H^{1.5}$ flow distribution. This sub-type may be specified to constrain scalar mass inflow into 3D models, where sub-type 1/3 may allow simultaneous inflow and outflow at different depths.
QC, QG:	1	Tracer outflow defined by internal concentration
	2	Tracer outflow defined by BC concentration
QN	1	(default) The flux calculation uses a new boundary calculation method added to the 2019.01. This method should limit warning messages that could occur when using QN boundaries: " <i>Unable to solve for specified inflow conditions at bc.</i> "
	2	Pre-2019 TUFLOW FV Releases QN boundary flux calculation method and can be used for legacy simulations.
Wave	1	Correct ubot values based on discrepancies between wave model and TUFLOW FV model depth (if available). Factor = $1/(COSH(MIN(kh/(h*h-depth)))), 10$
	2	Do not correct waves based on depth.
	3	Calculate wave radiation stress gradients internally and correct for depth as in subtype 1.
	4	Calculate wave radiation stress gradients internally and do not correct for depth as in subtype 2.
WL, OBC	1	Water level is specified, velocity is calculated internally based on a radiation condition.
	2	Incoming wave height is specified and ghost cell properties are calculated by superimposing outgoing wave form.
	3	Velocity is specified, water level is calculated internally based on a radiation condition.
	4	Both water level and velocity are specified. Note that any changes between the OBC condition and the TUFLOW FV model may result in waves reflecting off this boundary due to the overspecification.
	5	Apply water levels as incremental. The change in water level is added to any existing water level BC. This sub-type can be useful for including tidal forcing on top of a previously specified OBC specifying the non-tidal water levels and/or currents.
	6	Water level is specified, velocity is specified but barotropically relaxed based on the internal model conditions (by applying a barotropic Flather condition). This sub-type will be preferred where boundary forcing is from an Ocean General Circulation Model (e.g. HYCOM).

10.7 Initial Conditions

Initial conditions set the starting parameter state values of every cell within the model. Users must take care to set these to appropriate values. If the initial conditions differ vastly from the specified boundary conditions then this can create an initial ‘shock’ within the model and cause instabilities. Even if such shocks do not occur, some variables (such as salinity and temperature) can take a long time to ‘warm up’ (reach appropriate values spatially) and selecting initial conditions that are as close as possible to the desired values can decrease this warm up time.

Initial conditions can be specified equally across the whole model domain, spatially varying, or using a restart file from a previous model run.

10.7.1 Globally Specified

The simplest initial conditions are those that are globally specified and apply to all cells equally. The options available are as follows:

- Initial water level: Sets the water level to a constant value globally. To minimise initial shocks in the model this should be as close to possible as the initial timesteps of any water level boundary conditions.

```
Initial Water Level == 2.0
```

- Initial salinity: Sets the salinity to a constant value globally. As salinity changes slowly with time, this can take a long period to ‘warm up’ if poorly specified.

```
Initial Salinity == 35.0
```

- Initial temperature: Sets the temperature to a constant value globally, similar in use and consideration to the initial salinity.

```
Initial Temperature == 25.0
```

- Initial sediment concentration: sets the initial concentration for each sediment fraction.

```
Initial Sediment Concentration == 10, 20, 20, 10
```

- Initial tracer concentration: sets the initial concentration for each tracer. This can be used to initialise areas for flushing studies.

```
Initial Tracer Concentration == 100, 100, 100
```

- Initial WQ concentration: Sets the initial concentration for water quality constituents.

```
Initial WQ Concentration == 1.0, 0.02, 0.05, 0.087, 1.15, 1.15, 200
```

- Initial scalar profile: Allows the user to set initial scalar values globally but varying by depth. This uses a CSV file to define this profile structure.

```
Initial Scalar Profile == ..\bc_dbase\IC\InitialProfile_000.csv
```

InitialProfile_000.CSV

Depth	Sal	Temp	TRACE_1	TRACE_2	SED_1	SED_2
0	0	25	0	100	0	0
5	32	24	0	100	0	0
10	35	24	0	100	0	0
20	35	18	0	100	10	10

- [Initial condition quiescent](#); Sets the initial current velocity to zero globally. This is most commonly used to overwrite other initial conditions (OGCM or restart files) to ease any shocks due to initialising models with high currents.

```
Initial Condition Quiescent
```

10.7.2 Spatially Varying

Alternative initial conditions can be specified that allow the variables to differ spatially over the domain. This allows for greater control of appropriate initial conditions where there are distinct different areas of a model (such as offshore marine environment, and inshore estuarine or fresh water regions).

- Initial condition OGCM; If an OBC_GRID boundary is present (refer Section 10.5.3), then the values from this boundary condition are interpolated to all cells for the first timestep to use as an initial condition.

```
Initial Condition OGCM
```

Often the OGCM is used in combination with '[initial condition quiescent](#)' to limit initial instabilities.

- Initial condition 2D; Can specify a CSV file with an initial condition for every 2D cell within the model.

```
Initial Condition 2D == ..\bc_dbase\IC\Intial2D_000.csv
```

ID	WL	U	V	SAL	TEMP	TRACE_1	SED_1
1	1.001	0.01	0.13	35	25	100	0
2	1.001331	0.01	0.125	35	25	100	0
3	1.002012	0.02	0.13	35	25	100	0
4	1.002127	0.02	0.13	35	25	100	0
5	1.002802	0.01	0.13	35	25	100	0

- Initial condition 3D; Similar to the initial condition 2D, but allows for specification of the initial conditions for all 3D cells (noting 3D cells have different indices to the 2D cells, you can review the IDX2 and IDX3 attributes of the [mesh check file](#) to review your model mesh indices).

```
Initial Condition 3D == ..\bc_dbase\IC\Intial3D_000.csv
```

ID	WL	U	V	SAL	TEMP	TRACE_1	SED_1
1	1.1	0.25	0.13	5	25	100	0
2	1.1	0.15	0.12	10	25	100	0
3	1.1	0.1	0.12	25	25	100	0
4	1.1	0.1	0.08	35	20	100	0
5	1.1	0.01	0.01	35	20	100	10

10.7.3 Restart Files

[Restart](#) files are written by TUFLOW FV and write the current values of the parameters in each cell of the model at regular timesteps as specified by the [write restart dt](#) flag. The restart file can also be optionally overwritten at the restart dt using the [restart overwrite](#) command. Restart files can only be used by models that use the same mesh, bathymetry, vertical layering and number of conserved variables (using salinity and/or temperature and the same numbers of tracers, sediments, etc). If required, editing of restart files can be completed using existing tools, please contact support@tuflow.com for these.

Using a restart file as an initial condition is as simple as providing a path to an existing restart file:-

```
restart == log\PreviousRun_000.rst
```

This will automatically read the existing restart file, and any matching turbulence restart file ('_turb.rst', automatically written by previous runs also, but not necessary to run further models). By default, this will also adjust the starting time of the current run to the timestamp in the restart file/s. This can be changed by using the [use restart file time](#) flag.

For users of sediment transport models, a bed mass restart file ('_bed.rst') will also be written by each model run. This specifies the mass of each sediment class in each sediment layer of each cell. This file is not automatically used by specifying a restart in the main control file (.fvc) but requires an additional restart specification in the sediment control file.

```
restart == log\PreviousRun_000_bed.rst
```

10.8 Transport File

Transport files are a special kind of boundary condition that can be used to improve the runtimes of tracer, sediment transport, particle tracking or water quality models. Transport files store information of the hydrodynamics to be read by subsequent model runs where only tracer properties are changing. This is a common feature of many tracer studies (including sediment transport, water quality and particle tracking), where numerous scenarios with equal hydrodynamics but different tracer inputs are required. To implement a transport file, the hydrodynamics are simulated and saved first, outputting a transport file as follows:

```
output == transport
  output interval == 900
end output
```

The output interval must be small enough to capture any important variations in the hydrodynamics within the area of interest (for example, tidal variation).

The resulting transport file (a NetCDF file with the suffix ‘_trans’) can then be inspected onto the subsequent runs using a special BC type as follows:

```
bc == transport, ..\bc\PreviousRun_000_trans.nc
end bc
```

It is important that all other boundary conditions that can affect the hydrodynamics (including salinity and temperature BCs if using baroclinic coupling) must stay the same as the initial run that created the transport file. Any conflicts between the transport file and the subsequent model runs using it can cause inaccuracies and/or instabilities.

Transport files may improve runtimes for subsequent runs though improvements can often still be limited by CFL timestep restrictions, particularly in wetting and drying areas. In order to overcome these timestep restrictions the [transport mode depth](#) can be set to a value below which the transport flux is limited (rather than the timestep) to avoid exceeding the CFL stability constraint. This numerical treatment can lead to some loss of solution accuracy in shallow regions, however in practice these will usually not be significant. An appropriate value for the transport mode depth limit would be between the [cell wet depth](#) as a lower limit and 0.5-1.0m as an upper limit.

11 Hydraulic Structures

Chapter Contents

11.1	Introduction	11-2
11.2	Structure Types	11-3
11.2.1	Nodestring and Linked Nodestrings	11-4
11.2.2	Autoweir	11-5
11.2.3	Linked Zones	11-5
11.2.4	Cell or Zone	11-6
11.2.5	Defining a Nodestring, Linked Nodestring or Linked Zones Structure Block	11-7
11.2.6	Defining a Cell or Zone Structure Type	11-8
11.2.7	Structure Block Example	11-8
11.3	Flux Functions	11-10
11.3.1	Non-Linear Shallow Water Equations	11-10
11.3.2	Timeseries	11-11
11.3.3	Matrix	11-12
11.3.4	Weir Types	11-14
11.3.4.1	<i>Fixed Weirs</i>	11-14
11.3.4.2	<i>Variable Height Weirs</i>	11-15
11.3.5	Wall	11-16
11.3.6	Porous	11-16
11.3.7	Culvert	11-17
11.4	Energy Losses	11-21
11.4.1	Energy Loss Functions	11-21
11.4.2	Energy Loss Considerations	11-21
11.4.3	Form Loss Coefficient	11-22
11.4.4	Energy Table	11-22
11.5	Scalar Functions	11-24
11.6	Structure Controls	11-26
11.6.1	Control Block	11-26
11.6.2	Control Types	11-26
11.6.3	Control Parameters	11-26
11.6.4	Structure Logging File	11-27
11.6.5	Structure Control Examples	11-27

11.1 Introduction

TUFLOW FV includes a wide range of options for simulating hydraulic structures such as bridges, culverts, weirs, pumps, water treatment devices etc. Structures can be static, be modified over time, or respond according to flow behaviour at other locations in the model. The ability to modify selected cell properties such as topography/bathymetry is also supported

11.2 Structure Types

Hydraulic structures allow the user to describe flow behaviour not easily represented via the NLSWE alone by substituting specific equations for the exchange of water and other conserved variables.

Structures are defined using a [structure](#) block and require a unique structure block definition for each structure within the model (autoweirs being the exception to this rule). When developing complex models with many structures, [include](#) files are recommended to manage the data inputs and to keep control files as succinct as possible.

To assist with review and visualisation of flow behaviour through the structures, users are encouraged to use the structflux and structcheck outputs detailed in Section 12.8.

Structure types can be categorised into four broad groups as follows:

- Hydraulic structures applied as a flux term. These structures transfer fluxes from one cell to another via a nodestring or a pair of linked nodestrings (refer Section 11.2.1);
- Globally specified as ‘auto weir’ nodestring structures (refer Section 11.2.2);
- Source/sink hydraulic structures that transfer and modify mass via a linked zone (refer Section 11.2.3); and
- Area property ‘structures’ that modify the hydraulic properties of a cell (refer Section 11.2.4).

The [structure](#) command is used to define the location of a structure and its type. Nested within a given structure block, there are a wide range of options to modify, or replace the generic NLSWE. Each is explored within subsequent sections of this chapter. The key command types nested within a structure block include:

- [Flux functions](#) used to define weirs, culverts, bridges, porous breakwaters, walls and user-specified flow relationships;
- [Scalar functions](#) to enable the user to modify scalar variables such as salinities, temperatures, water quality variables or sediments; and
- [Control blocks](#) to allow a structure to modify its state over time, or in response to model variables or events.

11.2.1 Nodestring and Linked Nodestrings

In general, nodestring hydraulic structures work by averaging the upstream and downstream primary variables (water depth, current velocity, etc.) and using these to predict the flow exchange based on the selected [flux function](#). In addition, for users of the advection/dispersion module, conserved scalars such as salinity, temperature and tracers can be manipulated using a [scalar function](#).

The distribution of flow (and other conserved variables) extracted from the ‘upstream’ side and input to the ‘downstream’ side are weighted differently depending on the structure type, but generally weighted according to the cross-sectional area of a cell face, the cell area the depth of water within a given cell. This allows for some distribution of the flow in 2-dimensions but may remove any complex stratification or vertical changes between the upstream and downstream. This is important to remember when undertaking 3D modelling and using structure links.

Nodestring structure types can be specified using the following options:

- 1) Single Location (**Nodestring**): Defined using a single nodestring. Mass and momentum are transported between neighbouring cells, regulated through the cell face by the flow controls defined by the structure. This option is commonly used to model bridges, weirs and in some cases culverts (depending on the model resolution). Upstream and downstream conditions are inspected for each cell face, with a corresponding flow calculated. Exceptions to this rule are when using the *matrix*, *timeseries* or *culvert flux function* (refer Section 11.3) and/or with use of the [energy loss method](#), cell [width file](#) or [blockage file](#), all of which use cross sectional average conditions. Momentum is transferred according to the flow and the upstream velocity. An example nodestring structure is provided below:

```
! New Bridge
structure == nodestring, 12
  flux function == nlswe
  Energy loss function == coefficient
  Form loss coefficient == 0.4
  Blockage file == ..\model\geo\MultSpan.csv
end structure
```

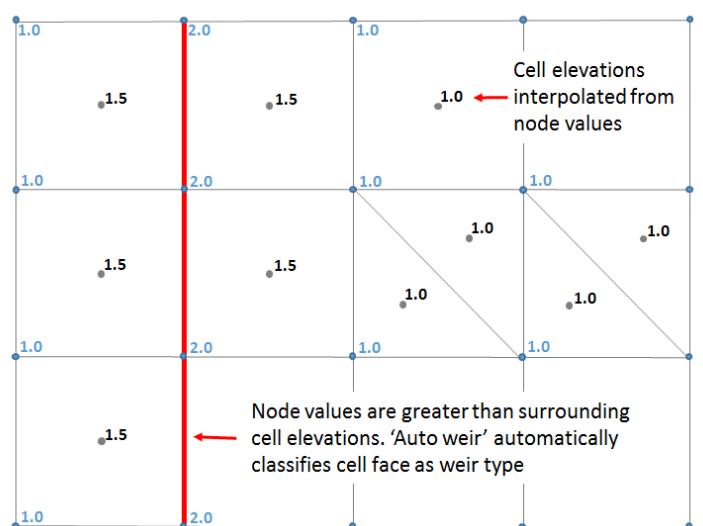
- 2) Multiple Locations (**Linked Nodestrings**): Defined by two separate nodestrings. Mass and momentum are transported between two locations in the model, regulated by the flow controls defined by the structure. This option is commonly used to represent culverts within a model, or structures that may be overtopped. Upstream and downstream conditions are averaged along each nodestring, weighted by the face length and/or flow depth that each cell contributes. These conditions are used to derive an overall flow through the structure which is distributed according to cell face lengths. Momentum is transferred according to the flow and the upstream velocity. An example linked nodestring structure is provided below:

```
! New Bridge
structure == linkednodestrings, 12,13
  flux function == nlswe
  Energy loss function == coefficient
  Form loss coefficient == 0.4
  Blockage file == ..\model\geo\MultSpan.csv
end structure
```

11.2.2 Autoweir

The globally assigned **Autoweir** structure is a single nodestring structure type. This structure type instructs TUFLOW FV to identify all cell faces in the model domain that are elevated above the adjacent cell's centroid elevations. These cell faces are then assigned a weir flow condition. This feature ensures that critical floodplain hydraulic controls such as roads and perched riverbanks regulate floodplain flows accurately. This command is recommended for catchment flood modelling. In combination with this command, the geometry log files can be used to identify all locations within the model where this function has been assigned.

NOTE: bathymetry at the faces can only be higher than the adjacent cells when bathymetry at the cell nodes has been specified. This is usually achieved by specifying the bathymetry on the mesh file (.2dm), or using the [Read GRID Zpts](#) command and not using a separate [cell elevation file](#).



```
Structure == Autoweir      ! Global weir flow check/calculation
End Structure
```

Figure 11-1 Auto Weir Example

11.2.3 Linked Zones

Multiple locations can be linked as a source/sink term defined by two separate polygons (**Linked Zones**). Mass is transported between the two locations in the model, regulated by the flow controls defined within the structure block (flux functions, scalar functions and/or structure controls). The upstream and downstream variables that might control this flow will be averaged over the cells in each polygon. The mass is transported as a source/sink across the whole polygon without the momentum being transferred. While this option can theoretically be used for bridges and culverts where hydraulic momentum is low, it is more commonly used for pump-style structures to move water from one area of the model to another. An example linked zone structure is provided below:

```
Structure == LinkedZones, 3,8
  polygonfile == example_polyfile.csv          ! Structure Location (from NS 5 to NS 6)
  flux function == timeseries                 ! File with polygons in it
  flux file == example_FluxFile.csv           ! Specified flow from csv
end structure                                    ! Timeseries of flow between zones
```

11.2.4 Cell or Zone

Cell or **Zone** structures can change properties of a cell, or group of cells, as a function of time or based on a reaction to model parameters at other locations in the model. The primary property that can be controlled is the topography (bathymetry), however, destratification ‘bubbler’ structures can also be simulated by locally modifying turbulent mixing. A description and example of the Cell and Zone area property structures are provided as follows:

- 1) Single Cell (**Cell**): Topography and other property controls are applied to a single cell location defined by a cell ID. This example raises cell ID 197 from -1m at the start of the simulation (time=0) to 1m at time=1hr:

```
!
!STRUCTURE COMMANDS
Structure == cell, 197
  Bed adjust == zb_adjust
    ! CELL ID Coordinates not supported
    ! Adjust bed level

  control == timeseries
    control parameter == ZB
    control file == ..\model\geo\increase_bed.csv
      ! Change bed as a function of time
      ! Modify elevation
      ! How the bed changes vs. time
  end control

end structure
!
```

The format of the file *Increase_bed.CSV* is provided as follows:

Time	ZB
0	-1
1	1
3	1

- 2) Multiple Cells (**Zone**): Topography and other property controls are applied to any cells with cell centroids within a polygon, as specified in a [polygon file](#). The example below is analogous to the cell example above, albeit applied across a group of cells IDs:

```
!
!STRUCTURE COMMANDS
Structure == zone, 1
  polygon file == ..\model\geo\zone_poly.csv
  Bed adjust == zb_adjust
    ! Zone definition
    ! Polygon location
    ! Adjust bed level

  control == timeseries
    control parameter == ZB
    control file == ..\model\geo\increase_bed.csv
      ! Change bed as a function of time
      ! Modify elevation
      ! How the bed changes vs. time
  end control

end structure
!
```

The format of the file *Zone_poly.CSV* is provided as follows:

X	Y	ID
197.5	62.5	1
210.5	62.5	1
210.5	42.5	1
197.5	42.5	1
197.5	62.5	1

11.2.5 Defining a Nodestring, Linked Nodestring or Linked Zones Structure Block

A typical decision-making process for setting up nodestring, linked nodestring or linked zones structure types is as follows:

1. Decide on the most suitable structure type as mentioned in Section 11.2. If momentum transfer is important, a nodestring type is likely required, if overtopping is possible then a linked nodestring with an elevated topography in between may be more suitable. If a source/sink type is appropriate that affects multiple cells, in a low momentum area, then a linked zones structure may be the best approach.
2. Decide on the flux regime by setting the [flux function](#) within the [structure](#) block. Either based on known parameterisations ([culvert](#), [weir](#)) or based on the NLSWE with some additional losses, using a [width file](#) and [energy loss function](#) for example. Or full specification using a flow matrix developed already.
3. If modelling in 3D, decide on depth-averaging via the [vertical coordinate type](#) and [vertical distribution file](#) commands. The flow can be sampled from the upstream and placed downstream with different depth-averaging options. This is sometimes important for the representation of currents downstream but is particularly important when using the advection dispersion module for controlling where conserved tracers move.
4. Decide on any [control](#) needs (refer Section 11.6). The flow rate can be adjusted based on a timeseries or based on sampling parts of the model and comparing it to some specified logic. This can be useful when the real system has some operational control based on gauged data, such as pumping when a lake falls below a certain level or triggering a gated culvert if upstream temperatures exceed a limit.

5. Decide on scalar regime (if appropriate) using a [scalar function](#). Scalars can either be adjusted or conserved as they pass through the structure (refer Section 11.5).

11.2.6 Defining a Cell or Zone Structure Type

Comparatively, cell and zone ‘structures’ have a simpler set of options and considerations than nodestrings, linked nodestrings and linked zones. They can only act on specific properties of a cell basis. **Importantly, flux function or scalar functions are not used for Cell or Zone structure types.**

General considerations are as follows:

1. Decide on the area property to apply. If updating the topography, then the [bed adjust](#) command needs to be specified. Alternatively, if representing a ‘bubbler’ use the [destratification unit](#) command.
2. Choose whether this area property should be completed on an individual cell or group of cells using a zone via a [polygon file](#).
3. If using the [bed adjust](#) command, select the appropriate [control](#) block commands to modify your bed during the simulation. Alternately, if using the [destratification unit](#) select the required [properties](#) for the ‘bubbler’ type.

11.2.7 Structure Block Example

Figure 11-2 provides an example of how nesting of flux functions and structure controls can be completed within a structure block to represent coastal inundation pumping system that is engaged during times of extreme weather. This example, whilst somewhat verbose, provides an overview of the powerful features that can be implemented within a structure block.

The [structure](#) block begins on line 42 of the control file and is closed via the ‘end structure’ command on line 64. This structure is defined to transfer flow at cell faces along a single nodestring, nodestring number 13. The structure has been optionally given the name ‘st_pump’ to assist in locating model results in any tabulated output data the model produces.

A [flux function](#) is defined on line 45 as a time series. For this setup, we are assigning the peak capacity of the pump 200m³/s over the entire simulation. We will regulate this flow later via the control block.

For this structure we are not modifying any scalar variables via the [scalar function](#) command on line 49. As ‘none’ is the default, this line could have been omitted.

To operate the structure, on line 52 a nested [control](#) block is opened using the sample rule approach. The [control](#) block is closed on line 62 with an end control command. Within the nested control block are a series of commands that increase or decrease the pump operation as a function of model water level at a storm tide gauge downstream of the structure, its location defined by a [sample point](#).

On line 54 and 55 we tell TUFLOW FV to return water level [sample type](#) at the [sample point](#), that will be used as an input to determine the pump operation. On line 57 the ‘fraction_open’ [control parameter](#) compares the water level at the [sample point](#) with a relationship specified within the [control file](#) ‘*pump_rate.CSV*’ as defined on line 60. The file ‘*pump_rate.CSV*’ determines whether the pump is steady, its speed being increased or decreased (and thus flow rate being increased or decreased) as a function of the water level at the sample point. For example, when the water level reaches 1m at the storm tide gauge a fraction_open of 0.5 is returned via the sample rule, which will turn the pump onto half capacity at 100 m³/s. This flowrate is then passed through the structure.

At the beginning of this model simulation, the pump is switched off via the [start control state](#) command on line 59. Updating the operation of the structure is limited to a user defined time increment via the [control update dt](#), in this case every six minutes. While this is example is quite complex, the structure block can be kept as simple as a few lines or optionally built up to complete more complex operations. The possible combinations are explored throughout the chapter.

```

40 !
41 !Structure Commands
42 structure == nodestring, 3
43
44     name == st_pump
45     flux function == timeseries
46     flux file == pump_capacity.csv      ! 200 m^3/s peak capacity
47
48     ! Scalar Function Specification - No modification of scalars
49     scalar function == none
50
51     ! Control block. Open or close gate based on downstream WL
52     control == sample_rule           ! nested control block
53
54         sample point == 300, 50      ! Storm tide gauge location
55         sample type == WL          ! Water level
56         sample dt == 0.0166667     ! 1min
57         control parameter == fraction_open ! Open or close
58         control update dt == 0.1    ! Update structure movement (hrs)
59         start control state == 0   ! Start with pump off
60         control file == pump_rate.csv ! Proporation of pump rate to water level
61
62     end control
63
64 end structure
--
```

pump_rate.csv	
sample_value	fraction_open
0	0
1	0.5
2	1
3	1
4	1

pump_capacity.csv	
time	q
0	200
100	200

Figure 11-2 Structure Block Example

11.3 Flux Functions

[Flux functions](#) are used to transfer flow through nodestring, linked nodestring and linked zones [structure](#) types. The available flux functions for each [structure](#) type and their relevant sections are summarised in Table 11-1.

Table 11-1 Flux Function Types

Flux Function Type	Structure Types Supported	Relevant Section
None/NLSWE	Nodestring, Linked Nodestring	11.3.1
Timeseries	Nodestring, Linked Nodestring, Linked Zones	11.3.2
Matrix	Nodestring, Linked Nodestring, Linked Zones	11.3.3
Weir (Dz, Adjust, DZ_Adjust, autoweir)	Nodestring, Linked Nodestring	11.3.4
Wall	Nodestring	11.3.5
Porous	Nodestring, Linked Nodestring	11.3.6
Culvert	Nodestring, Linked Nodestring	11.3.7

11.3.1 Non-Linear Shallow Water Equations

The flow through the structure is based on the non-linear shallow water equations, however several commands allow the user to assign additional terms, controls or geometric modifiers at the structure. The primary variables are averaged on the upstream and downstream sides of the structure and the flow is otherwise calculated in the same way as the cell fluxes throughout the model normally.

These are commonly combined with a combination of [width](#) (or [blockage](#)) file and [energy loss functions](#) to simulate sub-grid scale effects (refer to Section 11.4 for detailed discussion on the use of energy loss and width modifiers).

The below example shows a basic link between two nodestrings in different parts of the model domain.

```
Structure == linkedNodestrings, 5,6      ! Structure flow from NS 5 to NS 6
  flux function == nlswe                  ! Not actually required as is default
end structure
```

11.3.2 Timeseries

The flow through the structure is specified in a separate timeseries CSV file. This method is commonly used to simulated pumping structures with specified times and flow rates. A secondary command for the [flux file](#) is required to specify the time and flow rate.

The following example shows a set of linked nodestrings with a specified flow rate between them.

```

Structure == LinkedZones, 3,8
  polygonfile == example_polyfile.csv
  flux function == timeseries
  flux file == example_FluxFile.csv      ! Structure Location (from NS 5 to NS 6)
                                         ! File with polygons in it
                                         ! Specified flow from csv
                                         ! Timeseries of flow between zones
end structure

```

example_polyfile.csv			
X	Y	ID	Comments
0	0	1	! Polygon 1
10	0	1	! Polygon 1
10	10	1	! Polygon 1
0	10	1	! Polygon 1
900	0	2	! Triangle
905	5	2	! Triangle
910	2	2	! Triangle
20	0	3	! First in structure
25	1	3	! First in structure
25	11	3	! First in structure
20	10	3	! First in structure
53	20	8	! Second in structure
60	25	8	! Second in structure
50	20	8	! Second in structure

example_FluxFile.csv		
Time	Q	comments
0	0	! No flow
10	5	! Positive flow
12	5	! Constant
15	10	! Ramping Up
20	10	! Constant
30	0	! No Flow
50	-5	! Reverse Flow

11.3.3 Matrix

The hQh structure option ([Flux Function == Matrix](#) combined with a [Flux File](#)) leaves the calculation of flow through a structure to the user. This makes the hQh structure extremely flexible in its application (any structure, be it weir, culvert, pipe, or an irregular structure *etc...* can be applied). However, the user is required to define the hQh relationship to be used within the model.

Structure flow is determined from an “hQh” relationship; flow (Q) across the nodestring is determined by the upstream water level (h_{us}) and downstream water level (h_{ds}), as specified in a user defined matrix of values (the hQh table). The following figure illustrates this (please refer to the [Flux File](#) command description for formatting details of the hQh matrix CSV file).

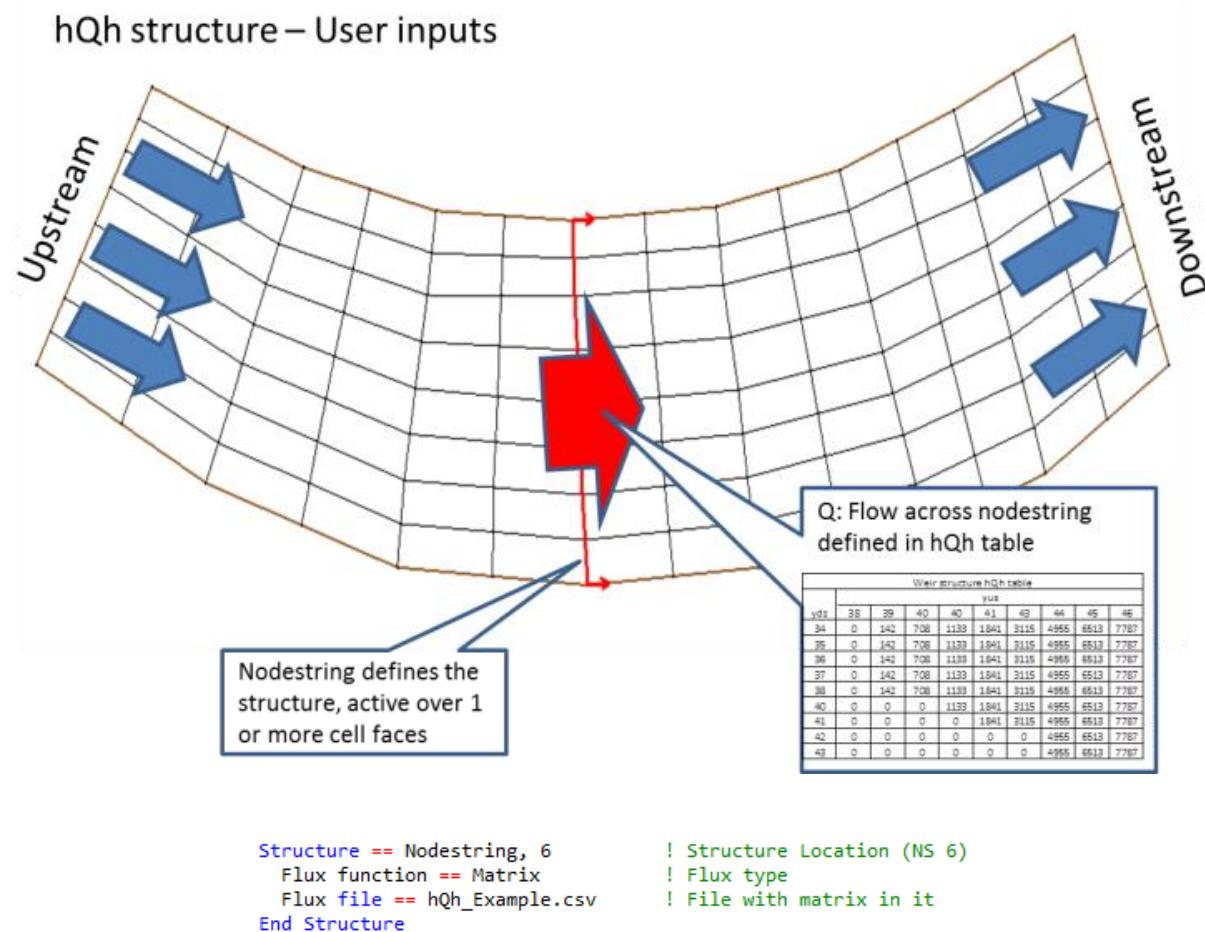


Figure 11-3 hQh Structure Example

The logic process for computing structure flow is as follows:

- 1 Flows in the hQh table are distributed across the nodestring according to the relative widths of each individual cell face (a cell face being the connecting line between two cells). Thus, each individual cell face has a unique hQh table with Q values factored from the original hQh table according to the cell face width.
- 2 During a model simulation step, at each cell face the upstream and downstream water levels are used to obtain Q from the hQh matrix.

