TUFLOW FV
User Manual

Flexible Mesh Modelling

2012 (Build 2012)

www.TUFLOW.com
www.TUFLOW.com/forum
wiki.TUFLOW.com
support@tuflow.com

Contents

TABLE OF CONTENTS

List of Figures

List of Tables

1 Navigating the Manual

1.1 About This Manual

This document is a User Manual for the TUFLOWFV.exe hydrodynamic computational engine. This engine is driven through a Console (DOS) Window and relies on third party software to provide the interface to the user and the engines. These software are typically a text editor (eg. Notepad++), a mesh generator (eg. SMS), result viewing (eg. SMS) and also possibly a GIS platform (eg. MapInfo). Please also refer to the user documentation or help for the third party software you have chosen to use in addition to this manual.

Setting up a TUFLOW FV model generally requires building a flexible mesh, and the quality of the mesh can have a significant influence on model performance. Recognising this, the manual provides guidance for developing a flexible mesh and an example of creating a flexible mesh using our preferred mesh generator, SMS.

1.2 How to Use This Manual

This manual is designed for both hardcopy and digital usage. Section, table and figure references are *hyperlinked* (click on the Section, Table or Figure number in the text to move to the relevant page).

Similarly, text file commands are hyperlinked and accessed through relevant lists (front page and Section 8). There are also command hyperlinks in the text (normally blue and underlined). Command text can be copied and pasted into the text files to ensure correct spelling.

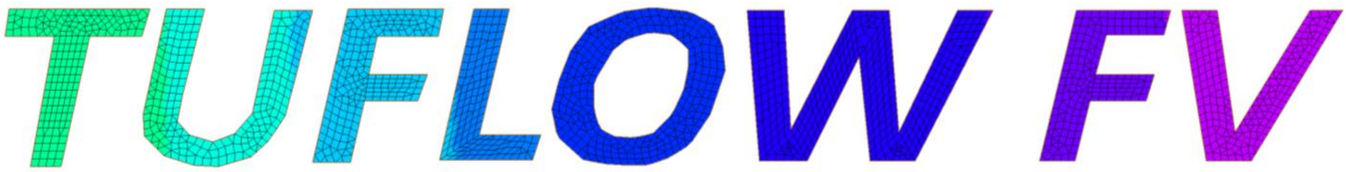
Some useful keys to navigate backwards and forwards are *Alt Left / Right arrow* to go backwards / forwards to the last locations. *Ctrl Home* returns to the front page, which contains useful hyperlinks. Also, *Ctrl End* provides quick access to the end pages, which contain all the hyperlinks to the text file commands.

Any constructive suggestions are very welcome (support@tuflow.com).

1.3 Sections

- **Introduction:** Section 0 provides some basic information about TUFLOW FV and flexible mesh modelling in general. Reading this section should provide a good overall impression of the modelling approach and under what circumstances use of TUFLOW FV is most appropriate.
- **Installation:** Section 3.3 provides the steps needed to install TUFLOW FV on your computer.
- **Running the TUFLOW FV executable:** TUFLOW FV is run using a file called TUFLOWFV.exe. This can be activated in a variety of ways that are described in Section 3.4.
- **Before starting:** Once installed, Section 0 describes a few administrative steps that should be followed prior to running TUFLOW FV, such as naming conventions, how dongles work, etc.
- **The modelling process:** What are the steps needed to setup and run a successful modelling exercise? Section 0 provides an overview of the steps.

- **Quick tutorial on mesh generation and TUFLOW FV model setup:** Section 0 demonstrates the development of a very simple model mesh. Follow the steps performed here and expand upon them to develop more complex, real-world models.
- **Tips, tricks, troubleshooting:** Section 7 contains suggested solutions to commonly encountered problems plus tips to help you make the most of TUFLOW FV. This includes tips on flexible mesh generation.
- **Command file references:** Sections 8 and 0 provide the reference to each command available in TUFLOW FV. Use the lists of available commands in Sections 8.1 and 9.1 as a starting point for navigating the reference. Section 0 provides a description of the 2dm mesh file.
- **References:** The references include those identified in the document, plus additional references where TUFLOW FV has been documented in scientific literature.



2 Introduction

2.1 What is TUFLOW FV?

TUFLOW FV is an engine for performing 2D and 3D hydrodynamic simulations. The model solves the Non-linear Shallow Water Equations (NLSWE) on a flexible mesh using a finite-volume numerical scheme.

Specific features and capabilities:

- Finite volume explicit
  - Fully dynamic
  - Timestep dependent upon (CFL) Courant number
- Flexible mesh
  - More flexibility when designing a mesh
  - Can use a fixed grid if preferred
- Parallelised
  - Can run on multiple processors on a single computer
- Stable numerical scheme
  - Shock capturing capability (stable in supercritical flows, steep gradients, etc)
  - Very stable wetting and drying
- Applications
  - Traditionally coastal and estuarine applications.
  - FV engine perfectly suited to dambreak simulation
- Modules
  - Hydrodynamic, 2D and 3D
  - Advection dispersion, including atmospheric heat balance and density coupling of temperature, salinity and sediment concentration
  - Sediment transport
  - Water quality

2.2 Flexible meshes and mesh generation

TUFLOW FV is a ***flexible mesh model***. Compared to other approaches (using fixed grids, etc) the design of the flexible mesh tends to have a greater influence on model performance. Thus, more time and effort should be spent preparing the model 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 (ie – fine resolution in the area of interest, coarser resolution in the regional extents). Thus, a range of spatial scales can be modelled without resorting to nesting.
- Mesh alignment can fit bathymetric contours and boundary extents, optimising mesh resolution. This is particularly relevant in regions with complex bathymetric features.
- Specific features, such as man-made developments, infrastructure etc can be included in the model mesh precisely.

To exploit these advantages, the mesh needs to be designed carefully and appropriately for the specific model application. There are a number of mesh generators available to construct a model mesh, however BMT uses the SMS package, provided by Aquaveo (see www.aquaveo.com/sms). We use SMS for the following reasons:

- SMS (previously Fasttabs of Brigham Young University) has been a commercially available mesh generation package for decades – it has been extensively tested, improved and adjusted.
- SMS strikes a good balance between manual and automatic mesh generation techniques; in our experience setting up a mesh still needs some manual inputs!

An illustrative comparison of fixed grid vs flexible mesh is shown in Figure 2-1.

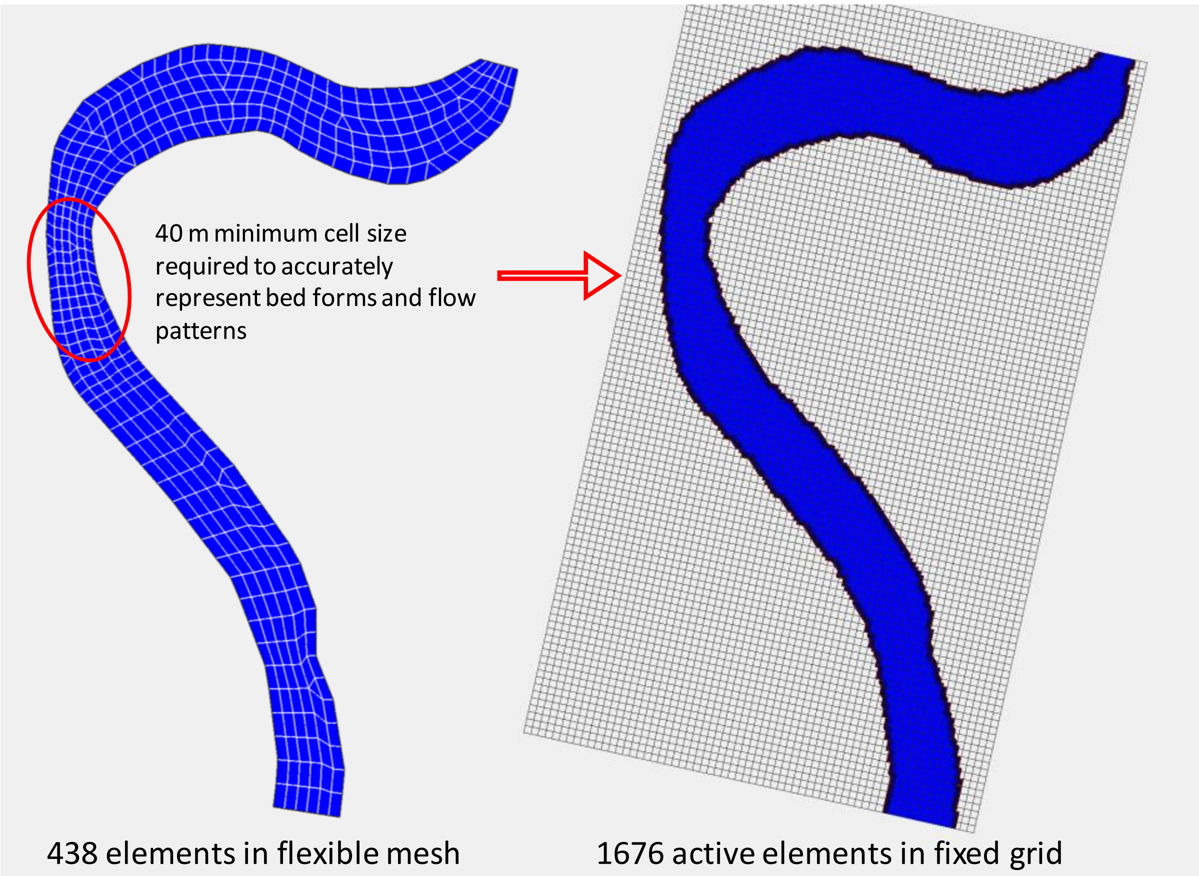


Figure 2-1 Flexible mesh vs fixed grid

The flexible mesh shown has 438 elements with a typical cell size of 40 to 140 m. In the narrowest bend of the river the cells are smaller and elongated (ie longer in the direction of flow, shorter across the channel) – the cross-channel width in this location is the critical cell distance in this situation, because this resolution is necessary to accurately describe the bed forms and flow conditions. Thus, the required minimum cell width must be greater than 40 m.

To achieve a similar degree of accuracy the corresponding fixed grid requires a cell size of 40 m. Within the computational domain (ie in the river channel) there are 1676 cells. This is around a four-fold increase in the number of active cells.

Without parallelisation across multiple cores, the computational performance of a finite volume scheme is slower compared to a finite difference scheme; nevertheless, in situations such as illustrated in Figure 2-1 there are good reasons for opting for a flexible mesh approach.

2.3 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 is not quite linear with the number of threads, as demonstrated in Figure 2-2.

Unless the user decides otherwise, TUFLOW FV will run using the maximum number of threads available to it (limited by the licence). This means that, by default, TUFLOW FV will run as fast as the host computer permits it to.

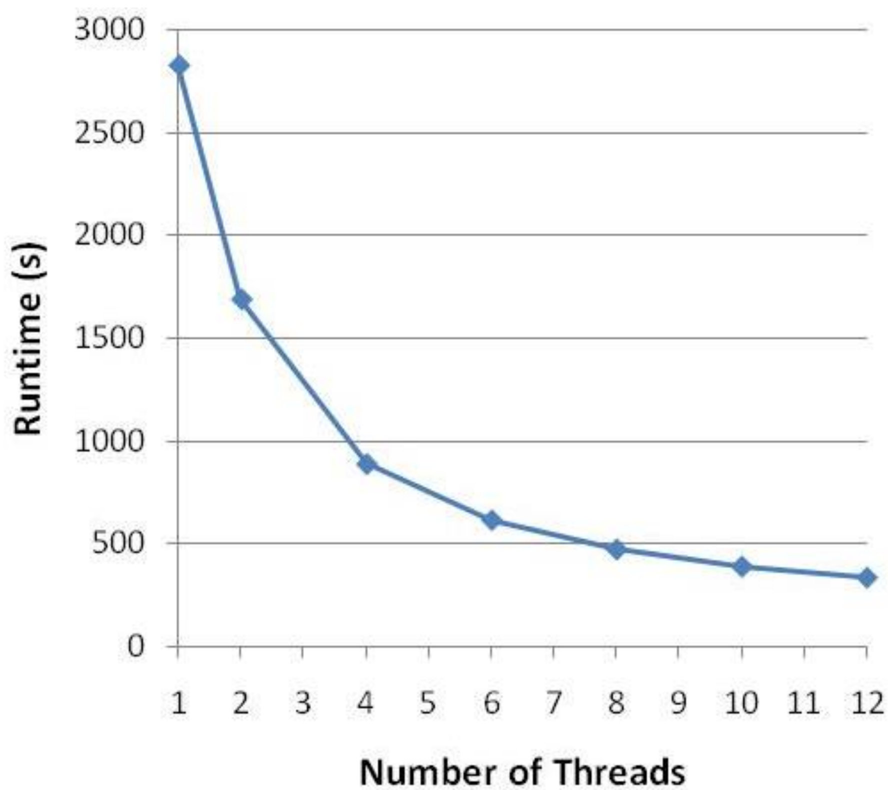


Figure 2-2 Typical runtime decrease / computational speed increase with multi-core processing with TUFLOW FV

2.4 TUFLOW Classic or TUFLOW FV?

Table 2-1 provides a summary of some of the fundamental differences between TUFLOW Classic and TUFLOW FV.

Each have their core applications; TUFLOW has traditionally been applied for floodplain and urban stormwater management and TUFLOW FV has traditionally been applied to coastal and estuarine applications. The available additional modules for each can limit their applicability.

That said, both TUFLOW and TUFLOW FV are broadly applicable to a range of hydrodynamic and related situations. Choice of one over the other depends upon the specific problem to be solved and the modeller's preference and prior experience.

Note that TUFLOW FV can be run using a fixed grid if preferred.

Table 2-1 Comparison of TUFLOW and TUFLOW FV

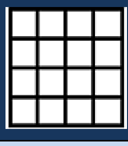
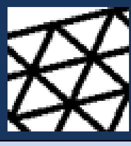
Category	Feature	TUFLOW 	TUFLOW FV 
General	Solution Scheme	Finite difference semi-implicit • Fully dynamic • Timestep not entirely dependent upon Courant number (but wetting and drying + 1d links are explicit features)	Finite volume explicit • Fully dynamic • Timestep dependent upon Courant number
	Domain	Fixed grid • Nesting if higher grid size needed • Diagonal flows can be a pain • 1D link workarounds	Flexible mesh • Needs a modeller to design
	Speed	Computationally efficient	More computationally intensive (but can parallelise)
	Stability	Handles wetting and drying well	Very stable wetting and drying • Shock capturing capability (stable in supercritical flows, steep gradients, etc)
	Geometry	Relatively straightforward to import from DEM • Less manual adjustment, less reliance on modeller to design	More effort required to design the mesh • More flexibility when designing the mesh, more dependence upon modeller to do a good job
	Typical applications	TUFLOW has traditionally been applied for floodplain and urban stormwater management. • But, TUFLOW originally was created for estuarine application	TUFLOW FV has traditionally been applied to coastal and estuarine applications. • But, FV engine is perfectly suited to dambreak simulation
		Both can be applied to a range of applications. **The available additional modules for each limit their application to some extent.**	

Table continued (see note for symbols)

Category	Feature	TUFLOW	TUFLOW FV
Dimensionality	1D / 2D Links	🍏	🚧
	2D	🍏	🍏
	3D	X	🚧
Hydrodynamic (HD)	Structures	🍏	✓🚧
	Pipes and urban drainage	🍏	X
	Precipitation input	🍏	✓
	Wind field input	✓	🍏
	Wave field input	X	🍏
	Links to global ocean circulation models	X	🍏
Advection Dispersion (AD)	Plumes and pollutants	🍏	🍏
	Decay coefficients	🍏	🍏
Sediment Transport (ST)	Mud transport	X	🍏
	Sand transport	🍏	🍏
	Sediment plumes	X	🍏
	Morphological Update (Rivers)	🍏	🚧
	Morphological Update (Coastal)	✓	🚧
Water Quality and Ecology (WQ)	Water Quality processes	🚧	🚧
Other	Emergency flood response and evacuation	🍏	X
Interface	GIS Environment	🍏	✓
	SMS	🍏	🍏
	Xp	🍏	X
	ISIS	🍏	X

For specific features listed there are icons as follows:

🍏 = core capability, feature used frequently, ✓ = able to do within product, feature used less frequently, 🚧 = A feature considered for future releases, X = does not have the capability.

3 Before starting

3.1 TUFLOW FV program

The TUFLOW FV executable, **tuflowfv.exe**, is a command console program. A model is started by calling the executable with the control file (.fvc) as the first and only argument. If no argument is specified the command line will request the user input one. See Section 3.4.

TUFLOW FV is a multi-threaded program based on the OpenMP shared-memory model. It will automatically spawn multi-threaded simulations, where the number of threads can be set by specifying an OMP_NUM_THREADS environment variable. If not explicitly specified the OMP_NUM_THREADS value will be assumed to equal to the NUMBER_OF_PROCESSORS environment variable.

Example:
```
C:\>set OMP_NUM_THREADS=4
C:\>Tuflowfv.exe
```

3.2 TUFLOW FV dongles

Performing TUFLOW FV simulations will require the presence of a suitably licensed hardware lock. TUFLOW FV supports both local license and network license versions of the WIBU codemeter system dongles. An FV dongle will have one or more engine licenses and typically twice as many thread licenses as engines. For instance, a 4 license hardware lock would permit 4 simultaneous simulations utilising 2 threads each, or it would permit 1 simulation utilising 8 threads.

In addition to the basic TUFLOW FV engine license, various optional modules can be licensed via the WIBU codemeter dongles. The number of module licenses can be less than or equal to the number of engine licenses available on a dongle.

Network dongles are also available, which then licences TUFLOW FV simulations across an office network.

3.3 Installing TUFLOW FV

The following description provides a brief overview of installation; a full description is provided by software support when TUFLOW FV is purchased, or is available on the website at http://www.tuflow.com/ProductDownload.aspx?tuffv.

Installing TUFLOW FV is mostly about installing dongle drivers and licence files. As for all TUFLOW products, TUFLOW FV uses a hardware lock (or dongle). This requires a dongle driver on your computer, then a software licence, to run.

The actual model is TUFLOWFV.exe. It doesn’t need to be installed, just placed in a folder on your computer. See Section 3.4. There are also several dll files (dynamic link libraries) which are also placed in the same folder.

The steps to installation are broadly described below. If you have any queries or problems, or have concerns about the steps, please contact support@tuflow.com.