- 3 A check is performed between the tabulated flow (Q_{hqh}) and that calculated using the Shallow Water Equations (Q_{SWE}).

If the tabulated flow (Q_{hqh}) is less than the Shallow Water Equations (Q_{SWE}) equivalent, it is applied. Conversely, if the tabulated flow (Q_{hqh}) is greater than the Shallow Water Equations (Q_{SWE}) value, the Shallow Water Equations flow is used.

- 4 Momentum is actively transferred through the structure based on Q_{hqh} and the upstream velocity.

A hQh relationship can be calculated either from first principles or using other modelling packages.

When deriving the hQh relationship, care must be taken to ensure that entry and exit losses are being applied appropriately. Depending upon the layout of the structure in the mesh design, the hQh relationship can represent all the losses that occur in a structure (macro and micro) or only internal (micro) losses. This concept is described illustrated in Figure 11-4.

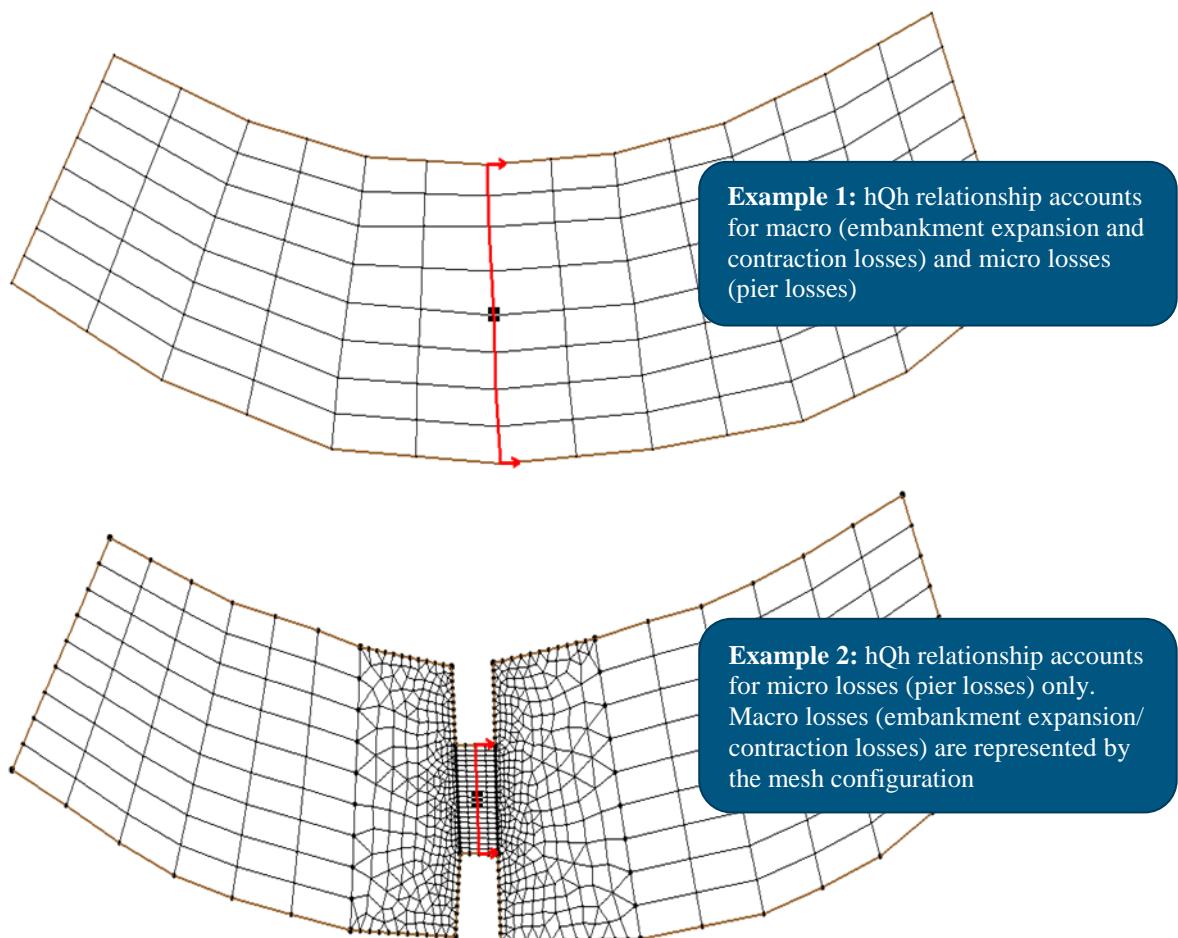


Figure 11-4 hQh Calculation Design Options

11.3.4 Weir Types

Weir flow can be specified within the model in a variety of ways depending on the intended application. In some situations, using the NLSWE to calculate weir flow (just letting TUFLOW FV simulate weir flow without any direct specification of a weir structures) may be entirely acceptable. Where model resolution is insufficient (the structure width is represented by less than 5 cells), manual specification of the weir is recommended.

When manually specified, TUFLOW FV uses the standard weir equation (shown below), where the coefficient C and crest level P are user inputs. TUFLOW automatically calculates the weir width b based on the cell face across which the weir is being specified.

$$Q = C b h_1^{3/2}$$

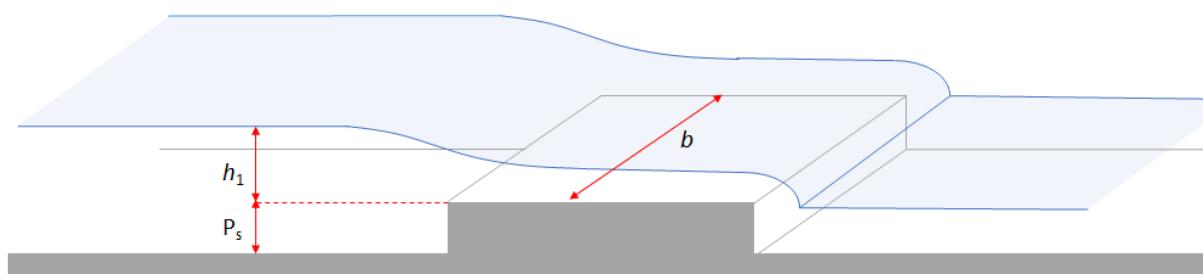


Figure 11-5 Weir Flow

A variety of weir options are available (see [flux function](#)). These weir commands are applied at cell faces within the model at specified nodestring locations. For all these options, using default values for C provides an exact solution to the broad crested weir equation. Two fixed and variable weir [flux function](#) options are available, summarised in Sections 11.3.4.1 and 11.3.4.2.

11.3.4.1 Fixed Weirs

A Fixed Weir Options is used to model flow control structures or levees. The fixed weir has the following options:-

- 1) Weir: A weir structure with a fixed crest level. Used when absolute bathymetry and weir crest is certain.
- 2) Weir_dz: A weir structure with a crest level (dz) above the existing cell elevation. Used when the weir crest above the bed level is certain, but the bed level may differ between different scenarios.

[Properties](#) input are required both fixed weir options. Example inputs are shown below in Figure 11-6.

```

Structure == Nodestring, 5           ! Structure location (NS 5)
  Flux function == Weir             ! Weir flag
  Properties == 9.5, 1.6            ! Weir elevation (mAHD), weir coefficient
End Structure

Structure == Nodestring, 6           ! Structure location (NS 6)
  Flux function == Weir_dz          ! Weir flag
  Properties == 2.0, 1.6            ! Weir elevation above cell elevation (m), weir coefficient
End Structure

```

Figure 11-6 Fixed Elevation Weir Example

11.3.4.2 Variable Height Weirs

Variable Weir used to modify weir height over time. These are useful to assess levee breach scenarios. The variable weir options are:-

- 1) Weir_adjust: A weir structure with a time-varying adjustable crest level relative to the model datum.
- 2) Weir_dz_adjust: A weir structure with a time-varying adjustable crest level (weir_dz) above the existing cell elevation.

Control types are used to specify how the weir elevation should be varied, either by time series from a trigger location/water level from a location within the model domain, or from the start of the model simulation. Examples of both input types are provided in Figure 11-7.

```
Structure == Nodestring, 5           ! Structure location (NS 5)
Flux function == Weir_adjust        ! Weir flag
Properties == 9.5, 1.6              ! Weir elevation (mAHD), weir coefficient

Control == Trigger                 ! Control Option, Change after WL reached
  Control Parameter == Weir_Crest   ! Control the weir crest
  Sample point == 125332.0, 542533.0 ! Sample Location (x,y)
  Sample dt == 0.0166               ! Sample time interval (hr)
  Trigger value == 9.7             ! Trigger value (mAHD)
  Control file == adjustableWeir1.csv ! Weir crest timeseries after trigger
End Control

End Structure
```

adjustableWeir1.csv		
Time	weir_crest	comment
20	2.0	! Prior to this its constant
22	1.5	! Drops then constant

```
Structure == Nodestring, 6           ! Structure location (NS 6)
Flux function == Weir_dz_adjust     ! Weir flag
Properties == 2.0, 1.6              ! Weir elevation above cell elevation (m), weir coefficient

Control == timeseries              ! Control Option, adjust according to timeseries
  Control Parameter == Weir_Crest   ! Control the weir crest
  Control file == adjustableWeir2.csv ! Weir level with time
End Control

End Structure
```

adjustableWeir2.csv		
Time	dz	comment
0.0	9.7	! Initial Level
2.0	9.2	! Starts dropping
4.0	8.7	
6.0	8.2	

Figure 11-7 Variable Elevation Weir Example

11.3.5 Wall

Wall type flux-functions block all flow through the structure. They are only available for nodestring type [structure](#) types and do not work for linked nodestrings or linked zones. Wall type structures can be used to block flow along cell edges where the model resolution would otherwise not allow the bathymetry to do so.

```
Structure == Nodestring, 2 ! Structure Location (NS 2)
  Flux function == Wall    ! Blocks all flow across this line
End Structure
```

11.3.6 Porous

Porous structures allow the user to control the flow between two areas based on Darcy's law of flow through porous media. It requires the user to specify the hydraulic conductivity and structure length using the [properties](#) flag. This type of structure cannot be used for linked zone structures but is available for nodestring and linked nodestring types.

Darcy's law is as follows:

$$Q = -KA \frac{dH}{dL}$$

Where K is the hydraulic conductivity (user specified), dL is the length of the structure (user specified), A is the average cross-sectional area of the water column and dH is the difference between the upstream and downstream water levels.

```
Structure == Nodestring, 2      ! Structure Location (NS 2)
  Flux function == porous      ! Darcy flow through structure
  Properties == 0.5, 10         ! Hydraulic conductivity (m/s), Length of flowpath (m)
End Structure
```

11.3.7 Culvert

Sub-grid sized culverts can be modelled as 1D structures. The standard structure equations which have been used by TUFLOW Classic the late 1990's have been included in TUFLOW FV¹². The calculations of culvert flow and losses are carried out using techniques from "Hydraulic Charts for the Selection of Highway Culverts" and "Capacity Charts for the Hydraulic Design of Highway Culverts" together with additional information provided in Henderson 1966. The calculations have been compared and shown to be consistent with manufacturer's data provided by both "Rocla" and "Armco". The equations allow for a range of different flow regimes, as summarised in Figure 11-8, Figure 11-9 and Table 11-2

Table 11-2 1D Culvert Flow Regimes

Regime	Description
A	Unsubmerged entrance and exit. Critical flow at entrance. Upstream controlled with the flow control at the inlet.
B	Submerged entrance and unsubmerged exit. Orifice flow at entrance. Upstream controlled with the flow control at the inlet.
C	Unsubmerged entrance and exit. Critical flow at exit. Upstream controlled with the flow control at the culvert outlet.
D	Unsubmerged entrance and exit. Sub-critical flow at exit. Downstream controlled.
E	Submerged entrance and unsubmerged exit. Full pipe flow. Upstream controlled with the flow control at the culvert outlet.
F	Submerged entrance and exit. Full pipe flow. Downstream controlled.
G	No flow. Dry or flap-gate active.
H	Submerged entrance and unsubmerged exit. Adverse slope. Downstream controlled.
J	Unsubmerged entrance and exit. Adverse slope. Downstream controlled.
K	Unsubmerged entrance and submerged exit. Critical flow at entrance. Upstream controlled with the flow control at the inlet. Hydraulic jump along culvert.
L	Submerged entrance and exit. Orifice flow at entrance. Upstream controlled with the flow control at the inlet. Hydraulic jump along culvert.

¹² For benchmarking of culvert flow to the literature, see Huxley (2004):
<http://www.tuflow.com/Downloads/TUFLOW%20Validation%20and%20Testing,%20Huxley,%202004.pdf>

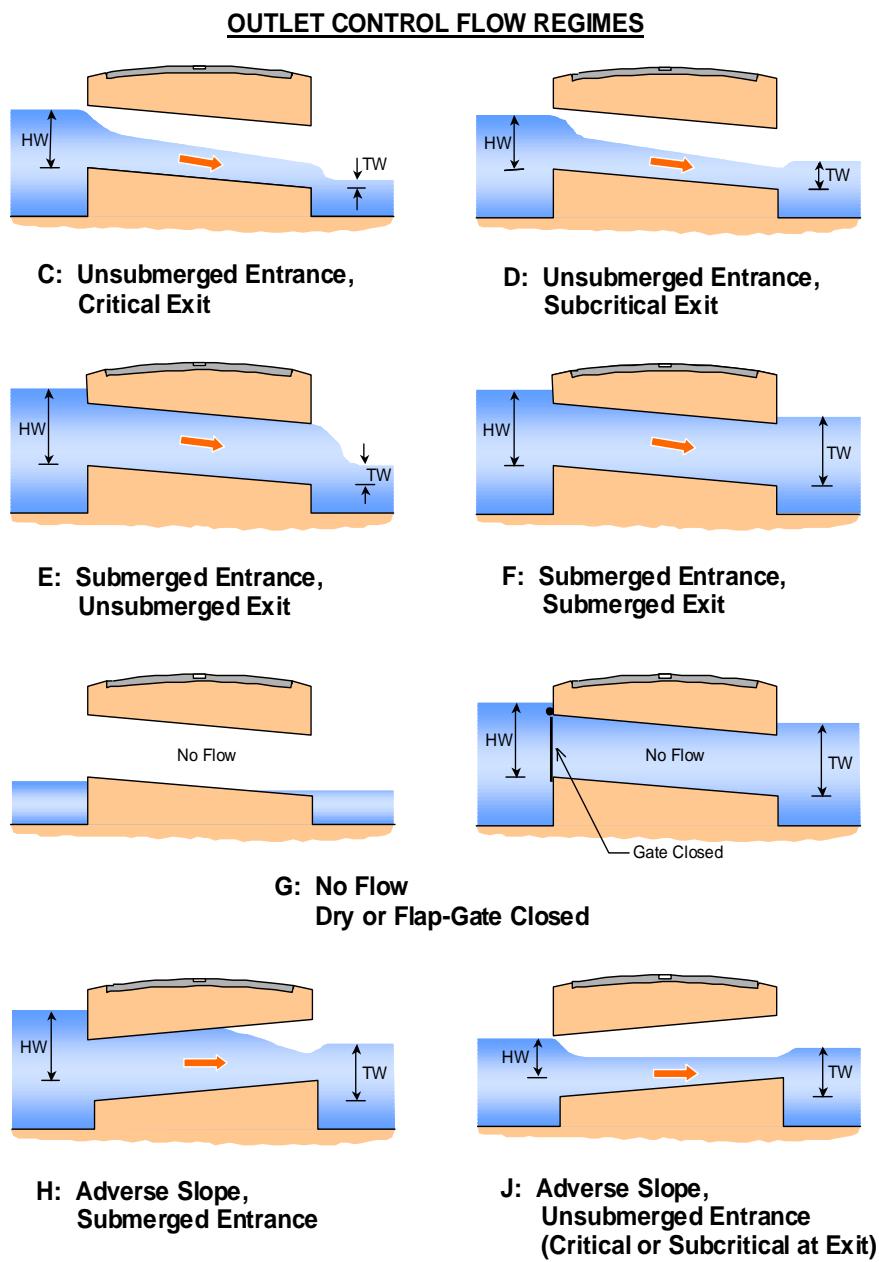


Figure 11-8 1D Outlet Control Culvert Flow Regimes

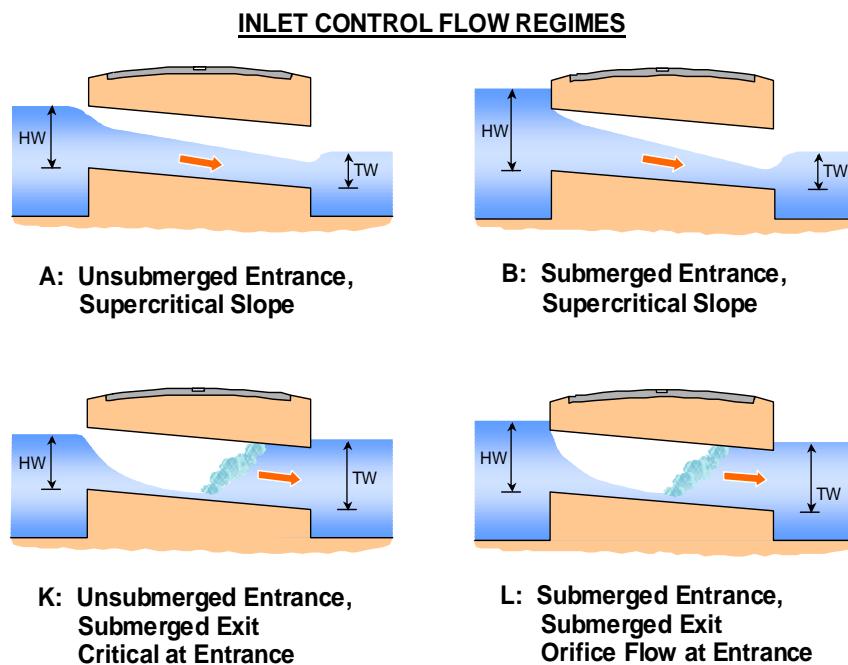


Figure 11-9 1D Inlet Control Culvert Flow Regimes

Culverts are defined using the [Flux function](#) command. In combination with the Flux function command, culvert structure information is defined within the model via a culvert database (see [Culvert file](#)). The culvert file contains a list of the culvert attributes, such as the culvert ID, culvert type, dimensions, length, upstream and downstream inverts, number of barrels, Manning's n , entrance and exit losses. A description of the required culvert file inputs is summarised in Table 11-3

Multiple culverts can be listed within a culvert file (i.e. a unique culvert file for each culvert is not required). The culvert ID within the culvert file is used to associate a specific structure with the command line input. Example inputs are shown below in Figure 11-10.

```

Structure == Linked Nodestrings, 7, 8      ! Structure Location (from NS 7 to NS 8)
  Flux function == Culvert                 ! Culvert flow between nodestrings
  Culvert File == Culvert_dbase.csv, 2     ! Culvert details, ID 2 in culvert file
End Structure

Structure == Nodestrings, 9                  ! Structure Location (NS 9)
  Flux function == Culvert                 ! Culvert flow between nodestrings
  Culvert File == Culvert_dbase.csv, 3     ! Culvert details, ID 2 in culvert file
End Structure
  
```

Culvert_database.csv															
ID	Type	Ignore	Len_or_ANA	n_or_n_F	US_Invert	DS_Invert	Form_Loss	pBlockage	Width_or_Dia	Height_or_WF	Number_of_	Height_Cont	Width_Cont	Entry_Loss	Exit_Loss
1	2	0	15	0.015	2.1	2.1	0	0	0.9	1.4	1	0.6	1.0	0.5	1
2	1	0	20	0.015	5.0	4.3	0	0	1.8	0.0	1	0.6	1.0	0.5	1
3	1	0	15	0.015	3.0	2.9	0	0	0.6	0.0	1	0.6	1.0	0.5	1
4	2	0	16	0.015	3.0	2.8	0	0	2.4	1.4	6	0.6	1.0	0.5	1
5	2	0	12	0.015	1.1	1.0	0	0	2.4	1.4	3	0.6	1.0	0.5	1
6	2	0	12	0.015	1.1	1.1	0	0	2.4	1.4	3	0.6	1.0	0.5	1

Figure 11-10 1D Culvert Example

Table 11-3 Culvert File Inputs

Header Column	Description
ID	Culvert identifier. Links to “culvert file” command line.
Type	1 = circular, 2 = rectangular, 4 = gated circular (unidirectional), 5 = gated rectangular (unidirectional).
Ignore	If = 1 culvert is ignored (Default = 0).
UCS	Currently not used.
Len_or_ANA	Culvert length (m or ft – see units).
n_or_n_F	Friction (Manning’s ‘n’).
US_Invert	Upstream invert level (relative to model datum: m or ft – see units).
DS_Invert	Downstream invert level (relative to model datum: m or ft – see units).
Form_Loss	Form loss coefficient; an additional dynamic head loss coefficient applied when culvert flow is not critical at the inlet.
pBlockage	% blockage (for 10%, enter 10). For rectangular culverts, the culvert width is reduced by the % Blockage, while for circular culverts the pipe diameter is reduced by the square root of the % Blockage. (Default = 0).
Inlet_Type	Currently not used.
Conn_2D	Currently not used.
Conn_No	Currently not used.
Width_or_Dia	Width for rectangular culverts or diameter for circular culverts (m or ft – see units).
Height_or_WF	Height for rectangular culverts (m or ft – see units).
Number_of	Number of culvert barrels.
Height_Cont	Height contraction coefficient for orifice flow at the inlet. Recommended values, 0.6 for square edged entrances to 0.8 for rounded edges. Not used for unsubmerged inlet flow conditions or outlet controlled flow regimes. Not used for C channels.
Width_Cont	The width contraction coefficient for inlet-controlled flow. Usually 0.9 for sharp edges to 1.0 for rounded edges for R culverts. Normally set to 1.0 for C culverts. If value exceeds 1.0 or is less than or equal to zero, it is set to 1.0. Not used for outlet controlled flow regimes.
Entry_Loss	The entry loss coefficient for outlet controlled flow (recommended value of 0.5).
Exit_Loss	The exit loss coefficient for outlet controlled flow (recommended value of 1.0).

11.4 Energy Losses

Please note: The commands within this Section are **recommended for use only with the NLSWE flux function for nodestring and linked nodestring structure types.**

11.4.1 Energy Loss Functions

When using the NLSWE [flux function](#) (which is also the default if not explicitly specified via the [flux function](#) command) sub-element scale energy losses can be represented via two alternative methods as follows:

1. Energy losses calculated by the model using a combination of user specified cell [form loss coefficient energy loss function](#) and optional application of a [width](#) (or [blockage](#)) file (refer to Section 11.4.3); or
2. A lookup [table energy loss function](#) of flows and pre-calculated energy losses (refer to Section 11.4.4).

11.4.2 Energy Loss Considerations

When adapting structure loss coefficients from a 1D model or from coefficients that apply across the entire waterway, for example, from Hydraulics of Bridge Waterways (FHA 1973), the following should be noted:

- The 2D solution automatically predicts most “macro” losses due to the expansion and contraction of water through a constriction, or round a bend, provided the resolution of the grid is sufficiently fine. It is recommended that raised bridge approaches/abutments should be represented by the model topography (i.e. not included as a contribution to the form loss coefficient). A breakline using the [Read GIS Z Line](#) command may be useful for defining these topographic features. Where the waterway width varies slightly from the cumulative width of the cells across which the structure is being applied, a [width file](#) or [blockage file](#) can be used to refine the flow area and define the structure soffit within the model¹³.
- Where the 2D model is not at a fine enough resolution to simulate the “micro” losses (e.g. from bridge piers, vena contracta, losses in the vertical (3rd) dimension), additional form loss coefficients and/or modifications to the cells widths and flow height need to be added. This can be done by using a the [form loss coefficient](#) or [energy file](#) commands.
- The additional or “micro” losses, which may be derived from information in publications, such as Hydraulics of Bridge Waterways, should be distributed evenly across the waterway (i.e. rather than being too specific about the representation of each individual cell).
- The head loss across key structures should be reviewed, and if necessary, benchmarked against other methods (e.g. Hydraulics of Bridge Waterways or a secondary model). Note that a well-designed 2D model will be more accurate than a 1D model if any “micro” losses are

¹³ Width or blockage files are only applicable if used in combination with the form loss coefficient energy loss function.

incorporated. Ultimately, the best approach is to calibrate the structure through adjustment of the additional “micro” losses – but this, of course, requires good calibration data!

11.4.3 Form Loss Coefficient

The coefficient [energy loss function](#) allows the user to specify a [form loss coefficient](#) (FLC) as a function of velocity head ($v^2/2g$), providing a the modeller with a mechanism to allow for sub-element scale energy losses around piers and similar entry/exit losses that are not captured in the 2D/3D domain.. For slowly moving flow, the head loss is minimal, but for fast moving water it can become significant.

$$\Delta H = FLC \frac{v^2}{2g}$$

[Width files](#) or [blockage files](#) can optionally be used in combination with the coefficient [energy loss function](#). They allow the user to specify the approximate width of the structure as a function of elevation, for example to account for flow area reduction due to abutments, piers or other flow obstructions not represented by the model mesh. These files alter the effective cross-sectional area at the structure, modifying the velocity in accordance with the laws of continuity: $v' = Q/A'$ where A' is the modified cross-sectional area of the structure and Q is the flowrate. This increased velocity v' is used to assign kinetic energy losses via FLC $v'^2/2g$.

```
Structure == Nodestring, 6           ! Structure location (NS 6)
Energy loss function == coefficient ! Loss type
Form loss coefficient == 0.2        ! Pier losses (hydraulics of bridged waterways)
Width file == Width_Example.csv    ! Width file
End Structure
```

Width_Example.csv		
Z	WIDTH	Comment
-5	0	!Bed Invert
-4.9	100	!Bridge Opening
5	100	!Bridge Opening
5.1	0	!Bridge Deck and Railing
7.1	0	!Bridge Deck and Railing
7.2	100	
12	100	

11.4.4 Energy Table

An energy table [energy loss function](#) gives users a higher level of control over the losses applied for different flow rates. The downside to this additional level of control is that the user is required to pre-calculate the expected energy losses and input via an [energy file](#). TUFLOW FV interpolates these energy losses (constant value extrapolation if outside the bounds set in the [energy file](#)) based on the modelled flowrate through the structure. Please note [width files](#) or [blockage files](#) should not be used with this method.

```
Structure == Nodestring, 6          ! Structure location (NS 6)
Energy loss function == table      ! Loss type
Energy loss file == energyloss.csv ! Pier losses (hydraulics of bridge waterways)
End Structure
```

energyloss.csv		
Q	dH	Comment
-10	0.1	! Flow against structure direction
0	0	! No head loss for no flow
1	0.1	! Nlarger head losses for +ve flow
5	0.2	
10	0.3	

11.5 Scalar Functions

The [scalar function](#) can be used to control the transfer of scalar variables through the structure. This is often (but not necessarily) used to simulate pumped structures with some filter or treatment of the scalar variables (salinity, temperature, sediments, tracers and water quality constituents) through them.

The available scalar functions are as follows:

- **None** – scalar variables are unaffected by flow through the structure (the default).
- **Timeseries** – scalar concentrations through the structure are controlled by a separate CSV file that specifies them as changing in time.
- **Scale_factor** – scalar concentrations are multiplied by a relevant scale factor from the upstream to the downstream.
- **Treatment** – scalar concentrations are multiplied by a relevant scale factor from the upstream to downstream. If scaling reduces concentrations to a specified lower threshold [min conc](#) then no more scaling is done. Similarly, if concentrations are already below this threshold, no scaling is done at all. If a maximum concentration is specified via [max conc](#), then the concentrations will be limited to this, regardless of the scale factor.

NOTE: when these structure types are used, the mass of scalar variables through the structures is not conserved (i.e. this is typically the purpose of these structures, to add or remove mass from the model). If reviewing the model MASS.CSV you may find that the overall mass in the model is increasing or decreasing, and if this is the case it will likely be due to one of these structures. Care should be taken to review the structcheck and/or structflux model outputs to ensure that the structure is operating as intended.

Several examples of scalar functions are provided below for the ‘timeseries’, ‘scale factor’ and ‘treatment’ functions.

‘Timeseries’ [flux function](#) example applied at a linked zones structure type with no [scalar function](#):

```
Structure == LinkedZones, 3,8           ! Structure Location (from NS 5 to NS 6)
Polygon file == example_polyfile.csv    ! File with polygons in it
Flux function == timeseries            ! Specified flow from csv
Flux file == example_FluxFile.csv      ! Timeseries of flow between zones

Scalar function == None                 ! Structure does not affect scalar concentration
end structure
```

‘Timeseries’ [flux function](#) example applied at a linked zones structure type with a ‘timeseries’ [scalar function](#):

```
Structure == LinkedZones, 3,8           ! Structure Location (from NS 5 to NS 6)
Polygon file == example_polyfile.csv    ! File with polygons in it
Flux function == timeseries            ! Specified flow from csv
Flux file == example_FluxFile.csv      ! Timeseries of flow between zones

Scalar function == Timeseries          ! Structure does not affect scalar concentration
Scalar file == scalar_tseries.csv      ! Timeseries of the concentration of the outflow of scalars
end structure
```

scalar_tseries.csv					
Time	Sal	Temp	SED_1	SED_2	Comment
0	15	25	1	0	
20	15	25	1	0	
40	15	25	1	0	
48	15	25	1	0	
48.001	20	25	5	0	Different scalars
60	20	25	5	0	
80	20	25	5	0	

‘Timeseries’ [flux function](#) example applied at a linked zones structure type with a ‘scale_factor’ [scalar function](#):

```
Structure == LinkedZones, 3,8           ! Structure Location (from NS 5 to NS 6)
Polygon file == example_polyfile.csv   ! File with polygons in it
Flux function == timeseries            ! Specified flow from csv
Flux file == example_FluxFile.csv     ! Timeseries of flow between zones

Scalar function == Scale_Factor        ! Structure does not affect scalar concentration
min_conc == -1, -1, 5, 5               ! dont limit sal, temp, do limit sediments
max_conc == 100, 100, 15, 15            ! dont limit sal, temp, do limit sediments
scal_fact == 1, 1, 0.5, 0.5             ! dont scale sal, temp, halve sediments
end structure
```

‘Timeseries’ [flux function](#) example applied at a linked zones structure type with a ‘treatment’ [scalar function](#):

```
Structure == LinkedZones, 3,8           ! Structure Location (from NS 5 to NS 6)
Polygon file == example_polyfile.csv   ! File with polygons in it
Flux function == timeseries            ! Specified flow from csv
Flux file == example_FluxFile.csv     ! Timeseries of flow between zones

Scalar function == Treatment          ! Structure does not affect scalar concentration
min_conc == -1, -1, 5, 5               ! dont limit sal, temp, do limit sediments
max_conc == 100, 100, 15, 15            ! dont limit sal, temp, do limit sediments
scal_fact == 1, 1, 0.5, 0.5             ! dont scale sal, temp, halve sediments
end structure
```

11.6 Structure Controls

11.6.1 Control Block

Sometimes it is desirable to adjust the flow through a structure or adjust the topography according to values calculated internally in the model in real-time. [Control](#) structures allow the user to specify internal model conditions and their outcome on the structure. Examples for this are a gated weir that closes according to downstream salinity. Or a bund that ‘fails’ when water levels reach a certain height.

Logic controls adjust flow conditions through a structure according to a series of logical rules specified by the user. This is particularly useful for applications with adjustable structures, such as drop gates / sluices / pumps or for levee breach/failure assessments.

Logic controls are specified in a [control block](#) nested within a [structure](#) block. Structures can have multiple controls, with the contributions of these combined in different ways depending on the controlled parameter.

11.6.2 Control Types

The different [control](#) types can adjust one of the following [control parameters](#):

- *Trigger*: Trigger controls define a condition that will commence after the first exceedance of a specific [trigger value](#) within the model at a specified [sample point](#). Currently TUFLOW FV is only able to sample water levels for trigger controls.
- *Timeseries*: Timeseries controls define an adjusting condition based on a separate [control file](#) CSV timeseries. The current timestep is interpolated into the timeseries to calculate a corresponding [control parameter](#) value.
- *Sample rule*: A sample rule control can adjust the [control parameter](#) by sampling a variable within the model and comparing it to a [control file](#) CSV file. Currently TUFLOW FV can sample the following variables using the [sample type](#) command at a [sample point](#) at a time interval specified by [sample dt](#):
 - $H, V_x, V_y, SAL, TEMP, SED_1 - SED_N, TRACE_1 - TRACE_N, WQ_1 - WQ_N$
- *Target rule*: A target rule control can adjust the [control parameter](#) to reach the target value of the [sample type](#) at a [sample point](#) at a time interval specified by [sample dt](#). The flow through the structure is adjusted based on the difference between the sampled value and the target value. The target value is specified in a [target file](#), which is a timeseries to allow for time-varying target conditions.

11.6.3 Control Parameters

The [control parameter](#) modifies the state of the control structure command with the following options:

- *Fraction_Open*: Effectively a scale factor (between 0 and 1) of the flow rate through the structure. Multiple fraction_open control parameter commands within the same structure will multiply together for the overall value. Fraction_Open is often used to simulate culverts or

pumps with a gate control that is automated based on the water level elsewhere in the region. Please note this control parameter is not supported with [flux function](#)== nlswe.

- *Min_Flow*: Provides a minimum flow rate that must pass through the structure (physical limitations permitting). This is often used in conjunction with dam release structures where a minimum flow is required for environmental releases. When multiple *min_flow* controls are used in the same structure, the maximum of these will be the adopted value. Please note this control parameter is not supported with [flux function](#)== nlswe.
- *Weir_Crest*: Used with *weir_adjust* or *weir_dz_adjust* [flux functions](#) (refer Section 11.3.4.2) and allows the crest height of the weir (absolute level or relative to the adjacent cells) to be adjusted based on the control. Often used to simulate a lowerable or failing levee system (please refer to Figure 11-7 for an example).
- *ZB*: Can only be used with cell or zone structure types using [bed_adjust](#) (refer Section 11.2.4) and adjusts the bathymetry of the structure area based on a condition. Often used to artificially adjust a model bed for scour with time, or to simulate a bund failure after a trigger condition.
- *DZB*: As per *ZB* above, but with all elevation changes relative to the existing bed level of the structure region/cell.

11.6.4 Structure Logging File

Operation of structure controls can be reviewed using the structure logging file (.slf output to the [logdir](#) folder), enabled via the [structure logging](#) command.

11.6.5 Structure Control Examples

Example 1: Levee Breach

Levee failure at cell 1041 once water level reaches 24m at the [sample point](#). Period of failure occurs over 10 hours.

```
Structure == Cell, 1041           ! Structure Location (in cell 1041)
  Bed Adjust == zb_adjust        ! Adjusts Bed Level

  Control == Trigger             ! Adjust after trigger level
    Sample point == 622943, 4334061 ! Location of sampling (x,y)
    Sample type == WL            ! Trigger can only use WL
    Sample dt == 0.05             ! hours
    Trigger Value == 24.0         ! Trigger level (m)
    Control file == BreachCondition.csv ! Timeseries of zb after trigger reached
  End Control

End Structure
```

BreachCondition.csv	
Time	Zb
0.0	24.0
2.0	23.0
4.0	22.0
6.0	21.0
8.0	20.0
10.0	19.0

Figure 11-11 Trigger Control Example

Example 2: Floodgate Control - Timeseries

Historic event modelling. Floodgate operation timeseries specification.

```

Structure == Nodestring, 3           ! Structure Location (NS 3)
Flux function == matrix              ! Flux type
Flux file == hQh.csv                 ! File with flux matrix

Control == timeseries               ! Control Type, uses a csv timeseries
Control parameter == fraction_open  ! Will scale flow rate
Control update dt == 1.              ! HRs
Control file == tseries.csv         ! csv timeseries of fraction_open
max opening increment == 0.1         ! Never open by more than this per timestep
End Control

End Structure

```

tseries.csv	
Time	Fraction_Open
0.0	0.0
2.0	0.0
4.0	0.2
6.0	0.4
8.0	1.0
10.0	1.0

Figure 11-12 Timeseries Control Example

Example 3: Irrigation Control - Sample Control - Salinity

Floodgate operation defined using sample control (i.e. the gate operation is based on salinity monitoring at the sample point location).

```

Structure == Nodestring, 3           ! Structure Location (NS 3)
Flux function == matrix              ! Flux type
Flux file == hQh.csv                 ! File with flux matrix

Control == sample_rule               ! Adjust based on sample in model
Control parameter == fraction_open  ! Will scale flow rate
Control update dt == 1.              ! HRs
Sample type == SAL                  ! Sample salinity
Sample point == 364352., 8139574.    ! Sample Location (x,y)
Vertical coordinate type == height ! Sample vertical averaging type. Above bed
Vertical coordinate == 0, 2           ! Sample between 0 and 2m above bed
Sample dt == 0.1                     ! HRs
max opening increment == 0.1         ! Never open by more than this per timestep
Sample file == sample_control.csv   ! File relating sample to fraction_open
End Control

End Structure

```

sample_control.csv	
Sample_Value	Fraction_Open
10.0	0.0
10.2	0.1
10.4	0.3
10.6	0.5
10.8	0.8
11.0	1.0
12.0	0.8
13.0	0.5

Figure 11-13 Sample Rule Control Example

Example 4: Irrigation Control - Sample Control - Salinity

Flow across a nodestring regulated by the surface salinity in a part of the model (no flow if salinity>20 ppt).

```

Structure == Nodestring, 5           ! Structure Location (NS 5)
flux function == timeseries          ! Specified flow rates with time
flux file == flow_pump.csv           ! csv with flow rates in it

control == Sample_Rule               ! Sample a point to control structure
sample point == 502645, 6962207      ! Sample location (this eg. in UTM)
vertical coordinate type == depth   ! Sample vertical coordinate reference
vertical coordinate == 0,1            ! Sample over top 1 meter
sample type == SAL                  ! Sample Salinity
sample dt == 300                     ! Sample over a 5 minute period (moving average)

control parameter == fraction open  ! Adjust opening based on control
control update dt == 900             ! Only update every 15 minutes
start control state == 1             ! Start Open
control file == example_sample.csv   ! CSV of control relationship
end control

end structure

```

example_sample.csv	
Sample_Value	Fraction_Open
0	1
20	1
20.01	0

flow_pump.csv	
Time	Q
0	10
99999	10

Figure 11-14 Sample Rule Control Example

12 Model Outputs

Chapter Contents

12.1	Introduction	12-2
12.2	Output Formats	12-3
12.3	Point Output	12-4
12.4	Profile Output	12-5
12.5	Flux/Mass Output	12-6
12.6	Map Output	12-8
12.6.1	Overview	12-8
12.6.2	2D Only Map Output Formats	12-8
12.6.3	3D Map Outputs	12-9
	<i>3D Vertical Averaging Options</i>	12-9
12.6.4	2D and 3D NETCDF Format	12-10
12.6.5	Map Output Statistics	12-15
12.7	Map Output Parameters	12-16
12.8	Structure Outputs	12-23
12.8.1	Structure Flux	12-23
12.8.2	Structure Check	12-23
12.9	Check Files and GIS Layers	12-25
12.9.1	GIS Workspaces (.wor and .qgs files)	12-25
12.10	GIS Check Files	12-26

12.1 Introduction

TUFLOW FV outputs cross-check information and calculation results in a range of different file formats for use within a wide choice of text, plotting, GIS and GUI visualisation tools and software.

This chapter discusses options to configure and customise TUFLOW FV's output prior to carrying out a simulation.

12.2 Output Formats

TUFLOW FV offers various model output options. The output type, parameter(s)¹⁴ and time interval are specified using [output](#) block commands. Results are saved to a location defined via the [output dir](#) command. Common model output types include:

- Point time-series in text format (2D or vertically averaged 3D)
- Mapped output in SMS Data File (2D or vertically averaged 3D, DATV or XMDF) or NetCDF (2D, vertically averaged 3D or full 3D) formats
- Mapped statistical output in SMS Data File (2D or vertically averaged 3D) or NetCDF (2D, vertically averaged 3D or full 3D) formats
- Vertical profile time-series output at point locations in NetCDF format (full 3D)

The types of output will typically depend on the TUFLOW FV hydrodynamic calculation mode (2D or 3D), the available model calibration/validation data, the objectives of the modelling exercise and the modeller's preferred method of communicating assessment results.

The range of output formats available are summarised in Table 12-1.

Table 12-1 Output Data Formats

Output Format	Description	Data Type	Relevant Section
Points	Time-series results for specific point locations.	CSV	Section 12.3
Profile	Outputs the time history of the vertical profile at a point location.	NetCDF	Section 12.4
Flux and/or Mass	Outputs flux across all model nodestrings. This will output flow, salinity, temperature, sediment and scalar fluxes if they are being simulated.	CSV	Section 12.5
Map Outputs	Time-series 2D or 3D model output for the entire mesh.	DATV XMDF NetCDF	Section 12.6
Transport	A file containing the conserved variables stored on the TUFLOW FV mesh to be used in a subsequent advection-diffusion, sediment or particle tracking simulation.	NetCDF	Section 10.8

¹⁴ Parameters are referred to as Map Output Data Types if you're a TUFLOW Classic or HPC User.

12.3 Point Output

The TUFLOW FV 2D ‘points’ option provides model parameter time-series at point locations, each defined by a 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. This type of model output is often used to compare with time-series data recorded at a fixed location as part of a model calibration/validation exercise. Example points [output block](#) commands are provided in Figure 12-1. Results are saved to a location defined via the [output dir](#) command.

```
output == points
  output points file == ..\geo\points\output_points.csv           !Output points file
  output parameters == H,V                                         !Output parameters
  output interval == 900.                                         !Output interval in seconds
end output
```

X	Y	ID
50	50	ADCP1
500	50	ADCP2
900	50	ADCP3

Figure 12-1 Points Output Commands and ‘Output_points.CSV’ Example

The TUFLOW FV 2D points output file is a comma separated variable format (*.POINTS.CSV) that can be opened and viewed in a text editor or spreadsheet software such as Microsoft Excel. The file contains the time-series of model parameters at the point locations defined in the [output points file](#). Following the example in Figure 12-1, a sample of the corresponding *.POINTS.CSV output file is shown in Figure 12-2.

TIME	ID_ADCP1_H [m]	ID_ADCP1_VX [m s^-1]	ID_ADCP1_VY [m s^-1]	ID_ADCP2_H [m]	ID_ADCP2_VX [m s^-1]	ID_ADCP2_VY [m s^-1]	ID_ADCP3_H [m]	ID_ADCP3_VX [m s^-1]	ID_ADCP3_VY [m s^-1]
0.00	-3.50	0.00	0.00	-3.50	0.00	0.00	-3.50	0.00	0.00
0.17	-3.45	0.18	0.00	-3.48	0.15	0.00	-3.50	0.12	0.00
0.34	-3.47	0.37	0.00	-3.49	0.27	0.00	-3.50	0.21	0.00
0.50	-3.46	0.56	0.00	-3.49	0.46	0.00	-3.50	0.38	0.00
0.67	-3.44	0.73	0.00	-3.48	0.59	0.00	-3.50	0.49	0.00
0.84	-3.43	0.91	0.00	-3.48	0.75	0.00	-3.50	0.62	0.00
1.00	-3.40	1.06	0.00	-3.47	0.90	0.00	-3.50	0.75	0.00

Figure 12-2 TUFLOW FV 2D Points Output File Example

12.4 Profile Output

For 3D simulations, profile output at point locations may be requested and output to a NetCDF file. Example use of the output profile command within an [output block](#) is provided in Figure 12-3.

An example profile output is shown in Figure 12-4 whereby salinity at a point location is plotted as a function of depth for a given timestep. Results are saved to a location defined via the [output dir](#) command.

```
output == profiles           !Output type
  output points file == ..\geo\points\profile_points_000.csv !Output points file
  output parameters == H, V !Output parameters
  output interval == 900    !Output interval
end output
```

Figure 12-3 Profile Output Example Commands

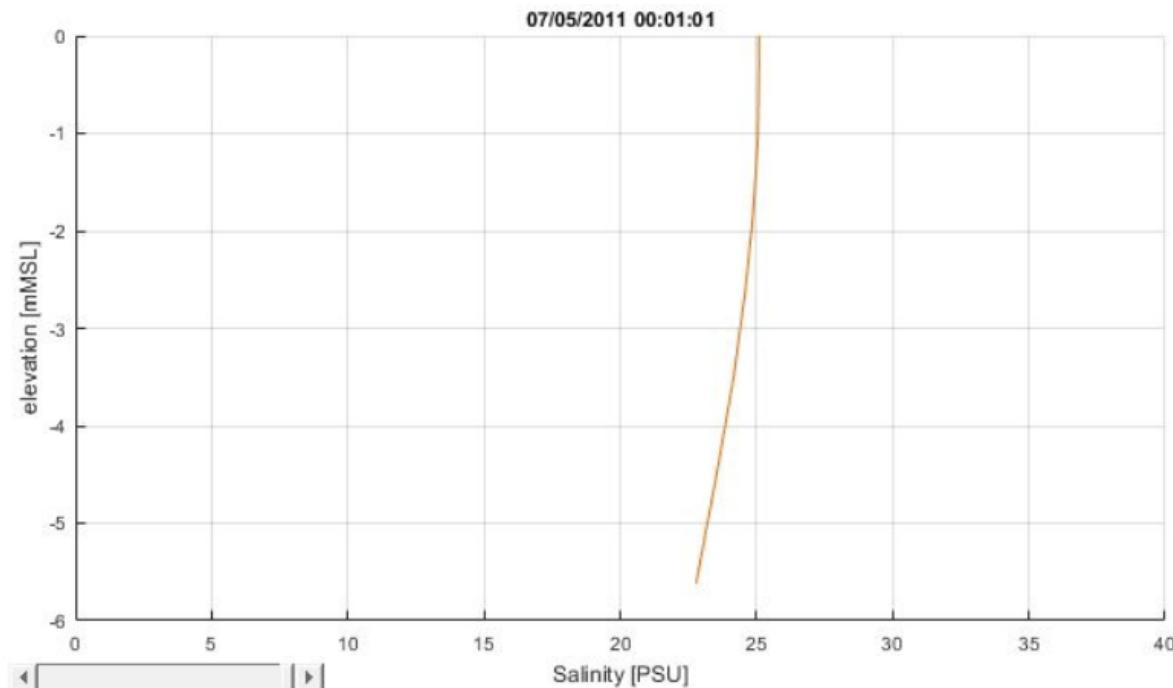


Figure 12-4 Example profile output

12.5 Flux/Mass Output

The flux or mass of various parameters (refer to Table 12-2) can be output using the flux or mass output commands. Results are saved to a location defined via the [output dir](#) command.

The values of each parameter in the mass and flux files are made up of the concentrations of each scalar conserved variable multiplied by the volume of water or the flow rate respectively. These units are not always immediately intuitive (for example the heat ‘mass’) but if divided by either the flow rate or volume (for flux terms and mass terms respectively) they will be equivalent to the average concentration of the parameter that is either passing through a given nodestring, or within the whole domain. These files can be used to ensure that no instabilities are causing spurious changes in conserved quantities, that source/sink terms are operating as expected and that external boundaries (such as an offshore ocean boundary) are not influencing the conserved mass of parameters.

- The simulation ‘mass’ file output is used to check the volume of fluid and, where applicable, other simulated quantities within the model domain. The command to output the mass file is simply:

```
output == mass
  output interval == 900           !Output interval in seconds
end output
```

- The simulation flux file used to check the rates of fluid and, where applicable, other simulated quantities entering/exiting the model boundaries or crossing specified nodestrings within the model domain. A separate set of time-series is provided for each nodestring in the model domain. The command to output the flux file is simply:

```
output == flux
  output interval == 900           !Output interval in seconds
end output
```

Table 12-2 Flux and Mass Outputs

Variable Name	Quantity Measured	Units
FLOW	The flow rate of water passing the nodestring.	m ³ /s
SALT_FLUX	The mass rate of salt passing the nodestring.	PSU x m ³ /s approximately kg/s
TEMP_FLUX	The ‘mass’ rate of temperature passing the nodestring.	Degrees C x m ³ /s (similar to heat energy of the moving water without factoring in pressure and heat capacity)
TRACE_#_FLUX	The mass rate of a tracer ‘n’ passing the nodestring. For example, TRACE_1_FLUX.	concentration units x m ³ /s Tracers can be thought of as any chosen concentration unit, so if mg/L then this would be in g/s.

Variable Name	Quantity Measured	Units
SED_#_FLUX	The sediment fraction # mass rate passing the nodestring. For example, SED_1_FLUX.	kg/s
VOLUME	The total volume of water in the whole model domain.	m ³
SALT_MASS	The mass of salt in the whole model domain.	PSU x m ³ approximately kg
TEMP_MASS	The ‘mass’ of temperature in the whole model domain.	Degrees C x m ³
TRACE_#_MASS	The mass of tracer # in the whole model domain. For example, TRACE_1_MASS.	concentration units x m ³
SED_#_MASS	The mass of sediment fraction # in the whole model domain. For example, SED_1_MASS.	kg

12.6 Map Output

12.6.1 Overview

Map outputs are time and spatially varying results that can either be:

- 2D outputs from a 2D depth-averaged model, typically in DATV or XMDF format (refer Section 12.6.2);
- A 2D visualisation of a 3D model output typically in DATV or XMDF format (refer Section 12.6.3); and/or
- Direct 3D model output, typically in cell-centred NetCDF format (refer Section 12.6.3);

These outputs are commonly used to complete statistical analysis, mapping, result checking and visualisation and are used to produce a range of plot types. This section describes the available options and parameters available in TUFLOW FV.

12.6.2 2D Only Map Output Formats

The TUFLOW FV ‘DATV’ or ‘XMDF’ options provide output on the mesh nodes at the specified output time interval. The output format is the SMS Data File binary formats and can be viewed using the SMS Generic Mesh Module or via the QGIS TUFLOW Viewer Plugin.

In builds 2019.01 and later, TUFLOW FV produces a SMS Super File’ (.sup) that links the model mesh and result outputs, allowing the user to open a single .sup file to view all model parameters at once from a given run.

If using DATV [output](#) a separate output file is created for each specified model parameter (e.g. *_H.dat, *_V.dat, *_ZB.dat). The Super file will have the name <fvfilename>.ALL.sup.

If using XMDF, a single output file with .xmdf extension is produced that contains all output parameters. XMDF also allows file compression, which may be useful if dealing with large models and needing to reduce the storage footprint of your model results. The Super file produced will have the file name <fvfilename>.xmdf.sup. Results are saved to a location defined via the [output dir](#) command.

Example datv and xmdf [output block](#) commands and are provided in Figure 12-5.

```

! XMDF Output
output == xmdf
  output parameters == h,v,d      ! Output Parameters
  output interval == 600.          ! Output interval in seconds
end output

! DATV Output
output == xmdf
  output parameters == h,v,d      ! Output Parameters
  output interval == 600.          ! Output interval in seconds
end output

```

Figure 12-5 SMS Mapped Output Commands Example

The screenshot in Figure 12-6 provides an example of TUFLOW FV ‘datv’ output opened in the SMS Generic Mesh Module environment.

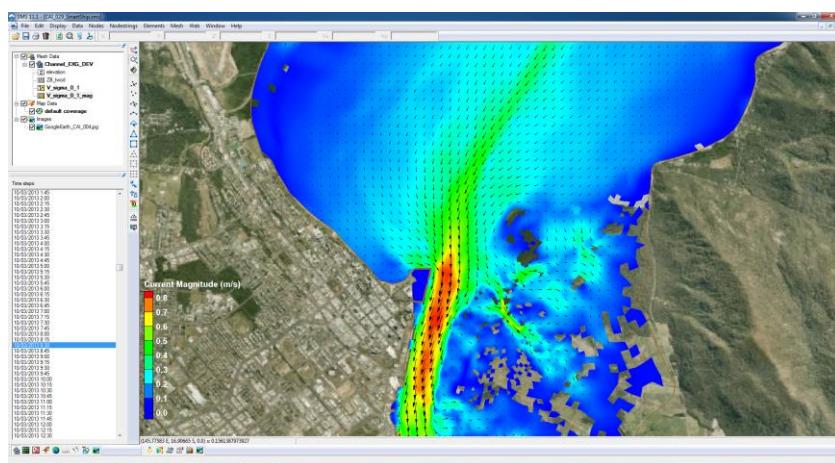


Figure 12-6 TUFLOW FV Mapped Current Velocity Output in the SMS Generic Mesh Module Environment

Various post-processing tasks can be undertaken using SMS, including data extraction and creating animations. The TUFLOW FV Wiki and Aquaveo SMS website provide post-processing examples and tips:

- TUFLOW FV Wiki: <http://fvwiki.tuflow.com>
- Aquaveo SMS website: <http://www.aquaveo.com/software/sms-learning-tutorials>

12.6.3 3D Map Outputs

TUFLOW FV 3D offers various vertically averaged output options intended to simplify 3D post-processing tasks and allow output based on 3D calculations to be viewed using the SMS Generic Mesh Module or QGIS TUFLOW Viewer Plugin. TUFLOW FV 3D vertically averaged model output can be generated in the following formats:

- Time-series at a point location (or multiple point locations) defined in an [output points file](#);
- Mapped output DATV or XMDF format; or
- Mapped output in NetCDF format (see Section 12.6.4).

The 3D vertical averaging options and commands are described below.

3D Vertical Averaging Options

The following vertical averaging options are available when using TUFLOW FV 3D to allow the 2D visualisation of 3D datasets:

- depth-all – averaging over entire water column
- depth-range – averaging between specified minimum and maximum absolute depths measured downward from water surface

- height-range – averaging between specified minimum and maximum absolute heights measured upward from the bed
- elevation-range – averaging between specified minimum and maximum elevations relative to model vertical datum
- sigma-range – averaging between specified decimal fraction of the water column where 0 is the bed and 1 is the water surface
- layer-range-top – averaging between layers referenced from the water surface (i.e. surface layer is 1, positive downwards)
- layer-range-bot – averaging between layers referenced from the bed (i.e. bottom layer is 1, positive upwards)

The depth-all option simply averages the 3D output over the entire water column, giving 2D depth averaged output based on the 3D calculations. The other options allow the modeller to specify a range of the water column to vertically average over. Examples of each depth averaging command are described further in the [Vertical averaging](#) command section or the [TUFLOW FV Wiki](#).

- The [suffix](#) command is often used with the [Vertical averaging](#) command to add a clear identifier to the output filename.
- Multiple model [output parameters](#) may be specified in a single [output block](#).
- Multiple vertical averaging outputs are supported but must be added to separate [output block](#) commands for each vertical averaging specification.

12.6.4 2D and 3D NETCDF Format

The NetCDF (network Common Data Form) software was developed at the Unidata Program Centre in Boulder, Colorado. It is an interface for array-oriented data access and a library that provides implementation of the interface. The NetCDF library also defines a machine-independent format for representing scientific data. Together, the interface, library and format support the creation, access and sharing of scientific data (Unidata, 2014).

TUFLOW FV NetCDF output adopts the NetCDF-4/HDF5 format. The output file is self-describing and contains information regarding the model geometry (2D or 3D) together with the mapped output at the specified time interval. The NetCDF format offers the following advantages:

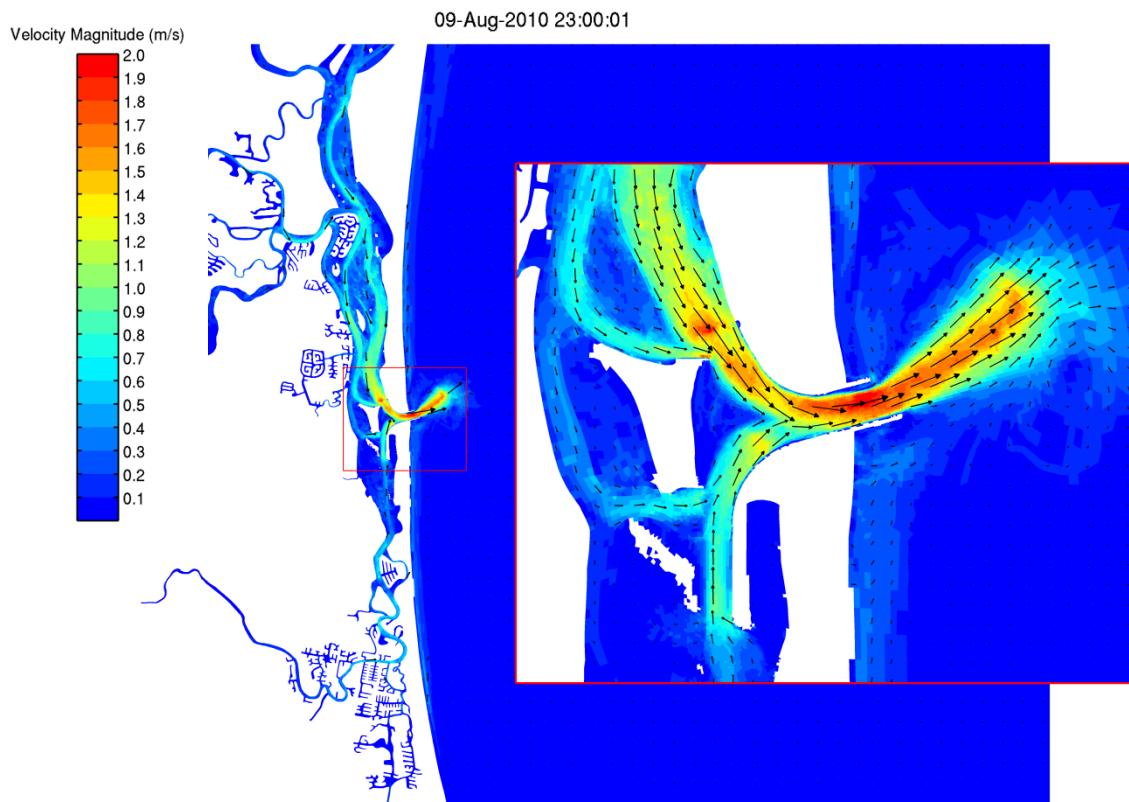
- Storage of the “cell-centred” output as calculated by TUFLOW FV (i.e. no interpolation to the cell nodes as required by the SMS Data File Formats DATV and XMDF);
- The files are machine-independent and can be viewed using any numerical analysis package with a NetCDF library interface, including MATLAB, R, GNU Octave or Python NumPy;
- An ability to store full 3D output in a single compressed file format; and
- An ability to view the vertical distribution of modelled parameters.

MATLAB or Python is typically used to view TUFLOW FV NetCDF output, extract data and generate animations. Some commonly used post-processing functions include:

- Sheet plots of vertically averaged, cell-centred model output (example in Figure 12-7);
- Curtain plots (longitudinal or cross-sectional) showing the vertical distribution of model output (examples in Figure 12-8 and Figure 12-9); and
- Conversion of TUFLOW FV NetCDF output to vertically averaged SMS Data File Format.

Both [MATLAB](#) and [Python](#) TUFLOW FV specific visualisation toolboxes are available to assist with the handling of 3D datasets.

The dimensions, variable definitions and attributes of a TUFLOW FV NetCDF output file are provided in Appendix DAppendix C. This information is intended to assist advanced users wishing to develop functions and scripts to post-process and/or view TUFLOW FV NetCDF output.