1 Install dongle drivers and hardware dongle

- Both 32 and 64 bit dongle drivers are available on the TUFLOW website:
  - http://www.tuflow.com/ProductDownload.aspx?tuffv
- Once installed you will need a hardware dongle. The TUFLOW support team will provide this.

2 Create licence request

- If the dongle is not licenced you will need to create a licence request. This is a file (extension .WibuCmRaC), which needs to be emailed to us, then is returned with the updated licence file.
- Contact support@tuflow.com for detailed instructions.

3 Update Licence File

- The updated licence file is installed using the Codemeter software installed on your computer.

4 Update TUFLOWFV.EXE Path

- The model itself is called TUFLOWFV.exe. This can be placed in any folder on your computer.

5 Check TUFLOWFV.EXE

- Once the previous steps are completed, check that all is working by double clicking on the TUFLOWFV.EXE executable, then pressing RETURN when prompted for an input file. The licence information should be presented. Any problems, contact support@tuflow.com.

6 Accessing TUFLOWFV.EXE from your project folder

- See Section 3.4.

3.4 Running TUFLOWFV.exe

There are several different ways available to run TUFLOW FV, ranging from simple double-click to advanced batch files.

Keep in mind that for all approaches the executable is a single file "tuflowfv.exe"; the operating system, console program or 3rd party program simply accesses this file with associated command line arguments.

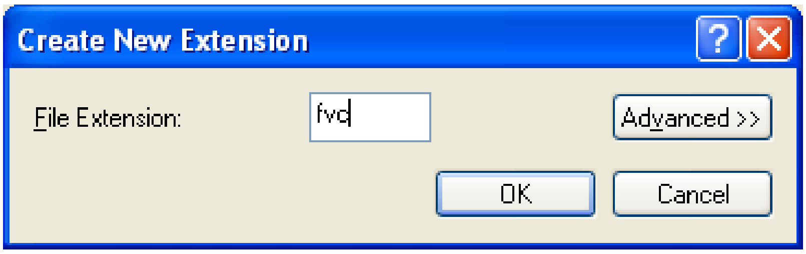
3.4.1 Double click tuflowfv.exe

The TUFLOW FV executable, tuflowfv.exe, is a command console program. A model is started by calling the executable with the control file (.fvc) as the first and only argument. If no argument is specified the command line will request the user input one.

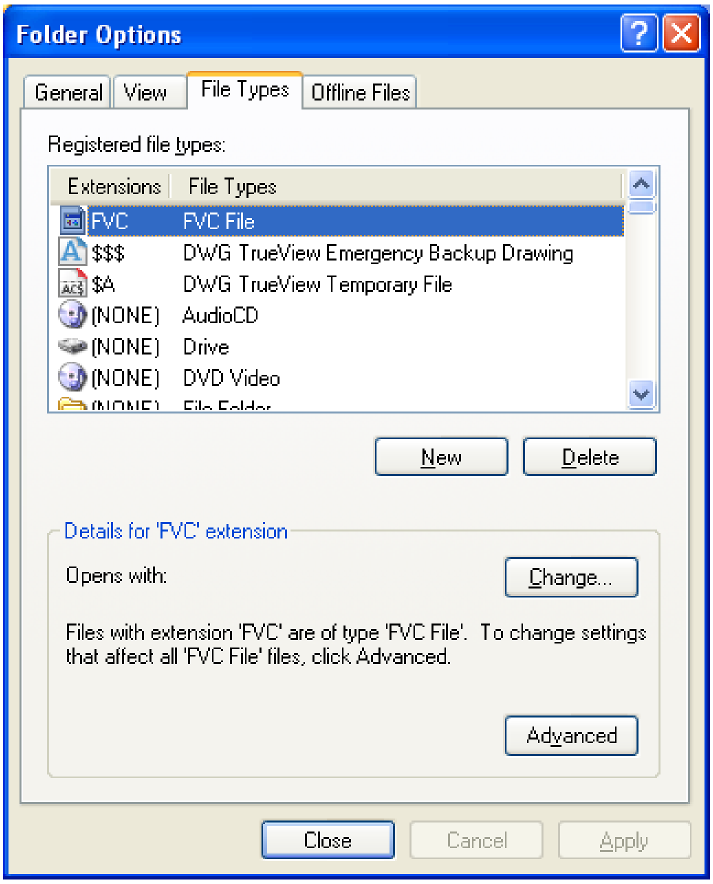
3.4.2 Right Mouse Button in Microsoft Explorer

To start a simulation in Microsoft Explorer by using the right mouse button, first follow the following steps to set up a file association:

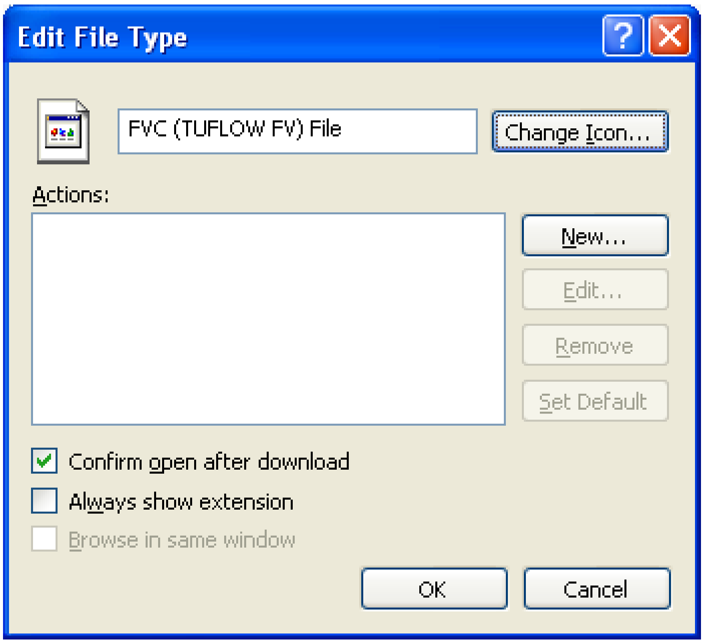
1. In Explorer, open the "View" (or "Tools"), "Folder Options…" menu and select the "File Types" tab. If . fvc files are not in the "Registered file types:" list box, choose "New Type…", otherwise select the .fvc file entry under "Registered file types:" as shown in Step 3 below.
2. If adding a new type, enter in a description (eg. "TUFLOW FV Control File") and "fvc" as the associated extension (see below) and press OK.



3. The Folder Options dialog should appear something like the below.

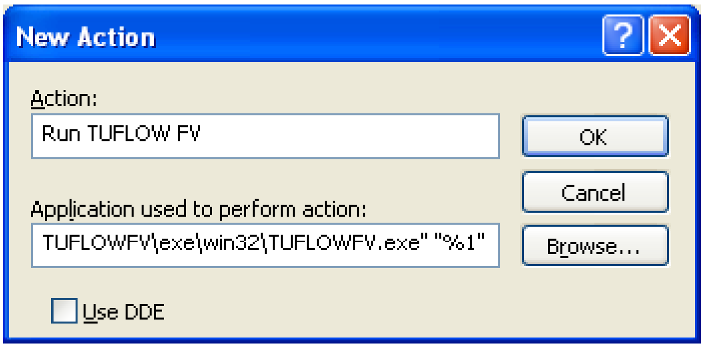


4 Click "Advanced" to bring up the dialog below (you can add a new icon and change the file type description here).



5 Choose "New…" and enter text to describe the "Action:" (eg. "Run TUFLOW FV") – this text appears on the pop-up menu when you click the right mouse button on an .fvc file in Explorer. Enter or use "Browse…" to specify the path to TUFLOWfv.exe; note the need for quotes if the path has any spaces. After "TUFLOWfv.exe", add a space then "%1" including the quotes, as shown below. Choose "OK". The "Application used to perform action:" field should be something like:

"C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe" "%1"



6 The action should now appear in the list under "Actions:". It is not recommended that a "Run TUFLOW FV" action be set as the default action as it is easy to accidentally start a simulation, which instantly overwrites any existing result files. You may wish to set up other associations at this point, for example, to access your preferred editor.

7 Choose "OK" or "Close", then "Close" on the "Folder Options" menu.

8 Check the file association, by clicking the right mouse button on an .fvc file in Windows Explorer. The "Run TUFLOW FV" action should appear in the list.

Once the file association is complete, clicking the right mouse button on an .fvc file in Explorer, and selecting the "Run TUFLOW FV" action, starts a simulation. A Console Command Window opens and TUFLOW FV starts.

3.4.3 From a Console (DOS) Window or "Run"

A single simulation can be started directly from an open Console Window (also called "Command Prompt" in the list of programmes in Windows) 

or from the "start" then "run" commands 

.

For example, at the Console prompt enter:

```
"C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe" run01.fvc
```

You can use the various switches and Windows NT/2000/XP/7 priority settings as discussed in Section 3.4.4 and 3.4.4.1.

3.4.4 Using a Batch File

One or many simulations, and other associated operations, can all be specified within a batch file. The simplest format is to specify each simulation one after another. The following shows the contents of a 4 line batch file (which could be named "TUFLOW FV Simulations.bat"):

```
"C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe" run01.fvc
"C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe" run02.fvc
"C:\Program Files\TUFLOWFV\exe\win32\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.

Comment lines are specified in a .bat file using "#" in the first column, or alternatively a "REM". For example, if you want to re-run only the first simulation in the examples above and include a description for the batch file, edit the file as follows:

```
REM TUFLOW FV Model simulations for demonstration project
REM CFN 09/11/2011
"C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe" run01.fvc
REM "C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe" run02.fvc
#"C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe" run03.fvc
pause
```

3.4.4.1 Changing priority

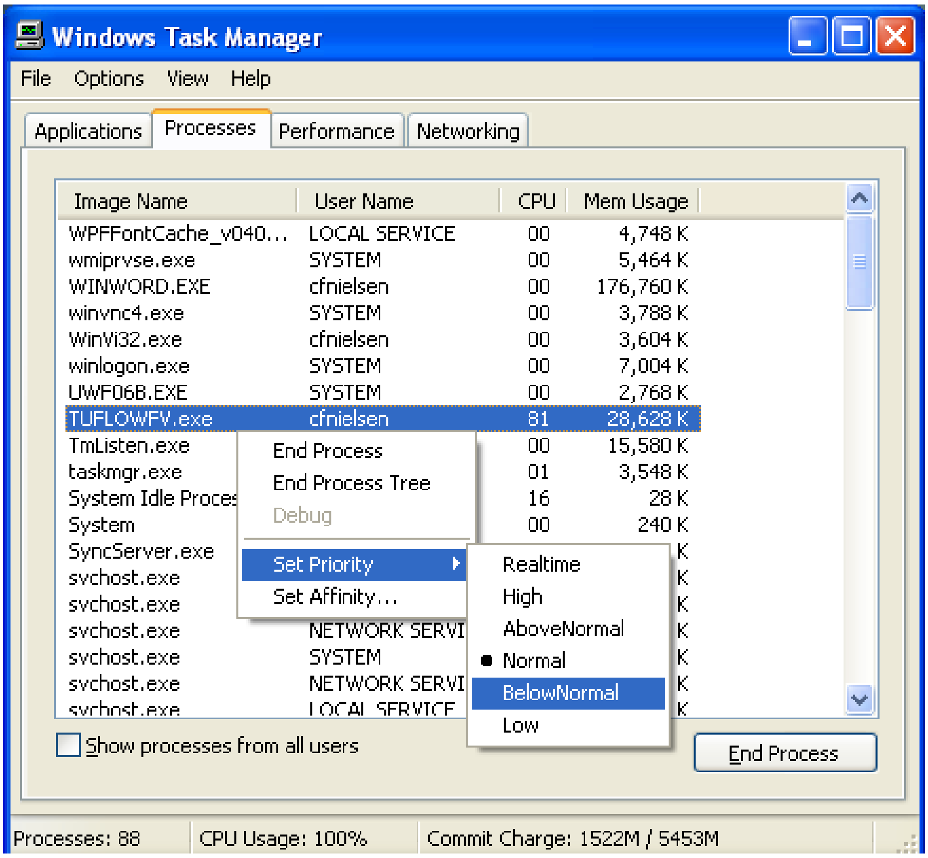
Windows NT/2000/XP/7 can assign a process a different priority level using the Task Manager. This is very useful for running simulations in the "background" without slowing down other computer work you need to do.

3.4.4.2 Manually

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 Processes 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 the TUFLOW process to take over your CPU and you may not able to do much until the simulation is finished.



3.4.4.3 From the batch file

The DOS command "start" is required to execute a simulation at a different priority.

Precede each of the lines in the example in Section 3.4.4 with "`start "TUFLOW FV" /wait /low`" as shown below. This will:

- Initiate a separate Console Window for each simulation
- Give the new console window a title called "TUFLOW FV" (or whatever you choose)
- /wait will ensure that the next line in the batch file is not executed until this line is completed (ie – one simulation after another; without the /wait option all simulations will start at once)
- /low will run the simulation on low priority

```
start “TUFLOW FV” /wait /low “C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe” run01.fvc
start “TFV R2” /wait /low “C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe” run02.fvc
start “Hello World” /wait /low “C:\Program Files\TUFLOWFV\exe\win32\TUFLOWFV.exe”
run03.fvc
pause
```

Other useful switches available are:

- /belownormal and /abovenormal to set these priority levels
- /min to minimise the process once started

3.4.4.4 Advanced .bat Files

Further details on using a batch file are available in the wiki:

- http://wiki.tuflow.com/index.php?title=Run_TUFLOW_From_a_Batch-file

3.4.4.5 Pause, restart and cancel a simulation

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”.

3.4.5 From UltraEdit

The benefits of running TUFLOW FV from UltraEdit 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 is very similar to setting up TUFLOW Classic. This is described in the wiki:

- http://wiki.tuflow.com/index.php?title=Run_TUFLOW_From_UltraEdit

3.4.6 From Notepad++

As for UltraEdit, Notepad++ provides the option to run TUFLOW FV from the editor, either from a shortcut key or from the menu. Follow the instructions in the Notepad++ wiki:

- http://wiki.tuflow.com/index.php?title=NotepadPlusPlus_Run_TUFLOW

3.5 Mesh Development Tools

The “Surfacewater Modelling System” (SMS) by Aquaveo is a powerful environment for developing TUFLOW FV flexible mesh models and visualising model results. A trial version of SMS can be downloaded from www.aquaveo.com/sms. It’s a useful mesh development environment because it offers a blend of automated mesh generation tools in combination with intuitive manual operators, which is ideal for making a TUFLOW FV flexible mesh.

3.6 Pre and Post Processing

Modelling of any kind requires significant processing and presentation of input and output information. BMT does not sell pre and post processing tools. We do however use a range of commercially available tools that best suit our needs. Generally, TUFLOW FV users employ the following:

- A text editor of some description to edit input text files (UltraEdit and Notepad++ are popular, although Notepad does suffice).
- SMS for mesh generation, pre and post processing. The "SMS Tips" wiki contains some useful information:
    - http://wiki.tuflow.com/index.php?title=SMS_Tips
- Excel spreadsheets.
- Geographic Information Systems (GIS) such as MapInfo or ArcGIS provide powerful environments for developing model components and building blocks, such as Digital Elevation Models.
- MatLab is used extensively for manipulating input data and model results. BMT provides a series of compiled executable MatLab scripts on the website; see www.tuflow.com/ProductDownload.

TUFLOW FV input and output formats are designed to be as flexible as possible to accommodate other pre and post processing tools (see Section 3.7).

3.7 File Types

A variety of file types are used to specify a TUFLOW FV model. Importantly, all of the input and output file types used to specify or produced by a TUFLOW FV model simulation are open formats that can be readily interrogated and manipulated. Many TUFLOW FV components are simple ascii (text) files that can be easily created and manipulated in text editor and spreadsheet environments.

Table 3-1 File formats and file extensions used by TUFLOW FV

• Ascii Control File (e.g. .fvc)
• SMS Generic Mesh File (.2dm)
• Comma Delimited Text File (.csv)
• Network Common Data Format - netcdf (.nc)
• Hierarchical Data Format – hdf5 (.h5)
• SMS Binary Data File (.dat)

3.8 Recommended Directory Structure

It is highly recommended that a directory structure similar to that specified in the following table is adhered to when setting up a TUFLOW FV modelling project. For complex modelling projects it may help if more sub-directories are created. For instance the BC directory could be further split into "Meteorological", "Tidal" and "Catchment" sub-directories.

In many cases the "Output" directory will be specified on a local computer hard-disk as TUFLOW FV output files can sometimes be too large for network storage.

Table 3-2 Recommended TUFLOW FV Directory Structure

Level 1	Level 2	Level 3	File types	Comments
TuflowFV			SimulationLog.xls	A list of all the simulations performed, their relevant control files and reasons for running them.
	geo		*.2dm, *.csv	Model geometry, often linked to a separate folder containing spatial data generation (such as from GIS and/or mesh generation packages).
	bc		*.csv, *.nc	Boundary conditions.
	input		*.fvc, *.fvm	Input control files, often where TuflowFv.exe is executed from. Batch files are also stored here when performing multiple simulations.
		Log	*.log, *_geo.nc, *_cfl.csv, *.rst	Log files, recording pre-processor outputs and performance during simulation.
	output		*.dat, *.csv, *.nc	Can be a large folder, often placed on local drive rather than a network drive.
	results		*.xls, *.jpg, etc.	Processing of model output
	exe (or bin)		tuflowfv.exe	Optionally, placing the executable (and associated dlls) within the folder structure may be a worthwhile measure if archiving.
	Matlab		*.m, *.mat	Optional, storing Matlab scripts.
	SMS		*.sms, *.2dm, *.mat	Optional, storing intermediate mesh generation files.

3.9 Recommended "fvc" Structure

TUFLOW FV control files are simple ASCII based scripts or recipes for how a simulation is to be performed. The control file includes information about the simulation configuration, when the simulation is to start and end, the value of model variables, where to find the model geometric layout, what initial and boundary conditions are specified and what model output to produce.

As with the directory structure, a standardised layout should be adopted for preparing the TUFLOW FV control file, as shown in Table 8-1 (Section 8). It is also recommended that a fully-featured text editor be used, as these provide various useful features such as "command colour coding", "file hyper-linking" and "macros". One such editor is Notepad++ (this is open source, www.notepad-plus-plus.org).

3.10 SMS Interface (Beta)

An interface for TUFLOW FV that allows the user to build and run a model within SMS is being developed. At present this allows for only limited boundary condition types, but this is planned to be expanded in the future. In this section of the manual, the installation of the interface is described. A tutorial example is provided in Section 6.

3.10.1 Installation

When the TUFLOW FV interface for SMS in downloaded the following files are included:

1 Convert_and_run.bat

2 Mesh_to_fv.exe

3 TUFLOW_FV.2dm

The interface does not need to be installed, however it does need to be configured.

The convert_and_run.bat file is a batch file that will be initialised by SMS. This needs to be configured to your machine, to do this, edit the .bat file in a text editor. An example of commands in the .bat file are show in Figure 3-1 below, the 8th line defines the location of the TUFLOW FV executable, the highlighted text needs to be replaced with the location of the TUFLOW FV executable on your machine.

Tip: In Windows 7 if you explore to the path of the executable, hold shift down and then right click on the executable "Copy As Path" should be an option. This copies the pathname to the clipboard and can be pasted into the text editor.

Similarly line 7 needs to be edited to define the location of 'mesh_to_FV.exe' file.