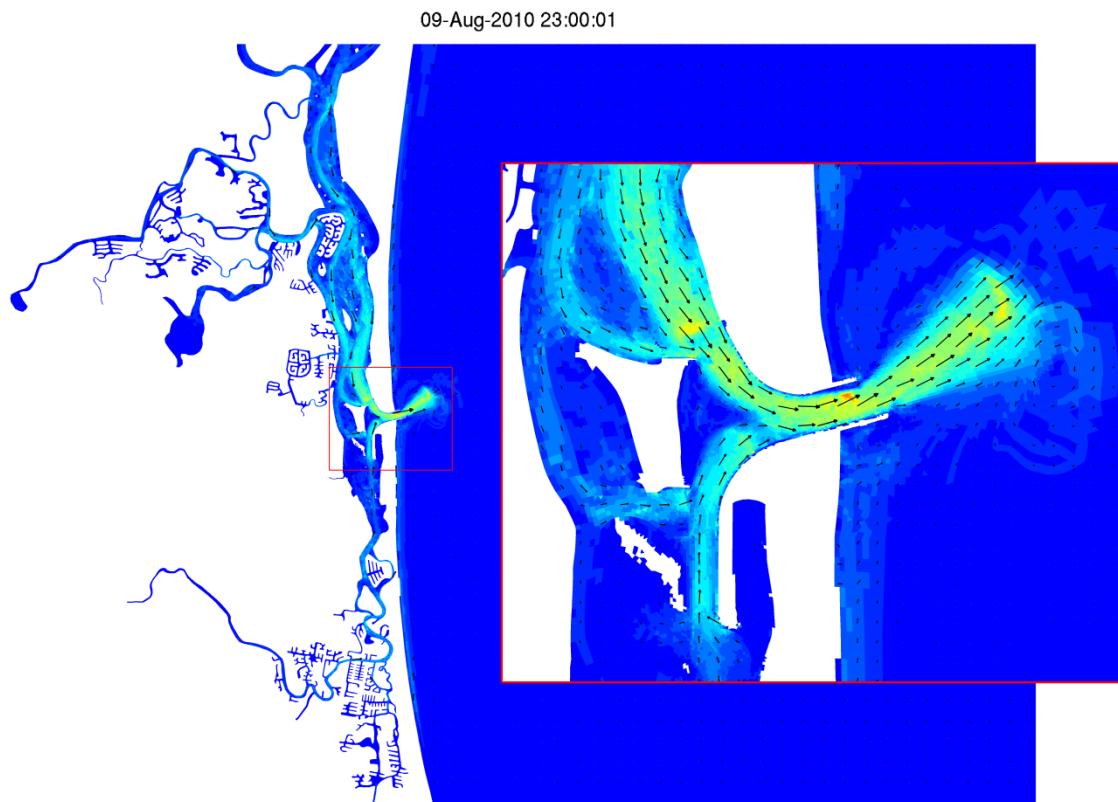
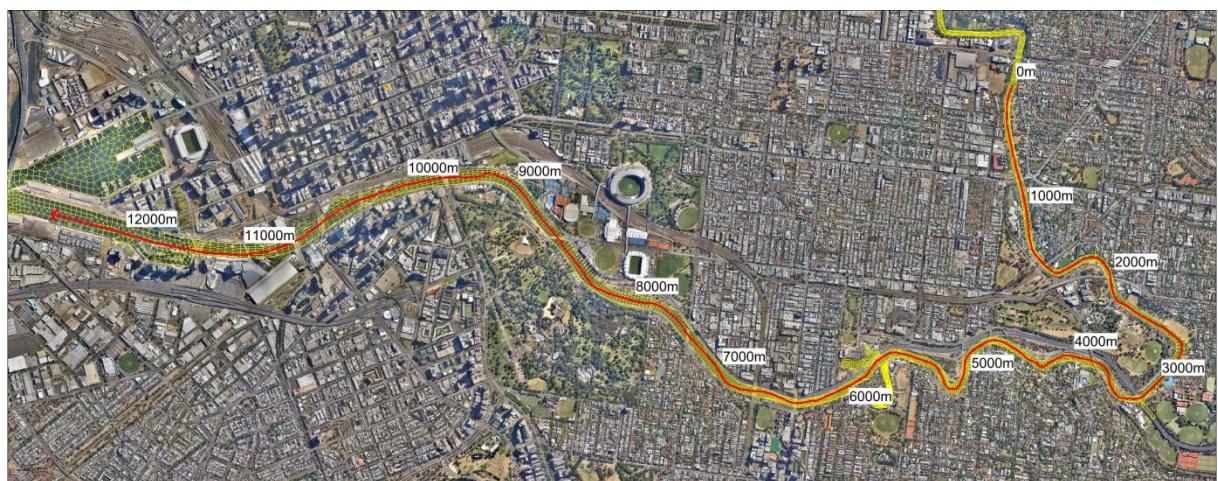


Figure 12-7 TUFLOW FV Sheet Plot with Zoom Example: Velocity Magnitude Top 50% Water Column (top); Velocity Magnitude Bottom 50% Water Column (bottom)



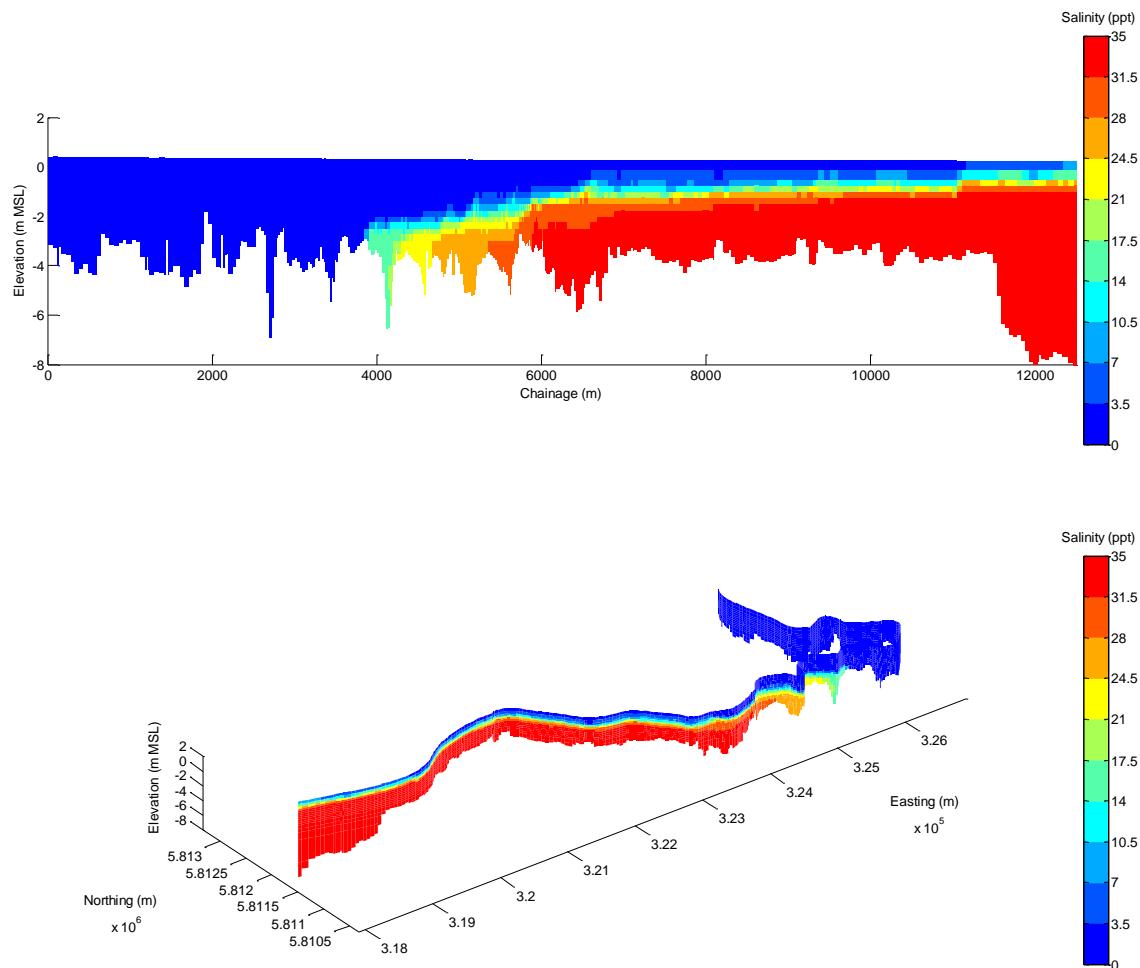


Figure 12-8 TUFLOW FV Salinity Vertical Distribution: Model Mesh and Curtain Polyline (top); Salinity Curtain Plotted with Polyline Chainage; Salinity Curtain Plotted with Polyline Coordinates (bottom)

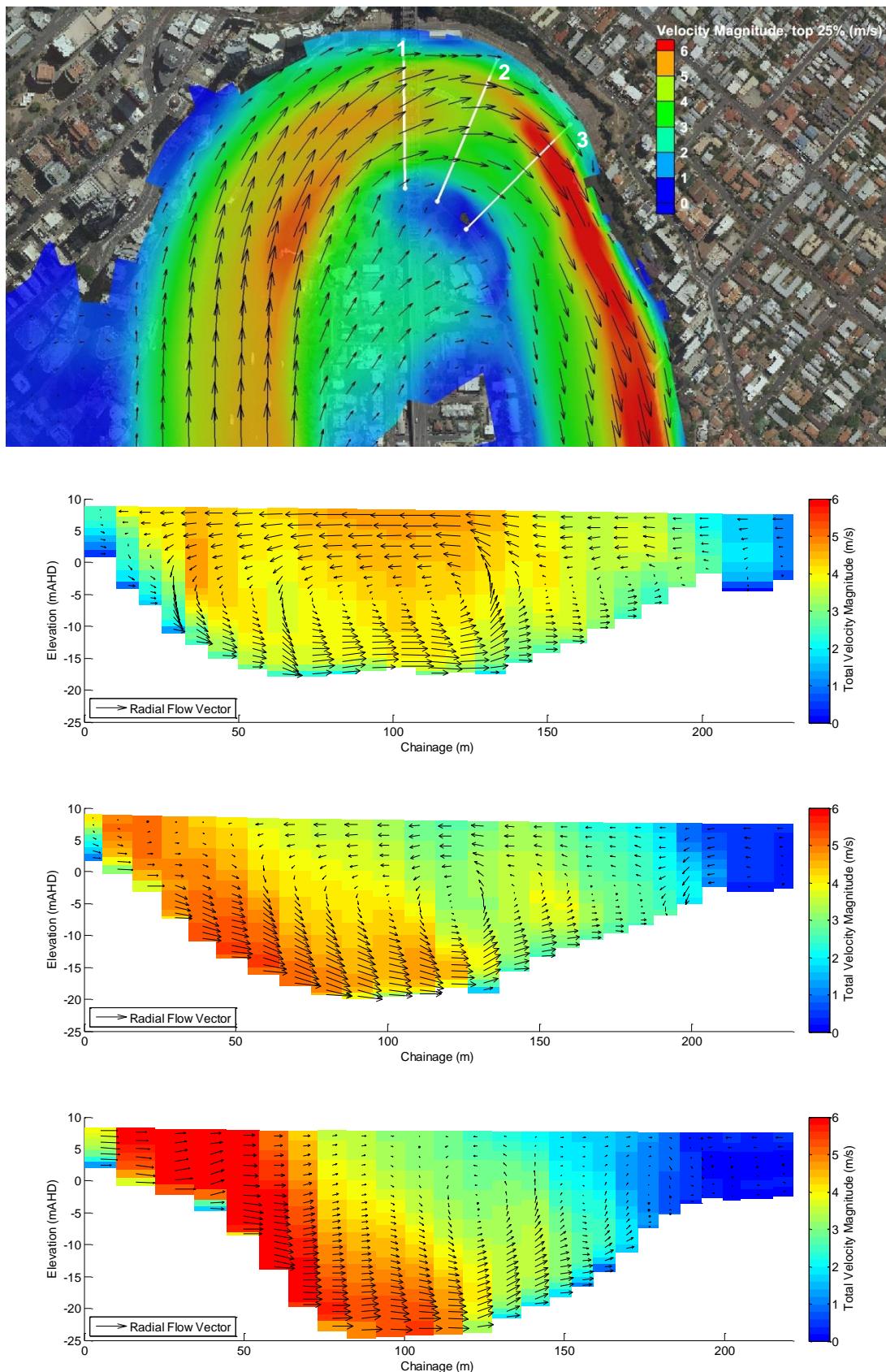


Figure 12-9 TUFLOW FV Velocity Vertical Distribution: River Bend Flood Flow and Cross-Section Locations (top); Total Velocity Magnitude (contours) with Radial Flow Vectors Cross-Sections (bottom three)

12.6.5 Map Output Statistics

The maximum and/or minimum values of the specified model parameters can be tracked during a TUFLOW FV simulation using the ‘output statistics’ command. Statistical output can be generated for the following output formats:

- Mapped output in XMDF or DATV format; or
- Mapped output in NetCDF format (see Section 12.6.4).

Example use of the output statistics command within an [output block](#) is provided in Figure 12-10. The ‘output statistics dt’ must be specified in addition to the ‘output interval’. In the example below, DATV output files would be created at 900 second intervals with the maximum tracked at a 1 second interval.

```
output == datv          !Output type (datv or netcdf)
output parameters == H,V !Output parameters
output interval == 900   !Output interval in seconds
output statistics == max !Output statistics type, max or min
output statistics dt == 1 !Output statistics dt in seconds
end output
```

Figure 12-10 Statistical Output Example Commands

12.7 Map Output Parameters

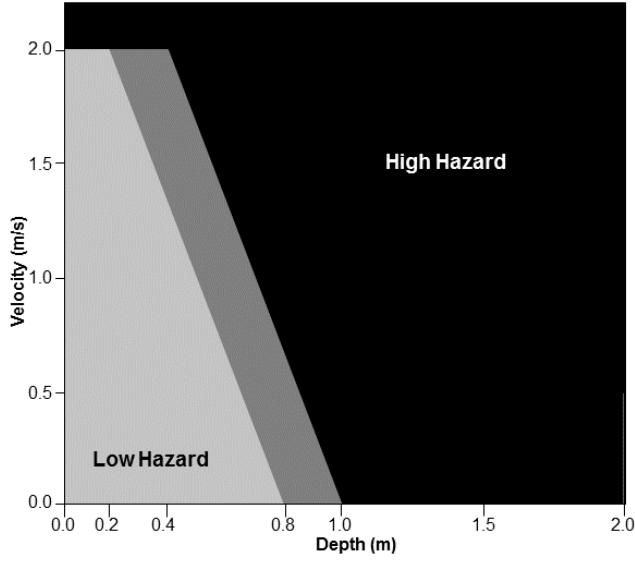
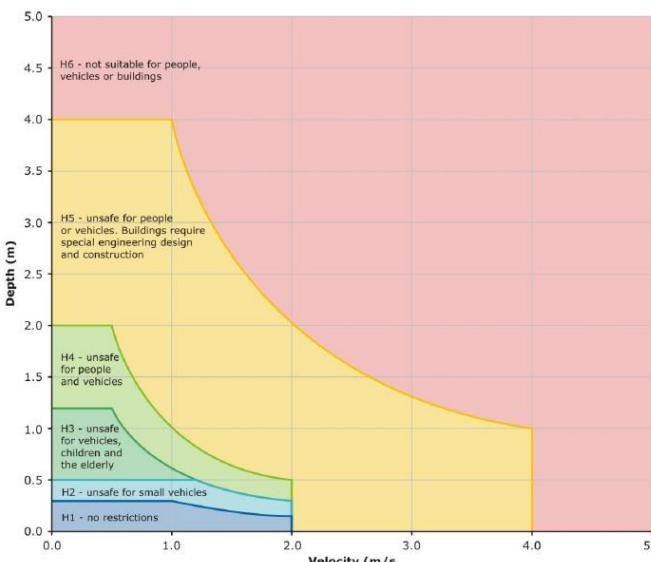
Common and advanced model parameters are provided in Table 12-3, Table 12-4, and Table 12-5.

Table 12-3 Common Map Output Types

Output Parameter	Description
D	Water depth (m)
H	Water surface elevation (m)
Taub	Bed shear stress (N/m ²) (Hydrodynamic module)
Taus	Surface shear stress (N/m ²) (Hydrodynamic module)
V	Velocity vector and magnitude (m/s)
Vmag	Velocity magnitude only (m/s)
W	Vertical Velocity (m/s)
ZB	Bed Elevation (m)

Table 12-4 Advanced Map Output Types

Output Parameter	Description
Air_temp	Air temperature (degrees Celsius)
Evap	Evaporation rate (m/day)
DZB	Bed elevation change (m)
Hazard_z1	Flood hazard category based on the Australian NSW Floodplain Management Manual (NSWG, 2005). The output is an integer number from 1 to 3 as follows and as illustrated in the figure below. 1 Low Hazard 2 Intermediate Hazard (dependent on-site conditions) 3 High Hazard Note: The maximum hazard value is monitored throughout the simulation and is not necessarily when the maximum water level occurs as with some other output.

Output Parameter	Description														
															
hazard_zaem1	<p>Flood hazard category as outlined by Australian Emergency Management Institute in 2014. ZAEM1 output values are 0 (zero) for no hazard and 1 to 6 for H1 to H6 respectively.</p>  <table border="1" data-bbox="547 1628 1246 1875"> <thead> <tr> <th data-bbox="547 1628 690 1684">Hazard Classification</th><th data-bbox="690 1628 1246 1684">Description</th></tr> </thead> <tbody> <tr> <td data-bbox="547 1684 690 1718">H1</td><td data-bbox="690 1684 1246 1718">Relatively benign flow conditions. No vulnerability constraints.</td></tr> <tr> <td data-bbox="547 1718 690 1740">H2</td><td data-bbox="690 1718 1246 1740">Unsafe for small vehicles.</td></tr> <tr> <td data-bbox="547 1740 690 1763">H3</td><td data-bbox="690 1740 1246 1763">Unsafe for all vehicles, children and the elderly.</td></tr> <tr> <td data-bbox="547 1763 690 1785">H4</td><td data-bbox="690 1763 1246 1785">Unsafe for all people and all vehicles.</td></tr> <tr> <td data-bbox="547 1785 690 1830">H5</td><td data-bbox="690 1785 1246 1830">Unsafe for all people and all vehicles. Buildings require special engineering design and construction.</td></tr> <tr> <td data-bbox="547 1830 690 1875">H6</td><td data-bbox="690 1830 1246 1875">Unconditionally dangerous. Not suitable for any type of development or evacuation access. All building types considered vulnerable to failure.</td></tr> </tbody> </table>	Hazard Classification	Description	H1	Relatively benign flow conditions. No vulnerability constraints.	H2	Unsafe for small vehicles.	H3	Unsafe for all vehicles, children and the elderly.	H4	Unsafe for all people and all vehicles.	H5	Unsafe for all people and all vehicles. Buildings require special engineering design and construction.	H6	Unconditionally dangerous. Not suitable for any type of development or evacuation access. All building types considered vulnerable to failure.
Hazard Classification	Description														
H1	Relatively benign flow conditions. No vulnerability constraints.														
H2	Unsafe for small vehicles.														
H3	Unsafe for all vehicles, children and the elderly.														
H4	Unsafe for all people and all vehicles.														
H5	Unsafe for all people and all vehicles. Buildings require special engineering design and construction.														
H6	Unconditionally dangerous. Not suitable for any type of development or evacuation access. All building types considered vulnerable to failure.														
hazard_zqra	<p>Hazard categories for the Queensland Reconstruction Authority. Refer to http://qldreconstruction.org.au/u/lib/cms2/resilient-</p>														

Output Parameter	Description																								
	<p>floodplains-part2-full.pdf (Queensland Reconstruction Authority, 2011-2012).</p> <div style="border: 1px solid black; padding: 10px;"> <p>Schedule 4 - Flood hazard criteria</p> <p>Indicative flood hazard criteria</p> <p>The following indicative flood hazard criteria have been prepared for use in preparing flood investigations (level 2), and planning evaluations based on latest available engineering guidance. In the absence of other more appropriate flood hazard definitions, the criteria below may be used.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">Rules</th> </tr> <tr> <th>Depth</th> <th>Low</th> <th>Significant</th> <th>High</th> <th>Extreme</th> </tr> </thead> <tbody> <tr> <td>Depth</td> <td><0.5</td> <td><2</td> <td><2</td> <td>>2</td> </tr> <tr> <td>Velocity</td> <td><1.5</td> <td><2</td> <td><2</td> <td>>2</td> </tr> <tr> <td>Dv Ratio</td> <td><0.6</td> <td>0.6 to <0.8</td> <td>0.8 to <1.2</td> <td>1.2+</td> </tr> </tbody> </table> <p>Rationale</p> <ol style="list-style-type: none"> 1. Low – self evacuation possible for adults and children, vehicle stability within tolerance for large 4WD 2. Significant – working limit for trained safety workers, Vehicle evac unsuitable, Building Code limitation 3. High – limit of uncompromised stability for adults (dangerous to most) 4. Extreme – in excess of known stability limits <p>References</p> <ol style="list-style-type: none"> 1. ARR Revision Project 10: Appropriate Safety Criteria for People <ul style="list-style-type: none"> a. Children – Significant Hazard $D \leq 0.6$ & $D \leq 0.5$ b. Adult – Moderate Hazard $DV \geq 0.6$ c. Working limit for trained safety workers or experienced and well equipped persons $DV \leq 0.8$ 2. ARR Revision Project 10 State 2 Report: Appropriate Safety Criteria for Vehicles (Draft) <ul style="list-style-type: none"> a. Large 4WD $DV \leq 0.6$ & $D \leq 0.5$ 3. Dale et al. (2004) Structural flood vulnerability and the Australianisation of Black's Curves <ul style="list-style-type: none"> a. Fibro/Tile construction $D < 0.5$ & $V < 2$ b. Draft QDC for flood hazard areas for Deemed to Satisfy provisions – $V < 1.5$ 4. BMT WBM (2012) Newcastle City-wide Floodplain Risk management Study and Plan P81-82 <ul style="list-style-type: none"> a. Hydraulically suitable for wading by able-bodied adults $V < 2$ & $D < 0.8$ b. Hydraulically suitable for light construction (e.g. timber frame and brick veneer) $V < 2$ and $D < 0.5$ 5. Jonkman et al. (2008) Methods for the estimation of loss of life due to floods: A literature review and proposal for a new method Natural Hazards P. 364 <ul style="list-style-type: none"> a. Level of hazard to people can be categorized as low, moderate, significant or extreme. </div>	Rules				Depth	Low	Significant	High	Extreme	Depth	<0.5	<2	<2	>2	Velocity	<1.5	<2	<2	>2	Dv Ratio	<0.6	0.6 to <0.8	0.8 to <1.2	1.2+
Rules																									
Depth	Low	Significant	High	Extreme																					
Depth	<0.5	<2	<2	>2																					
Velocity	<1.5	<2	<2	>2																					
Dv Ratio	<0.6	0.6 to <0.8	0.8 to <1.2	1.2+																					
Heat_content	(Degrees Celsius m ³)																								
LW_rad	Downward long-wave radiation flux (W/m ²)																								
MSLP	Mean sea level pressure (hPa)																								
PRECIP	Precipitation rate (m/day)																								
Rel_hum	Relative humidity (%)																								
Rhow	Water density (kg/m ³)																								
Sal	Salinity concentration (PSU)																								
SW_rad	Downward short-wave radiation flux (W/m ²)																								
Temp	Temperature (degrees Celsius)																								
Trace_#	Tracer concentration (units/m ³)																								
TURBZ	Output of vertical turbulence parameters, which includes: <ul style="list-style-type: none"> • TURBZ_TKE (m²/s²) • TURBZ_EPS (m²/s³) • TURBZ_L (m) 																								

Output Parameter	Description
	<ul style="list-style-type: none"> • TURBZ_SPFSQ ($/\text{s}^2$) • TURBZ_BVFSQ ($/\text{s}^2$) • TURBZ_NUM (m^2/s) • TURBZ_NUH (m^2/s) • TURBZ_NUS (m^2/s)
W10	10 m wind speed vector (m/s)
WQ_ALL	Output of water quality parameters
WQ_DIAG_ALL	Output of water quality diagnostics
Wvht	Wave height (m) – typically significant wave height
Wvper	Wave period (s) – typically peak wave period
Wvdir	Wave direction (degrees cartesian)
Wvstr	Wave stress vector (N/m^2)

Table 12-5 Sediment Transport Map Output Types

Output Parameter	Data Type	Description	Supported Formats
Hydraulics			
TAUC	Scalar	Current related effective bed shear stress component (N/m ²)	All
TAUW	Scalar	Wave related effective bed shear stress component (N/m ²)	All
TAUCW	Scalar	Combined effective current/wave bed shear stress (N/m ²)	All
Morphological			
DZB	Scalar	Change in elevation between current output time and model start time (Current ZB – Start ZB m)	All
ZB	Scalar	Bed elevation at current ouput time (m)	All
Suspended sediment			
SED_#	Scalar	Suspended concentration of sediment fraction # (mg/L)	All
TSS	Scalar	Total suspended solids concentration of all fractions (mg/L)	All
DEPOSITION	Array of Scalars (Number of sediment fractions x Number of 2d cells)	Deposition rate of each suspended sediment fraction (g/m ² /s)	NetCDF only.
PICKUP	Array of Scalars (Number of sediment fractions x Number of 2d cells)	Pickup rate of each suspended sediment fraction (g/m ² /s)	NetCDF only.
NETSEDRATE	Array of Scalars (Number of sediment fractions	Net rate of each suspended sediment fraction (Deposition-Pickup (g/m ² /s)	NetCDF only.

Output Parameter	Data Type	Description	Supported Formats
	x Number of 2d cells)		
DEPOSITION_TOTAL	Scalar	Total deposition rate of suspended sediment (g/m ² /s)	All
PICKUP_TOTAL	Scalar	Total pick up rate of suspended sediment (g/m ² /s).	All
NETSEDRATE_TOTAL	Scalar	Sum of the total deposition and pick up rates (g/m ² /s).	All
SUSPLOAD	Array of Vectors (Number of sediment fractions x Number of 2d cells)	Suspended load (g/m/s) components in X and Y directions with output names of "SUSPLOAD_X" and "SUSPLOAD_Y".	NetCDF only.
SUSPLOAD_TOTAL	Vector	Total Suspended load (g/m/s)	All
Bed Material and Load			
BEDLOAD	Array of Vectors (Number of sediment fractions x Number of 2d cells)	Bed load (g/m/s) components in X and Y directions with output names of "BEDLOAD_X" and "BEDLOAD_Y".	NetCDF only.
BEDLOAD_TOTAL	Vector	Total Bed load (g/m/s)	All
BED_MASS	Array of Scalars (Number of sediment fractions x Number of 2d cells)	Bed mass of each sediment fraction over all bed layers (kg/m ²)	NetCDF only.
BED_MASS_TOTAL	Scalar	Total bed mass (kg/m ²)	All
BED_MASS_LAYER_#	Array of Scalars (Number of sediment fractions x Number of 2d cells)	Bed mass of each sediment fraction in bed layer # (kg/m ²)	NetCDF only
BED_MASS_LAYER_#_TOTAL	Scalar	Bed mass of all fractions in layer # (kg/m ²)	NetCDF only.

Output Parameter	Data Type	Description	Supported Formats
BED_MASS_LAYER_#_SED_#	Scalar	Bed mass of sediment fraction # in bed layer # (kg/m ²)	All
Combined suspended and bed load			
SEDLOAD	Array of Vectors (Number of sediment fractions x Number of 2d cells)	Total sediment load (bed plus suspended) (g/m/s) components in X and Y directions with output names of "SEDLOAD _X" and "SEDLOAD _Y". (g/m/s)	NetCDF only.
SEDLOAD_TOTAL	Vector	Total Sediment load (g/m/s)	All

12.8 Structure Outputs

12.8.1 Structure Flux

The ‘structflux’ output can be used to provide the flux that passes through a structure in a similar format to that stored in the standard flux output files. The ‘structflux’ output provides a time-series of the water mass flux (flow rate), the salt flux (salt mass per second), temperature flux (heat per second), and all other conserved tracer fluxes. If a [scalar function](#) is being used, then these scalar fluxes are the total fluxes that make their way through the structure, and are effectively the amount output at the downstream side of the structure.

If users wish to further investigate the difference between upstream and downstream conditions going through the structure (for example when using the aforementioned scalar functions) then a ‘structcheck’ file can be used (refer to Section 12.8.2). An example of the first few lines of a ‘structflux’ file for a simulation with two tracers (but no other conserved scalar variables) is shown below.

TIME	channel_FLOW [m^3 s^-1]	channel_TRACE_1_FLUX [units m^3 s^-1]	channel_TRACE_2_FLUX [units m^3 s^-1]	c
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.020503	9.46E+00	4.73E+01	6.15E+01	
0.036742	1.91E+01	9.55E+01	1.24E+02	
0.052968	1.44E+01	7.14E+01	9.28E+01	
0.069502	7.90E+00	3.88E+01	5.05E+01	
0.085937	9.34E+00	4.55E+01	5.91E+01	
0.10224	1.63E+01	7.75E+01	1.01E+02	

The output file has the suffix _STRUCTFLUX.CSV and is located in the folder set by the [output dir](#) command. An .fvc snippet for this output is provided as follows:

```
!Structure Flux
output == structflux
  output interval == 900.
end output
```

12.8.2 Structure Check

The structure check file outputs a time-series of the flow and scalar concentrations on the upstream and downstream sides of the structure, as seen by the model when determining any processing internal to the structure. For most basic structures, the upstream and downstream flows will be equal, as will the scalar concentrations. Where these might differ are for structures with some losses, sources, or scaling of some of these parameters. An example is the ‘treatment’ type structure, that can be used to reduce the salinity through the structure by a constant scale factor (to simulate a desalination plant for example), this would show the concentration flowing into the structure (and flow rate), and the treated concentration coming out of the structure, which users can use to check that the structure has the desired behaviour.

An example of the first few lines of a ‘structcheck’ file for a simulation with two tracers (but no other conserved scalar variables) is shown below:

TIME	channel_FLOW A [m^3 s^-1]	channel_FLOW B [m^3 s^-1]	channel_TRACE_1 A [units m^-3]	channel_TRACE_1 B [units m^-3]
0	0.00E+00	0.00E+00	0.00E+00	0.00E+00
0.020503	9.46E+00	9.46E+00	1.00E+01	5.00E+00
0.036742	1.91E+01	1.91E+01	9.98E+00	4.99E+00
0.052968	1.44E+01	1.44E+01	9.92E+00	4.96E+00
0.069502	7.90E+00	7.90E+00	9.83E+00	4.92E+00
0.085937	9.34E+00	9.34E+00	9.73E+00	4.86E+00
0.10224	1.63E+01	1.63E+01	9.54E+00	4.77E+00

The output file has the suffix _STRUCTCHECK.CSV and is located in the folder set by the the [output dir](#) command. An .fvc snippet for this output type is provided as follows:

```
!Structure Check
output == structcheck
|   output interval == 3600.
end output
```

12.9 Check Files and GIS Layers

Check files are produced so that modellers and reviewers can readily check that the constructed model is as intended. These are provided as the model initialises so that the modellers does not have to wait for the completion of the simulation to check the model geometry and boundary conditions. Advanced models draw upon a wide variety of data sources. The check files represent the final data set after all data inputs, allowing the model construction to be viewed in its final form. The check files are typically in text format and include:

- The simulation .log file which is automatically generated at the beginning of (and continuously written to during) a TUFLOW FV simulation.
- GIS check files (refer Section 12.10).
- The simulation timestep files which contain the minimum and mean timestep required for calculation of the external (free-surface) and internal (advective) terms within each model cell are automatically written at the end of a TULFLOW FV simulation (refer echo geometry). This information can be used to identify model cell(s) constraining the simulation timestep. A model ‘timestep review’ example is provided on the TUFLOW FV Wiki:

http://fvwiki.tuflow.com/index.php?title=A_Model_Runs_Slow

12.9.1 GIS Workspaces (.wor and .qgs files)

A MapInfo workspace (.wor) and QGIS workspace (.qgs) is automatically created for every simulation if GIS integration is enabled (refer Section 5.3). They are named <fvc_filename>.wor / .qgs and are written to the same folder as the .log file. The workspace contains all GIS layers used as input to the simulation and is an excellent way of ascertaining which GIS layers were used to set up a model, particularly large models with many GIS inputs, or those with multiple events or scenarios.

The .wor file when opened in MapInfo simply opens the .tab layers. No Map or Browser windows are automatically opened. If the simulated model contains any .shp files, these are not opened, however the .shp file layers used by TUFLOW can be viewed when opening the .wor file in a text editor.

Opening the .qgs file in QGIS will open all GIS input and output check layers (.mif and/or .shp). Note that the visibilities of the output layers are unchecked so that the display time is quick.

For ArcMap users, the .mxd files are not directly written by TUFLOW. The format is proprietary and can't be directly written. However, the [ArcTUFLOW toolbox](#) can be used to load the simulation input files into ArcMap.