```
echo off
setLocal
set input=%1
set dir=%cd%
set temp=%1:.2dm=.fvc%
set output=%input:.2dm=.fvc%
set parser="C:\Program Files (x86)\TUFLOW\mesh_to_FV.exe"
set tf_fv="C:\Program Files (x86)\TUFLOW\TUFLOWFV.exe"
set fvcpath=%dir%\TUFLOWFV

echo Current Directory: %dir%
echo Input 2d mesh file: %input%
echo Output control file: %output%
echo Path to 2dm convertor: %parser%
echo Path to TUFLOWFV exe: %tf_fv%
echo Press any key to convert to TUFLOW-FV Format
echo (where's the any key)
pause
```

Figure 3-1 SMS Interface: Configuring Batch File

In SMS the interface needs to be configured to utilise the batch file that we just modified, to do this, in SMS select:

Edit → Preferences…

Navigate to the file locations tab and then in the Model Executables under the Generic entry , select "Browse", navigate to the correct directory, select "All Files" from the files of type dropbox and then and select the convert_and_run.bat. Screen images are provided in Figure 3-2.

The SMS interface is now ready to use.

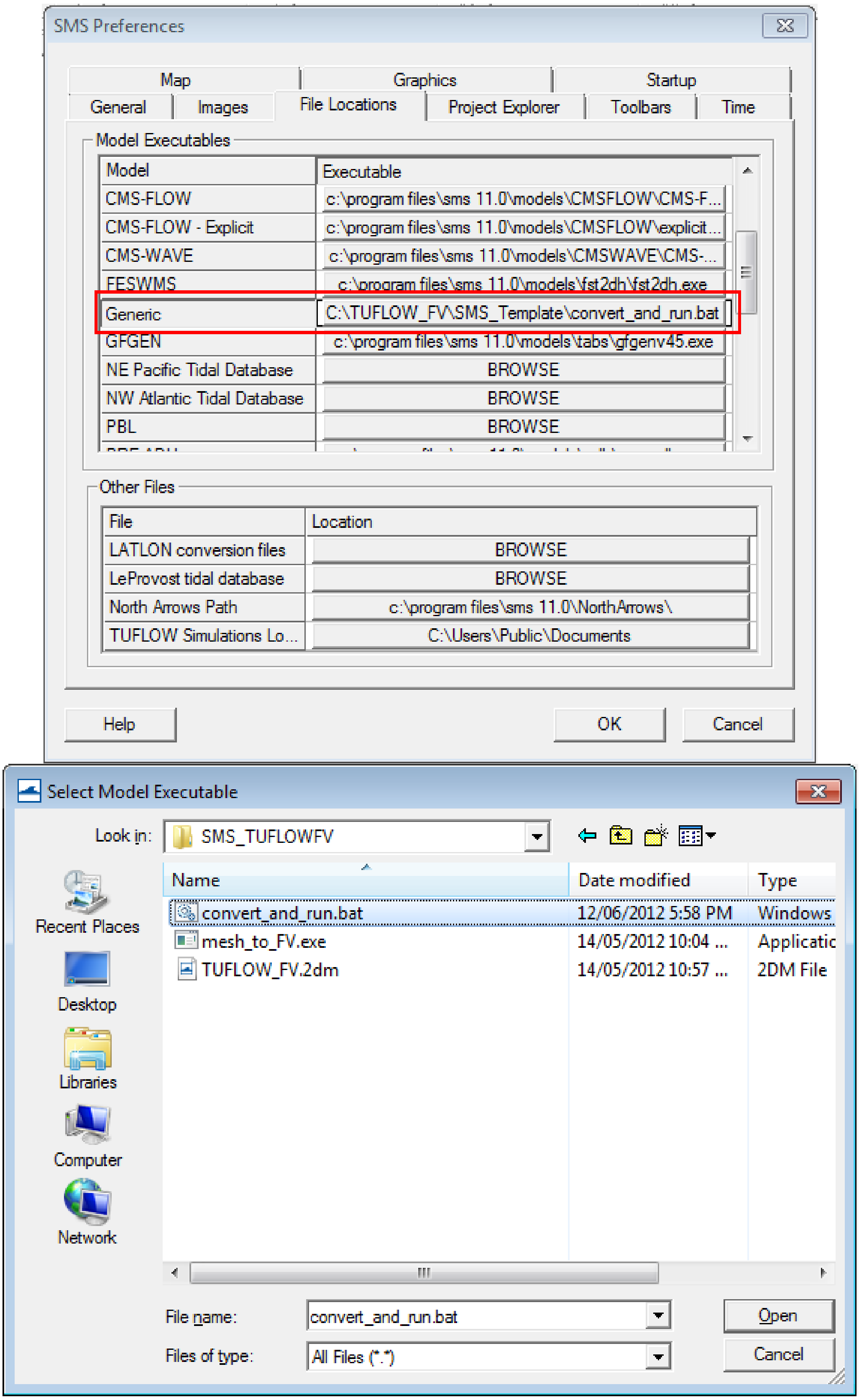


Figure 3-2 SMS Interface: Setting Generic Interface Location

3.10.2 Loading the Interface

When using the SMS interface for TUFLOW FV the steps involved in creating the model are:

- Create the model mesh (see Section 4.4 and the SMS tutorials)
- Set the model boundaries and parameters
- Select Run TUFLOW FV (in the menu choose TUFLOW FV → Run TUFLOW FV). This:
  (a) Creates the TUFLOW FV directory structure and converts the SMS .2dm file to TUFLOW FV format inputs.
  (b) Runs TUFLOW FV on the newly created inputs.

When RUN-TUFLOW FV is selected this starts the batch file, which firstly converts the model and then runs the model.

An example model using the interface is provided in Section 6.2. Before starting the creation of the model mesh, the TUFLOW FV definition needs to be loaded into SMS, this is done by opening the TUFLOW_FV.2dm provided with the download (see Figure 3-3). Once loaded a TUFLOW FV menu item is visible, as shown in Figure 3-4. At this stage with no model mesh created most of the options are un-selectable (grey).

NOTE: The Define Model is used to create / modify the interface, this should not be modified by the user and is password protected. If this is modified, the conversion process is highly likely to fail.

With the TUFLOW FV model definition loaded we are now ready to create the TUFLOW FV mesh and model. This is described in the tutorial model in Section 6.2.

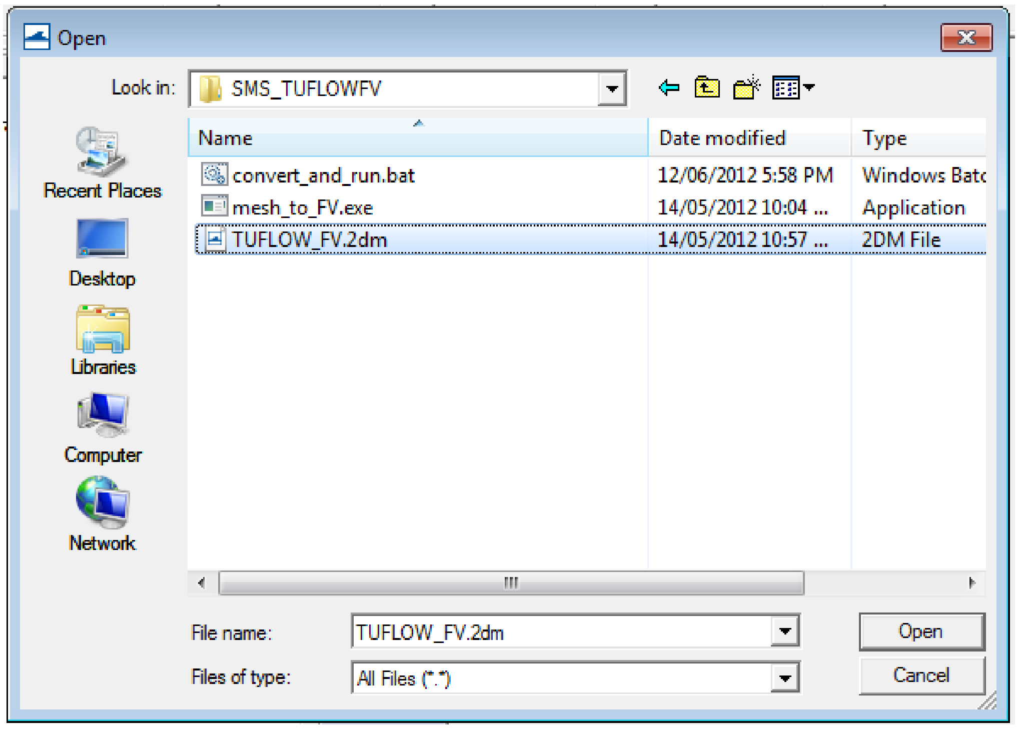


Figure 3-3 SMS Interface: Loading Model Definition

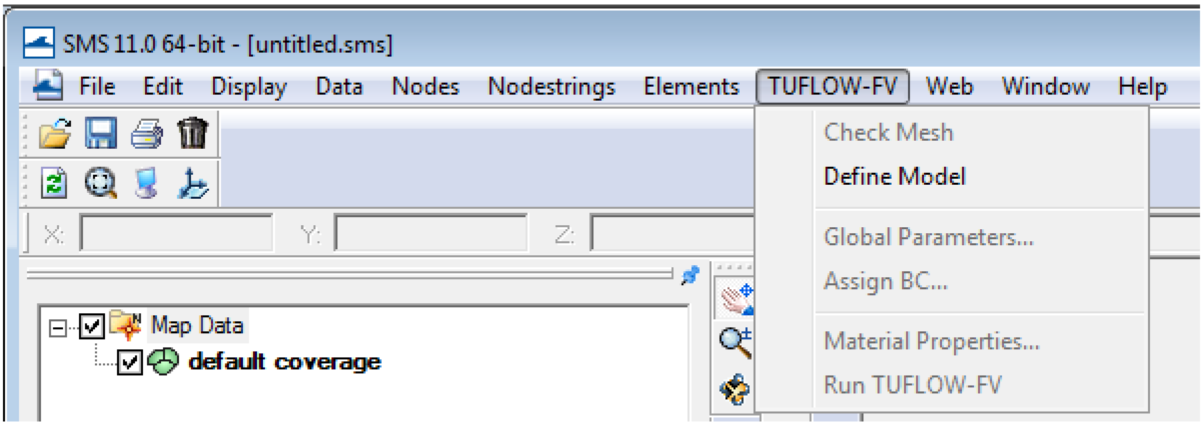


Figure 3-4 SMS Interface: TUFLOW FV Menu Item

3.11 Excel Interface

Is coming….

4 Recommended steps in the modelling process

4.1 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:
        - o time scales (hours, months, years, decades, etc)
        - o 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.

4.2 Establish model domain, spatial and temporal scales

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 timestep, which in turn is limited by the element in the model domain that limits the CFL number. Section 7.3 provides further discussion on this aspect.
- How complex the numerical processes are (eg an HD + ST simulation will require additional computational effort compared to a HD simulation).
- The number of active, wet elements (or cells) in the model domain (note that this can vary from one timestep to another).
    - ➢ The spatial extent of a TUFLOW FV model (ie the area to be modelled) is typically guided by:
        - o the spatial extent of the problem to be solved
        - o the availability and locality of data with which to define boundary conditions
        - o the spatial extent of the key physical processes to be represented
- The specified start and end time.
    - ➢ The temporal extent of a TUFLOW FV model (ie the duration of model simulations) is typically guided by the temporal extent of the key physical processes to be represented. Examples include:
        - o a flood assessment requires simulation of individual flood events of hours duration
        - o an estuarine assessment, where tidal forces dominate, requires simulations of semi-diurnal and diurnal tidal cycles and possibly spring /neap cycles
        - o a morphological assessment may require simulation periods of decades

4.3 Consolidate and prepare base data

Consolidate and prepare base data, especially bathymetry / topography but also boundary conditions.

Spatial and time series data is normally relatively easy to collate, especially with pre-processing tools such as spreadsheets, GIS, MatLab, etc. Quality checking of data is important (yes, the often quoted garbage in, garbage out phrase cannot be left out of any modelling manual).

4.3.1 Bathymetry/Topography

A good description of bathymetry (below the water surface in rivers, seas, etc) and topography (above the water surface on land) is crucial for all hydrodynamic modelling exercises.

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.

In most modelling exercises an early step will be to develop a Digital Elevation Model (DEM) of the study area using the available sources of bathymetry/topography data and GIS software. DEMs can be directly imported to some mesh building environments (such as SMS) and used to guide the mesh construction prior to interpolating the elevation data to the TUFLOW FV mesh. Alternatively, and depending on the capability of the mesh building software, the digitised bathymetry/topography x,y,z scatter datasets may be directly imported to the mesh building environment and interpolated to the mesh.

Various bathymetry and topography datasets are freely available online. Note that these datasets are typically of a regional scale and may not resolve local features. An example DEM constructed using MapInfo software from a combination of hydrographic survey, LIDAR and digitised nautical chart data sources is shown in Figure 4-1.

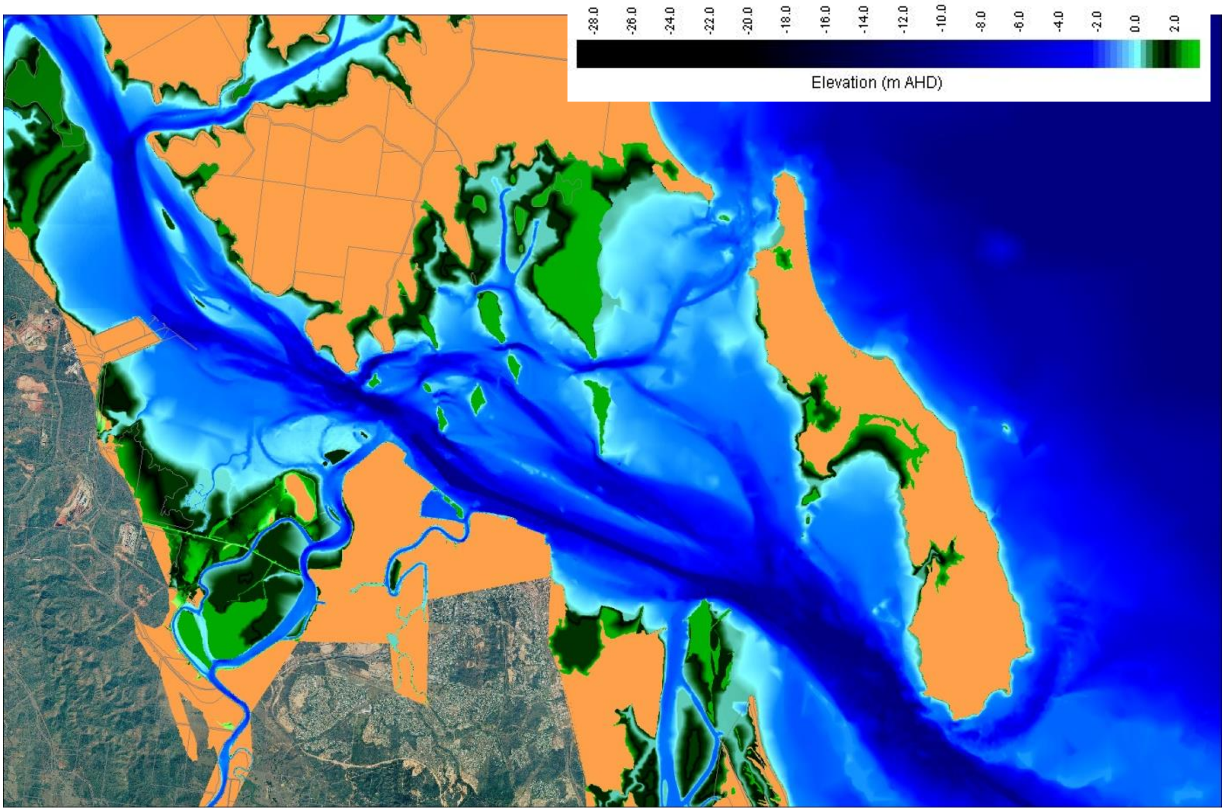


Figure 4-1 Digital Elevation Model of Port Curtis, Queensland, Australia

4.4 Mesh construction

Using mesh generation software, create a model mesh. Design a mesh that takes full advantage of the flexible mesh approach and also avoids pitfalls and disadvantages.

Section 7.1 provides further information.

TUFLOW FV solves the NLSWE on unstructured meshes comprising of triangular and/or 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.

Figure 4-2 shows a TUFLOW FV mesh and DEM of Port Curtis (the DEM without the mesh is shown in Figure 4-1). 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 4-2.

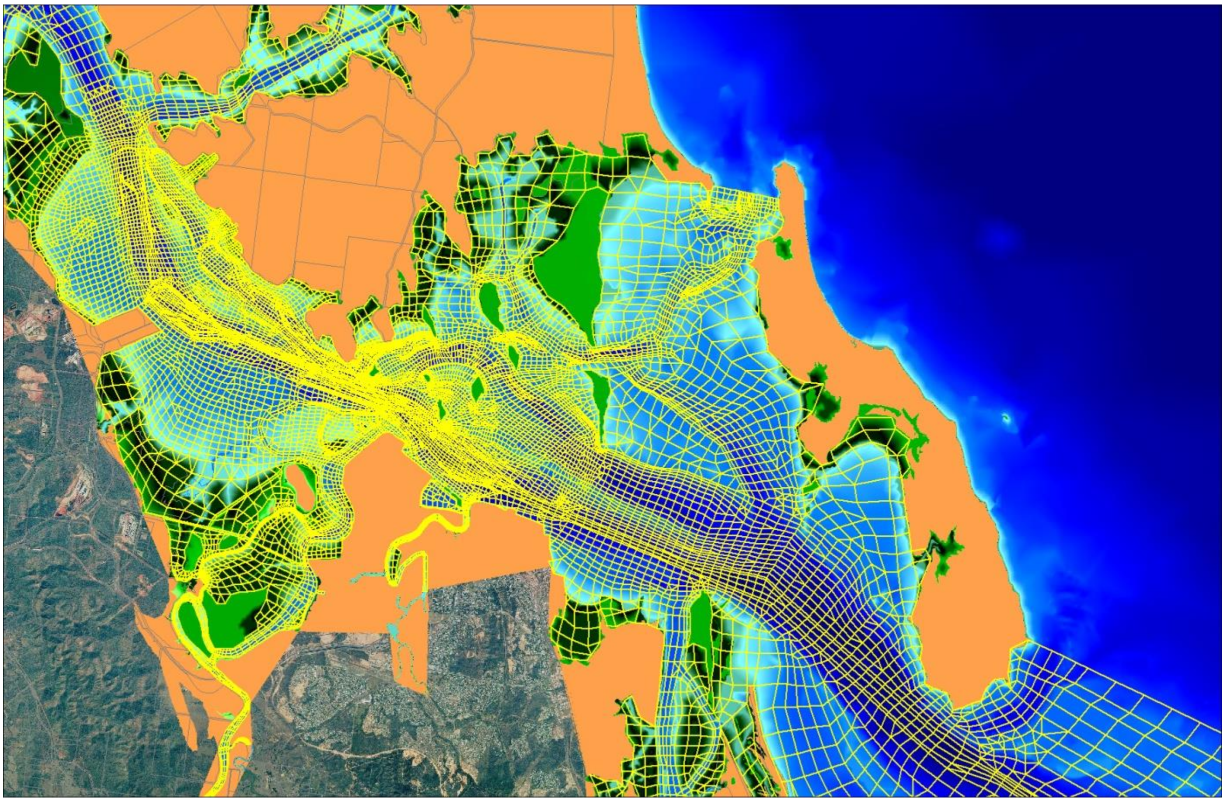


Figure 4-2 TUFLOW FV Mesh of Port Curtis, Queensland, Australia

Unstructured mesh geometries can be created using any suitable mesh generation tool. BMT WBM staff generally use the SMS Generic Mesh Module (www.aquaveo.com/sms) 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 included with a SMS installation or can be accessed via the Aquaveo SMS website.

A TUFLOW FV mesh is constructed using nodes, arcs and vertices. 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 etc. A good mesh is constructed using the mesh controls (nodes, vertices and arcs) to neatly resolve the important features within the model domain.

Figure 4-3 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 and based on the positions of the mesh controls.

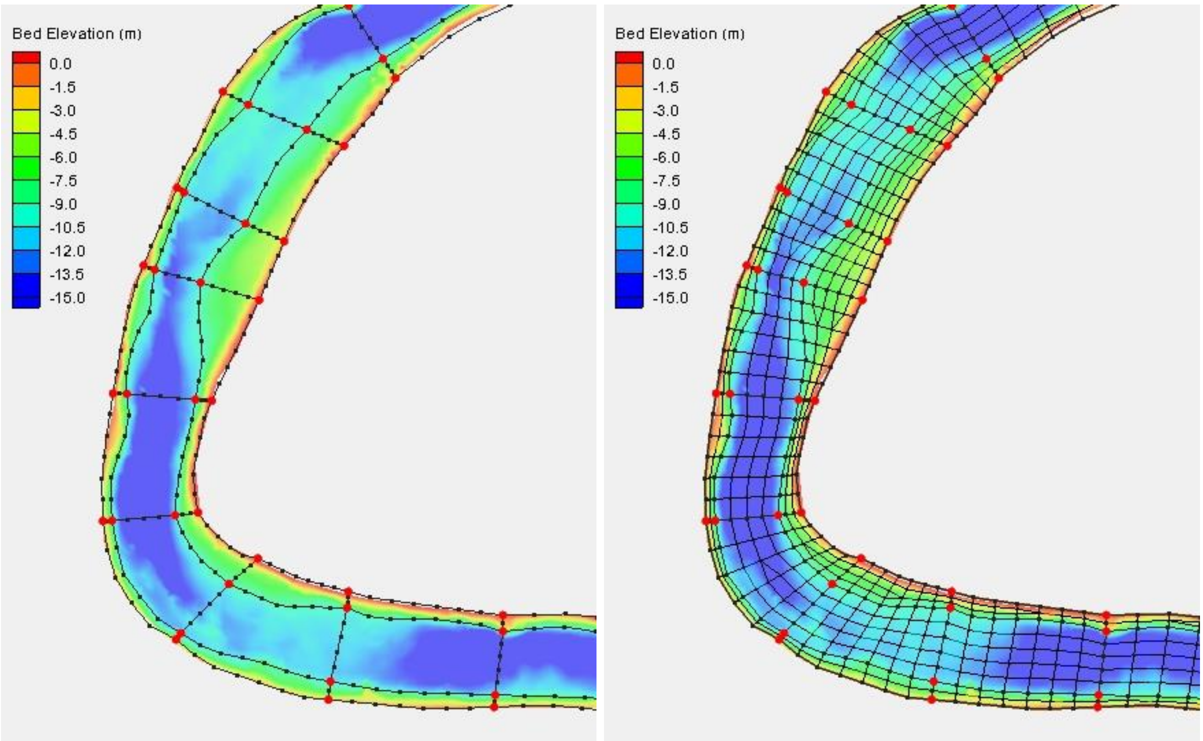


Figure 4-3 Mesh nodes, arcs and vertices (left) and the resulting mesh (right)

Bed levels / bathymetry are normally assigned to the mesh once the mesh design is completed.

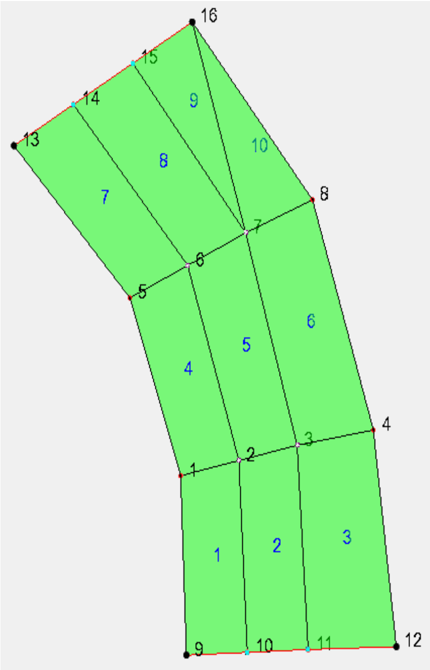
For more discussion on mesh generation, see the tutorial exercise in Section 5.1 and tips in Section 7.1.

4.4.1 2dm file format

The 2dm file format is used to define the TUFLOW FV mesh. It is an ASCII format from the SMS Generic Mesh Module. The contents of the file relevant to TUFLOW FV simulations (see also the example in Figure 4-4) are:

- 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 (ie bed level) coordinate.
- 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.
- 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.
- Lines that commence with a “NS” are nodestrings, which are used to define boundary conditions. Each NS line defines the series of nodes that form the string, the last node number is assigned as negative.
- Other components of the 2dm file are not used by TUFLOW FV.

For more information see Section 0.




```
MESH2D
E4Q 1 2 1 9 10 1
E4Q 2 3 2 10 11 1
E4Q 3 4 3 11 12 1
E4Q 4 6 5 1 2 1
E4Q 5 7 6 2 3 1
E4Q 6 8 7 3 4 1
E4Q 7 14 13 5 6 1
E4Q 8 15 14 6 7 1
E3T 9 16 15 7 1
E3T 10 16 7 8 1
ND 1 2.48000000e+001 4.02800000e+001 0.00000000e+000
ND 2 3.30421270e+001 4.21236640e+001 -1.00000000e+001
ND 3 4.12871134e+001 4.39683219e+001 -1.00000000e+001
ND 4 5.20800000e+001 4.58200000e+001 0.00000000e+000
ND 5 1.76200000e+001 6.18200000e+001 0.00000000e+000
ND 6 2.57990034e+001 6.57578467e+001 -1.00000000e+001
ND 7 3.39997467e+001 6.96992724e+001 -1.00000000e+001
ND 8 4.34600000e+001 7.37100000e+001 0.00000000e+000
ND 9 2.56200000e+001 1.85400000e+001 0.00000000e+000
ND 10 3.42333333e+001 1.88833333e+001 -1.00000000e+001
ND 11 4.28466667e+001 1.92266667e+001 -1.00000000e+001
ND 12 5.53200000e+001 1.95700000e+001 0.00000000e+000
ND 13 1.21000000e+000 8.02700000e+001 0.00000000e+000
ND 14 9.62000000e+000 8.52633333e+001 -1.00000000e+001
ND 15 1.80300000e+001 9.02566667e+001 -1.00000000e+001
ND 16 2.64400000e+001 9.52500000e+001 0.00000000e+000
NS  9 10 11 -12
NS  16 15 14 -13
BEGPARAMDEF
GM  "Mesh"
SI  1
DY  0
TU  ""
TD  0  0
NUME  3
BCPGC  0
BEDISP  0 0 0 0 1 0 1 0 0 0 0 1
BEFONT  0 2
BEDISP  1 0 0 0 1 0 1 0 0 0 0 1
BEFONT  1 2
BEDISP  2 0 0 0 1 0 1 0 0 0 0 1
BEFONT  2 2
ENDPARAMDEF
BEG2DMBC
MAT  1 "material 01"
END2DMBC
```

Figure 4-4 Example 2dm file, showing spatial layout (left) and 2dm file contents (right)

4.5 Boundaries

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 (see also Section 8.4.1 and 8.4.10):

- 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

4.5.1 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.

Descriptions of the various boundary conditions, their commands and associated inputs are provided in Section 8.4.10.

4.5.2 Bed friction

For hydrodynamic simulations (without sediment transport) the bed boundary is simply described using a bed roughness model. The default model is that attributed to Manning, in which case a Manning's "n" coefficient should be specified. An alternative model assumes a log-law velocity profile and requires specification of a surface roughness length-scale, "ks". A single bed roughness can be set globally or the modeller can assign different roughness values to particular mesh cells within the model domain. See Section 8.4.7.1.

4.5.3 Forcings

Boundary conditions can be applied to the water surface and 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. Wind, pressure 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.

4.5.4 Wetting and Drying

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 more extreme over land flows associated with a flood, storm surge or tsunami. 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, 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.

4.5.5 Initial conditions

For models that simulate tidal hydrodynamics only, the modeller may choose to start the model with an initially flat, stationary sea and allow the open boundary input to “warm-up” the model. 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 by output from a previous simulation (using a TUFLOW FV restart file).

4.6 Model Parameterisation

Define what processes and parameter values are to be assigned to the model, ensuring that their values lie within scientifically justifiable ranges.

4.6.1 Turbulent Mixing

Unresolved mixing processes are modelled as gradient-diffusion, where the eddy-viscosity for momentum mixing and the diffusivity for scalar mixing can be parameterised using various options.

4.6.1.1 Eddy viscosity

The horizontal-mixing eddy-viscosity can be defined as a constant value or can be calculated using the Smagorinsky model. The Smagorinsky model sets the diffusivity proportional to the local strain rate.

The vertical-mixing eddy-viscosity can be defined as a constant value or can be calculated using a parametric model. The parametric model is based on a parabolic eddy-viscosity profile and applies the Munk & Anderson limiters in the case of stable stratification.

Upper and lower bound values can be specified for the horizontal and vertical eddy-viscosities.

4.6.1.2 Scalar diffusivity

The horizontal-mixing scalar diffusivity can be defined as a constant value or can be calculated using the Smagorinsky or Elder models. The Elder model calculates an an-isotropic diffusivity tensor with principal axes aligned with the flow direction and which scales on the local friction velocity. The Elder model allows the user to specify higher mixing in the longitudinal flow direction than transverse to the flow.

The vertical-mixing scalar diffusivity can be defined as a constant value or can be calculated using a parametric model. The parametric model is based on a parabolic eddy-viscosity profile and applies the Munk & Anderson limiters in the case of stable stratification.

Upper and lower bound values can be specified for the horizontal and vertical scalar diffusivities.

4.6.2 First or Second Order

Higher order spatial schemes will produce more accurate results in the vicinity of sharp gradients due to reduced numerical diffusion, however they will be more prone to developing instabilities and are more computationally expensive. 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.

As a general rule of thumb, 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.

4.6.3 2D/3D

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.

As a step in the development of a 3D model a 2D simulation should be performed first. Once the 2D model has been optimised and output verified the modeller may then choose to perform a 3D simulation.

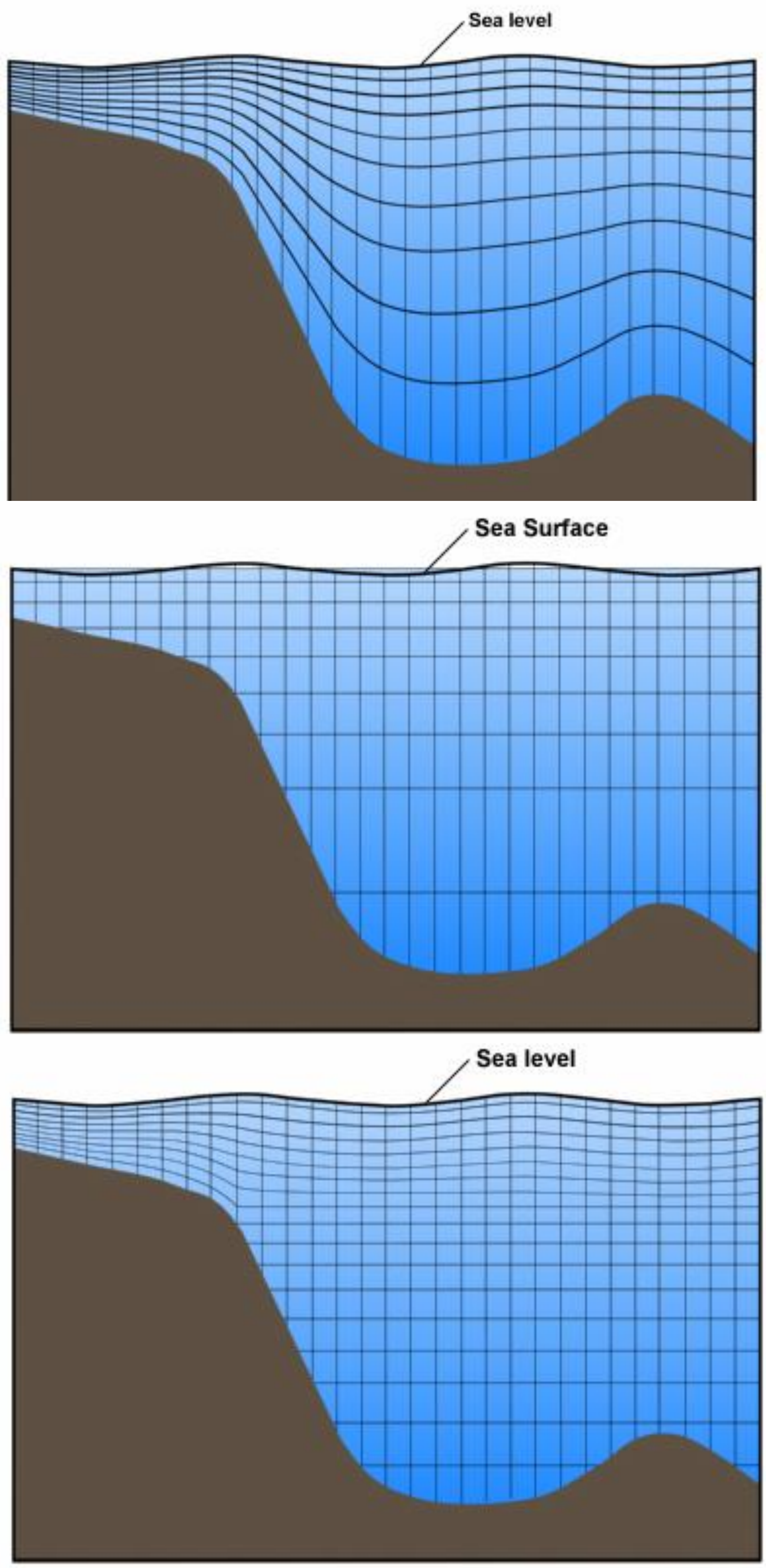


Figure 4-5 Illustration of vertical discretisation options; sigma coordinates (top), z coordinates (middle) and hybrid z-sigma coordinates (bottom) (from publicwiki.deltares.nl)

4.6.4 Baroclinic

Baroclinic pressure-gradient terms can be optionally activated to allow the hydrodynamic solution to respond to temperature, salinity and sediment induced density gradients.

4.6.5 Atmospheric Exchange

Atmospheric heat and momentum exchange can also be calculated given standard meteorological parameter inputs by an integrated module.

4.7 Test Model performance

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

- Mesh accurately represents the bathymetry / topography
- Key physical processes are suitably represented
- There are no strange element shapes or sizes
- The model does not require unnecessarily short timesteps to run
- Any unexpected outputs or model features are explained and justified.

TUFLOW FV has a number of pre-simulation checks and log outputs that can be used to assist; see Section 8.

4.8 Calibration / validation / sensitivity testing

Calibrate the model to available data.

Verify the model to another set of independent 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 results match measurements. Validation is the process where a calibrated model is compared to measurements from a different period with different physical conditions. In combination, calibration and verification prove that the model can replicate the physical processes and is a useful tool. An uncalibrated and unvalidated model is also called a “computer game” (except the graphics aren’t usually as good!).

Choice of measurement periods for calibration depends upon the physical processes that need to be captured in the model. Typically, time series of response (for example river discharge / stage or tidal variations) 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 time series of current speed, direction and water level at two separate locations, performed over a 3 day period during a spring tidal range
- Validation: A time series of current speed, direction and water level at two separate locations, performed over a 3 day period during a neap tidal range

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

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 is not a complete waste of time. 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.

4.9 Application

Apply the model to the problem to be solved. Description of existing conditions, impact scenarios and comparison of differences, etc are common applications. Keep in mind the quality and clarity of your post processing; communicating your modelling efforts to your audience effectively is a key part of using TUFLOW FV.

5 Quick SMS and TUFLOW FV Tutorial

5.1 A quick SMS tutorial – trapezoidal channel

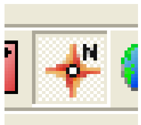
The following example demonstrates the development of a very simple model mesh. Follow the steps performed here and expand upon them to develop more complex, real-world models.

The example is a trapezoidal channel, dimensions as shown:

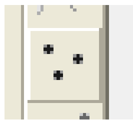
- Top width = 100 m
- Bottom width = 50 m
- Depth = 5 m
- Length of channel = 1,000 m
- Grade of channel = 1 in 1,000
- The model domain should have a resolution of 12.5 m across the channel and 25 m along the channel.