12.10 GIS Check Files

There are several new GIS check files and GIS workspaces that can assist with the review of model inputs. These include:

Table 12-6 GIS Check Files

Filename prefix / suffix	Description
<u>mesh_check_R.shp</u>	<p>GIS layer of the final 2D or 3D mesh. Represents the final mesh including modifications from the .fvc file, such as ZB updates.</p> <p>Please note reported ZB values do not include ZB modifications due to morphological changes when morphology is enabled. Model material, ZB, bed roughness and cell 2D and 3D ID are reported.</p>
<u>_ns_check_L.shp</u>	<p>Shows the cell faces selected, the direction and ID of nodestrings. This includes both nodestrings assigned directly to the .2dm or by external nodestrings.</p>
<u>_bc_check_L.shp</u>	<p>Provides the cell faces selected to input nodestring boundary information. Returns the nodestring ID and the input CSV data file assigned to that boundary.</p>
<u>_zln_zpt_check_P.shp</u>	<p>GIS layer containing Zpts (ZB values at cell centroids) that have been modified by Read GIS Z Line == commands, the type of Z Line and the Z Line filename. This feature is very useful for checking which Zpts that the Z Lines have modified.</p>
<u>_sa_check_P.shp</u>	<p>If using QC or other cell based input boundaries returns the cell centroid of each cell selected by the boundary.</p>
<u>_input_layers.mif</u>	<p>GIS layer containing full file paths to all input layers used to compile the model.</p>

Appendix A .fvc File Commands

SIMULATION COMMANDS

Bottom drag model	Include bed friction	Spherical
Cell 3D depth	Include Coriolis	Stability limits
Cell dry/wet depths	Include wind	Start time
Cell elevation file	Initial condition 2d	Structure logging
Cell elevation polygon file	Initial condition 3d	Surface sigma layers
Cell elevation polyline	Initial condition OGCM	Time format
CFL	Initial condition quiescent	Timestep Limits
CFL external	Initial water level	Transport Mode Depth
CFL internal	Kinematic viscosity	Turbulence update dt
Constant Timestep	Latitude	Tutorial Model
Display dt	Layer faces file	Units
Echo geomtry	Logdir	Use restart file time
Echo geometry CSV	MI Projection	Vertical Alpha R
Echo geometry NetCDF	Min bottom layer thickness	Vertical gradient limiter
Echo spatial	Mode split	Vertical mesh type
End time	Momentum mixing model	Vertical mixing model
External turbulence model	Nodestring polyline file	Vertical mixing parameters
dir	Output dir	Wind stress model
g	Read File	Wind stress params
Geometry 2d	Read GIS Mat	Write Check Files
GIS Format	Read GIS Nodestring	Write Empty GIS Layers
Global bed elevation limits	Read GIS Z Line	Write restart dt
Global horizontal eddy viscosity	Read GRID Zpts	
Global horizontal eddy viscosity limits	Reference MSLP	
Global vertical eddy viscosity limits	Reference time	
Horizontal AlphaR	Restart file	
Horizontal gradient limiter	Restart overwrite	
Include	Set Mat	
	Set Zpts	
	SHP Projection	
	Spatial order	

Bottom drag model == <'Manning' ; 'ks'>

(Optional; Default == Manning)

This command can be used to specify the bottom drag model to be used in the simulation.

The default model is Manning, in which case a Manning's 'n' coefficient(s) should be specified using the [global bottom roughness](#) or material [bottom roughness](#) command.

An alternative model assumes a log-law velocity profile and requires specification of a surface roughness length-scale, in which case 'ks' values should be specified using the [global bottom roughness](#) or material [bottom roughness](#) command.

Cell 3D depth == <threshold depth (m)>

(Optional 3D)

An optional command to set the threshold water depth for 3D calculations. In areas where the depth is less than the threshold value vertical fluxes between layers are reduced to a 2D solution. Note this doesn't speed up model run times but can reduce potential instability in cells with shallow depths.

Cell wet/dry depth == <cell dry depth (m) , cell wet depth(m)>

(Optional; Default == 1.0e-6, 1.0e-2)

Sets the cell wetting and drying depths in metres:

- The drying value corresponds to a minimum depth below which the cell is dropped from computations (subject to the status of surrounding cells). If set to zero or a negative value the drying is set to zero.
- The wet value corresponds to a minimum depth below which cell momentum is set to zero in order to avoid unphysical velocities at very low depths. If less or equal to the drying depth the value is set to the drying depth.

Cell elevation file == <cell elevation file (.CSV), xytype, ztype>

(Optional, Legacy command)

This command is provided for legacy models. The recommended alternative is [Read GIS Z Line](#).

This optional command can be used to set the bed elevations for some or all cells in the model domain, overriding the elevations defined by the .2dm file.

If xytype = cell_ID (Default):

- The .CSV file must contain a header line:
 - ID, Z
 - with cell ids and corresponding elevations entered within the rows below the header line

If xytype = coordinate:

- The .CSV file contains a first line header:

-
- X,Y,Z
 - with the x-coordinate, y-coordinate and corresponding elevations entered within the rows below the header line

If ztype = overwrite (Default):

- the cell elevation will correspond to the last z value read (i.e. overwrite the elevations defined by the .2dm file).

If ztype = average:

- the elevation will be the average of the z values read for a given cell (i.e. the average elevation defined by the .2dm file and read from the cell elevation file).

Note that more than one cell elevation file can be listed in the simulation control file (.fvc). Please refer to Section 8.3.4 for more details. See also [Read GIS Z Line](#).

Cell elevation polygon file == <polygon file (.CSV), ZB, ID>

(Optional)

This optional command reads a polygon defined in a comma separated variable file. All cell centres that lie within the polygon are assigned an elevation ZB, overriding the corresponding elevations defined by the .2dm and/or cell elevation file.

Polygon file:

- The .CSV file contains a first line header:
 - X,Y,ID (ID is optional)
 - Input data is entered within the rows below the column header.
 - Points within the CSV file define the perimeter of the polygon. The definition of points needs to be consecutively listed and can be either clockwise or counter-clockwise.

ZB:

- The elevation that all cells within the polygon will be assigned.

ID:

- Only those vertices listed in the .CSV file with an ID value matching the specified value in the command line will be read by the model.
 - note: If ID = 0 or is blank, all vertices will be read from the polygon file

Please refer to Section 8.3.4 for more details.

Cell elevation polyline file == <polyline file (.CSV), ID>

(Optional)

This optional command will set bed elevations for cells that are intersected by a polyline. Cell Z values will be interpolated from the z values specified at vertices along the polyline, overriding the corresponding elevations defined by the .2dm and/or cell elevation file.

Polyline file:

- The .CSV file contains a first line header:
 - X,Y,Z,ID

-
- Input data is entered within the rows below the column header

ID:

- Only those vertices listed in the .CSV file with an ID value matching the specified value in the command line will be read by the model.
 - note: If ID = 0 or is blank, all vertices will be read from the polyline file

Please refer to Section 8.3.4 for more details.

CFL == <global maximum Courant-Friedrichs-Lowy number>

(Mandatory; Default == 1)

Sets the Courant–Friedrichs–Lewy (CFL) condition used for the calculation of both the internal (advective) and external (free-surface) terms.

The default value is 1, which is the theoretical stability limit. In practise this value is commonly lowered to provide additional stability for models that exhibit large gradients in flow, density or constituent concentrations such as the assessment of Tsunami, dam break or lakes with strong vertical stratification. Also see [CFL external](#) and [CFL internal](#).

CFL external == <external maximum Courant-Friedrichs-Lowy number>

(Optional; Default == 1)

Optional command to specify a Courant–Friedrichs–Lewy (CFL) condition for the calculation of external (free-surface) terms only. Will overwrite the command used by [CFL](#).

CFL internal == <internal maximum Courant-Friedrichs-Lowy number>

(Optional; Default == 1)

Optional command to specify a Courant–Friedrichs–Lewy (CFL) condition for the calculation of internal (advective) terms only. Will overwrite the command used by [CFL](#).

Constant Timestep == <constant timestep (s)>

(Optional, not typically recommended)

Specifies the value of a constant timestep that is to be used during the simulation.

If the command is not entered then a variable timestep is applied according to the CFL stability criterion, see [CFL](#) and [Timestep Limits](#)). Due to variable time-stepping using the TUFLOW FV explicit scheme the use of a constant timestep is not typically recommended.

Using this option will ignore [Timestep Limits](#), [CFL](#), [CFL Internal](#) and [CFL External](#) commands.

Display dt == <display timestep (s)>

(Optional; Default == 300)

Allows the user to specify the simulation time interval (in seconds) for displaying timestep information to the log and terminal window.

Echo geometry == <0;1>

(Optional, Default == 0)

Setting this to 0 stops the model from writing a series of comma separated variable files and NetCDF output that include various relevant geometry and spatial features of the simulation, including:

- Mesh details
- Cell elevations and material types
- Cell elevations that have been updated externally to the .2dm file (for example by using the [cell elevation file](#) command)
- Nodestring locations and their purposes (boundary, structure, etc.)
- Output locations

See also [echo geometry CSV](#) and [echo geometry NetCDF](#).

Echo geometry CSV == <0;1>

(Optional, Default == 0)

Setting this to 0 stops the model from writing a series of comma separated variable files that include various relevant geometry and spatial features of the simulation, including:

- Mesh details
- Cell elevations and material types
- Cell elevations that have been updated externally to the .2dm file (for example by using the [cell elevation file](#) command)
- Nodestring locations and their purposes (boundary, structure, etc.)
- Output locations

See also [echo geometry](#) and [echo geometry NetCDF](#).

Echo geometry NetCDF == <0;1>

(Optional, Default == 0)

Setting this to 0 stops the model from writing a *_geo.nc (NetCDF format geometry) check file.

Contents of the *_geo.nc file include:

- Cell geometry including connectivity, coordinates, layers, areas, elevations, materials, bottom roughness, vertices and faces.
- Face geometry including connectivity, coordinates, elevations, lengths, vertices and CFL values.
- Node geometry including connectivity, coordinates, elevations and weightings of adjacent cells.

As of Release 2018 this information is output when using NetCDF output file formats by default.

See also [echo geometry](#) and [echo geometry CSV](#).

Echo spatial == <0;1>

(Optional, Default == 0)

Legacy command (see [echo geometry](#)).

End Time == <simulation end time>

(Mandatory; No Default)

Specifies the end time for the simulation:

-
- For [Time Format](#) == Hours, units are in decimal hours.
 - For [Time Format](#) == ISODate, inputs are in date form *dd/mm/yyyy HH:MM:SS* (or some truncation thereof).

External Turbulence Model Directory == <directory path>

(Optional)

Optional command to specify the directory for external turbulence model definition files if an external [vertical mixing model](#) is used. If not specified, external turbulence model files must be in the same directory at the simulation control file.

g == <gravitational acceleration (m/s²) or (ft/s²)>

(Optional)

Gravitational acceleration. If not specified, then the default value depends on the specified [units](#):

- 9.81 m/s²
- 32.174 ft/s²

Geometry 2d == <mesh file location and name (.2dm)>

(Mandatory)

Specifies the model 2D geometry input file. The input file must be in a format consistent with the SMS generic mesh module format (see Section 8.1.2).

When following the suggested TUFLOW FV folder structure (refer Section 4.2.1):

geometry 2d == ..\model\geo\mesh_name.2dm

GIS Format == <MIF;SHP >

(Optional, Default == MIF)

Specifies the output format for GIS check layers and GIS outputs. If the command [GIS Format](#) is not specified, the GIS format used for check layers and other GIS outputs is based on whether [MI Projection](#) or [SHP Projection](#) has been specified. If neither or both commands have been specified, and [GIS Format](#) has not been specified, the default of using .mif files is adopted.

Note that the format of an input layer is solely controlled by the file extension (i.e. .mif for the MIF format and .shp for the SHP format).

Global Bed Elevation Limits == <zbmin, zбmax>

(Optional)

Provides option to apply limits to the bed elevations. Can also be applied to specific material types (see [Bed Elevation Limits](#)).

Global Horizontal Eddy Viscosity == <eddy viscosity (m²/s) ; coefficient/s (-)>

(Mandatory)

Globally sets a constant horizontal eddy viscosity (m²/s) or the horizontal eddy viscosity coefficient. This is dependent on the momentum mixing model set using the [momentum mixing model](#) command:

- Constant: specify a constant eddy viscosity; Default == 0
- Smagorinsky: specify the Smagorinsky coefficient; Default == 0.2

For further information please refer to Section 7.2.

Global Horizontal Eddy Viscosity Limits == <min eddy viscosity (m²/s)>, max eddy viscosity (m²/s)>

(Optional)

For use with Smagorinsky [momentum mixing model](#), globally sets the minimum and maximum horizontal eddy viscosity (m²/s) limits.

Not applicable if a Constant horizontal eddy viscosity is set using the [global horizontal eddy viscosity](#) command. For further information please refer to Section 7.2.

Horizontal AlphaR == <alphaH (depth), alphav (velocity), alphaS (scalars)>

(Optional; Default == 1.0, 1.0, 1.0)

This command can be used to apply a reduction factor to high-order cell reconstruction gradients, which may be useful in stabilising a 2nd order simulations. This has no effect on when running in first order (refer [Spatial Order](#)).

Default is <1.0, 1.0, 1.0>, which corresponds to no gradient reduction, whereas <0.0, 0.0, 0.0> would revert to a first-order scheme.

This may be implemented to solve for water level and velocity in 1st order whilst solving plume/constituents in 2nd order as they may exhibit high spatial gradients (or vice-versa). This is useful if 2nd order hydrodynamics are not required. For example, Horizontal AlphaR == 0.0,0.0, 1.0

Horizontal Gradient Limiter == <LCD; MLG>

(Optional; Default == LCD)

Sets the Total Variation Diminishing (TVD) limiting scheme for 2nd order horizontal spatial integration scheme, the options are:

- LCD is the less compressive option and the least computationally intensive.
- MLG is the most compressive option and the most computationally intensive.

For further information please contact support@tuflow.com.

Include == <file path>

(Optional)

At any location in the simulation control file (.fvc) an ‘include file’ can be used. Commands contained in the include file will be read as if they are listed in the .fvc file. Provides the same functionality as the [Read File](#) command.

Include Bed Friction == <0;1>

(Optional; Default == 1)

Optional command used to switch off bed friction and thereby simulate an ‘ideal’ fluid:

- 0 = False (i.e. bed friction is not included).
- 1 = True (i.e. bed friction is included).

Include Coriolis == <0;1>

(Optional; Default == 1)

Optional command used to switch off the Coriolis force source term from the momentum conservation equations:

- 0 = False (i.e. Coriolis forces source term is not included).
- 1 = True (i.e. Coriolis forces source term is included).

Include Wind == <0;1>

(Optional; Default == 1)

An optional command used to remove the wind stress terms from the momentum and mass transport equations (only relevant if wind is a specified input using the [BC](#) command):

- 0 = False (i.e. wind stress terms are not included).
- 1 = True (i.e. wind stress terms are included).

Initial Condition 2D == <initial condition file (.CSV)>

(Optional)

Optional command to read the initial conditions from a comma separated variable file. The .CSV file contains initial conditions for each cell of the mesh.

As a minimum, the following column headers are required in this file:

ID, WL, U, V

If salinity, temperature or other scalars are included in the simulation they should also be specified in the .CSV file (e.g. Sal, Temp, Scal_1,...). An example of the command usage and corresponding .CSV file is given below:

Initial condition 2d == ..\bc\initial_conditions_001.CSV

and the contents of initial_conditions.CSV:

ID, WL, U, V, Scal_1, Scal_2, Scal_3

1, 0.300, 0.000, 0.000, 1.000, 0.000, 0.000

Please refer to Section 10.7.2 for further details.

Initial Condition 3D == <initial condition file (.CSV)>

(Optional)

Optional command to read the initial conditions from a comma separated variable file. The .CSV file contains initial conditions for each cell of the mesh.

As a minimum, the following column headers are required in this file:

ID, WL, U, V

If salinity, temperature or other scalars are included in the simulation they should also be specified in the .CSV file (e.g. Sal, Temp, Scal_1,...). An example of the command usage and corresponding .CSV file is given below:

Initial condition 2d == ..\bc\initial_conditions_001.CSV

and the contents of initial_conditions.CSV:

ID, WL, U, V, Scal_1, Scal_2, Scal_3

1, 0.300, 0.000, 0.000, 1.000, 0.000, 0.000

Please refer to Section 10.7.2 for further details.

Initial Condition OGCM

(Optional)

Ocean General Circulation Model. Derives initial water level, currents, salinity and temperature from an input NETCDF grid. This is commonly used in combination with the OBC boundary curtain type to apply global ocean model conditions to a local mesh, for example [HYCOM](#).

Initial Condition Quiescent

(Optional)

Sets a quiescent initial condition i.e. zero initial velocity.

Initial Water Level == <water level (m)>

(Optional)

Command to globally set an initial, quiescent water level.

Kinematic Viscosity == <kinematic viscosity value (m²/s)>

(Optional; Default == 1.05e-6)

Optionally specifies the background water kinematic viscosity.

Latitude == <latitude in degrees (-ve for Southern Hemisphere)>

(Optional; Default == 0.0)

Sets the latitude for Coriolis calculations when a Cartesian coordinate system is used. If Coriolis forcing is important for your study, care should be taken to set a representative latitude, or preferably use Spherical == 1. Note that the equator is used as the default latitude.

Layer Face File == <file specifying layer interface levels (.CSV)>

(Mandatory if using Sigma or for Z-coordinate vertical mesh type)

Specifies the location and name of the comma separated variable file containing the 3D layer face information, depending on the Vertical Mesh Type:

- In the case of Sigma-coordinates, the layer faces are optionally specified as decimal fractions between 1 (water surface) and 0 (bed level) in a ‘SIGMA’ column. The file must be monotonically decreasing from the surface (1) to the bed (0). Also, values are required to be between (and not include) the values 0 or 1 otherwise an error code will be produced. This command is used if a non-uniform vertical distribution of sigma layers is desired.

```
1 SIGMA
2 0.8
3 0.6
4 0.4
5 0.2
```

- In the case of Z-coordinates, fixed layer face elevations are specified in a ‘Z’ column.

```
1 z
2 -3
3 -4
4 -5
5 -6
6 -7
7 -8
8 -10
```

In tidal environments, the user may wish to adopt a hybrid z-sigma-coordinate mesh. In this instance, the Z-coordinate Vertical Mesh Type is set and “always wet” fixed layer face elevations are specified in a ‘Z’ column. The number of sigma layers between the maximum “always wet” elevation and the free-surface is then specified using the surface sigma layers command.

Please refer to Section 9.3 for further details on 3D layering approaches.

Logdir == <filepath>

(Optional)

This command specifies the directory for TUFLOW FV simulation log file (.log) output. A log file is automatically generated for each simulation, the contents of which are the same as that displayed in the simulation window. The log filename has the same prefix as the simulation control file.

If not specified, the log file will be written to either the same location at the simulation control file or to the input\log sub-directory if this has been first created by the user (see suggested sub-folder structure in Section 4.2).

Other check files (*geo.nc, *cfl_dt.CSV) and [restart](#) files (if specified) are also written to the specified log directory location.

**MI_Projection == [<.mif_file> |
<Projection_line_from_MIF_file>]**

(Optional but recommended)

Sets the geographic projection for all GIS input and output in MID/MIF format. If this command is omitted, TUFLOW FV searches for a file “Header.mif” in each folder it opens GIS files, and extracts the projection from this file. The “Header.mif” file is any GIS layer in the correct projection exported in MID/MIF format. If no “Header.mif” file is found, non-earth coordinates are assumed.

Alternatively, a projection line extracted from a mif file may be entered (although the previously described approach of specifying a MIF file is the recommended approach). Follow these steps:

1. In a GIS, create or open a layer in the Cartesian projection to be used for the model. For non-geographic models (e.g. a test model), use the Non-Earth (metres) projection.
2. Export the layer in MIF/MID format.
3. Open the .mif file in a text editor, copy the whole line starting with “CoordSys” (usually the 4th or 5th line) and paste after “MI_Projection ==” in the .fvc file.

Examples:

```
MI_Projection == ..\model\mi\Model_Projection.mif
MI_Projection == CoordSys Earth Projection 8, 13, "m", 153, 0, 0.9996, 500000,
10000000 Bounds (-7745874.38492, 1999.40969607) (8745874.38492, 19998000.5903)
```

Note: If using GIS Integration All MID/MIF GIS layers read by TUFLOW FV MUST USE this projection. The projection must be a Cartesian based projection and in metres if using [Spherical == 0](#). If using [Spherical == 1](#) the coordinate system needs to be spherical such as Latitude/Longitude.

Further examples of creating a projection file using various software packages can be found on the [TUFLOW Wiki](#).

See also the corresponding command [SHP Projection](#), for GIS input and output files in .shp format. If a model has a mixture of .mif and .shp files as input, then both [MI_Projection](#) and [SHP Projection](#) should

be specified. Please also refer to the command [Spherical](#) for more information on using Cartesian or [Spherical](#) spatial reference systems.

Min Bottom Layer Thickness == <dzmin>

(Optional 3D)

Optionally specify the minimum thickness of the lowest layer (i.e. layer at the bed). This command is used to avoid a thin vertical layer (and associated small timestep) at the bed.

Note: Syntax Minimum Bottom Layer Thickness == <dzmin> is also supported.

Please refer to Section 9.3 for further details on 3D layering approaches.

Mode Split == <0;1>

(Optional for 2D; Default == 1)

TUFLOW FV uses a mode-splitting approach to efficiently solve the external (free-surface) mode in 2D at a timestep constrained by the surface wave speed while the internal 3D mode is updated less frequently. This command is used to disable mode-splitting:

- 0 = False (i.e. mode-splitting is disabled).
- 1 = True (i.e. mode-splitting is enabled).

For further information please refer to Section 6.8.2.

Momentum Mixing Model == <None; Constant; Smagorinsky>

(Optional; Default == None)

Sets the horizontal eddy viscosity calculation method. See also [global horizontal eddy viscosity](#).

- None: horizontal momentum mixing is not represented.
- Constant: specify a constant eddy viscosity using the [global horizontal eddy viscosity](#) command.
- Smagorinsky: the horizontal eddy viscosity is calculated according to the Smagorinsky model
 - specify the Smagorinsky coefficient using the [global horizontal eddy viscosity](#) command.

For further information please refer to Section 7.2.

Nodestring Polyline File == <nodestring file (.CSV), ID , Boundary>

(Optional, superseded by [Read GIS Nodestring](#))

Specifies a comma separated variable file that contains the vertex coordinates of a polyline defining a nodestring path.

The nodestring path is identified by (1) finding the nearest node to each vertex and (2) identifying internal nodes between them.

Two vertices per nodestring are required as minimum inputs. If additional vertices are defined between the start and end vertices, the definition needs to be consecutively listed

Polygon file:

- The .CSV file contains a first line header:
 - X,Y,Z, ID (Z, ID are optional)
 - Input data is entered within the rows below the column header
 - Z inputs will assign elevations to the vertices along the nodestring. This will not influence cell elevations (use the [cell elevation polyline](#) for this task), but can be used to assign elevations for [nodestring structures](#).

ID:

- If an ID column exists, only those vertices listed in the .CSV file with an ID value matching the specified value in the command line will be read by the model.
 - note that if ID = 0 or is blank, all vertices will be read from the polyline file.

Boundary (optional input):

- If “boundary” is specified, the nodestring path is restricted to being along the boundary

Nodestring numbering is as follows:

- If ID = 0 or is blank, the nodestring ID is the next incremental number after the nodestring IDs in the 2dm file.
- If ID /= 0, the nodestring ID is assigned the ID value. Existing nodestrings will be overwritten if the ID is the same as an existing nodestring.

Output Dir == <filepath>

(Optional)

Command to specify the location where simulation output files are to be written. The first example below specifies the output directory assuming the TUFLOW FV sub-folder structure recommended in Section 4.2.

```
output dir == ..\output
```

Alternatively, the user may wish the output directory to be located on a local drive, for example:

```
output dir == D:\project12345\tuflowfv\output
```

Output is written to the same location at the simulation control file (.fvc) if this command is not used.

From Windows build 2019.01 TUFLOW FV will attempt to automatically create the output directory if it doesn't exist.

Read File == <file path>

(Optional)

At any location in the simulation control file (.fvc) a ‘read file’ can be used. Commands contained in the read file will be read as if they are listed in the .fvc file. Provides the same functionality as the [Include](#) command.

Read GIS Mat == <gis_layer>

(Optional, No Default)

Reads .mif/.mid or .shp formatted files containing regions assigned with a material ID. The material ID must coincide with a value assigned within a [material](#) block.

Please refer to Section 8.4 for further information.

Read GIS Nodestring == <gis_layer>

(Optional)

This command provides an alternative to the previous .CSV file method [Nodestring Polyline File](#). Reads .mif/.mid or .shp formatted files containing polylines assigned with a nodestring ID and boundary option flag. Please refer to Section 8.2 for more details.

Read GIS Z Line == <gis_layer>

(Optional)

Reads .mif/.mid or .shp formatted files containing polylines that are treated as breaklines in the model's bathymetry. The breakline can vary in height along its length (i.e. a 3D breakline).

This is a powerful feature for quickly and easily entering a breakline feature such as a road, railway, levee, creek, drain, etc. It is particularly useful where TUFLOW FV cell discretisation does not guarantee that the crest along, for example, a road, is picked up from the DTM, or the lowest point along a drain. It saves having to incorporate roads, levees, etc. into the DTM.

The modified Zpts are output to the 2d_zln_zpt_check layer (see Section 12.10) if [Write Check Files](#) has been set.

A variable height polyline is created in the GIS by snapping the polyline to points in the same layer. The first attribute column must be a number (real or integer) representing the elevation of the points. Other attributes are ignored. If the polyline is not snapped with a point at its beginning and end, the polyline is assumed to be horizontal (the height is taken from the polyline's attribute). Otherwise, the polyline's grade is determined by the height of the points snapped to the polyline nodes. It is not necessary to snap a point at every polyline node – the minimum requirement is a point snapped to each polyline end.

See Section 8.3.3 for further information.

Read GRID Zpts == <grid_file>

(Optional)

Directly interrogates an ESRI ASCII (.asc) or binary (.flt) grid to set the cell centroid elevations.

The use of this command has significant advantages over the previous method of manually carrying out a manual cell inspection with the [Cell Elevation File](#) command.

Like other topographic update commands, [Read Grid Zpts](#) may be specified more than once. See Section 8.3.2 for further information.

Reference MSLP == <Mean Sea Level Pressure (hPa)>

(Optional; Default == 1013.25)

Optionally sets the reference mean sea level pressure value. Please refer to Section 7.3 for further information on reference values.

Reference Time == <Input/Output reference time>

(Optional; For [Time Format](#) == Hours, Default == 0;
For [Time Format](#) == ISODate, Default == 01/01/1990 00:00:00)

Optional command to set the simulation reference time.

Restart file == <restart file name (.rst)>

(Optional)

Optional command to load the simulation initial conditions from a restart file (.rst) generated by a previous TUFLOW FV simulation.

Unless the use restart [file time](#) command is used the simulation start time will be set to the timestamp in the restart file. See also [write restart](#) command.

Restart overwrite == <0;1>

(Optional, Default == 1)

Option to overwrite the restart file at the time interval specified using the [write restart dt](#) command (default) or create a series of restart files for each timestep:

- 0 = False (i.e. the restart file will not be overwritten, and a series of restart files will be generated).
- 1 = True (i.e. the single restart file will be overwritten).

Set Mat == <material ID>

(Optional, No Default)

Sets the material ID at all cells in the model domain. The Material ID value must correspond to a value within a [material block](#).

Set Zpts == <elevation>

(Optional, No Default)

Sets the value of all cell centroid ZBs in the model domain to this value.

SHP Projection == < >

(Optional)

This command is similar to the [MI Projection](#) command that sets the .shp file projection for checking whether input layers are in the same projection, and for setting the projection of all output layers (e.g. check layers).

Example:

```
SHP Projection == ..\model\shp\Projection.prj
```

If a model has a mixture of .mif and .shp files as input, then both [MI Projection](#) and [SHP Projection](#) should be specified.

Spatial Order == <1;2 (horizontal), 1;2 (vertical)>

(Default == 1,1)

Specifies the spatial order of accuracy of the solution schemes used in the simulation:

- 1 = first order scheme
- 2 = second order scheme

The first-order schemes assume a piecewise constant value of the modelled variables in each cell, whereas the second-order schemes perform a linear reconstruction.

Higher order spatial schemes will produce more accurate results in the vicinity of sharp gradients; however, they will be more prone to developing instabilities and are more computationally expensive.

Generally, initial model development should be undertaken using low-order schemes, with higher-order spatial schemes tested during the latter stages of development. If a significant difference is observed between low-order and high-order results then the high-order solution is probably necessary, or alternatively further mesh refinement is required.

Second order spatial accuracy will typically be required in the vertical direction when trying to resolve sharp stratification.

See also the [horizontal gradient limiter](#) and [vertical gradient limiter](#) commands, which may be used to specify the Total Variation Diminishing (TVD) limiting schemes employed during the higher-order reconstructions.

When running in second order the [Horizontal AlphaR](#) and [Vertical AlphaR](#) horizontal and vertical gradient reduction factor commands may be of use for regions of high spatial gradients or to assist with improving model stability.

Spherical == <0;1>

(Optional; Default == 0)

Flag to specify that the model is in spherical coordinates:

- 0 = Cartesian where geometry inputs and computational coordinates are in metres / feet.
- 1 = Spherical where geometry inputs and computational coordinates are in degrees.

Note: If Coriolis forcing is likely to be important than Spherical coordinates are recommended. However, if Cartesian coordinates are required then please take care that a representative [Latitude](#) is set as the default is 0 (the equator).

Stability Limits == <maximum WL, maximum velocity>

(Optional);

Optional command to specify a maximum water level and maximum velocity which indicate an unstable model. The simulation will stop if these limits are exceeded. For example, <max cell ZB value plus 10m, 10m/s>.

Start Time == <simulation start time>

(Mandatory; No Default)

Specifies the start time for the simulation:

- For [Time Format](#) == Hours, units are in decimal hours.
- For [Time Format](#) == ISODate, inputs are in date form *dd/mm/yyyy HH:MM:SS*.

Structure logging == <0;1>

(Optional, Default == 0)

Setting this to 1 will write a structural log file (.slf) that contains the operational behaviour of included structures through time.

Surface Sigma layers == <Nsigma>

(Optional 3D)

Depending on the [Vertical Mesh Type](#), this command is optionally used to:

-
- In the case of Sigma-coordinates, specify the number sigma layers to be uniformly distributed over the entire water column. Alternatively, a non-uniform vertical distribution of sigma layers is specified using the [layer face file](#) command.
 - In the case of Z-coordinates, specify the number sigma layers to be uniformly distributed between the maximum “always wet” fixed layer elevation and the free-surface. This creates a hybrid z-sigma-coordinate vertical mesh.

Time Format == <Hours;ISODate>

(Mandatory; Default == Hours)

Specifies the simulation time format:

- ‘Hours’ time in decimal hours (e.g. Start Time == 3.0)
- ‘ISODate’ requires a date specification in the form dd/mm/yyyy HH:MM:SS.

Subsequent simulation time commands and simulation inputs must be in the specified time format. Simulation outputs will be in the specified time format.

Timestep Limits == <min timestep (s) , max timestep (s)>

(Mandatory; No Default)

Specifies the maximum and minimum variable timestep allowed according to the CFL stability criterion. See also [CFL](#).

Transport Mode Depth == < depth (m)>?

(Optional)

Transport inputs can be disabled for shallow cells to limit instabilities by using the transport mode depth. This can be useful for areas where wetting and drying are leading to issues with application of the transport boundary. Refer Section 10.8

Turbulence update dt == <time (s)>

(Optional)

Optional command to specify the timestep for updating the vertical turbulence mixing eddy-viscosity and scalar-diffusivity terms. If not specified, this will occur at every timestep.

Tutorial Model == [ON, OFF]

(Optional; Default == OFF)

When set to ON, allows simulation of the Tutorial Models without the need for a TUFLOW license. For further information refer to Section 2.4.1.

Units == <metric; Imperial; US Customary>

(Optional; Default == metric)

Optional command to apply Imperial or US Customary units (if not specified the default is metric). All simulation inputs, model parameters and outputs will follow the specified units

Note that currently the units are valid only for 2D hydrodynamics; please contact support@tuflow.com if considering using customary units for additional modules.