1 Map Coverage (points and arcs defining the model layout)

A The first step is to setup the SMS Map coverage. In this module the “outline”, or specific points and curves that describe the geometry to be meshed, is defined.

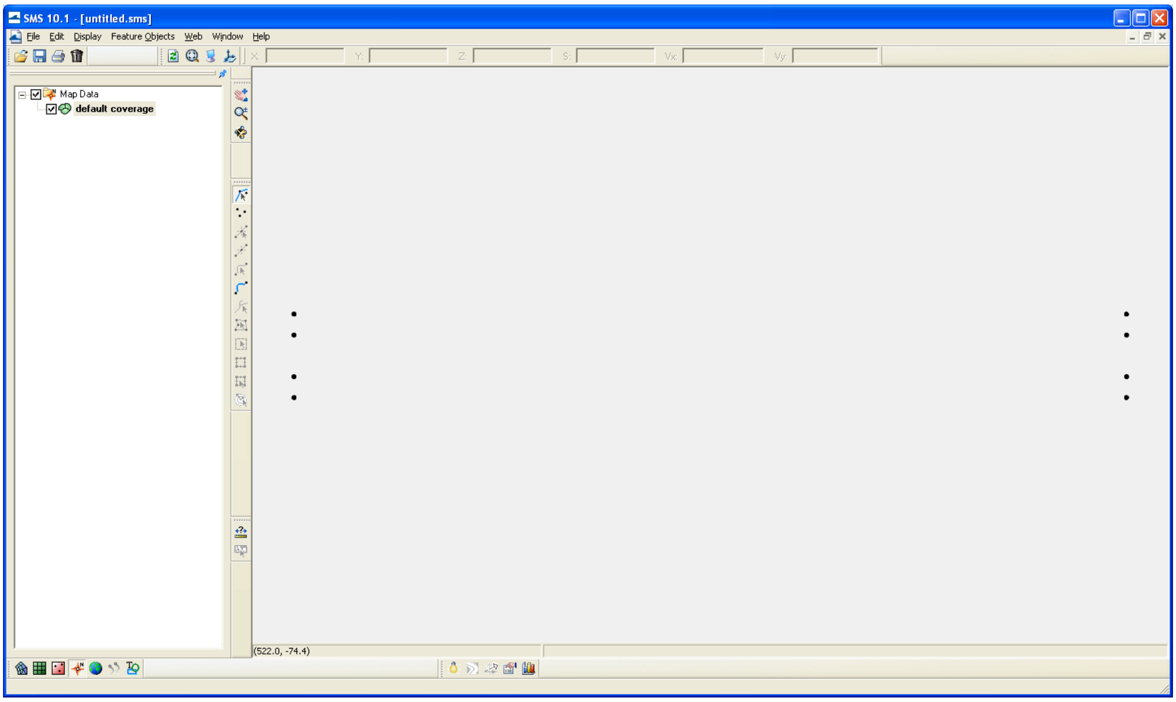


B The map coverage works in a plan view. Using the “create feature point” button, create the 8 points that define the channel.

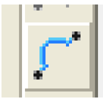


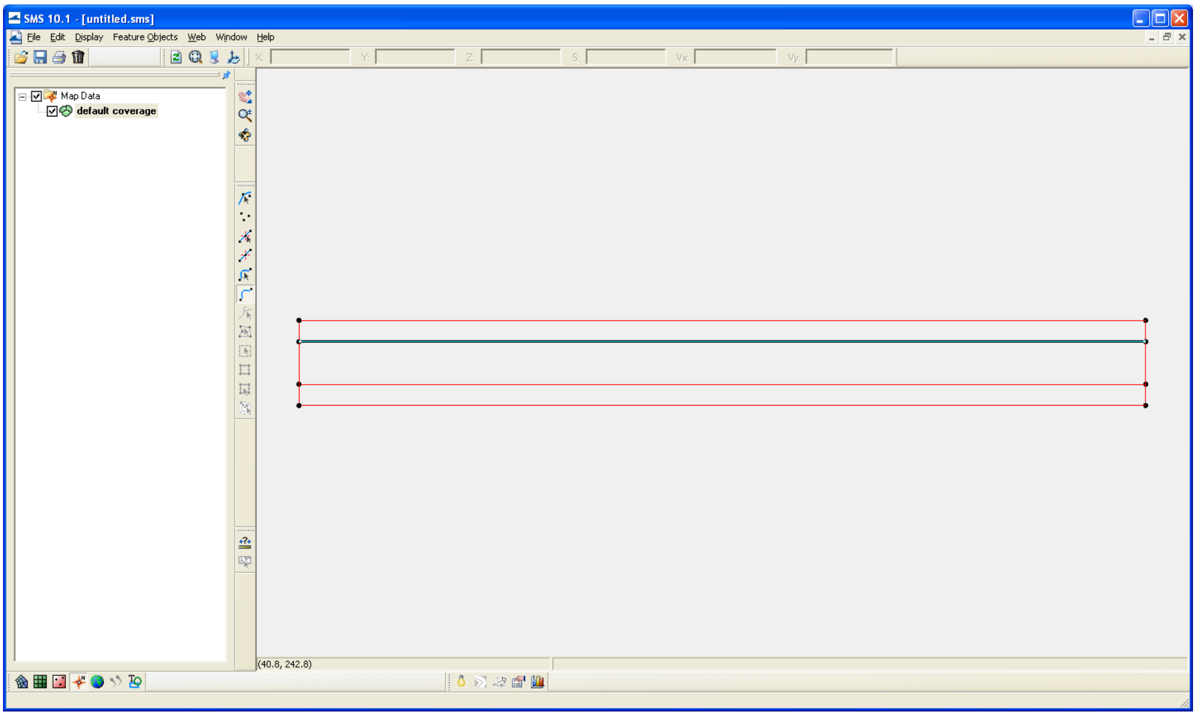
C As each feature point is created, use the coordinate boxes in the toolbar to specify the precise coordinates (x and y). Also insert the z value.

The feature points should then look like:



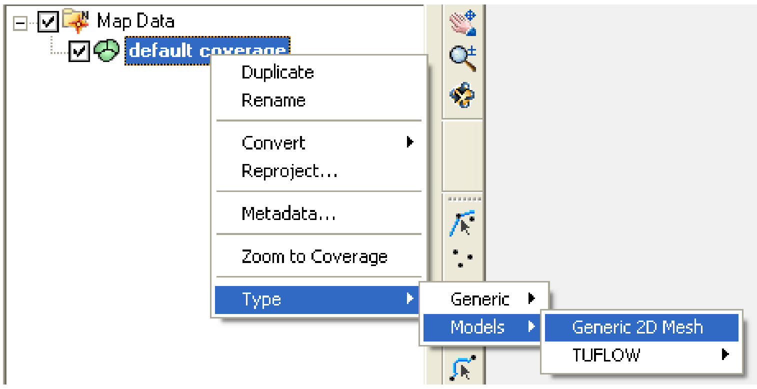
D Now use the "Create feature arc" button to join the dots together.





E At this point, it's a good idea to **SAVE**.

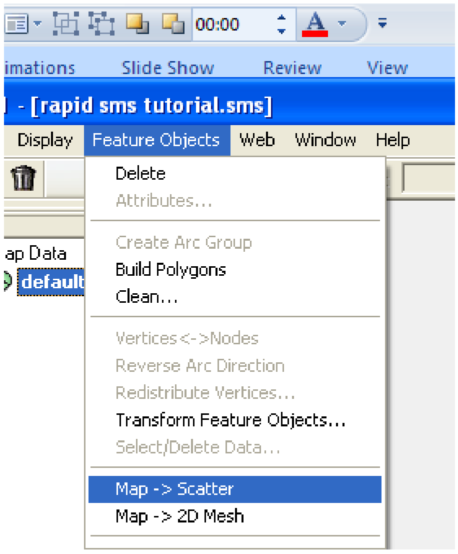
F Also at this point (if not before), it is time to set the coverage type for the map data. The coverage type must be "Generic 2D Mesh". Right click on the coverage label in the explorer bar as shown:



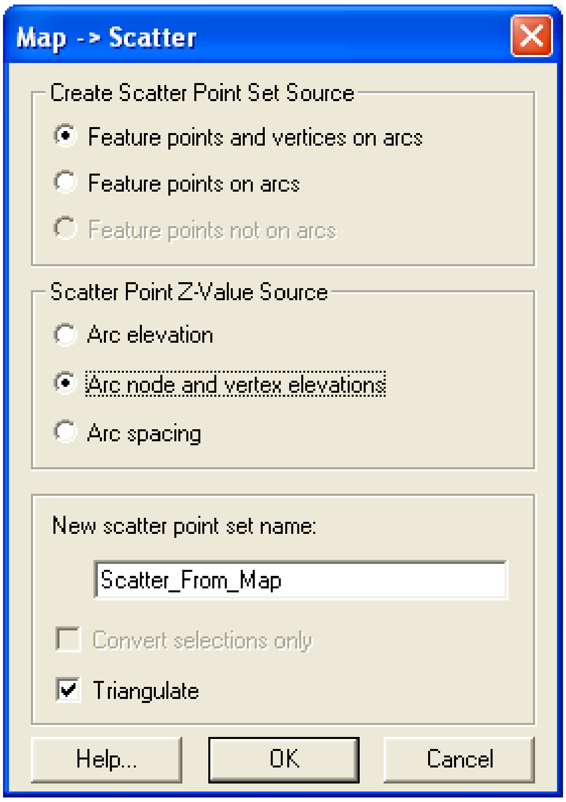
You have now created the basic map layout that will define the model geometry.

2 Create Scatter points (from which bed levels will be interpolated from)

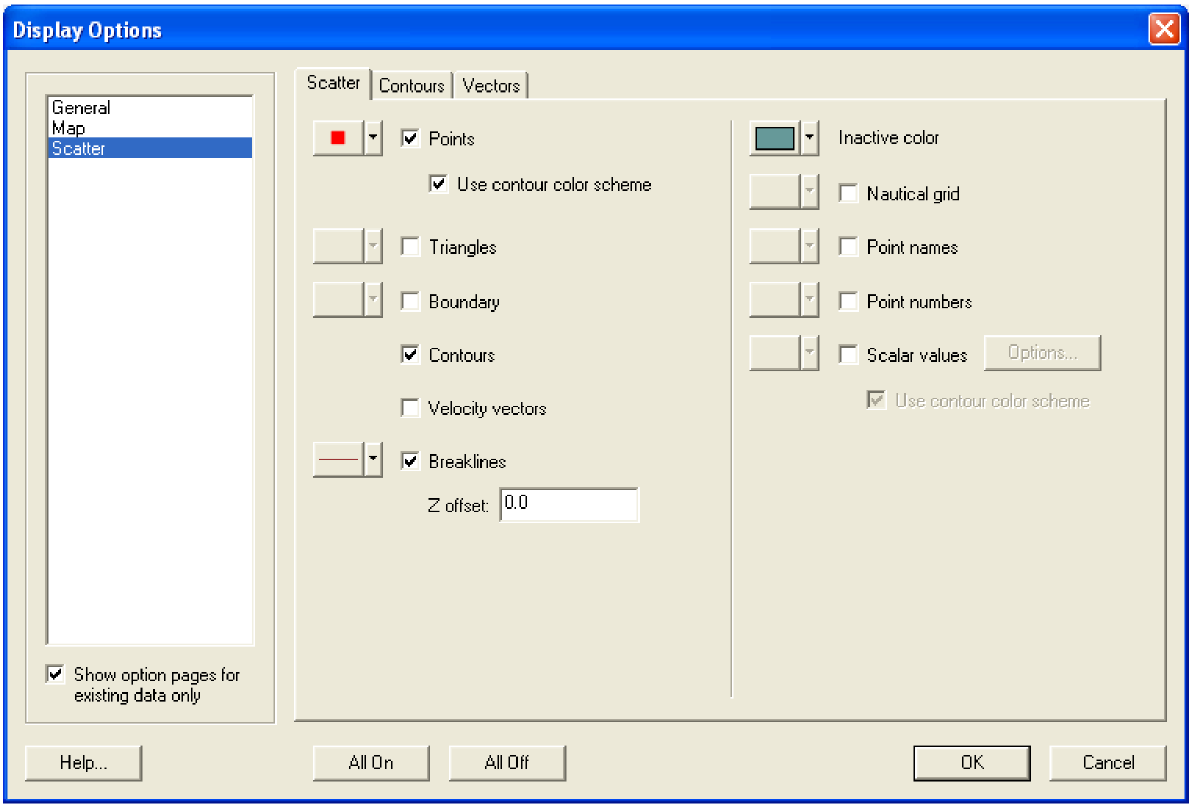
A If you have entered "z" values you can also specify the bathymetry to be used in the model. Do this from the menu "Feature objects" – "Map -> Scatter".



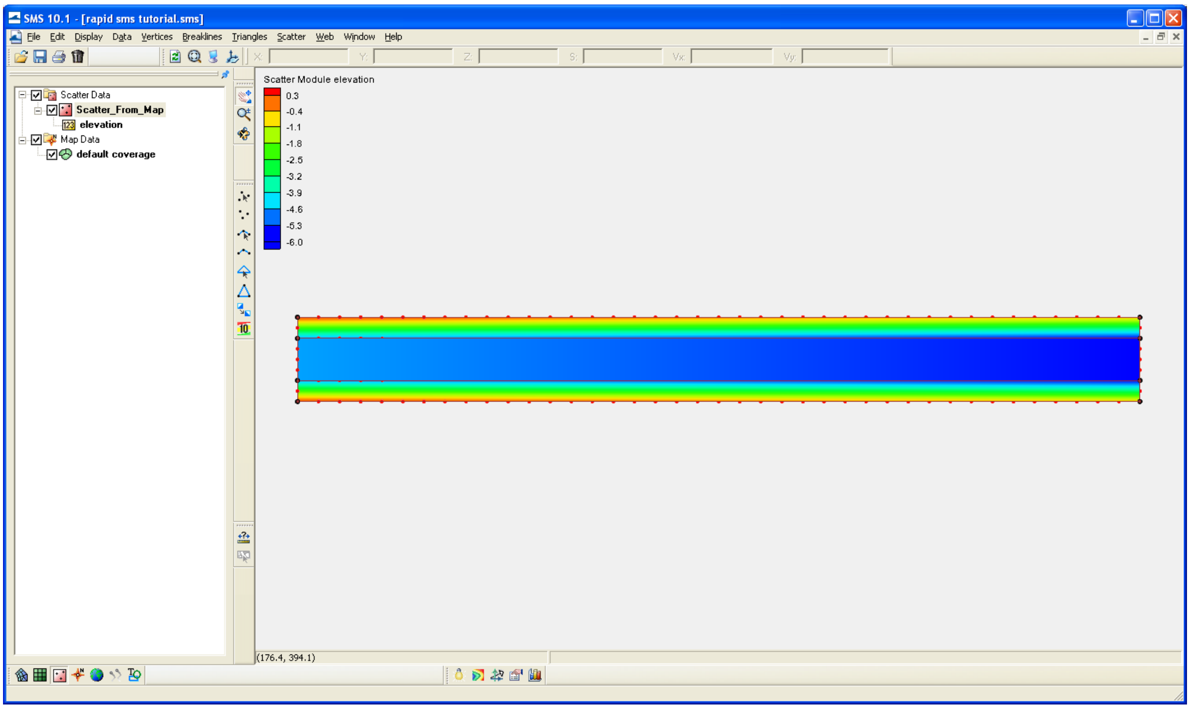
B Make sure you specify the z value source from the “Arc node and vertex elevations” in the dialog box:



C To see your handiwork, use the display button to turn on contours in the scatter data module:



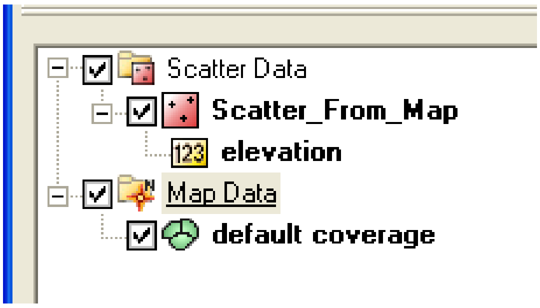
Then the shaded z values are then visible:



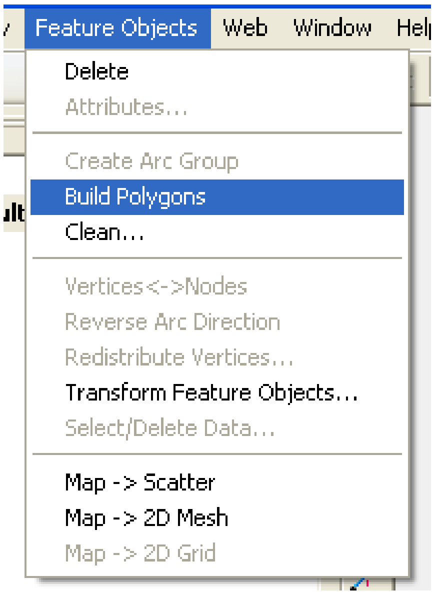
These steps (1 and 2) have replaced the often complex steps associated with inputting GIS layers, scatter datasets, etc to create the base geometry for the model. The approach demonstrated is fine for a simple test case, but real world applications are often more complex and contain a variety of data sources, etc. This can be done in SMS in a more rigorous manner (discussed in the SMS manuals) but is also done using other software such as GIS and CAD.

3 Build polygons

A Back to the mesh module – click on the "Map Data" entry in the explorer window to do this.



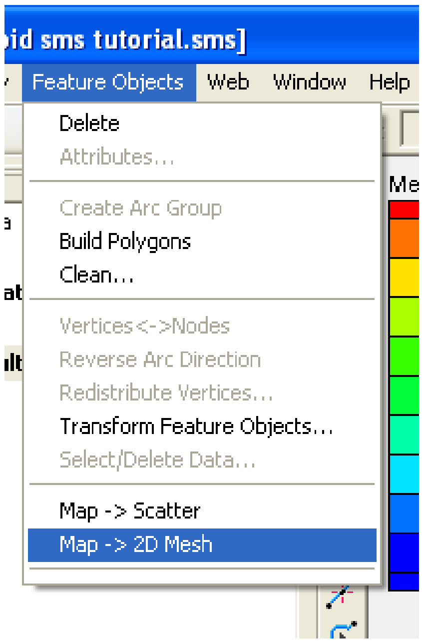
B The next step is to build polygons, which is done from the menu “Feature objects” – “Build polygons”.



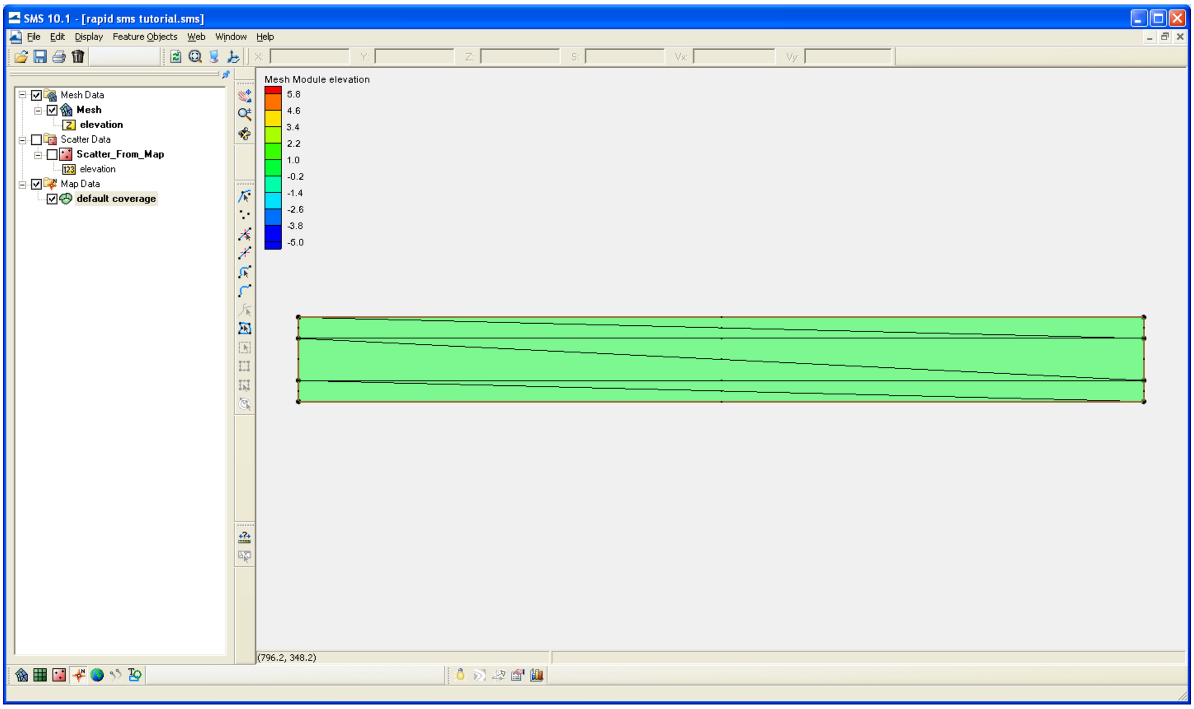
This takes the feature arcs and creates a series of polygons. It is these polygons that we can now individually investigate and specify mesh properties for.

4 Build the mesh (but need to go back and increase vertex resolution!)

A Now we can build a mesh. To build the mesh, use the menu commands “Feature objects” – “Map -> 2D Mesh”.



The resulting mesh, as shown, doesn’t look very good. But it is a mesh! A mesh has been created using 6 triangular elements (pave), connected by the nodes which, in this case, are the 8 points used to define the extents of the trapezoidal channel.



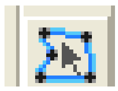
Note in the image that the scatter data set has been "unticked" in the explorer window – this hides the scatter data in the display, which makes the other information easier to see. Also unclick the mesh data set to better inspect the mesh module information.

This model geometry is not good enough; we require a much higher resolution than this. To make it better we need to go back to the mesh module and adjust the polygons to create more vertices and hence more elements.

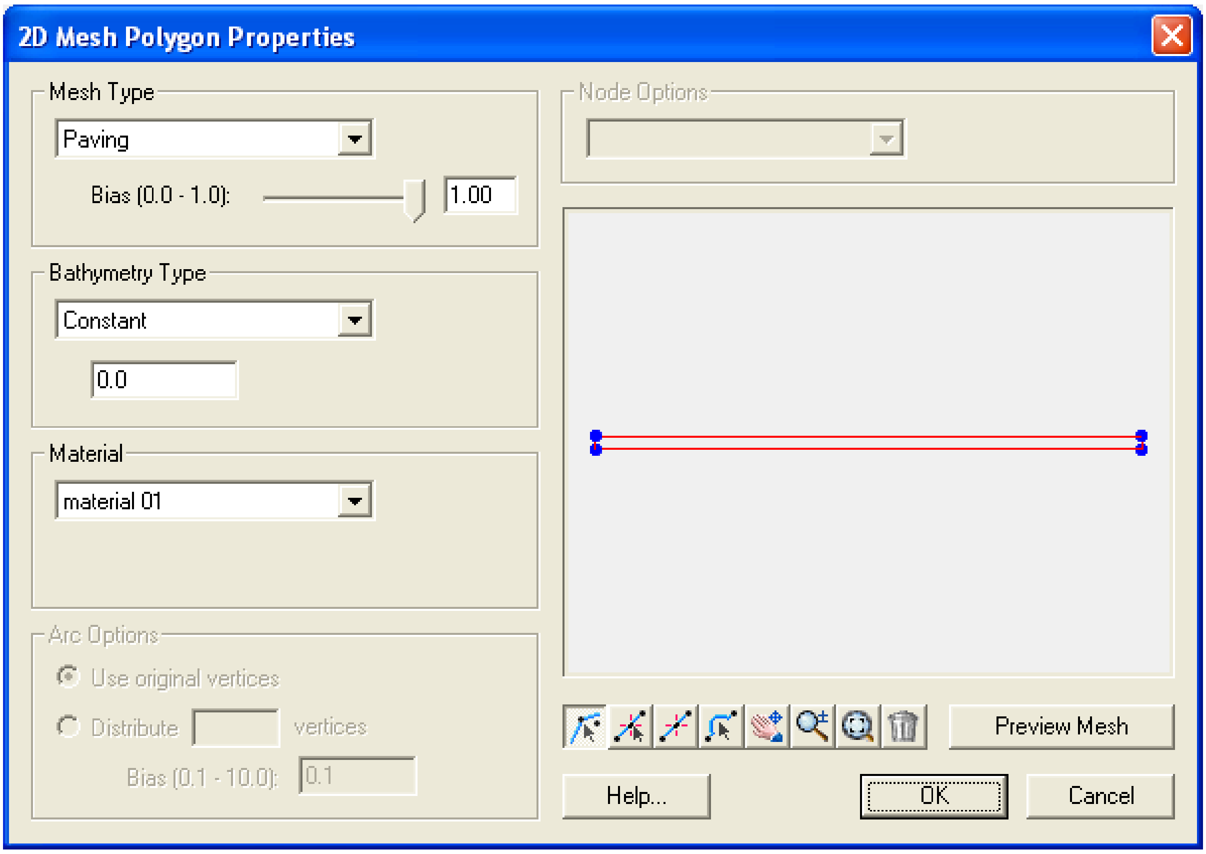
Note: more vertices along the polygon arcs = higher mesh resolution.

5 Modify polygons

A Using the "select feature polygon" button, double click on each of the polygons.



A dialog box will appear with many options. Check out the SMS manual for a better description, or try a few different options and see what happens.

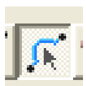


Some of the key options worth noting are:

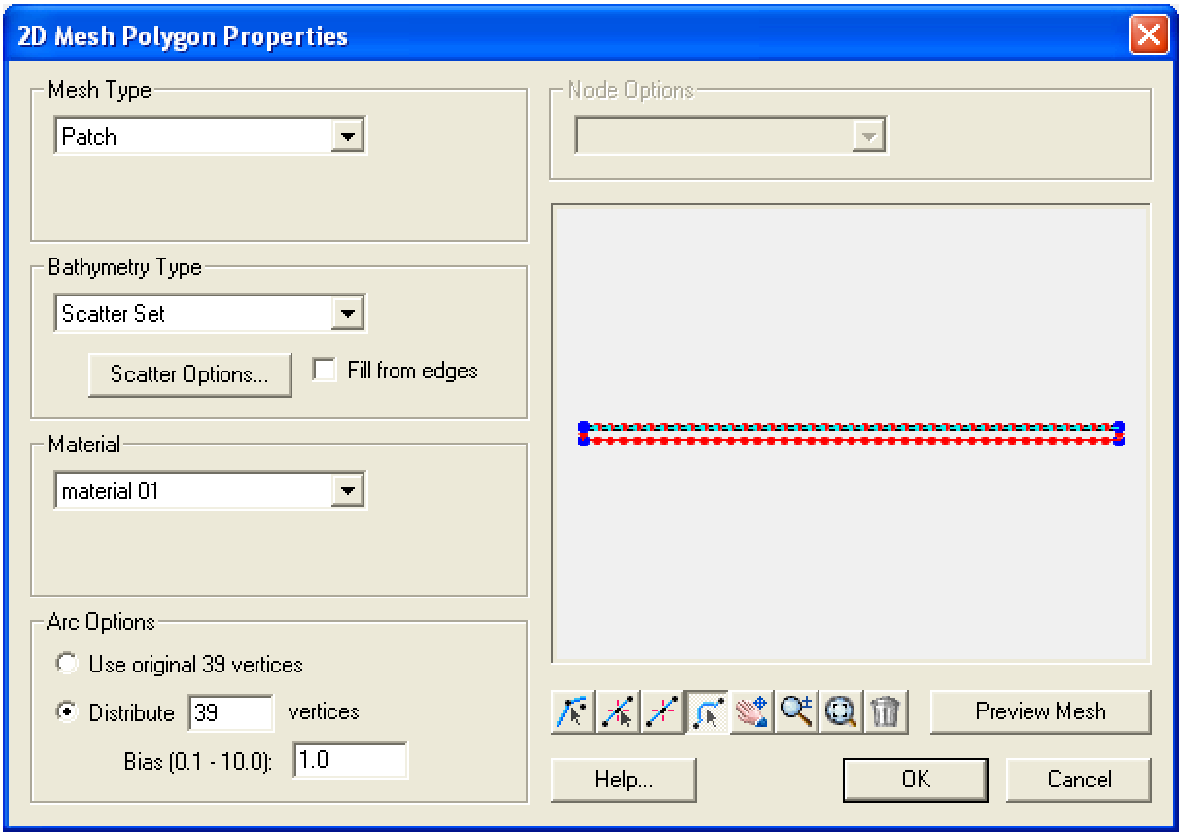
- Mesh type:
  - ➢ Paving is the classic triangular mesh, where triangles are used to fill the polygon area.
  - ➢ Patch fills the polygon area with a patch of quadrilateral (rectangular) elements. There are some limitations to using this mesh type (like having 4 arcs defining the polygon).
- Bathymetry Type:
  - ➢ Scatter Set will use the scatter data we have created in step ? to set the z values in the mesh
- Preview Mesh:
  - ➢ Use this to see how your mesh design looks for this polygon area.

Along the bottom of the display image is a series of buttons which let you adjust arc lines and the vertices that define them.

B For this model example we should adopt a resolution of 5 m across the channel and 25 m along the channel. Using the "Select Feature Arc" button, select the top arc.

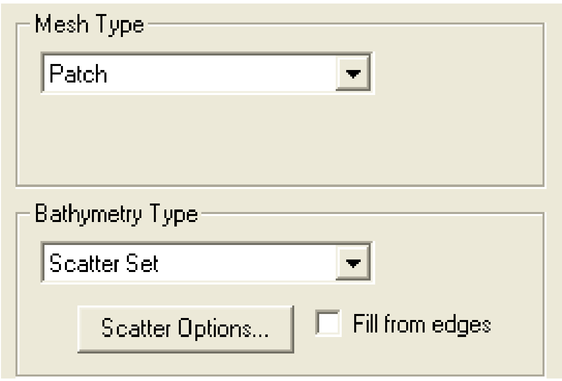


Then, adjust the number of vertices (the "Arc options" buttons) to suit the desired mesh resolution. In this instance, there should be 1,000 / 25 - 1 = 39 vertices. Repeat this for the bottom arc.

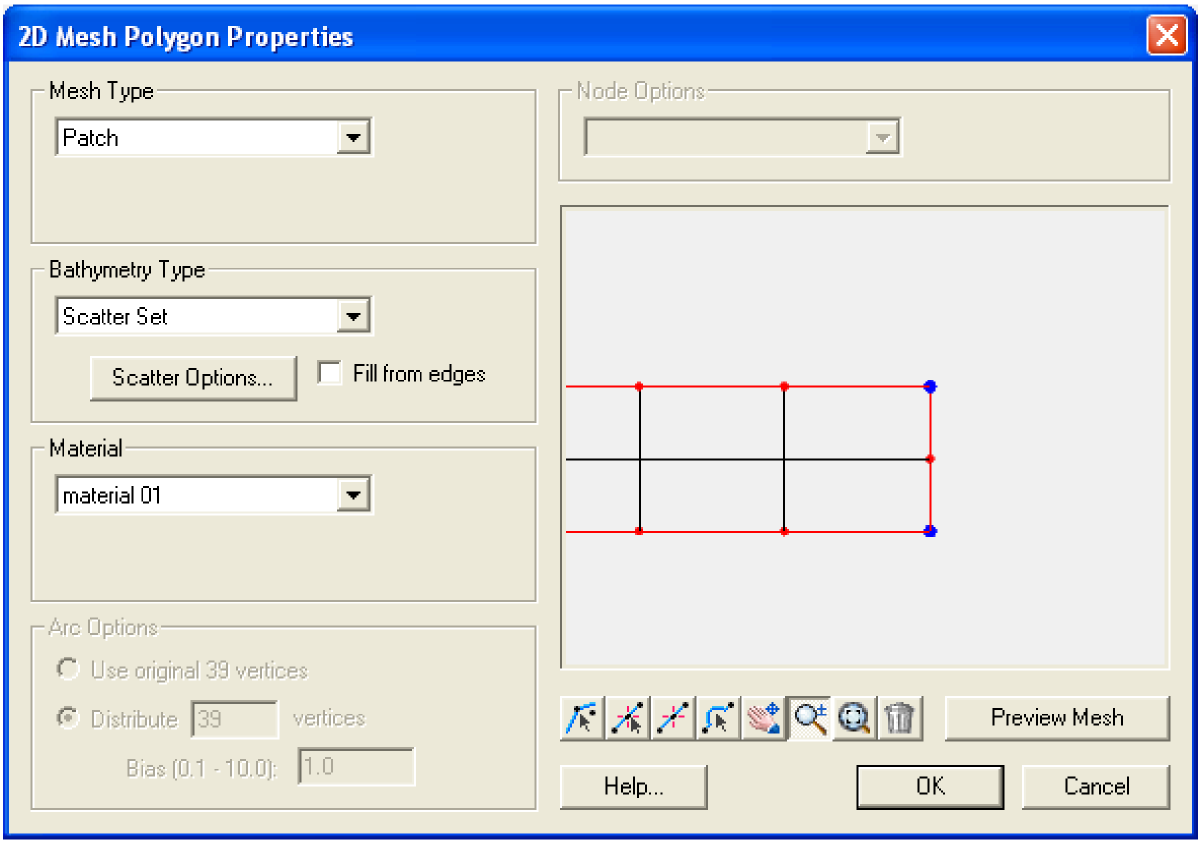


C Repeat for the left and right arcs, which will have 25 / 12.5 - 1 = 1 vertices. You may need to use the "zoom" button to assist with arc selection.

D Once this is done, check the "Bathymetry type" to be "scatter set". This will ensure that the z values previously entered into the scatter data will be interpolated onto the final mesh. For a straight trapezoidal channel such as this, a patch mesh type is the most efficient.

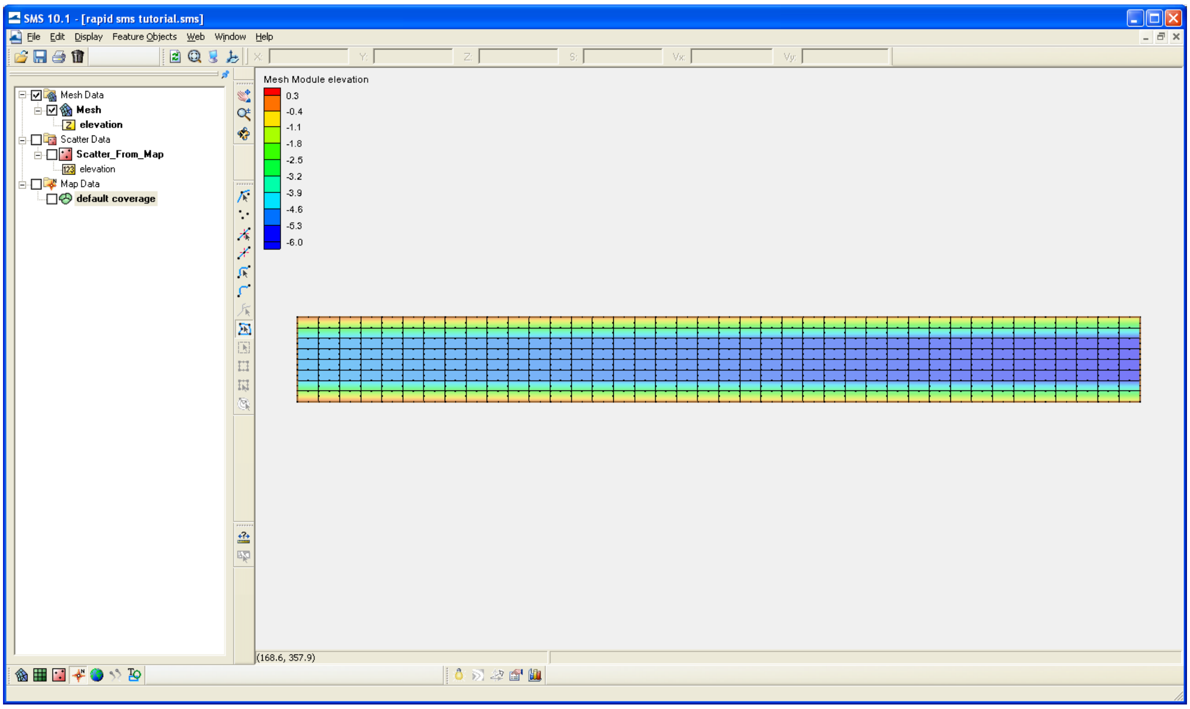


E Use the "Preview Mesh" to see what the mesh looks like. Use the zoom function to see the finer details.



F Once happy with the layout the mesh within this specific polygon, repeat with the remaining polygons. The middle polygon has 50 / 12.5 - 1 = 3 vertices across the channel and 1,000 / 25 -1 = 39 vertices along the channel. The lower polygon has the same vertice count as the top polygon. Note that as each polygon is edited, the arc vertices are updated – this highlights how the mesh generator tracks each polygon to ensure that the overall mesh is consistent.

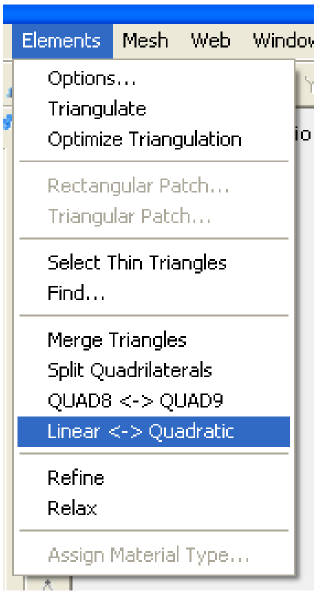
G Now repeat step 4, using the menu commands “Feature objects” – “Map -> 2D Mesh”, to create the mesh. This time, a reasonable looking mesh should appear.



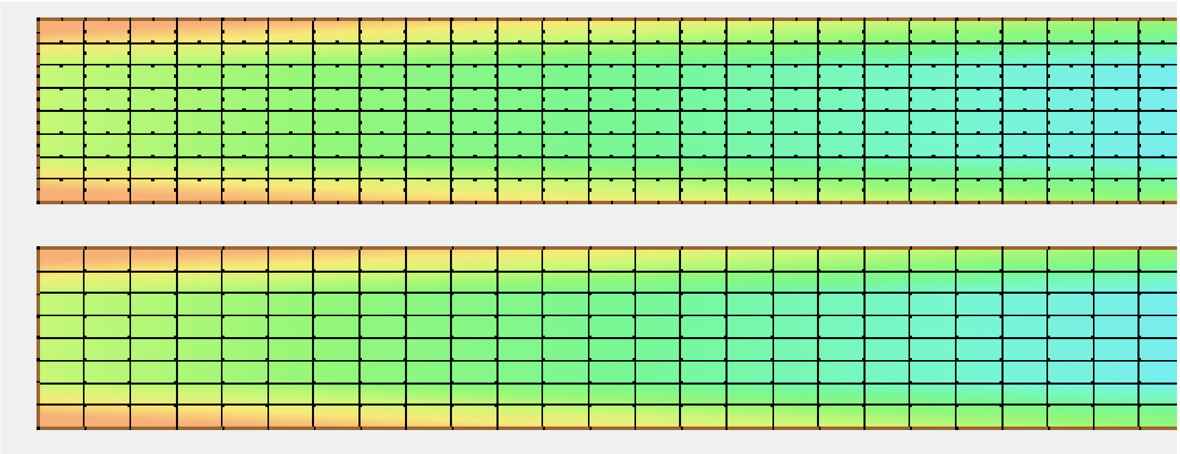
6 Linear elements

There are some final adjustments to be made prior to finishing the mesh creation.

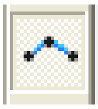
A The first is to switch the elements to be linear, rather than quadratic. Quadratic elements (not used by TUFLOW FV) are for finite element models that use mid-side nodes (such as RMA). Press the menu command "Elements" – "Linear <-> Quadratic" to remove the mid-side nodes.



The difference between linear and quadratic can be seen in the mesh display.



7 Nodestrings (boundary conditions)



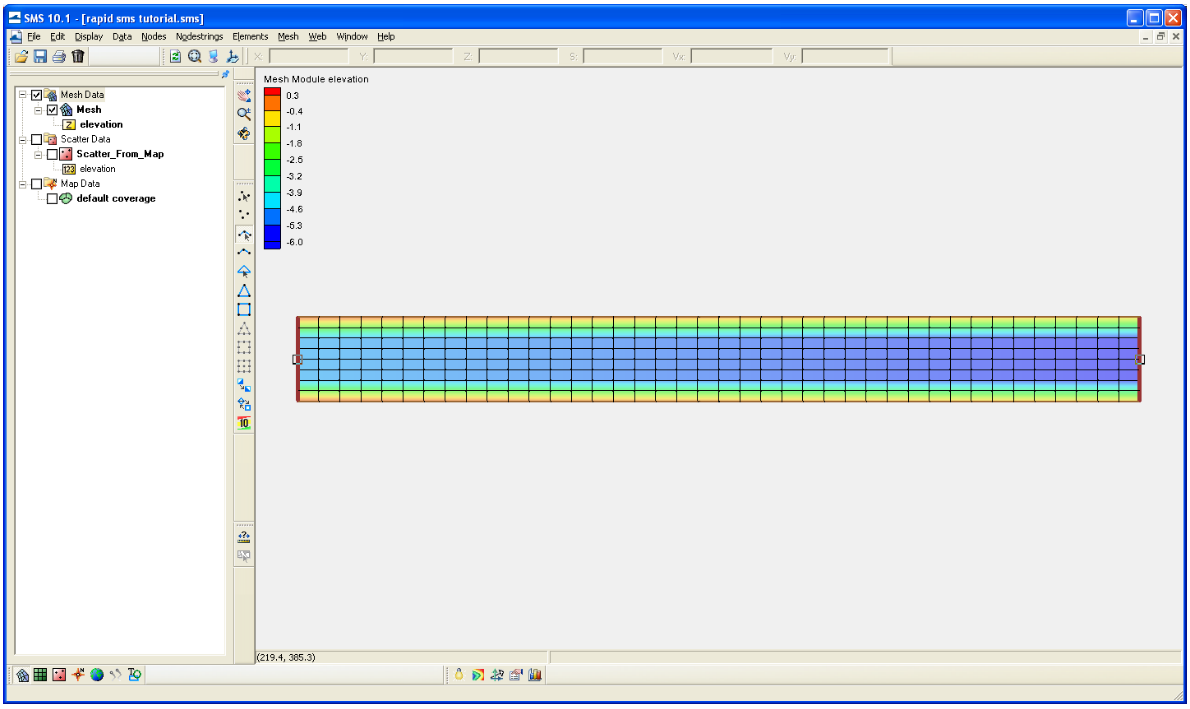
The last step is to insert nodestrings. Nodestrings are a string of nodes that can be used to define boundary conditions in TUFLOW FV (in SMS the nodestrings have a number of other functions not used by TUFLOW FV). For this example, there will be an upstream and a downstream boundary condition applied (ie along the left and right edges of the model domain).

A Press the "Create nodestring" button, then click along the nodes that make up the left edge of the mesh. Then create a second nodestring along the right edge of the mesh.

Hint – hold the "shift" button down to select all nodes between first clicked and second clicked nodes.

These nodestrings are used to specify boundary conditions at a later time (see Section 5.2).

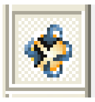
Nodestrings should all be created from right to left while looking downstream.



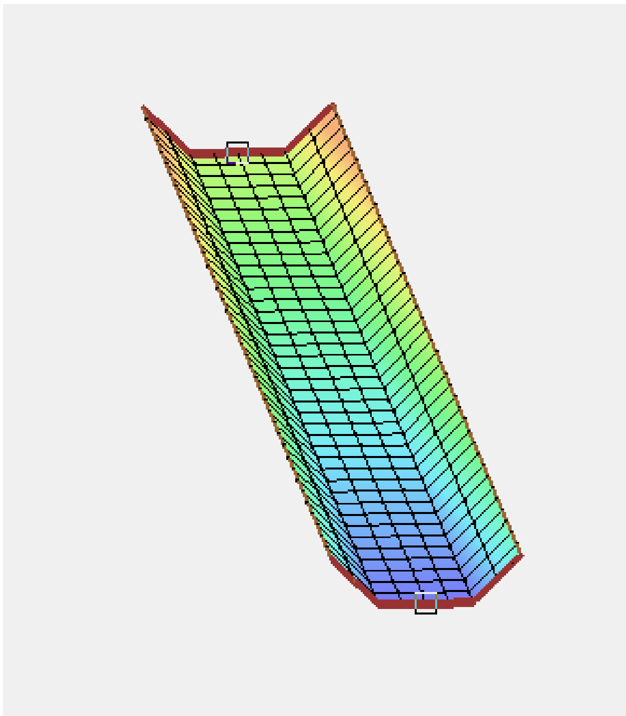
Save now. You have completed the construction of a mesh, congratulations.

8 Visualise

A The best way to admire your handiwork is to use the “Rotate” button.



This allows you to visualise the mesh in perspective view.



5.2 A quick TUFLOW FV model setup

The following example takes the model mesh of a trapezoidal channel created in Section 5.1 and sets up, runs and visualises a hydrodynamic simulation.

Specifications for the model setup of Flume 2 are as follows:

- The bed is lined with a coarse concrete; a Manning friction of 0.014
- There is a constant upstream inflow of 419.089m^3/s
- The downstream water level is 78.401 m above the bed in super-critical and the downstream water level is 77.937 m in subcritical.

1 Establish a folder structure

The first step is to establish a folder structure (see Section 3.8). So for a TUFLOW FV model project called “quick tutorial”, the folder structure will be:

```
\---quick tutorial
    +---bc
    +---geo
    +---input
    +---output
    \---results
```

Place the .2dm file created in Section 5.1 into the “geo” folder.

2 Work out nodestring order

TUFLOW FV uses the nodestrings as boundaries. The nodestring ID specified in TUFLOW FV is the same as the ID listed in the 2dm file. In the SMS interface, click on a nodestring (using the nodestring select tool). The info bar along the bottom of SMS will show you what the nodestring ID is.

Alternatively, open the 2dm file in a text editor and look for the nodestrings. Do this by searching for “NS” at the start of the line. For the 2dm file from Section 5.1, the NS lines are as follows:

```
NS  83 1 42 369 368 367 246 245 -244 1
NS  241 242 243 364 365 366 82 41 -123 2
```

The first NS string listed here has a negative number at the end (which signals that this is the end of the nodestring) and then the nodestring ID, which in this case is 1. Similarly, the second nodestring listed has an ID = 2.

Don’t worry if the nodes listed in the nodestring are in reverse order to that shown; this doesn’t influence their behaviour when used as boundary conditions (TUFLOW FV considers positive flow to be always entering a model domain, and negative flow leaving a model domain).

3 Create boundary condition files

For TUFLOW FV, separate csv format files contain boundary conditions. There is typically one file for each boundary. See Section 8.4.10.

In this case, the boundary conditions are very simple because the run is steady state.

The flow boundary (called “steadyQ.csv”) should contain the following:

```
Time,Flow
0,0
1,100
2,450
6,450
```

Note that the first column (time) is in hours. Note also that there is a warm-up period of 2 hours; see Section 7.4.3 for a discussion on this (or, try removing the warmup by putting a constant 450 m^3/s and see what happens!).

The water level boundary (called “steadyWL.csv”) should contain the following:

```
Time,WL
0,-3.50
24,-3.50
48,-3.50
```

Both files should be placed in the folder “bc”.

4 Create the FVC control file

Often, an fvc file is created from an earlier model or from a template. If using a template then it’s good practice to comment out the irrelevant commands. A “!” at the start of the line means that the line is not read by TUFLOW FV. This allows you to insert comments into your fvc file (this is recommended).

To simplify this example only those lines that are relevant to this simulation are shown in the fvc file. For this tutorial example, the file is called “trap_steady_01.fvc”.

The fvc file is shown below. A description of each entry is provided; for further information see Section 0.

5 FVC File Contents

FVC File Contents	Description
`! TUFLOW FV TUTORIAL` `! Flow along a trapezoidal channel`	The first 2 lines are a description of the model simulation. You may also wish to include the initials of the modeller, etc.

```
! TIME COMMANDS
start time == 0.0
end time == 6.0
cfl == 1.0
timestep limits == 0.0001, 10.
```

The time commands include the start and end times (the default time format is Hours). The CFL limit is 1 by default – TUFLOW FV then assigns a timestep at each computational step according to the CFL limit and between the ranges specified in the timestep limits.

```
! MODEL PARAMETERS
stability limits == 10. ,100.
momentum mixing model == Smagorinsky
global horizontal eddy viscosity == 0.2
```

The model parameters are those that control various physical and numerical processes.

When the stability limits are exceeded (water level first, then velocity), the model is considered to have crashed. Note that the velocity limit here is high – that's because the velocities along the wetting and drying boundary edges are high.

A Smagorinsky eddy viscosity approach has been specified, with a Smagorinsky factor of 0.2.

```
! GEOMETRY
geometry 2d == ..\geo\quick tutorial.2dm
```

The model geometry is the 2dm created in Section 5.1.

```
! MATERIAL PROPERTIES
material == 1
  bottom roughness == 0.018
end material
```

So far, material types haven't been highlighted. By default, SMS will create elements using a single material type (1). It is this material type that is assigned a bottom roughness of 0.018 (the default friction approach is a Manning's number).

```
! INITIAL CONDITIONS
initial water level == -3.5

units == english
```

The initial condition is 2.5 m above the bed at the downstream end (ie -3.5 m).

The unit is in feet. Roughness parameters will be automatically converted in the appropriated unit.

```
! BOUNDARY CONDITIONS
bc == Q, 1, ..\bc\steadyQ.csv
  bc header == time,flow
end bc

bc == WL, 2, ..\bc\steadyWL.csv
  bc header == time,WL
end bc
```

The boundary conditions link the csv files containing the actual flows and water levels to the nodestrings. Nodestring 1 is assigned a flow boundary and nodestring 2 is assigned a water level boundary.

```
! OUTPUT COMMANDS
output dir == ..\Output\
output == datv
```

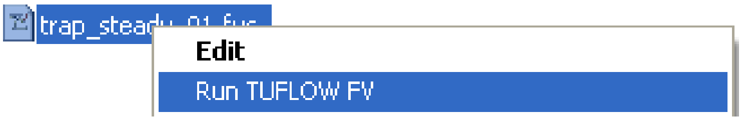
In this instance, a datv format file is specified. This format is easily read into SMS

```
  Output Parameters == h,v,d
  Output Interval == 600
end output
```

for viewing. The h, v and d mean that outputs files containing water level, velocity and water depth will be created.

6 Run TUFLOW FV

Once you’re happy with the fvc file contents, run TUFLOW FV. See Section 3.4 for information on how to do this – a right click from Explorer is a straightforward way.



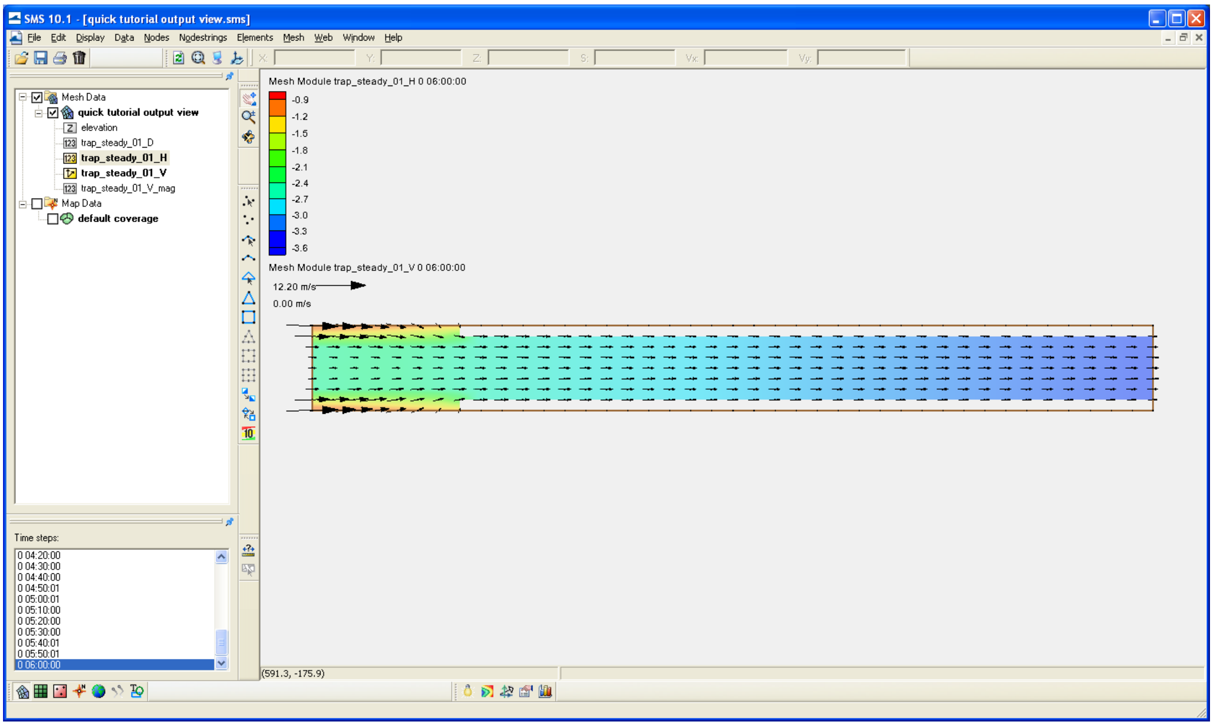
You may find that your simulation has crashed, or some other syntax error in the inputs has caused it to stop. If this happens, open the log file to see what may have gone wrong. Be logical and thoughtful in your model preparation; often it’s a simple mistake that causes the most frustration. See Section 7.4 for advice.

7 Check Results

During the model simulation the result files will be written; one for the water levels (“_H”.dat), one for velocities (“_V.dat”) and another for water depths (“_D.dat”). They will have the same prefix as the fvc file; in this example they will be called:

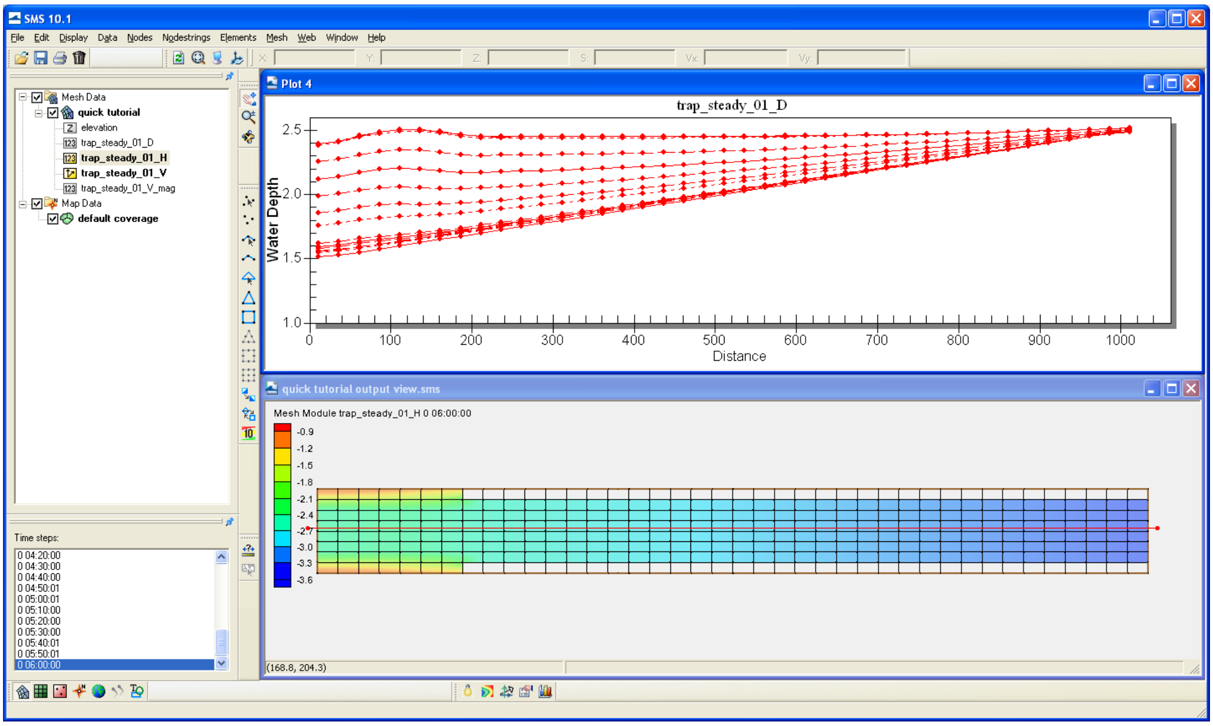
```
trap_steady_01_H.dat
trap_steady_01_V.dat
trap_steady_01_D.dat
```

To view them, start SMS and open the 2dm file. Then, from either the SMS menu or by dragging into the SMS window, open the dat files. See the SMS manual for advice on viewing results files.



In this instance the results can be checked by comparing to the Manning's equation ($Q = 1/n\ A\ R_h^{2/3}\ S^{1/2}$). For a flow rate of 450 m^3/s the normal water depth is approximately 2.5 m. Thus, in this example, there should be a reasonably constant water depth along the length of the channel at the end of the simulation.

There are a range of display options in SMS; the following display shows a longitudinal profile of water depths throughout the simulation. To do this, a feature arc needs to be created in the Map module (the type of this coverage needs to be "Observation"). Then, the "Display" – "Plot wizard" menu is used.



As shown, at the end of the simulation there is almost a constant water depth of 2.5 m – success, TUFLOW FV is replicating the Manning's equation!

5.3 Inclusion of Salinity

It is relatively straightforward to include a conservative tracer into the model simulation. The following additional components are required:

8 Update lines in FVC File — Description