Use Restart File Time == <1;0>

(Optional, Default == 1)

Setting this to 0 resets the model start time to be equal to the value specified using [start time](#) even when a restart file is used (refer [Restart file](#)). Without this command or when set to 1 (true), the start time is set equal to the restart file timestamp:

- 0 = False (i.e. use start time set with command [start time](#))
- 1 = True (i.e. use time equal to the [Restart file](#) timestamp)

Vertical AlphaR == <alphaV (velocity) , alphas (scalars)>

(Optional; Default == 1.0, 1.0)

This command can be used to apply a reduction factor to high-order cell reconstruction gradients, which may be useful in stabilising a higher-order simulation.

Default is <1.0, 1.0>, which corresponds to no gradient reduction, whereas <0.0, 0.0> would revert to a first-order scheme.

Global Vertical Eddy Viscosity Limits == <min eddy viscosity (m²/s) , <max eddy viscosity (m²/s)>

(Optional)

For use with Parametric or External [vertical mixing model](#), globally sets the minimum and maximum vertical eddy viscosity (m²/s) limits.

Not applicable if a Constant vertical eddy viscosity is set using the [vertical mixing parameters](#) command.

Vertical gradient limiter == <MINMOD;MC;Superbee>

(Optional 3D; Default == MC)

Sets the Total Variation Diminishing (TVD) limiting scheme for 2nd order vertical spatial integration scheme.

The options are MINMOD, MC (Monetized Central) and Superbee (ranging from least compressive to most compressive). For further information please contact support@tuflow.com.

Vertical mesh type == <sigma;z>

(Mandatory 3D; Default == sigma)

Specifies the type of discretisation applied to the 3D layer structure, either:

- Sigma-coordinates
- Z-coordinates

The number of Sigma-coordinate layers is specified using the [sigma layers](#) command or the [layer faces](#) command. Layer face elevations corresponding to the Z-coordinate mesh type are defined using the [layer faces](#) command.

Vertical mixing model == <Constant; Parametric; External>

(Optional; Default == Constant)

Sets the vertical momentum and scalar mixing model:

- Constant: a constant viscosity / diffusivity value is applied to the vertical mixing of both momentum and scalars.
- Parametric: a zero-equation parametric turbulence model in which a parabolic eddy viscosity / diffusivity profile is calculated. Stratification is represented using the Munk & Anderson stability formulae.
- External: any external turbulence model that has been built by the user to couple with TUFLOW FV through the tuflowfv_external_turb.dll.

See also the [Vertical mixing parameters](#), [Global vertical eddy viscosity limits](#), [Global vertical scalar diffusivity limits](#) [Turbulence update dt](#) and [External Turbulence Model Dir](#) commands.

Vertical mixing parameters == <eddy viscosity (m²/s); coefficients (-)>

(Optional)

Globally sets a constant vertical eddy viscosity (m²/s) or the vertical eddy viscosity coefficients. This is dependent on the vertical mixing model set using the [vertical mixing model](#) command:

- Constant: specify a constant eddy viscosity; Default == 0
- Parametric: specify the parametric model coefficients; Default == 0.4, 0.4

Not applicable if coupling with an External vertical mixing model.

Wind Stress Model == <1;2;3>

(Optional); Default == 1>

Globally sets the wind stress model to one of the following options:

- 1: Wu
- 2: Constant
- 3: Kondo

For further information please refer to Section 6.7. See also [wind stress parameters](#).

Wind Stress Parameters == <windpar>

(Optional); Default == <0.0, 0.8e-03, 50.0, 4.05e-03>

Optionally specifies the parameter values of the [wind stress model](#) as follows:

If [wind stress model](#) == 1:

Windpar = W_a (m/s), C_a (-), W_b (m/s), C_b (-)

Where:

$$Cd = Ca; [W_{10} < W_a]$$

$$Cd = Ca + (W_{10} - W_a) / (W_b - W_a) * (C_b - C_a); [W_a \leq W_{10} \leq W_b]$$

$$Cd = Cb; [W_{10} > W_b]$$

The default parameters correspond the Wu parameterisation (with a 50 m/s upper limit).

If [wind stress model](#) == 3:

Windpar = <scalefactor> with a default value of 1.

For further information please refer to Section 6.7.

Write Check Files == < >

(Optional)

Creates GIS check files in .mid/.mif or .shp format and text .CSV files for quality control checking of model input data. Refer to Section 12.10 for further details on the check files produced.

Write Empty GIS Files == < >

(Optional)

Creates empty GIS files in either mid/.mif or .shp format. Each layer as described in 5.3.3 is produced with the required attribute definitions pre-defined, but containing no geographic objects. Provided the [MI Projection](#) or [SHP Projection](#) command has been previously specified, each layer has the correct GIS projection.

Empty GIS layers are prefixed using the prefixes defined in Table 5-2 and are given a suffix of “_empty”. If <folder> is specified, the GIS files are located in the folder, which must already exist.

After writing the files, TUFLOW stops executing.

Write restart dt == <time (hours)>

(Optional)

Writes a restart file (.rst) to the [log directory](#) location at the time interval specified. The restart file is binary format and contains the spatially varying conserved variables at an instant in time.

A restart file is used to specify the initial condition for subsequent TUFLOW FV simulations using the [restart file](#) command.

MATERIAL BLOCK COMMANDS

[Bed elevation limits](#)

[Bottom roughness](#)

[Global bottom roughness](#)

[Horizontal eddy viscosity](#)

[Horizontal eddy viscosity limits](#)

[Horizontal scalar diffusivity](#)

[Horizontal scalar diffusivity limits](#)

[Inactive](#)

[Material](#)

[Spatial reconstruction](#)

[Surface roughness](#)

[Vertical eddy viscosity limits](#)

[Vertical scalar diffusivity limits](#)

Bed Elevation Limits == <zbmin, zbmax>

(Optional)

[Material](#) block command to specify limits to bed elevations for cells with material id#. Can also be set globally using the [global bed elevation limits](#) command.

Bottom roughness == <roughness value>

(Optional, Default == value set using [global bottom roughness](#) command)

[Material](#) block command used to set the bottom roughness value for cells with material id#. The bottom roughness specification depends on the [bottom drag model](#), and may be a Manning's 'n' coefficient (default) or an equivalent Nikuradse roughness, 'ks' (m).

Global bottom roughness == <bottom roughness>

(Optional)

Globally sets the bottom roughness value. The bottom roughness specification depends on the [bottom drag model](#), and may be a Manning's 'n' coefficient (default) or an equivalent Nikuradse roughness, 'ks' (m).

Horizontal eddy viscosity == <eddy viscosity (m²/s) ; coefficient (-)>

(Optional, Default == value set using [global horizontal eddy viscosity](#) command)

[Material](#) block command to specify the horizontal eddy viscosity Constant value (m²/s) or Smagorinsky model coefficient for cells with material id# (thereby overriding the default or corresponding globally parameters), depending on the turbulence model used. See [momentum mixing model](#) command to set momentum mixing turbulence model.

Horizontal eddy viscosity limits == <dv_limit1, dv_limit2>

(Optional)

[Material](#) block command for use with Smagorinsky [momentum mixing model](#) to set the minimum and maximum horizontal eddy viscosity (m²/s) limits for cells with material id# (thereby overriding the default or corresponding globally set parameters).

Horizontal scalar diffusivity == <diffusivity (m²/s) ; coefficient (-)>

(Optional, Default == value set using [global horizontal scalar diffusivity](#) command)

[Material](#) block command to specify the horizontal scalar diffusivity value (m²/s) or model coefficients for cells with material id# (thereby overriding the default or corresponding global turbulence parameters), depending on the scalar mixing model used. See [scalar mixing model](#) command to set momentum mixing turbulence model.

Horizontal scalar diffusivity limits == <ds_limit1, ds_limit2>

(Optional)

Material block command for use with Smagorinsky and Elder scalar mixing model to set the minimum and maximum horizontal scalar diffusivity (m^2/s) limits for cells with material id# (thereby overriding the default or corresponding globally set parameters).

Inactive == <0;1>

(Optional, Default == 0)

Material block command used to exclude cells with material id# from the computational domain:

- 0 = False (i.e. cells included).
- 1 = True (i.e. cells excluded).

Example material block commands:

```
material == 1
inactive == 1
end material
```

Material == <material id #>

...

End Material

(Mandatory)

This command indicates the beginning of a material block, specifying unique properties for cells with material id#. Material properties are listed in the following rows and the ‘end material’ command is used to indicate the end of the material block.

The following example material block specifies unique bottom roughness, horizontal eddy viscosity and horizontal scalar diffusivity values for all cells with material type 1 (thereby overriding the default or corresponding global turbulence parameters):

```
material == 1
bottom roughness == 0.020
horizontal eddy viscosity == 0.20
horizontal scalar diffusivity == 60.0, 6.0
end material
```

Note that several material types can be grouped into a single material block:

```
material == 2,3,4
bottom roughness == 0.1
```

```
end material
```

As a minimum, the roughness for each material type specified in the geometry file should be defined using material block commands (see also Section 8.5) or the [global bottom roughness](#) command. The commands that can be used to within a material block include:

- [Inactive](#)
- [Bottom roughness](#)
- [Surface roughness](#)
- [Horizontal eddy viscosity](#)
- [Horizontal scalar diffusivity](#)
- [Horizontal eddy viscosity limits](#)
- [Horizontal scalar diffusivity limits](#)
- [Vertical eddy viscosity limits](#)
- [Vertical scalar diffusivity limits](#)
- [Bed elevation limits](#)
- [Spatial reconstruction](#)
- [Shortwave radiation extinction coefficients](#)

Spatial reconstruction == <0;1>

(Optional, Default == 0)

[Material](#) block command used in high order simulations to optionally limit spatial reconstruction for cells with material id# (effectively reducing the spatial order of accuracy of the solution):

- 0 = False (i.e. no higher order reconstruction)
- 1 = True (i.e. higher order reconstruction)

Surface roughness == <roughness value>

(Optional, Default == 0)

[Material](#) block command used to set the surface roughness value, typically used to simulate ice cover.

Vertical eddy viscosity limits == <dv_limit1, dv_limit2>

(Optional)

[Material](#) block command for use with Parametric or External [vertical mixing model](#) to set the minimum and maximum vertical eddy viscosity (m^2/s) limits for cells with material id# (thereby overriding the default or corresponding globally set parameters). See also [global vertical eddy viscosity limits](#).

Vertical scalar diffusivity limits == <dv_limit1, dv_limit2>

(Optional)

Material block command for use with Parametric or External vertical mixing model. Used to control the vertical diffusion limits (m^2/s) i.e. to increase/decrease the effective vertical mixing if there is insufficient or too much vertical diffusion occurring in sections of your model. Please refer to Section 7.2.2. See also global vertical scalar diffusivity limits.

Vertical scalar diffusivity limits == <ds_limit1, ds_limit2>

(Optional)

Material block command for use with Smagorinsky and Elder scalar mixing model to set the minimum and maximum vertical scalar diffusivity (m^2/s) limits for cells with material id# (thereby overriding the default or corresponding globally set parameters).

BOUNDARY CONDITION BLOCK COMMANDS

[BC](#)

[BC default](#)

[BC default update dt](#)

[BC header](#)

[BC nodestrings](#)

[BC offset](#)

[BC reference time](#)

[BC scale](#)

[BC time units](#)

[BC update dt](#)

[Grid definition file](#)

[Grid definition variables](#)

[Includes MSLP](#)

[Sub-type](#)

[Vertical coordinate type](#)

[Vertical distribution file](#)

```
BC == <bc type, [id], [input file]>
```

```
...
```

```
End BC
```

```
<OR>
```

```
BC == <bc type, [xid], [yid], [input file]>
```

```
...
```

```
End BC
```

Typically (but not always), at least one boundary condition will be required for a TUFLOW FV simulation and often a number of different boundary condition types will be applied. Each boundary condition type is defined using a boundary condition (BC) block. The ‘BC’ and ‘End BC’ commands indicate the beginning and end of a boundary condition block.

See Table 10-2 and Table 10-3 for lists of boundary types.

Boundary conditions can be applied:

- Spatially (typically metrological conditions and/or wave fields)
- Along a nodestring (external boundaries such as water levels or flows)
 - The [id] value is the nodestring identifier included in the mesh geometry file (see Section 8.1.2)
- As a point source (within a single cell such as an outfall discharge or a moving point source such as a plume generated by dredger)
 - The [xid], [yid] values are the coordinates of the source location within the model domain.

The commands that can be used within a BC block are:

- [BC header](#)
- [Sub-type](#)
- [BC offset](#)
- [BC scale](#)
- [BC default](#)
- [BC update dt](#)
- [BC time units](#)

-
- [BC reference time](#)
 - [Includes MSLP](#)
 - [Include wave stress](#)
 - [Include stokes drift](#)
 - [Layer](#)
 - [Vertical distribution file](#)
 - [Vertical coordinate type](#)
 - [BC nodestrings](#)
 - [Polygon File](#)

For further details on boundary specification please refer to Section 10.2

BC default == <Var1_default, [Var2_default],...>

(Optional)

[BC](#) block command to specify a default boundary condition value if entry in the input file is empty.

BC default update dt == <time (s)>

(Optional)

A global command that allows the user to specify the update timestep for all boundary conditions. If not specified, the boundary condition is updated at every simulation timestep. See [BC update dt](#) for setting the update timestep for a specific boundary condition.

BC Header == <Header1,Header2,...>

(Optional)

[BC](#) block command that allows the user to specify the .CSV input file column headers or NetCDF file variable names (thereby overriding the defaults in described in Table 10-2 and Table 10-3). This command should immediately follow a [BC](#) command.

For example, the following lines apply a cell inflow (QC boundary condition type) at the cell which lies at the x,y coordinate 1025.5, 950.5. It looks in the specified .CSV file for columns Time, Tailwater_Flow and Turbidity:

```
BC == QC, 1025.5, 950.5, ..\bc\ tailwater_discharge.CSV
  BC header == Time,Tailwater_Flow,Turbidity
  End BC
```

Another example shows a water level boundary (WL) applied to nodestring 1, which looks in the specified .CSV file for columns Time and WL_Loc1:

```
BC == WL, 1, ..\bc\tidal_water_level.CSV
  BC header == Time, WL_Loc1
End BC
```

A final example shows a gridded wind field (W10_Grid) applied to a domain previously defined using the [grid definition variables](#) command, which looks in the specified NetCDF file for variables Time, Wind_X and Wind_Y:

```
BC == W10_Grid, 1, ..\bc\wind_10_grid.nc
  BC header == Time, Wind_X, Wind_Y
End BC
```

BC nodestrings == <id1, . . . , idn>
(Optional)

[BC](#) block command to apply the boundary condition to multiple nodestrings (only relevant to the OBC_GRID boundary type).

BC offset == <Var1_Offset, [Var2_Offset], . . . >
(Optional)

[BC](#) block command to apply an offset to boundary condition values.

BC offset == <Var1_Offset, Var2_Offset, Varn_Offset>

Please refer to Section 10.4.4 for further details.

BC reference time == <Hours;ISODate>
(Optional)

[BC](#) block command to set the boundary condition reference time. If not specified, the boundary condition is assumed to be consistent with the simulation [reference time](#).

BC scale == <Var1_ScaleFactor, [Var2_ScaleFactor], . . . >

[BC](#) block command to apply a scale factor to boundary condition values.

BC scale == <Var1_ScaleFactor, Var2_ScaleFactor, Varn_ScaleFactor>

Please refer to Section 10.4.4 for further details.

BC time units == <hours; . . . >
(Optional)

[BC](#) block command used to specify the unit of time for a boundary condition specified using a NetCDF file. The options are:

- Days

-
- Hours
 - Minutes
 - Seconds
 - Isotime

If not specified, the default is hours relative to the simulation [reference time](#).

BC update dt == <time (s)>

(Optional)

[BC](#) block command that allows the user to specify the update timestep for a boundary condition. If not specified, the boundary condition is updated at every simulation timestep.

Boundary gridmap == <0;1>

(Optional, Default == 0)

[Grid definition file](#) block command. Set to 1 to calculate interpolation weightings from the grid onto the boundary nodestrings (this is required for OBC_grid boundary conditions that are to be later applied to nodestrings).

Cell gridmap == <0;1>

(Optional, Default == 1)

[Grid definition file](#) block command. If set to 0, TUFLOW FV will not calculate the interpolation weightings of the cells in the model for this gridded boundary condition (may be more efficient for grids that are only going to be applied to the boundaries and not internal model domain).

Grid definition file == <NetCDF file defining grid coords (.nc)>

...
...
...
...

End Grid

(Optional)

Specifies a NetCDF location and filename that defines grid coordinates to be used in mapping input files to the model mesh. The commands that can be used within a grid definition file block include:

[grid definition label](#)

[grid definition variables](#)

[vertical coordinate type](#)

[cell gridmap](#)

[boundary gridmap](#)

[suppress coverage warnings](#)

End grid

For more information and examples please refer to Section 10.5.3.1.

Grid definition label == <name>

(Optional)

[Grid definition file](#) block command to specify the grid name and can be referred to by multiple boundary conditions later. This command is placed within the [grid definition file](#) command block.

Grid definition variables == <v1, v2, v3>

(Optional)

[Grid definition file](#) block command to specify the x,y coordinate variable names (typically Easting, Northing or Longitude, Latitude) contained in the NetCDF file defined using the [grid definition file](#) command.

Includes MSLP == <1;0>

(Optional, Default == 1)

[BC](#) block command that allows the user to specify whether a water level boundary condition (WL or WLS) includes an inverse barometer offset.

The default assumption (1) is that the boundary does already include an inverse barometer component.

If include MSLP == 0 then an offset determined by the local MSLP difference from the [reference MSLP](#) is applied at the boundary.

Sub-type == <sub-type number>

(Optional, Default == 1)

The sub-type [BC](#) block command is applicable for various boundary types (refer Table 10-4) and allows the user to control certain details of how these are numerically implemented.

For a Q type boundary condition:

- If sub-type == 1 (default), flow is:
 - Applied as a flux
 - Distributed across a nodestring by cell width
 - Note: While the net flow will match the input file specifications, using this sub-type with 3D simulations does not guarantee uniform inflow over the entire water column. In some cases, one part of the water column can be flowing in while another is flowing out. It is therefore recommended to use sub-type 2 or 4 for 3D models.
- If sub-type == 2, flow is:
 - Applied as a source term

-
- Distributed across a nodestring by cell width
 - Note: Boundary condition is specified as a reflective wall with a source distributed along the internal boundary cells.
 - If sub-type == 3, flow is:
 - Applied as a flux
 - Distributed across a nodestring by cell width and depth ($W \cdot H^{1.5}$)
 - Note: Boundary inflow is distributed according to depth along nodestring. This boundary condition treatment is otherwise the same as sub-type 1. This sub-type may not be suitable for use in 3D model simulations.
 - If sub-type == 4, flow is:
 - Applied as a source term
 - Distributed across a nodestring by cell width and depth ($W \cdot H^{1.5}$)
 - Note: Boundary inflow is distributed according to depth along nodestring. This boundary condition treatment is otherwise the same as sub-type 2. This sub-type is suitable for use in 3D model simulations.

Note: for overland application with inflows over an initially dry bed, subtype = 4 is recommended.

For a QN type boundary condition:

- If sub-type == 1 (default) the flux calculation uses a boundary calculation method added to the 2019 TUFLOW FV Release. This method should limit warning messages that could occur when using QN boundaries: ‘**Unable to solve for specified inflow conditions at bc**’.
- If sub-type == 2 the flux calculation uses the pre-2019 TUFLOW FV Release’s QN boundary flux calculation method and can be used for legacy simulations.

For all OBC boundary condition (i.e. OBC, OBC_PROF, OBC_CURT, OBC_GRID):

- If sub-type == 1 (default), the boundary is a specified water level. Boundary normal momentum flux is modified to avoid BC over-specification which can lead to boundary reflection of outgoing energy.
- If sub-type == 2, the boundary specifies an incoming wave form which is superimposed with the internally calculated outgoing wave form.
- If sub-type == 3, the boundary specified flow velocity. Water level is modified to avoid BC over-specification which can lead to boundary reflection of outgoing energy.
- If sub-type == 4, the boundary is over specified (both water level and velocity are specified). Water level and velocities are applied exactly as specified in the input files. This can lead to boundary reflection of outgoing energy.
- If sub-type == 5, specified water levels are treated as an increment to apply to the previously specified water level. This can for instance be used to add a tidal signal to a separately specified non-tidal OBC. This sub-type can also be used in conjunction with WL, WLS and WL_CURT BCs.

Note: for application with supercritical upstream boundaries, subtype = 4 is recommended.

QG and QC boundary conditions support the following sub-type specifications:

- If sub-type == 1 (default). When outflow is specified ($Q < 0$) the scalar flux is determined by the interior model concentration (the BC file value will be ignored).
- If sub-type == 2. When outflow is specified ($Q < 0$) the scalar flux is determined by the BC file specified value.

Suppress coverage warnings == <0;1>

(Optional, Default == 0)

[Grid definition file](#) block command. If set to 1 suppresses warnings if the grid does not cover the entire TUFLOW FV domain. It may be more efficient and result in smaller log file output in situations that the grid only needs to cover a small number of the cells within the model domain.

Vertical coordinate type == <elevation;depth;sigma;height>

(Optional 3D)

[BC](#) or [Structure](#) block command to specify the BC vertical coordinate type for vertically distributed boundary conditions. The options are:

- Elevation
- Depth
- Sigma
- Height

This command is followed by speciation of a [vertical distribution file](#) that defines the vertical distribution.

If not specified, the boundary condition is distributed evenly over the water column.

Vertical distribution file == <file path>

(Optional 3D)

[BC](#) or [Structure](#) block command used in conjunction with [vertical coordinate type](#) to specify a .CSV file that describes the boundary condition vertical distribution.

The .CSV file should contain two columns:

- The first column is the vertical coordinate (e.g. DEPTH) type reference.
- The second column is the weighting (between 0 and 1) at the corresponding vertical reference. The units of WEIGHT are irrelevant as the distribution is normalised.

The first example .CSV file corresponds to the [vertical coordinate type](#) “depth” and the boundary condition being applied to the top 2m of the water column:

DEPTH, WEIGHT

0.0, 1
2.0, 1
2.1, 0
9999.0, 0

The second example corresponds to the [vertical coordinate type](#) “height” and the boundary condition being applied to the bottom 1m of the water column.

HEIGHT, WEIGHT

0.0, 1
1.0, 1
1.11, 0
9999.0, 0

The final example corresponds to the [vertical coordinate type](#) “elevation” and the boundary condition being applied at -10 to -20 meters (below the model datum).

ELEVATION, WEIGHT

0.0, 0
-1.0, 0
-5.0, 0
-9.9, 0
-10.0, 1
-20.0, 1

STRUCTURE BLOCK COMMANDS

Bed adjust	Start control state
Blockage file	Structure
Control	Target file
Control file	Trigger value
Control parameter	Vertical coordinate type (see BC Block)
Control update dt	Vertical distribution file (see BC Block)
Culvert file	Width file
Destratification Unit	
Energy loss file	
Energy loss function	
Flux file	
Flux function	
Form loss coefficient	
Max conc	
Max open increment	
Min conc	
Name	
Polygon file	
Properties	
Sample dt	
Sample point	
Sample type	
Scal fact	
Scalar function	

Bed adjust == <celltype>

(Optional, Structure Block command)

If [Structure](#) = cell or zone then a bed adjust command can be used.

The bed adjust function type can be:

- ZB_adjust:
 - Adjustable bed elevations for a series of cells with a specified crest level
- DZB_adjust:
 - Adjustable bed elevations for a series of cells with a specified crest level dz above existing bed levels

A control specification is required to initiate bed adjustments (See [control](#)). Refer to Section 11.6.1 for more details. An example using a trigger control block is provided below:

```
Structure == Cell, 1041           ! Structure Location (in cell 1041)
Bed Adjust == zb_adjust          ! Adjusts Bed Level

Control == Trigger               ! Adjust after trigger level
  Sample point == 622943, 4334061 ! Location of sampling (x,y)
  Sample type == WL             ! Trigger can only use WL
  Sample dt == 0.05              ! hours
  Trigger Value == 24.0          ! Trigger level (m)
  Control file == BreachCondition.csv ! Timeseries of zb after trigger reached
End Control

End Structure
```

BreachCondition.csv	
Time	Zb
0.0	24.0
2.0	23.0
4.0	22.0
6.0	21.0
8.0	20.0
10.0	19.0

Blockage file == <blockfile(.CSV)>

(Optional, Structure Block Command)

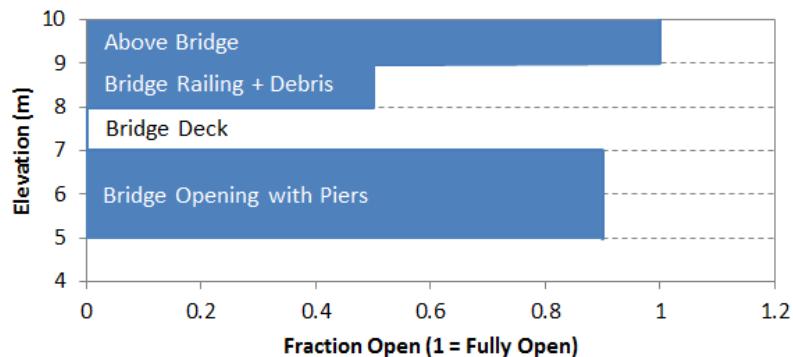
The blockage file is a comma separated variable file with a relationship of flow fraction and depth, commonly used during modelling of bridge structures (in conjunction with [Form loss coefficient](#)). Refer to in Section 11.4 for further details.

The file contains the header line with column labels “Z” and “FRAC”.

- The Z column is a list of elevations (lowest to highest).
- The FRAC column is the fraction of flow (0 to 1) for the vertical section between the corresponding Z value and its previous value.

An example of a CSV file is given below:

Z, FRAC
5.0, 0.9
7.0, 0.9
7.1, 0.0
7.9, 0.0
8.0, 0.5
8.9, 0.5
9.0, 1.0



Control == <controltype>

...

End Control

(Optional, Structure Block Command)

Specification of structure logic definition. Available control types include:

- *Trigger*: Trigger controls define a condition that will commence after the first exceedance of a specific [trigger value](#) within the model at a specified [sample point](#). Currently TUFLOW FV is only able to sample water levels for trigger controls.
- *Timeseries*: Timeseries controls define an adjusting condition based on a separate [control file](#) CSV timeseries. The current timestep is interpolated into the timeseries to calculate a corresponding [control parameter](#) value.
- *Sample_rule*: A sample rule control can adjust the [control parameter](#) by sampling a variable within the model and comparing it to a [control file](#) CSV. Currently TUFLOW FV can sample the following variables using the [sample type](#) command at a [sample point](#):
 - H , V_x , V_y , SAL , $TEMP$, $SED_1 - SED_N$, $TRACE_1 - TRACE_N$, $WQ_1 - WQ_N$
- *Target_rule*: A target rule control can adjust the [control parameter](#) to reach the target value of the [sample type](#) at a [sample point](#). The flow through the structure is adjusted based on the difference between the sampled value and the target value. The target value is specified in a [target file](#), which is a timeseries to allow for time-varying target conditions.

For quick reference, commands that reside in a control block are provided below:

[Control](#) == <controltype>

[Control parameter](#) == <controlparam>

[Control file](#) == <cfile>

[Control update dt](#) == <cdt (hours)>

[Sample point](#) == <spx, spy>

[Sample type](#) == <st>

[Max opening increment](#) == <moi>

[Trigger value](#) == <tv>

[Target file](#) == <tfile(.CSV)>

End Control

Please refer to Sections 11.6.1 and 11.6.2 for more details and Section 11.6.5 for examples.

Control file == <cfile(.CSV)>

(Optional, Nested Control Block Command)

Reads in a comma separated file (.CSV) with structure controls. The file contains a header line with specific column labels required for specific structure types:

If [flux function](#) == *weir_adjust* or *weir_dz_adjust* and [control parameter](#) == *weir_crest*

-
- Column headers = “Time**, weir_crest”

If bed adjust == zb_adjust and control parameter== zb

- Column headers = “Time**, zb”

If bed adjust == dzb_adjust and control parameter== dzb

- Column headers = “Time**, dzb”

If control == sample_rule:

- Column headers = “Sample_value, ControlParameter” (e.g. “*Sample_value, Fraction_open*”) or (e.g.:*Sample_value, Min_flow*)

If control == timeseries

- Column headers = “Time, ControlParameter” (e.g. “Time, Fraction_open”)

If control == target

- Column headers = “, Target_deficit, ControlParameter” (e.g. “Target_deficit, Fraction_open”)

**For and control files with “Time” values are specified:

- If control == trigger, Time (hr) from the moment that the structure adjustment commences.
- If control == timeseries, Time (hr) from the start of the model simulation.

Please refer to Sections 11.6.1 and 11.6.2 for more details and Section 11.6.5 for examples. See also control parameter and control.

Control parameter == <controlparam>

(Optional, Nested Control Block Command)

Specification of the parameter that will be controlled.

Options available are:

- *Fraction_open*: Effectively a scale factor (between 0 and 1) of the flow rate through the structure. Multiple fraction_open control parameter commands within the same structure will multiply together for the overall value. Fraction_Open is often used to simulate culverts or pumps with a gate control that is automated based on the water level elsewhere in the region. Please note this control parameter is not supported with flux function == nlswe.
- *Min_flow*: Provides a minimum flow rate that must pass through the structure (physical limitations permitting). This is often used in conjunction with dam release structures where a minimum flow is required for environmental releases. When multiple min_flow controls are used in the same structure, the maximum of these will be the adopted value. Please note this control parameter is not supported with flux function == nlswe.
- *Weir_crest*: Used with weir_adjust or weir_dz_adjust flux functions (refer Section 11.3.4.2) and allows the crest height of the weir (absolute level or relative to the adjacent cells) to be adjusted based on the control. Often used to simulate a lowerable or failing levee system (please refer to Figure 11-7 for an example).
- *Zb*: Can only be used with cell or zone structure types using bed adjust (refer Section 11.2.4) and adjusts the bathymetry of the structure area based on a condition. Often used to artificially adjust a model bed for scour with time, or to simulate a bund failure after a trigger condition.

-
- *Dzb*: As per *ZB* above, but with all elevation changes relative to the existing bed level of the structure region/cell.

Refer to Section 11.6.3 for more details and Section 11.6.5 for examples. See also [Control file](#) and [control](#).

Control update dt == <cdt (hours)>

(Optional, Nested Control Block Command)

The frequency of updating the control structure operation (hours).

Refer to Section 11.6.1 for more details.

Culvert file == <culvertfile(.CSV), ID>

(Optional, Structure Block Command)

Required if [Flux function](#) == Culvert

Reads in a comma separated variable file (CSV) with properties for a list of culverts.

- Culvertfile is the file containing a list of culvert properties.
- ID identifies the specific culvert properties from the list of culverts in culvertfile.

The file contains a header line with column labels. Each subsequent line contains the property values listed in Table 11-3. Refer to Section 11.3.7 for more details.

Destratification unit == <celltype>

(Optional, Structure Block command)

If [structype](#) = cell or zone then a destratification unit can be specified.

Destratification unit function type can be:

- Bubbler:
 - A bubbler structure, parameters as specified in in the [properties](#) command

Energy loss file == <energyfile(.CSV)>

(Optional, Structure Block command)

```
Structure == Nodestring, 6           ! Structure location (NS 6)
  Energy loss function == table    ! Loss type
  Energy loss file == energyloss.csv ! Pier losses (hydraulics of bridged waterways)
  Width file == Width_Example.csv   ! Width file
End Structure
```

energyloss.csv		
Q	dH	Comment
-10	0.1	! Flow against structure direction
0	0	! No head loss for no flow
1	0.1	! Nlarger head losses for +ve flow
5	0.2	
10	0.3	

Energy loss function == <energytyp>

(Optional, Structure Block command)

If structype = nodestring or structype = linked nodestrings then an energy loss function can be specified.

Energy loss function type can be:

- Coefficient:
 - Requiring specification of a [form loss coefficient](#).
- Table:
 - Requiring a hQh relationship to define the structure (see [flux file](#)).

Refer to Section 11.4 for more details.

Flux file == <hQhfile(.CSV)>

(Optional, Structure Block Command)

Required if fluxtype = matrix.

The flux file is a comma separated variable file with the hQh flux matrix, defining discharge for a combination of upstream and downstream water levels.

It contains header lines (as many header lines as desired but with no more than 2 commas in each line), then a matrix as follows:

- First row is a list of upstream water levels
- First column is a list of downstream water levels
- Matrix is discharge values corresponding to the listed water levels (corresponding row for downstream, corresponding column for upstream).
- The first value on the first line is a scale factor, which is applied to the Q values in the matrix.

An example of a CSV file is shown below. Refer to Section 11.3.3 for more details.

WL_DS,WL_US

1,0,2,4,10

0,0,0,3.6,5.1,8.1

1,-2.6,2.6,4.4,7.7

2,-3.6,0.0,3.6,7.2

10,-8.1,-7.2,-6.3,0.0

		Water Level Upstream (RL)			
		1	0	2	4
Water Level Downstream (RL)	0		0	3.6	5.1
	1		-2.6	2.6	4.4
	2		-3.6	0	3.6
	10		-8.1	-7.2	-6.3

Water Level Upstream (mRL of ftRL)
Water Level Downstream (mRL or ftRL)
Matrix Scale Factor
Flow (m^3/s or cfs)

Flux function == <fluxtype>

(Optional, Structure Block Command, Default == nlswe)

If structype = nodestring or structype = linked nodestrings or structype == linked zones then the flux function type defines the flux (or flow).

Flux function type can be:

- NLSWE:
 - The default flux function. Often used in combination with [Blockage file](#) or [width files](#) and an [energy loss function](#).
- Culvert:
 - A culvert structure.
 - See [culvert file](#) and Section 11.3.7 for more details.
- Porous:
 - A porous structure (Darcy flow conditions) (Section 11.3.6). See also [properties](#).
- Timeseries:
 - A specified timeseries of flow (see [flux file](#) and/or Section 11.3.2).
- Matrix:
 - A hQh relationship defines the structure (contained in the flux file). For this structure, flow is distributed across a nodestring by cell width and depth (WH^{1,5}). (see [flux file](#) and Section 11.3.3).
- Wall:
 - a solid wall (Q=0) (refer Section 11.3.5)
- Weir:
 - A broad crested weir structure with a fixed crest level
 - Crest level is specified in the [properties](#) command.
 - See Section 11.3.4.1 for more details.
- Weir_dz:
 - A broad crested weir structure with a crest level dz above existing bed levels
 - Weir crest levels are specified by:
 - A nodestring polyline with a “Z” column specification
 - If not using a nodestring polyline, weir crest is the highest of the 2dm file vertices and any additionally specified cell elevations.
 - An additional increment dz, in the [properties](#) command (can be dz = 0).
- Weir_adjust:
 - A broad crested weir structure with an adjustable crest level
 - Crest level is specified in the [properties](#) command.
 - [Control types](#) are used to specify how the weir elevation should be varied, either by time series from a trigger location/water level from somewhere within the model domain, or from the start of the model simulation.
 - See Section 11.3.4.2 for more details.
- Weir_dz_adjust:
 - A broad crested weir structure with an adjustable crest level dz above existing bed levels.
 - [Control types](#) are used to specify how the weir elevation should be varied, either by time series from a trigger location/water level from somewhere within the model domain, or from the start of the model simulation.
 - See Section 11.3.4.2 for more details.

Form loss coefficient == <flc>

(Optional, Structure Block command)

If [energy loss function](#) = Coefficient, form loss coefficient applied to structure. FLC applies a head loss across a cell face according to the equation:

$$\Delta h = FLC \cdot v^2 / 2g$$

Refer to Section 11.4.3 for more details.

Max_conc == <Sal, Temp, SED_1, ... SED_n, TRACE_1, ... TRACE_n>

(Optional, Structure Block command)

The maximum concentration of the tracers passing through the structure.. Used in combination with the [scalar function](#) command. See also [Min conc](#) and [Scal fact](#). For more information on Scalar functions please refer to Section 11.5).

Max opening increment == <moi>

(Optional, Nested Control Block Command)

Maximum change in structure Fraction open (See [control](#)) per update time step (see [Sample dt](#)). Refer to Section 11.6.1 for more details.

Min_conc == <Sal, Temp, SED_1, ... SED_n, TRACE_1, ... TRACE_n>

(Optional, Structure Block command)

The minimum concentration of the tracers passing through the structure. Used in combination with the [scalar function](#) command. See also [Max conc](#) and [Scal fact](#). For more information on Scalar functions please refer to Section 11.5).

Name == <sname>

(Optional, Structure Block Command)

Name of structure, used in some structure outputs later including the structflux, structcheck outputs.

Polygon file == <polyfile(.CSV)>

(Optional, Structure Block command)

Reads in a comma separated variable file defining the perimeter vertices of a polygon.

The file contains a header line with column labels “X”, “Y” and “ID, which define the coordinates of points describing the perimeter of the polygon and the polygon that each coordinate belongs to. The definition of points needs to be consecutively listed and can be either clockwise or counter-clockwise. TUFLOW FV searches for cell centres that lie within the polygon.

Commonly used with [Bed adjust](#) when [structure](#) == Zone.

Properties == <p1, . . . , pn>

(Optional, Structure Block Command)

If [flux function](#) == “Weir” or “Weir_dz”, then

- P1 = weir crest level (for a weir) or level above existing bed levels
 - for weir there is no default (level required)
 - for weir_dz, default = 0.0
- P2 = weir coefficient
 - default = 1.6

If [Structure](#) == “Autoweir”, then

-
- P1 = threshold elevation difference, where the autoweir is activated when the minimum of the face vertices elevations are P1 higher than the adjacent cell elevations
 - default = 0.1 m or ft
 - P2 = weir coefficient
 - default = 1.6

If [flux function](#) == “Porous” then

- P1 = Porous structure hydraulic conductivity (m/s)
- P2 = Porous structure width (m in the direction of flow)

If using [destratification unit](#) == “Bubbler” then

- P1 = elevation of bubbler
- P2 = air flow rate of bubbler (m³/s)
- P3 = alpha
- P4 = b1
- P5 = Lr
- P6 = gamma

Please contact support@tuflow.com for further information on the use of destratification unit structure types.

Sample dt == <sdt (hours)>

(Optional, Nested Control Block Command)

The frequency of updating the variable structure (hours). Required is using the

Refer to Section 11.6.1 for more details.

Sample point == <spx, spy>

(Optional, Nested Control Block Command)

spx, spy defines the location that controls the variable z value structure (i.e. the “control” point)

Commands commonly used in conjunction with [Sample point](#) are [Sample type](#) and [Sample dt](#).

Refer to Section 11.6.1 for more details.

Sample type == <st>

(Optional, Nested Control Block Command)

[Control](#) block command that defines the model parameter used for the sampling at a [Sample point](#).

Options available are:

- H: Water level (Please note “WL” is also supported);
- V_x: X-component of velocity
- V_y: Y-component of velocity
- SAL: Salinity
- TEMP: Temperature
- SED_1 – SED_N: Suspended sediment concentration for a selected fraction SED_X
- TRACE_1 - TRACE_N: Tracer concentration for a selected tracer TRACE_X
- WQ_1 – WQ_N: WQ constituent concentration for a selected constituent WQ_X

Refer to Section 11.6.1 for more details.

`Scal_fact == <Sal, Temp, SED_1, ... SED_n, TRACE_1, ... TRACE_n>`

(Optional, Structure Block command)

Used in combination with the [scalar function](#) command. The factor by which scalar concentrations are modified as they pass through the structure. Does not conserve mass.

For scale factor type scalar functions this scale factor will always be applied. If a [max conc](#) or [min conc](#) are also present then they will be applied afterwards.

For treatment type scalar functions this scale factor will be applied unless it results in concentrations below those specified in the [min conc](#) command.

`Scalar Function == <scalar type>`

(Optional, Structure Block command)

Scalar type options are:

- None;
- Timeseries;
- Scale_Factor; or
- Treatment.

Commands commonly used with the scalar function include: [Min conc](#), [Max conc](#) and [Scal_fact](#). For more information on scalar functions please refer to Section 11.5).

`Start control state == <scs>`

(Optional, Nested Control Block Command, Default == 1)

Value equal to or between 0 and 1 defining the fraction open of a structure at the start of the simulation.

Refer to Section 11.6.1 for more details.

`Structure == <Structype, ID>`

...

...

`End Structure`

(Optional)

Each structure type is defined using a structure block. The ‘Structure’ and ‘End Structure’ commands indicate the beginning and end of an structure block. The structure block specifies the type of structure using the structype input, and it’s location (if applicable) using the ID input.

The available structype options are:

- Nodestring
 - The structure is situated between one or more elements (i.e. – along the cell faces, defined by a nodestring)
 - The [id] value is the .2dm or external nodestring identifier.
- Linked nodestrings
 - The structure is situated between two nodestrings.
 - The first [ID1] nodestring is upstream and the second nodestrings [ID2] is downstream.

-
- For this structure, flow is distributed across a nodestring by cell width and depth ($WH^{1.5}$).
 - **Cell**
 - The structure is a single cell.
 - The [id] value the cell identifier.
 - **Zone**
 - The structure is a series of cells, defined by a polygon.
 - No [id] value is required.
 - **Linked zones**
 - The structure is situated between two zones.
 - No [id] value is required.
 - **Autoweir**
 - This structure identifies all cell faces (not nodestrings) in the model domain that are elevated above the adjacent cells. These cell faces are then assigned a weir flow condition.

Commands nested within a structure block

Once the structure type has been defined, a range of supporting commands can be nested into the block, depending on the structure type and if applicable: flux function, scalar function and need for structure controls.

Structure == Structype, ID

Name == <sname>
Flux function == <fluxtype>
Culvert file == <culvertfile(.CSV), ID> (if flux function == Culvert)
Flux file == <hQhfile(.CSV)> or <timeseries(.CSV)> (if flux function == Matrix or Timeseries)
Properties == <p1,....,pn>
Polygon file == <polyfile(.CSV)> (if structure == zone or linked zone)
Desratification unit == <celltype> (if structure == cell or zone)
Bed adjust == <celltype> (if structure == cell or zone)

Scalar function == <scalarmtype>
Min conc == <conc_1, conc_2, ... conc_n>
Max conc == <conc_1, conc_2, ... conc_n>
Scal_fact == <conc_1, conc_2, ... conc_n>

Control == <controltype>
Control parameter == <controlparam>
Control update dt == <cdt (hours)>
Sample point == <spx, spy>
Sample type == <st>
Sample dt == <sdt (hours)>
Max opening increment == <moi>
Trigger value == <tv>
Target file == <tfile(.CSV)>

End Control

End Structure

Refer to Section 11.2 for more details.

Target file == <tfile(.CSV)>

(Optional, Nested Control Block Command)

Reads in a comma separated file (.CSV) defining the target value for the [Sample Type](#)

- Column headers = "Time","Target_Value"

Refer to Section 11.6.1 for more details.

Trigger value == <tv>

(Optional, Nested Control Block Command)

The value of the specified model parameter at the [Sample point](#) spx, spy that, when exceeded, will trigger a change in structure elevations.

Note that currently the trigger value can only be an absolute water level. Refer to Section 11.6.1 for more details. Command is commonly used in conjunction with [Control file](#) and [Polygon file](#)

Vertical coordinate type == <elevation;depth;sigma;height>

(Optional, Structure Block command)

Analogous to [BC](#) block [vertical coordinate type](#).

Vertical distribution file == <file path>

(Optional, Structure Block command)

Analogous to [BC](#) block [vertical distribution file](#).

Width file == <widthfile(.CSV)>

(Optional, Structure Block command)

The width file is a comma separated variable file with a relationship of flow width and depth which is commonly used during modelling of bridge structures (in conjunction with [Form loss coefficient](#)).. Refer to in Section 11.4 for further details.

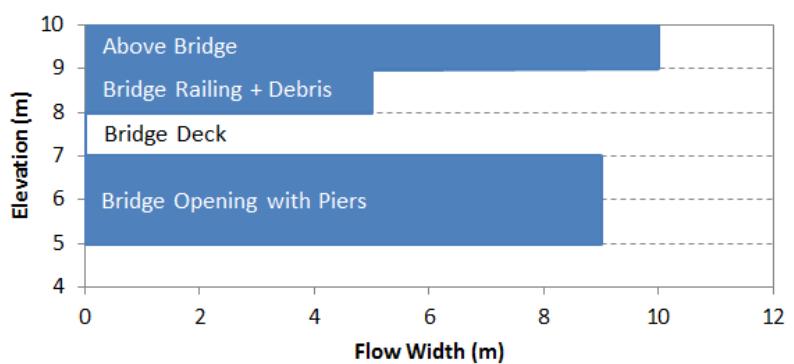
The file contains the header line with column labels "Z" and "WIDTH".

- The Z column is a list of elevations (lowest to highest).
- The WIDTH column is the width of flow (m or ft – depending on [units](#)) for the vertical section between the corresponding Z value and its previous value.

An example of a CSV file is given below:

Z, WIDTH

5.0, 9.0
7.0, 9.0
7.1, 0.0
7.9, 0.0
8.0, 5.0
8.9, 5.0
9.0, 10.0



OUTPUT BLOCK COMMANDS

[Final output](#)

[Output](#)

[Output compression](#)

[Output interval](#)

[Output parameters](#)

[Output points file](#)

[Output statistics](#)

[Start output](#)

[Suffix](#)

[Vertical averaging](#)

Final Output == <time>

(Optional)

[Output](#) block command to specify the final time for an output request. The time format must be consistent with the simulation [time format](#). If not specified, the output final time will be consistent with the simulation end time.

Output == <output format>

...

...

...

End output

(Mandatory)

Each output type is defined using an output block. The ‘Output’ and ‘End output’ commands indicate the beginning and end of an output block. The output block specifies the type of output and the output properties including the desired parameters and time definitions. Table 12-1 presents a summary of the the output types available, which include:

- dat (cell centred SMS data file output, not typically recommended).
- Datv (recommended for visualisation in SMS and QGIS).
- XMDF (recommended for visualisation in SMS and QGIS).
- Flux (time series of fluxes across all nodestrings, refer Section 12.5).
- Mass (time series mass in model refer Section 12.5).
- NetCDF (cell centred 2D and 3D output, used heavily for 3D model visualisation, refer Section 12.6.4).
- NetCDFv (vertex interpolated data file output, not typically recommended).
- Points (time series outputs, refer Section 12.3).
- Profiles (3D water column outputs at a location, refer Section 12.4).
- Transport (TUFLOW FV file used for decoupling hydrodynamics from following advection dispersion, sediment or water quality simulations, refer Section 10.8).
- Structflux (review flows through structures refer Section 12.8.1).
- Structcheck (Check the behaviour of a structure, commonly used in association with structure controls refer Section 12.8.2).

The commands that can be used within an output block include:

- [Output parameters](#)
- [Output interval](#)
- [Start output](#)

-
- [Final output](#)
 - [Suffix](#)
 - [Output points file](#)
 - [Output compression](#)
 - [Output statistics](#)
 - [Output statistics dt](#)
 - [Vertical averaging](#)
-

Output compression == <0 ; 1>

(Optional, Default == 1)

[Output](#) block command to compress NetCDF output files:

- 0 = False (i.e. no NetCDF file compression)
 - 1 = True (i.e. NetCDF file compression)
-

Output Interval == <timestep (s)>

(Optional)

[Output](#) block command used to specify the desired output interval in seconds. If this command is not specified output will be produced at each timestep. In many applications this will not be desired (possibly leading to extremely large output files) and an output interval at 10min (600s) or 30min (1800s) will be more appropriate.

Output parameters == <resulttype>

(Mandatory)

[Output](#) block command used within the output block to specify the required output parameters. The available output parameters are summarised in Table 12-3 and Table 12-4 (note that some output parameters are dependent on the output type).

Output points file == <filepath>

(Optional)

Mandatory command when points [output](#) type is required. This provides the location and name of a comma separated variable file with the coordinates of the required output points. The following column headers are required in .CSV file:

- X, Y, ID (where the ID field is optional but recommended for ease of result review)

Output statistics == <type 1, type 2>

(Optional)

Output additional requested statistics. The following statistics are currently supported: MAX & MIN. This feature is available with datv and NetCDF output types.

Start Output == <time>

(Optional)

Output block command to specify the start time for an output request. The time format must be consistent with the simulation [time format](#). If not specified, the output start time will be consistent with the simulation start time.

Suffix == <suffix>

(Optional)

Output block command to add a suffix to the output filename.

Vertical averaging == <type>

(Optional 3D, Default == depth-all)

Optional output block command to vertically average 3D results over a specified range. The vertical averaging types are:

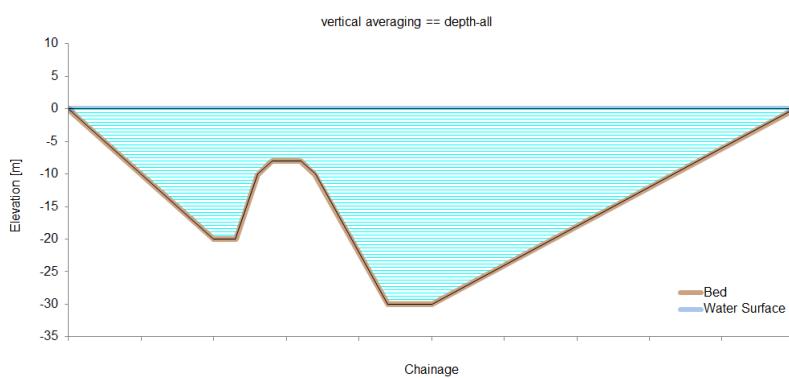
- depth-all – The ‘depth-all’ command vertically averages over the entire water column. Example ‘depth-all’ output block commands and a conceptual illustration are provided below.

```
output == points
output points file == ..\geo\points\output_points.csv
vertical averaging == depth-all
suffix == depth_all
output parameters == V
output interval == 900
end output

output == datv
vertical averaging == depth-all
suffix == depth_all
output parameters == V
output interval == 900
end output
```

!Output type
!Output points file
!Vertical averaging type, entire water column
!Output file identifier suffix
!Output parameters
!Output interval in seconds

!Output type (datv or netcdf)
!Vertical averaging type, entire water column
!Output file identifier suffix
!Output parameters
!Output interval in seconds



- depth-range – The ‘depth-range’ command vertically averages between specified minimum and maximum absolute depths measured downward from water surface. Example ‘depth-range’ output block commands and a conceptual illustration are provided below.

```

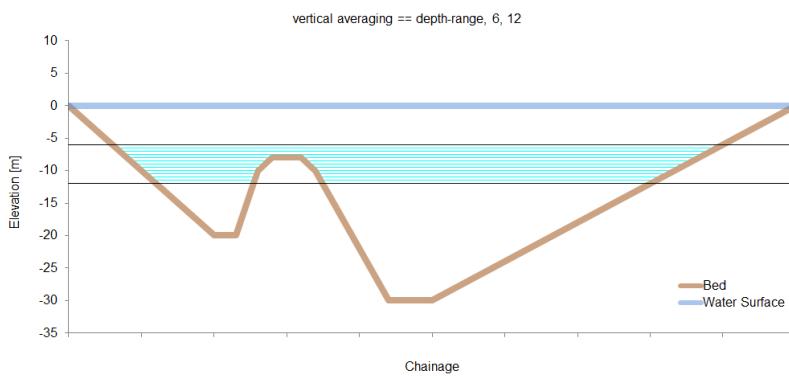
output == points
  output points file == ..\geo\points\output_points.csv
  vertical averaging == depth-range, 6, 12
  suffix == depth_6_12
  output parameters == V
  output interval == 900
end output

!Output type
!Output points file
!Vertical averaging type, absolute depth range minimum and maximum
!Output file identifier suffix
!Output parameters
!Output interval in seconds

output == datv
  vertical averaging == depth-range, 6, 12
  suffix == depth_6_12
  output parameters == V
  output interval == 900
end output

!Output type (datv or netcdf)
!Vertical averaging type, absolute depth range minimum and maximum
!Output file identifier suffix
!Output parameters
!Output interval in seconds

```



- height-range – The ‘height-range’ command vertically averages between specified minimum and maximum absolute heights measured upward from the bed. Example ‘height-range’ output block commands and a conceptual illustration are provided below.

```

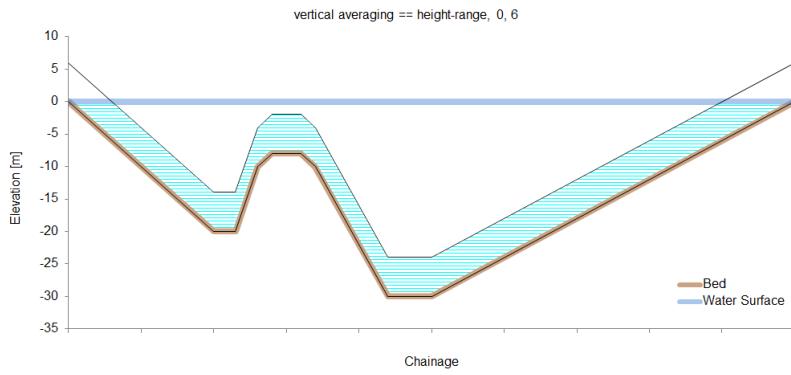
output == points
  output points file == ..\geo\points\output_points.csv
  vertical averaging == height-range, 0, 6
  suffix == height_0_6
  output parameters == V
  output interval == 900
end output

!Output type
!Output points file
!Vertical averaging type, absolute height range minimum and maximum
!Output file identifier suffix
!Output parameters
!Output interval in seconds

output == datv
  vertical averaging == height-range, 0, 6
  suffix == height_0_6
  output parameters == V
  output interval == 900
end output

!Output type (datv or netcdf)
!Vertical averaging type, absolute height range minimum and maximum
!Output file identifier suffix
!Output parameters
!Output interval in seconds

```



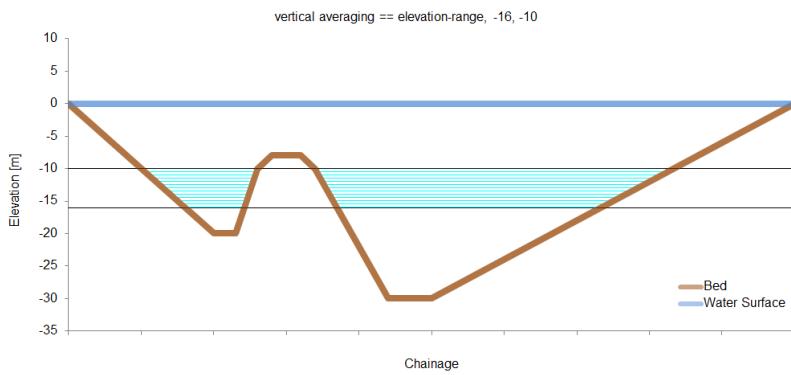
- elevation-range – The ‘elevation-range’ command vertically averages between specified minimum and maximum elevations relative to model vertical datum. Example ‘elevation-range’ output block commands and a conceptual illustration are provided below.

```

output == points
  output points file == ..\geo\points\output_points.csv
  vertical averaging == elevation-range, -16, -10
  suffix == elevation_-16_-10
  output parameters == V
  output interval == 900
end output

output == datv
  vertical averaging == elevation-range, -16, -10
  suffix == elevation_-16_-10
  output parameters == V
  output interval == 900
end output

```



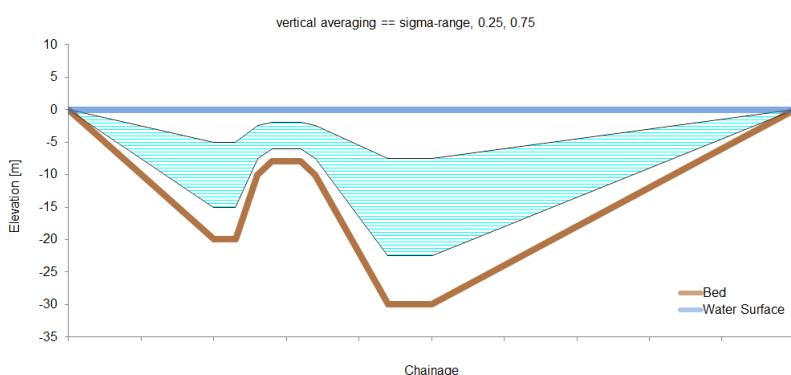
- sigma-range – The ‘sigma-range’ command vertically averages between specified decimal fractions of the water column where 0 is the bed and 1 is the water surface. Example ‘sigma-range’ output block commands and a conceptual illustration are provided below.

```

output == points
  output points file == ..\geo\points\output_points.csv
  vertical averaging == sigma-range, 0.25, 0.75
  suffix == sigma_0.25_0.75
  output parameters == V
  output interval == 900
end output

output == datv
  vertical averaging == sigma-range, 0.25, 0.75
  suffix == sigma_0.25_0.75
  output parameters == V
  output interval == 900
end output

```

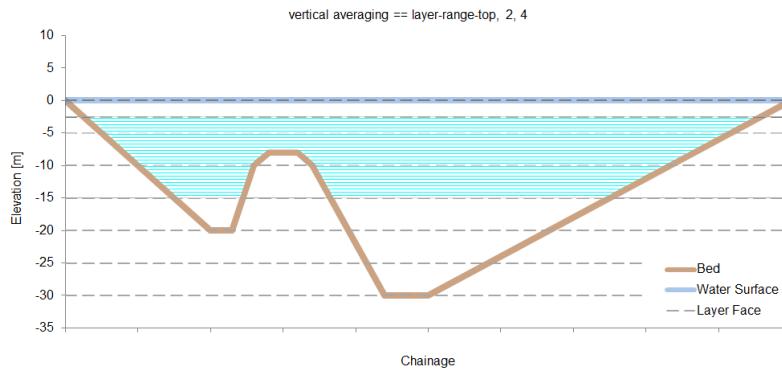


- **layer-range-top** – The ‘layer-range-top’ command vertically averages between layers referenced from the water surface (i.e. surface layer is 1, positive downwards). Example ‘layer-range-top’ output block commands and a conceptual illustration are provided below. Single layer output can be obtained by specifying an equal minimum and maximum.

```

output == points
  output points file == ..\geo\points\output_points.csv
  vertical averaging == layer-range-top, 2, 4
  suffix == layer_top_2_4
  output parameters == V
  output interval == 900
end output

output == datv
  vertical averaging == layer-range-top, 2, 4
  suffix == layer_top_2_4
  output parameters == V
  output interval == 900
end output
  
```

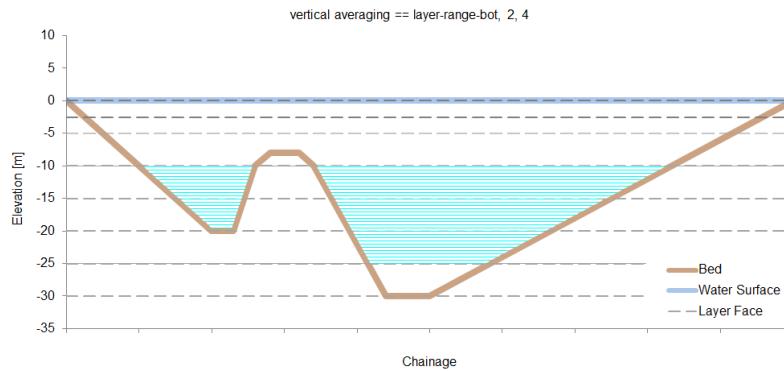


- **layer-range-bot** – The ‘layer-range-bot’ command vertically averages between layers referenced from the water surface (i.e. bottom layer is 1, positive upwards). Example ‘layer-range-bot’ output block commands and a conceptual illustration are provided below. Single layer output can be obtained by specifying an equal minimum and maximum.

```

output == points
  output points file == ..\geo\points\output_points.csv
  vertical averaging == layer-range-bot, 2, 4
  suffix == layer_bot_2_4
  output parameters == V
  output interval == 900
end output

output == datv
  vertical averaging == layer-range-bot, 2, 4
  suffix == layer_bot_2_4
  output parameters == V
  output interval == 900
end output
  
```



With the exception of depth-all, the vertical averaging type command must be followed by the minimum and maximum limits. For example, the commands for sigma vertical averaging over the top 25% of the water column:

```
output == datv
vertical averaging == sigma-range, 0,0.25      !top 25% of water column
suffix == sigma_0_0.25
output parameters == V
output interval == 1800.
end output
```

Or averaging over the bottom 2m measured upward from the bed:

```
output == datv
vertical averaging == height-range, 0,2      !bottom 2m measured from the bed
suffix == height_0_2
output parameters == V
output interval == 1800.
end output
```

In these examples the [suffix](#) command is used to distinguish between output files. This is particularly important when a simulation control file specifies multiple vertically averaged outputs.

Appendix B Advection Dispersion fvc Commands

AD SIMULATION CONFIGURATION COMMANDS

Atmospheric update dt	Longwave radiation albedo	Initial condition OGCM
Bulk latent heat coefficient	Longwave radiation model	
Bulk momentum transfer coefficient	Meteorological sensor height	
Bulk sensible heat coefficient	Ntracer	
Decay rate	Reference density	
Density air	Reference salinity	
Diffusivity limiter dt	Reference temperature	
Equation of state	Scalar mixing model	
Global horizontal scalar diffusivity	Settling velocity	
Global horizontal scalar diffusivity limits	Shortwave radiation albedo	
Global vertical scalar diffusivity	Shortwave radiation bed absorption	
Global vertical scalar diffusivity limits	Shortwave radiation extinction coefficients	
Heat relax dt	Shortwave radiation fractions	
Include heat	Shortwave radiation model	
Include salinity	Specific heat air	
Include temperature	Specific heat water	
Initial salinity	Tracer	
Initial scalar profile	Transport Mode Depth	
Initial temperature	Water emissivity	
Initial tracer concentration	See also:	
Latent heat model	Initial condition 2d	
	Initial condition 3d	

Atmospheric update dt == <timestep (s)>

(Optional)

Specifies the timestep for performing atmospheric parameter updating, if not specified atmospheric parameter updating occurs every hydrodynamic model timestep.

Bulk latent heat coefficient == <CLN>

(Optional, Default == 0.0013)

Bulk aerodynamic latent heat transfer coefficient under neutral conditions.

Bulk momentum transfer coefficient == <CDN>

(Optional, Default == 0.0013)

Bulk momentum transfer coefficient under neutral conditions.

Bulk sensible heat coefficient == <CSN>

(Optional, Default == 0.0013)

Bulk aerodynamic sensible heat transfer coefficient under neutral conditions.

Decay Rate == <Kd (units/day)>

(Optional)

Tracer block command to specify the scalar decay rate in concentration units/day. This results in a sink term flux, S :

$$S = K_d Ch$$

where C is the scalar concentration and h is the flow depth.

Density Air == <Air Density (kg/m³)>

(Optional; Default == 1.2)

Allows the user to specify the density of air used in atmospheric heat calculations. Please refer to Section 7.3 for further information on reference values.

Equation of state == <UNESCO; Direct>

(Optional; Default == UNESCO)

Sets the model for calculating the density of water in baroclinic simulations:

- UNESCO: use the UNESCO equation of state (Fofonoff and Miller, 1983).
- Direct: A tracer can be applied as a direct proxy for density. This approach is generally not recommended but can be useful for conditions where the user requires control over modelled

density. If using this option please contact support@tuflow.com to discuss your problem further and we can advise on its suitability for your specific modelling scenario.

Global Horizontal Scalar Diffusivity == <diffusivity (m²/s) ; coefficient/s (-)>

(Optional)

Globally sets the horizontal diffusivity (m²/s) or diffusivity model coefficients. This is dependent on the scalar mixing model set using the [scalar mixing model](#) command:

- Constant: specify a constant isotropic scalar diffusivity; Default == 0.
- Smagorinsky: specify the Smagorinsky coefficient; Default == 0 (typical value is 0.2)
- Elder: specify longitudinal and transverse coefficients – calculates a non-isotropic diffusivity; typically only used for 2D simulations; Default == 0, 0 (typical values 100, 10)

See [scalar mixing model](#) command to set scalar mixing turbulence model.

Global Horizontal Scalar Diffusivity Limits == <min diffusivity (m²/s)>, <max diffusivity (m²/s)>

(Optional)

For use with Smagorinsky or Elder [scalar mixing model](#), globally sets the minimum and maximum horizontal scalar diffusivity (m²/s) limits.

Not applicable if a Constant isotropic scalar diffusivity is set using the [global horizontal scalar diffusivity](#) command.

See [scalar mixing model](#) command to set scalar mixing turbulence model.

Global Vertical Scalar Diffusivity Limits == <min diffusivity (m²/s)>, <max diffusivity (m²/s)>

(Optional)

Command for use with Parametric or External [vertical mixing model](#). Used to control the vertical diffusion limits (m²/s) i.e. to increase/decrease the effective vertical mixing if there is insufficient or too much vertical diffusion occurring in sections of your model. Please refer to Section 7.2.2.2 for more information. See also [vertical scalar diffusivity limits](#).

Heat relax dt == <timestep (hour)>

(Optional, Default == 0)

Specifies the heat relaxation timestep in hours.

Include heat == <0;1>

(Optional, Default == 0)

Optional command to include atmospheric heat calculations:

-
- 0 = False (i.e. atmospheric heat calculations not included)
 - 1 = True (i.e. atmospheric heat calculations included)
-

Include salinity == <0;1, 0;1>

(Optional 3D; Default == 0,0)

Flag to specify salinity as a modelled parameter:

- 0 = False (i.e. salinity is not modelled).
- 1 = True (i.e. salinity is modelled).

The second flag specifies whether density is a function of the modelled salinity:

- 0 = False (i.e. density is not a function of the modelled salinity).
 - 1 = True (i.e. density is a function of the modelled salinity).
-

Include temperature == <0;1, 0;1>

(Optional 3D; Default == 0,0)

Flag to specify temperature as a modelled parameter:

- 0 = False (i.e. temperature is not modelled).
- 1 = True (i.e. temperature is modelled).

The second flag specifies whether density is a function of the modelled temperature:

- 0 = False (i.e. density is not a function of the modelled temperature).
 - 1 = True (i.e. density is a function of the modelled temperature).
-

Initial Salinity == <salinity (psu)>

(Optional; Default == 0.0)

Globally sets the initial salinity for simulations including baroclinic terms.

Initial Scalar Profile == <initial condition file (.CSV)>

(Optional 3D)

Command used in conjunction with [initial condition 2d](#) to specify a .CSV file that describes the initial scalar profile.

The .CSV file should contain two columns:

- The first column is the depth reference.
- The second column is the concentration at the corresponding depth reference.

If salinity, temperature are included in the simulation they should also be specified in the .CSV file (e.g. Sal, Temp, Scal_1,...). An example of the command usage and corresponding .CSV file is given below:

Initial condition 2d == ..\bc\initial_scalar_profile_001.CSV

and the contents of initial_conditions.CSV:

```
DEPTH, SAL, TEMP, SCAL_1
0.0, 30, 20, 100
2.0, 32, 19, 50
2.1, 35, 17, 25
9999.0, 35, 17, 25
```

Initial Temperature == <temperature (degrees Celsius)>

(Optional; Default == 0.0)

Globally sets the initial temperature for simulations including baroclinic terms.

Initial tracer concentration == <t_1, ..., t_Nwq> (parts/unit volume)

(Optional)

Globally sets the initial tracer concentration fields.

Latent heat model == <LHmodel>

(Optional, Default == 1)

Latent heat transfer model:

- 1 = Vapour pressure is calculated by the Magnus-Tetens formula (TVA, 1972) – requires air temperature and relative humidity inputs
- 2 = Vapour pressure is calculated via modified equations based on Lowe (1977) and Reed (1977) – requires air temperature and cloud cover inputs

Longwave radiation albedo == <alb_lw>

(Optional, Default == 0.03)

Mean long wave radiation albedo at the equator.

Longwave radiation model == <LWinput>

(Optional, Default == 2)

Long wave radiation heat transfer model:

-
- 1 = Net long wave radiation (accounting for both the incident long wave radiation and long wave radiation emitted by the water surface) – requires net downward long wave radiation input
 - 2 = Incident long wave radiation with long wave radiation albedo and water surface reflection are calculated following TVA (1972). Long wave radiation emitted by the water surface is calculated assuming the Stefan-Boltzmann law – requires incident downward long wave radiation input
 - 3 = Incident long wave radiation is calculated assuming the Stefan-Boltzmann law with a correction for cloud cover following TVA (1972) - requires incident downward long wave radiation and cloud cover inputs
 - 4 = Incident long wave radiation with corrections for cloud cover and the long wave radiation emitted by the water surface due to the air/water temperature difference following Zillman (1972) - requires incident downward long wave radiation, cloud cover and air temperature inputs
 - 5 = Based on air temperature and vapour pressure (Chapra, 2008) – requires air temperature and relative humidity inputs

Meteorological sensor height == <meters>

(Optional, Default == 10.0)

Meteorological sensor height.

Ntracer == <number of tracers>

(Optional, Default == 0)

Command used to specify the number of tracers in an AD simulation. The properties of each tracer are defined using the [tracer](#) block commands. Up to 100 tracers can be specified.

Reference Salinity == <Salinity (PSU)>

(Optional; Default == 0.0)

Optionally sets the model reference salinity for simulations including baroclinic terms.

Reference Density == <Density (kg/m³)>

(Optional; Default == 1000.0)

Sets the reference density of water value used in calculation of the baroclinic pressure terms.

Please refer to Section 7.3 for further information on reference values.

Reference Salinity == <Salinity (psu)>

(Optional; Default == 0.0)

Sets the reference density of water value used in calculation of the baroclinic pressure terms.

Please refer to Section 7.3 for further information on reference values.

Reference Temperature == <Temperature (°C)>

(Optional; Default == 20.0)

Optionally sets the model reference temperature for simulations including baroclinic terms.

Please refer to Section 7.3 for further information on reference values.

Scalar mixing model == <None; Constant; Smagorinsky; Elder; Warmup>

(Optional; Default == None)

Sets the scalar mixing calculation method. See also [global horizontal scalar diffusivity](#).

- None: horizontal scalar mixing is not represented.
- Constant: specify a constant isotropic scalar diffusivity using the [global horizontal scalar diffusivity](#) command.
- Smagorinsky: the horizontal non-isotropic scalar diffusivity is calculated according to the Smagorinsky model - specify the Smagorinsky coefficient using the [global horizontal scalar diffusivity](#) command.
- Elder: the horizontal non-isotropic scalar diffusivity is calculated according to the Elder model - specify the longitudinal and transverse coefficients using the [global horizontal scalar diffusivity](#) command.
- Warm up: can be used for initialising scalar distribution (diffusivity is set to maximum within [CFL](#) stability constraints).

Settling Velocity == < w_{s0} (m/s)>

(Optional)

[Tracer](#) block command to specify the scalar settling velocity in m/s. This results in a sink term flux, S :

$$S = -w_{s0}C$$

where C is the scalar concentration.

Shortwave radiation albedo == <alb swo>

(Optional, Default == 0.08)

Mean short wave radiation albedo at the equator.

Shortwave radiation bed absorption == <Bed abs>

(Optional, Default == 0.9)

Rate of light absorption by sediments.

Shortwave radiation extinction coefficients == < NIR eta, PAR eta, UVA eta, UVB eta>

(Optional, Default == 1.0m^{-1} , 0.25m^{-1} , 1.0m^{-1} , 2.5m^{-1})

Extinction coefficient of near-infrared (NIR) in short wave radiation.

Extinction coefficient of photosynthetically active radiation (PAR) in short wave radiation.

Extinction coefficient of ultraviolet A (UVA) in short wave radiation.

Extinction coefficient of ultraviolet B (UVB) in short wave radiation.

Shortwave radiation fractions == <NIR frac, PAR eta , UVA frac, <UVB frac>>

(Optional, Default == 0.43, 0.52, 0.048, 0.002)

Fraction of near-infrared (NIR) in short wave radiation.

Fraction of photosynthetically active radiation (PAR) in short wave radiation.

Fraction of ultraviolet A (UVA) in short wave radiation.

Fraction of ultraviolet B (UVB) in short wave radiation.

Shortwave radiation model == <SWinput>

(Optional, Default == 1)

Short wave radiation heat transfer model

- 1 = Incident short wave radiation estimated according to Jacquet (1983) – requires downward short wave radiation input
- 2 = Incident short wave radiation under clear sky estimated according to Zillman (1972) with cloud cover correction factor given by Reed (1977) – requires air temperature and cloud cover inputs

Specific heat air == <Air Specific Heat (J/kg/°C)>

(Optional, Default == 1005.0)

Specific heat capacity of dry air at 25°C. Please refer to Section 7.3 for further information on reference values.

Specific heat water == <Water Specific Heat (J/kg/°C)>

(Optional, Default == 4181.3)

Specific heat capacity of water at 25°C. Please refer to Section 7.3 for further information on reference values.

Tracer == <tracer id #>

...
...
...

end tracer

(Optional)

This command indicates the beginning of a tracer properties block, specifying the tracer id # that the properties should be applied to. Tracer properties are listed in the following rows and the ‘end tracer’ command is used to indicate the end of the tracer block.

Tracer properties include:

- [Settling Velocity](#)
- [Decay Rate](#)

Example Tracer Block:

```
Ntracer == 2
! Start Tracer Blocks
tracer == 1           ! Open Tracer Block for Tracer ID #1
settling velocity == 2.0e-5
decay rate == 0.05
end tracer          ! Close Tracer Block for Tracer ID #1
tracer == 2           ! Open Tracer Block for Tracer ID #2
settling velocity == 1.0e-5
decay rate == 0.05
end tracer          ! Close Tracer Block for Tracer ID #2
! End Tracer Blocks
```

Transport Mode Depth == <depth (m)>

(Optional)

An optional command to set the threshold water depth for transport mode AD calculations. AD calculations are not undertaken in areas where the depth is less than the threshold value.

Water emissivity == <EPS w>

(Optional, Default == 0.96)

Emissivity of the water surface

Appendix C Water Quality fvc Commands

WQ SIMULATION CONFIGURATION COMMANDS

[Cell WQ depth](#)

[Disable water quality model](#)

[External water quality
model dir](#)

[Initial WQ concentration](#)

[Water quality model](#)

[WQ update dt](#)

Cell WQ depth == <depth (m)>

(Optional)

An optional command to set the threshold water depth for water quality calculations. Water quality calculations are not undertaken in areas where the depth is less than the threshold value.

Disable water quality model == <0;1>

(Optional; Default == 0)

Optional command used to disable an external water quality model:

- 0 = False (i.e. if specified, external water quality model calculations are enabled).
- 1 = True (i.e. if specified, external water quality model calculations are disabled).

Disabling the water quality model can be a useful diagnostic tool if trying to troubleshoot a coupled TUFLOW FV/external water quality run.

External water quality model dir == <path>

(Optional)

Optional command to specify the directory for external [water quality model](#) definition files if an external water quality model is used. If not specified, external water quality model model files must be located in the same directory at the simulation control file.

Initial wQ concentration == <wq_1, ..., wq_Nwq>

(Optional)

Globally sets the initial water quality scalar concentration fields.

Water quality model == <external>

(Optional)

Optional command to link an external water quality model with TUFLOW FV such as [AED²](#).

WQ update dt == <timestep (s)>

(Optional, Default == 900)

Specifies the timestep for performing water quality parameter updating.

Appendix D TUFLOW FV NetCDF Output Format

The dimensions, variable definitions and attributes of a TUFLOW FV NetCDF 3D output file are provided below. This information is intended to assist advanced users wishing to develop functions and scripts to post-process and/or view TUFLOW FV output using a numerical analysis package with a NetCDF library interface (such as MATLAB, R, GNU Octave or Python NumPy).

```

Source: C:\TUFLOWFV\output\TUFLOWFV_NetCDF_3d_output.nc
Format: 64bit
Global Attributes:
    Origin      = 'Created by TUFLOWFV'
    Type        = 'Cell-centred TUFLOWFV output'
    spherical   = 'true'
Dimensions:
    NumCells2D      = 38839
    NumCells3D      = 386802
    NumVert2D       = 36790
    NumVert3D       = 378581
    MaxNumCellVert  = 4
    NumLayerFaces3D = 425641
    NumSedFrac     = 1
    Time           = 13441 (UNLIMITED)
Variables:
    RestTime
        Size:          13441x1
        Dimensions:   Time
        Datatype:      double
        Attributes:
            long_name = 'output time relative to 01/01/1990
00:00:00'
            units       = 'hours'
    cell_Nvert
        Size:          38839x1
        Dimensions:   NumCells2D
        Datatype:      int32
        Attributes:
            long_name = 'Cell number of vertices'
    cell_node
        Size:          4x38839
        Dimensions:   MaxNumCellVert, NumCells2D
        Datatype:      int32
        Attributes:
            long_name = 'Cell node connectivity'
    NL
        Size:          38839x1
        Dimensions:   NumCells2D
        Datatype:      int32
        Attributes:
            long_name = 'Number of layers in profile'
    idx2
        Size:          386802x1
        Dimensions:   NumCells3D
        Datatype:      int32
        Attributes:
            long_name = 'Index from 3D to 2D arrays'
    idx3
        Size:          38839x1
        Dimensions:   NumCells2D
        Datatype:      int32
        Attributes:
            long_name = 'Index from 2D to 3D arrays'
    cell_X
        Size:          38839x1

```

```
Dimensions: NumCells2D
Datatype:    single
Attributes:
    long_name = 'Cell Centroid X-Coordinate'
    units      = 'm'

cell_Y
Size:        38839x1
Dimensions:  NumCells2D
Datatype:    single
Attributes:
    long_name = 'Cell Centroid Y-Coordinate'
    units      = 'm'

cell_Zb
Size:        38839x1
Dimensions:  NumCells2D
Datatype:    single
Attributes:
    long_name = 'Cell Bed Elevation'
    units      = 'm'

cell_A
Size:        38839x1
Dimensions:  NumCells2D
Datatype:    single
Attributes:
    long_name = 'Cell Area'
    units      = 'm^2'

node_X
Size:        36790x1
Dimensions:  NumVert2D
Datatype:    single
Attributes:
    long_name = 'Node X-Coordinate'
    units      = 'm'

node_Y
Size:        36790x1
Dimensions:  NumVert2D
Datatype:    single
Attributes:
    long_name = 'Node Y-Coordinate'
    units      = 'm'

node_Zb
Size:        36790x1
Dimensions:  NumVert2D
Datatype:    single
Attributes:
    long_name = 'Node Bed Elevation'
    units      = 'm'

layerface_Z
Size:        425641x13441
Dimensions:  NumLayerFaces3D,Time
Datatype:    single
Attributes:
    long_name = 'Layer Face Z-Coordinates'
    units      = 'm'

stat
Size:        38839x13441
Dimensions:  NumCells2D,Time
Datatype:    int32
```

```
Attributes:
    long_name = 'Cell wet/dry status'
    units      = 'boolean'

H
Size:          38839x13441
Dimensions:   NumCells2D,Time
Datatype:     single
Attributes:
    long_name = 'water surface elevation'
    units      = 'm'

V_x
Size:          386802x13441
Dimensions:   NumCells3D,Time
Datatype:     single
Attributes:
    long_name = 'x_velocity'
    units      = 'm s^-1'

V_y
Size:          386802x13441
Dimensions:   NumCells3D,Time
Datatype:     single
Attributes:
    long_name = 'y_velocity'
    units      = 'm s^-1'

W
Size:          386802x13441
Dimensions:   NumCells3D,Time
Datatype:     single
Attributes:
    long_name = 'vertical velocity'
    units      = 'm s^-1'

SAL
Size:          386802x13441
Dimensions:   NumCells3D,Time
Datatype:     single
Attributes:
    long_name = 'salinity'
    units      = 'psu'

TEMP
Size:          386802x13441
Dimensions:   NumCells3D,Time
Datatype:     single
Attributes:
    long_name = 'temperature'
    units      = 'degrees celsius'
```

Appendix E Common NetCDF Input File Example Format

The dimensions, variable definitions and attributes of common TUFLOW FV NetCDF input files are provided below. This information is intended to assist users wishing to apply temporally and spatially varying boundary conditions. Two examples are provided:

1. SWAN wave model output
2. NCEP Re-analysis II atmospheric data
3. Curtain Boundary Example
4. Profile Boundary Example

1. SWAN Wave Model NetCDF Output Boundary Condition Example

Wave forcing is often important when simulating the advection-diffusion of sediments or water-borne constituents in estuarine or coastal environments. Mapped, time-varying SWAN wave model output in NetCDF format can be used as a boundary condition for a subsequent TUFLOW FV advection-diffusion simulation. The dimensions, variable definitions and attributes of a SWAN NetCDF output file are provided below and would typically contain the following variables:

- significant wave height (hs)
- surface peak period ‘smoothed’ (tps)
- surface mean direction (theta0)
- x-component of the wave induced force (xforce)
- y-component of the wave induced force (yforce)
- near bottom orbital velocity (ubot)
- near bottom period (tmbot)

The TUFLOW FV boundary condition block commands to read the SWAN NetCDF output file and include the parameters in the advection-diffusion calculations are:

```
grid definition file == ..\bc\waves\SWAN_output.nc
grid variables == longitude, latitude
grid definition label == SWAN_wavesRegional

bc == wave, SWAN_wavesRegional, ..\bc\waves\SWAN_output.nc
bc header == time, hs, tps, theta0, ubot, tmbot, xforce, yforce
bc reference time == 01/01/1970 00:00
bc time units == seconds
bc update dt == 3600
```

```
end bc
```

The variables listed following the ‘bc header’ command should follow the order in the example provided above. Specification of the ‘bc reference time’ and the ‘bc time units’ is crucial since SWAN and TUFLOW FV have differing default time formats. Without this information TUFLOW FV would not apply the SWAN output at the intended time. The ‘bc update dt’ specifies the time interval in seconds between wave field updates, generally consistent with the temporal resolution of the SWAN output.

It is not a requirement for ubot, tmbot, xforce and yforce to be included in the SWAN NetCDF output file. If not specified, these variables will be either approximated by TUFLOW FV or simply not included in the advection-diffusion calculation. As a minimum, the following variables should be specified:

```
bc header == time, hs, tps, theta0
```

In this example, TUFLOW FV would approximate the near bottom orbital velocity (ubot) using the available parameters and following linear wave theory, and the surface peak period (tps) would be applied as the near bottom period (tmbot). If not specified, the wave induced forces (xforce, yforce) are not approximated or used by TUFLOW FV.

In situations where ubot and tmbot are present in the SWAN NetCDF output file but the user wishes to use the TUFLOW FV approximations, the ‘dummy’ command should be used:

```
bc header == time, hs, tps, theta0, dummy, dummy, xforce, yforce
```

Recent versions of SWAN source code and many SWAN binary distributions for Windows support NetCDF model output. More information may be found on the SWAN website:

<http://swanmodel.sourceforge.net/>

SWAN NetCDF output file structure

Source:

C:\TUFLOWFV\bc\waves\SWAN_output.nc

64bit

Global Attributes:

```

Conventions      = 'CF-1.5'
History         = 'Created with agioncmd version 1.2'
Directional convention = 'cartesian'
project          = '001'
run              = '001'

```

Dimensions:

```

time = 18633 (UNLIMITED)
xc   = 251
yc   = 651

```

Variables:

```

time
  Size:        18633x1
  Dimensions: time
  Datatype:    int32
  Attributes:
    units      = 'seconds since 1970-01-01'
    calendar   = 'gregorian'
    standard_name = 'time'
    long_name   = 'time'

longitude
  Size:        251x651
  Dimensions: xc,yc
  Datatype:    single
  Attributes:
    units      = 'degrees_east'
    long_name   = 'longitude'
    standard_name = 'longitude'

latitude
  Size:        251x651
  Dimensions: xc,yc
  Datatype:    single
  Attributes:
    units      = 'degrees_north'
    long_name   = 'latitude'
    standard_name = 'latitude'

hs
  Size:        251x651x18633
  Dimensions: xc,yc,time
  Datatype:    int16
  Attributes:
    units      = 'm'
    standard_name =
'sea_surface_wave_significant_height'
    long_name   = 'hs'
    coordinates = 'longitude latitude'
    _FillValue   = -3.28e+04
    scale_factor = 0.000763
    add_offset   = 25

tps
  Size:        251x651x18633
  Dimensions: xc,yc,time

```

```

Datatype: int16
Attributes:
    units      = 's'
    long_name = 'tps'
    coordinates = 'longitude latitude'
    _FillValue = -3.28e+04
    scale_factor = 0.000763
    add_offset   = 25
theta0
Size: 251x651x18633
Dimensions: xc,yc,time
Datatype: int16
Attributes:
    units      = 'degrees'
    standard_name = 'sea_surface_wave_to_direction'
    long_name = 'theta0'
    coordinates = 'longitude latitude'
    _FillValue = -3.28e+04
    scale_factor = 0.00549
    add_offset   = 180
xforce
Size: 251x651x18633
Dimensions: xc,yc,time
Datatype: int16
Attributes:
    units      = 'N m-2'
    long_name = 'x-component of wave driven force
per unit surface area'
    coordinates = 'longitude latitude'
    _FillValue = -3.28e+04
    scale_factor = 0.0305
    add_offset   = 0
yforce
Size: 251x651x18633
Dimensions: xc,yc,time
Datatype: int16
Attributes:
    units      = 'N m-2'
    long_name = 'y-component of wave driven force
per unit surface area'
    coordinates = 'longitude latitude'
    _FillValue = -3.28e+04
    scale_factor = 0.0305
    add_offset   = 0
ubot
Size: 251x651x18633
Dimensions: xc,yc,time
Datatype: int16
Attributes:
    units      = 'm s-1'
    long_name = 'orbital velocity near bottom'
    coordinates = 'longitude latitude'
    _FillValue = -3.28e+04
    scale_factor = 0.000229
    add_offset   = 7.5
tbot
Size: 251x651x18633
Dimensions: xc,yc,time

```

```
Datatype:    int16
Attributes:
    units          = 's'
    long_name      = 'Bottom wave period'
    coordinates    = 'longitude latitude'
    _FillValue     = -3.28e+04
    scale_factor   = 0.000763
    add_offset     = 25
```

2. NCEP-DOE Reanalysis 2 Atmospheric Data Boundary Condition Example

For TUFLOW FV 3D simulations, the baroclinic pressure-gradient terms can be optionally activated to allow the hydrodynamic solution to respond to temperature, salinity and sediment induced density gradients. In addition, atmospheric (surface) heat exchange calculations can also be included for given standard meteorological parameter inputs.

In the example below mapped, time-varying atmospheric data derived as part of the NCEP-DOE Re-analysis 2 project (website: <http://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis2.html>) were downloaded and post-processed to create a single NetCDF file containing the following surface variables:

- u-component of the 10-minute averaged wind velocity (u)
- v-component of the 10-minute averaged wind velocity (v)
- air temperature (temp)
- relative humidity (rhum)
- downward shortwave solar radiation (dswr)
- downward long-wave non-penetrative radiation (dlwr)
- precipitation (rain)

The TUFLOW FV boundary condition block commands to read the NetCDF file containing the NCEP-DOE Re-analysis 2 data and include the parameters in the heat exchange calculations are:

```

grid definition file == ..\bc\atmospheric\NCEP_DOE_R2_data.nc
grid variables == lon, lat
grid definition label == NCEP

bc == W10_GRID, NCEP, ..\bc\atmospheric\NCEP_DOE_R2_data.nc
  bc header == time,u,v
  bc reference time == 01/01/1990 00:00
  bc time units == hours
  bc update dt == 14400
end bc

bc == AIR_TEMP_GRID, NCEP, ..\bc\atmospheric\NCEP_DOE_R2_data.nc
  bc header == time,temp
  bc reference time == 01/01/1990 00:00
  bc time units == hours
  bc update dt == 14400

```

```
end bc

bc == REL_HUM_GRID, NCEP, ..\bc\atmospheric\NCEP_DOE_R2_data.nc
bc header == time,rhum
bc reference time == 01/01/1990 00:00
bc time units == hours
bc update dt == 14400
end bc

bc == SW_RAD_GRID, NCEP, ..\bc\atmospheric\NCEP_DOE_R2_data.nc
bc header == time,dswr
bc reference time == 01/01/1990 00:00
bc time units == hours
bc update dt == 14400
end bc

bc == LW_RAD_GRID, NCEP, ..\bc\atmospheric\NCEP_DOE_R2_data.nc
bc header == time,dlwr
bc reference time == 01/01/1990 00:00
bc time units == hours
bc update dt == 14400
end bc

bc == PRECIP_GRID, NCEP, ..\bc\atmospheric\NCEP_DOE_R2_data.nc bc header == time,rain
bc reference time == 01/01/1990 00:00
bc time units == hours
bc update dt == 14400
end bc
```

With the exception of precipitation¹⁵, the variables provided in this example correspond to the inputs required by the TUFLOW FV heat exchange module in default mode. Additional inputs, such as cloud cover, may be required when activating non-default atmospheric module settings. A full description of the heat exchange calculation options is available in the TUFLOW FV Science Manual.

¹⁵ Precipitation is an optional input (assumed freshwater) for baroclinic mode simulations.

2. Meteorological Grid NetCDF Input File Example

```

Source:          C:\TUFLOWFV\bc\atmospheric\NCEP_DOE_R2_data.nc
Format:         classic
Global Attributes:
    origin = 'NCEP Reanalysis'
Dimensions:
    ni      = 11
    nj      = 14
    time   = 1464  (UNLIMITED)
Variables:
    time
        Size:          1464x1
        Dimensions:   time
        Datatype:     double
        Attributes:
            units      = 'hours'
            longname   = 'time in decimal hours since 01/01/1990
00:00'
            reference  = 'AEST'
    lon
        Size:          11x1
        Dimensions:   ni
        Datatype:     single
        Attributes:
            units      = 'degrees'
            longname   = 'longitude'
            projection = 'LL_WGS84'
    lat
        Size:          14x1
        Dimensions:   nj
        Datatype:     single
        Attributes:
            units      = 'degrees'
            longname   = 'latitude'
            projection = 'LL_WGS84'
    u
        Size:          11x14x1464
        Dimensions:   ni,nj,time
        Datatype:     single
        Attributes:
            longname = 'u'
            units    = 'm s^-1'
    v
        Size:          11x14x1464
        Dimensions:   ni,nj,time
        Datatype:     single
        Attributes:
            longname = 'v'
            units    = 'm s^-1'
    temp
        Size:          11x14x1464
        Dimensions:   ni,nj,time
        Datatype:     single
        Attributes:
            longname = 'air temperature'

```

```
                units      = 'degC'
rhum
Size:      11x14x1464
Dimensions: ni,nj,time
Datatype:   single
Attributes:
            longname = 'relative humidity'
            units    = 'percent'
dswr
Size:      11x14x1464
Dimensions: ni,nj,time
Datatype:   single
Attributes:
            longname = 'downward shortwave radiation'
            units    = 'W m^-2'
dlwr
Size:      11x14x1464
Dimensions: ni,nj,time
Datatype:   single
Attributes:
            longname = 'downward longwave radiation'
            units    = 'W m^-2'
rain
Size:      11x14x1464
Dimensions: ni,nj,time
Datatype:   single
Attributes:
            longname = 'precipitation'
            units    = 'm d^-1'
```

3. Curtain Boundary Example

```

Source:          C:\Users\tuflowfvuser\Desktop\Curtain_Example.nc
Format:         classic
Global Attributes:
    Type = 'example curtain BC NetCDF File'
Dimensions:
    x      = 500
    z      = 25
    time   = 3000  (UNLIMITED)
Variables:
    chainage
        Size:      500x1
        Dimensions: x
        Datatype:   single
        Attributes:
            units = 'meters'
    depth
        Size:      25x1
        Dimensions: z
        Datatype:   single
        Attributes:
            units = 'meters'
    time
        Size:      3000x1
        Dimensions: time
        Datatype:   single
        Attributes:
            units = 'hours since 01/01/1990 00:00'
    WL
        Size:      500x3000
        Dimensions: x,time
        Datatype:   single
        Attributes:
            units = 'meters'
    U
        Size:      500x25x3000
        Dimensions: x,z,time
        Datatype:   single
        Attributes:
            units = 'm/s'
    V
        Size:      500x25x3000
        Dimensions: x,z,time
        Datatype:   single
        Attributes:
            units = 'm/s'
    SAL
        Size:      500x25x3000
        Dimensions: x,z,time
        Datatype:   single
        Attributes:
            units = 'PSU'
    TEMP
        Size:      500x25x3000
        Dimensions: x,z,time
        Datatype:   single
        Attributes:
            units = 'degrees Celcius'

```

4. Profile Boundary Example

Source: C:\Users\tuflowfvuser\Desktop\Profile_Example.nc

Format: classic

Global Attributes:

```
Type = 'example profile BC NetCDF File'
```

Dimensions:

```
z      = 25
time  = 3000  (UNLIMITED)
```

Variables:

```
depth
  Size:      25x1
  Dimensions: z
  Datatype:   single
  Attributes:
    units = 'meters'

time
  Size:      3000x1
  Dimensions: time
  Datatype:   single
  Attributes:
    units = 'hours since 01/01/1990 00:00'

WL
  Size:      3000
  Dimensions: time
  Datatype:   single
  Attributes:
    units = 'meters'

U
  Size:      25x3000
  Dimensions: z,time
  Datatype:   single
  Attributes:
    units = 'm/s'

V
  Size:      25x3000
  Dimensions: z,time
  Datatype:   single
  Attributes:
    units = 'm/s'

SAL
  Size:      25x3000
  Dimensions: z,time
  Datatype:   single
  Attributes:
    units = 'PSU'

TEMP
  Size:      25x3000
  Dimensions: z,time
  Datatype:   single
  Attributes:
    units = 'degrees Celcius'
```