```
! TUFLOW FV TUTORIAL
! Flow along a trapezoidal channel + AD

! SIMULATION CONFIGURATION
include salinity == 1,0


! TIME COMMANDS
start time == 0.0
end time == 6.0
cfl == 1.0
timestep limits == 0.0001, 10.

! MODEL PARAMETERS
stability limits == 10. ,100.
momentum mixing model == Smagorinsky
global horizontal eddy viscosity == 0.2
```

Include salinity as a model parameter (the first number = 1), but decoupled from the density simulations (the second number = 0).

`Scalar mixing model == constant` `Global horizontal scalar diffusivity == 1`	The scalar mixing model and diffusivity are specified as model parameters.

`! GEOMETRY`
`geometry 2d == ..\geo\quick tutorial.2dm`
 	
`! MATERIAL PROPERTIES`
`material == 1`
`  bottom roughness == 0.018`
`end material`
 	
`! INITIAL CONDITIONS`
`initial water level == -3.5`	

`Initial Salinity == 0`	The initial concentration is 0.

`! BOUNDARY CONDITIONS`
`bc == Q, 1, ..\bc\steadyQS.csv`
`  bc header == time,flow,Sal`
`end bc`
`bc == WL, 2, ..\bc\steadyWLS.csv`
`  bc header == time,WL,Sal`
`end bc`
	An additional column in the boundary condition files is required, specifying the concentration at the boundary.

`bc == QC, 240,55, ..\bc\cellQ.csv`
`  bc header == Time,flow,Sal`
`end bc`
	A new boundary condition (QC) defines a constant inflow into an element (or cell). The numbers 240,55 are the x,y coordinates where the inflow will occur.

`! OUTPUT COMMANDS`
`output dir == ..\Output\`
`output == datv`
`  Output Parameters == h,v,d,Sal`
`  Output Interval == 600`
`end output`
	An additional output parameter is specified (Sal).

9 Update boundary condition files

The updated flow boundary (called “steadyQS.csv”) should contain the following:

```
Time,Flow,Sal
0,0,0
1,100,0
2,450,0
6,450,0
```

The water level boundary (called “steadyWLS.csv”) should contain the following:

```
Time,WL,Sal
0,-3.50,0
24,-3.50,0
48,-3.50,0
```

The cell inflow boundary (called “cellQ.csv”) should contain the following:

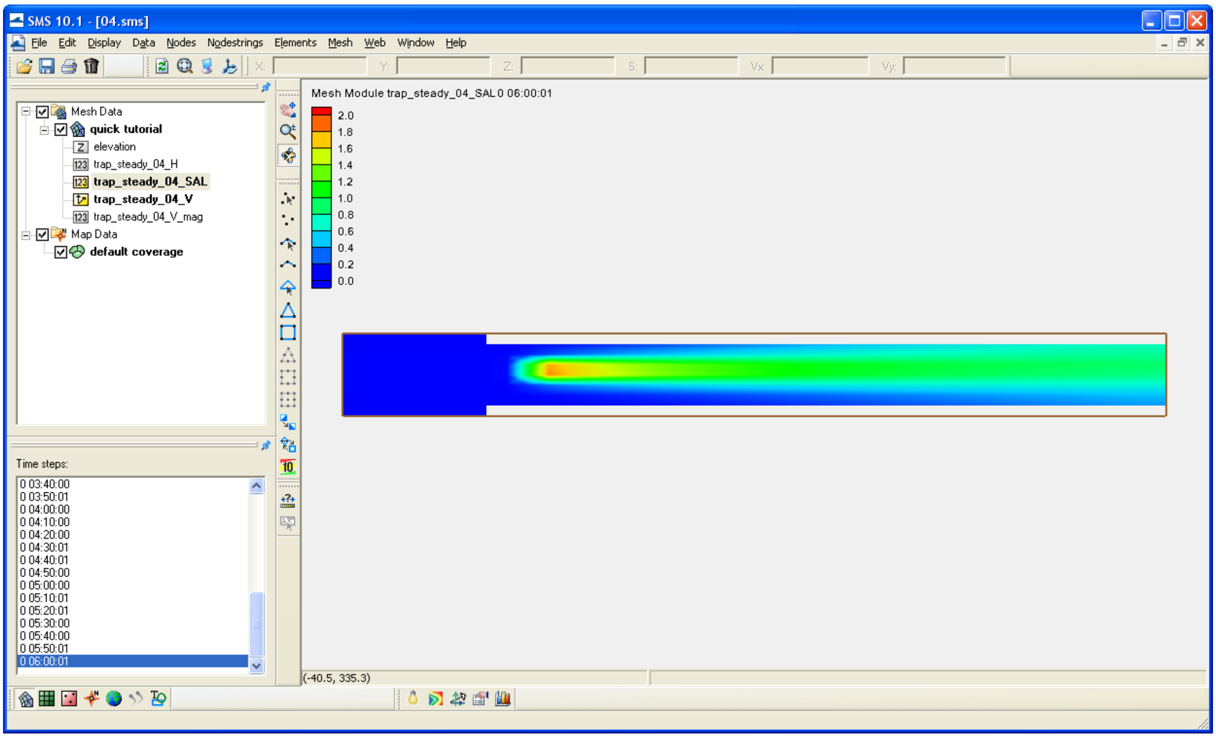
```
Time,Flow,Sal
0,0,30
1,10,30
2,10,30
6,10,30
```

10 View results

The output file with concentrations will have the extension “_SAL.dat”:

```
trap_steady_01_SAL.dat
```

The results view in SMS should look something similar to the following:

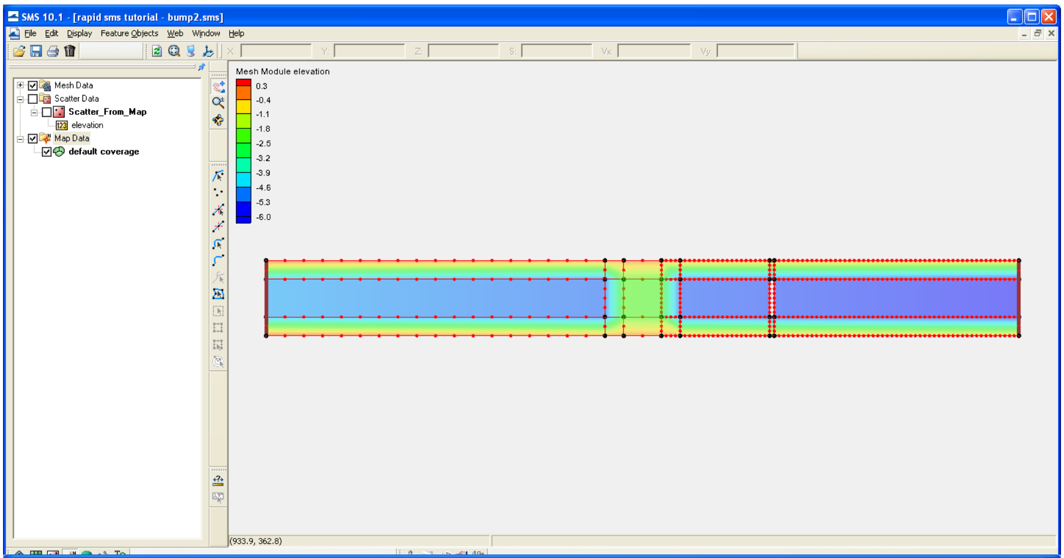


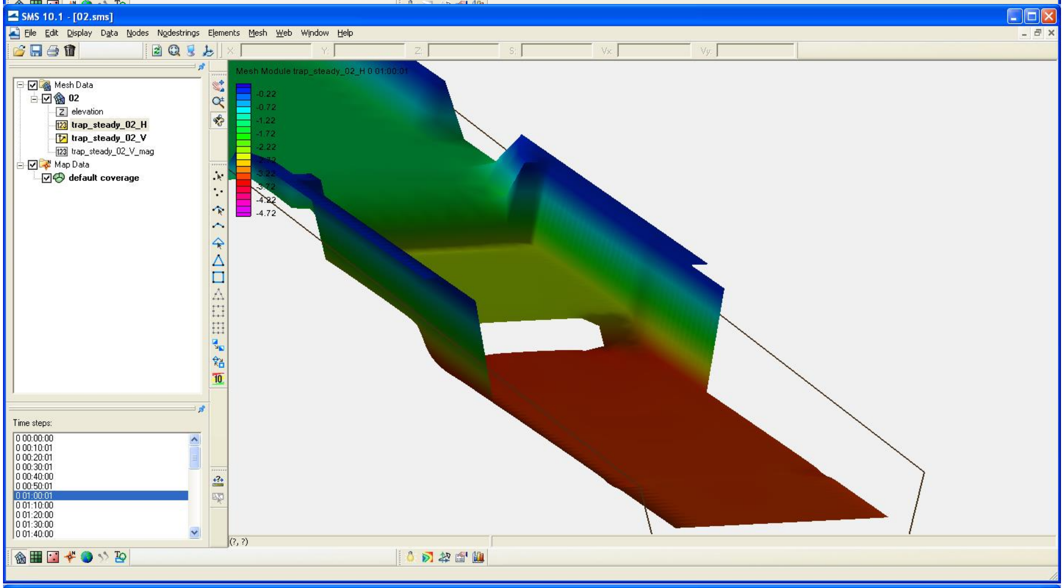
5.4 Going further

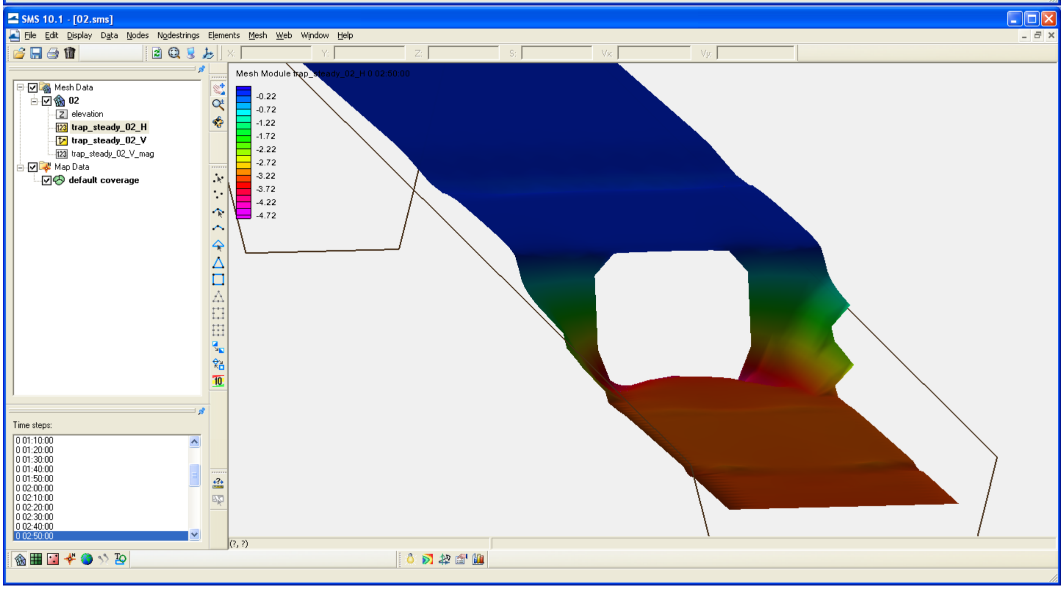
For modellers with a firm grasp of the basics of modelling, the best way to learn how to setup and run TUFLOW FV is to experiment. In the following example, the mesh created in Section 5.1 has been adjusted to include a “bump” in the centre and a constriction further downstream, which will induce transitions to supercritical flow. Mesh resolution has been increased around these features.

The results are far more interesting using this mesh design (see following page).

Additional tutorials are available on the website. Check out www.tuflow.com.


SMS 10.1 - [rapid sms tutorial - bump2.sms]
File Edit Display Feature Objects Web Window Help
Mesh Data
Scatter Data
Scatter_From_Map
elevation
Map Data
default coverage
Mesh Module elevation
0.3
-0.4
-1.1
-1.8
-2.5
-3.2
-3.9
-4.6
-5.3
-6.0
(933.9, 362.8)


SMS 10.1 - [02.sms]
File Edit Display Data Nodes Nodestrings Elements Mesh Web Window Help
Mesh Data
02
elevation
trap_steady_02_H
trap_steady_02_V
trap_steady_02_V_mag
Map Data
default coverage
-0.22
-0.72
-1.22
-1.72
-2.22
-4.72
Time steps:
0 00:00:00
0 00:10:01
0 00:20:01
0 00:30:01
0 00:40:00
0 00:50:01
0 01:00:01
0 01:10:00
0 01:20:00
0 01:30:00
0 01:40:00
(?, ?)


SMS 10.1 - [02.sms]
File Edit Display Data Nodes Nodestrings Elements Mesh Web Window Help
Mesh Data
02
elevation
trap_steady_02_H
trap_steady_02_V
trap_steady_02_V_mag
Map Data
default coverage
-0.22
-0.72
-1.22
-1.72
-2.22
-2.72
-3.22
-3.72
-4.22
-4.72
Time steps:
0 01:10:00
0 01:20:00
0 01:30:00
0 01:40:00
0 01:50:00
0 02:00:00
0 02:10:00
0 02:20:00
0 02:30:00
0 02:40:00
0 02:50:00
(?, ?)

6 Tutorial models

6.1 Where are they?

Check out www.tuflow.com for a series of tutorial models that can be downloaded and analysed. Note that the quick tutorial described in 5 is not available; the emphasis of the quick tutorial is to demonstrate the steps taken rather than the end result.

A description of the tutorial exercises is provided here.

6.2 Simple River Bend: Using SMS Interface

In this tutorial a simple model of a short section of river is created using the SMS TUFLOW FV interface. The setup of the SMS TUFLOW FV interface is described in Section 3.10. Please follow the configuration steps in this section before starting this tutorial.

For this model we will be building a mesh for an inbank area of a river, we will be applying an upstream inflow boundary and a downstream tidal boundary.

Before we start to create the TUFLOW FV mesh we need to load the TUFLOW FV model definition in SMS if it isn't already loaded. To do this open the TUFLOW_FV.2dm provided as part of the SMS interface.

NOTE: The Define Model is used to create / modify the interface, this should not be modified by the user and is password protected. If this is modified, the conversion process is highly likely to fail.

6.2.1 Data Provided

For this tutorial the following datasets have been provided.

- Bathymetry data, this is provided as a SMS Scatter TIN dataset
- Land-use areas, provided as SMS Map Coverage
- Boundary condition data, in comma separated variable (.csv) format

The SMS data (bathymetry and land use data) are shown below in Figure 6-1 and Figure 6-2 respectively. Load the bathymetry data (RiverBend_Bathymetry.tin) and land use data (RiverBend_LandUse.map) in SMS. When loaded correctly the table of contents in SMS, should contain the scatter dataset containing the bathymetry and a map dataset containing the land use polygons as shown in Figure 6-3.

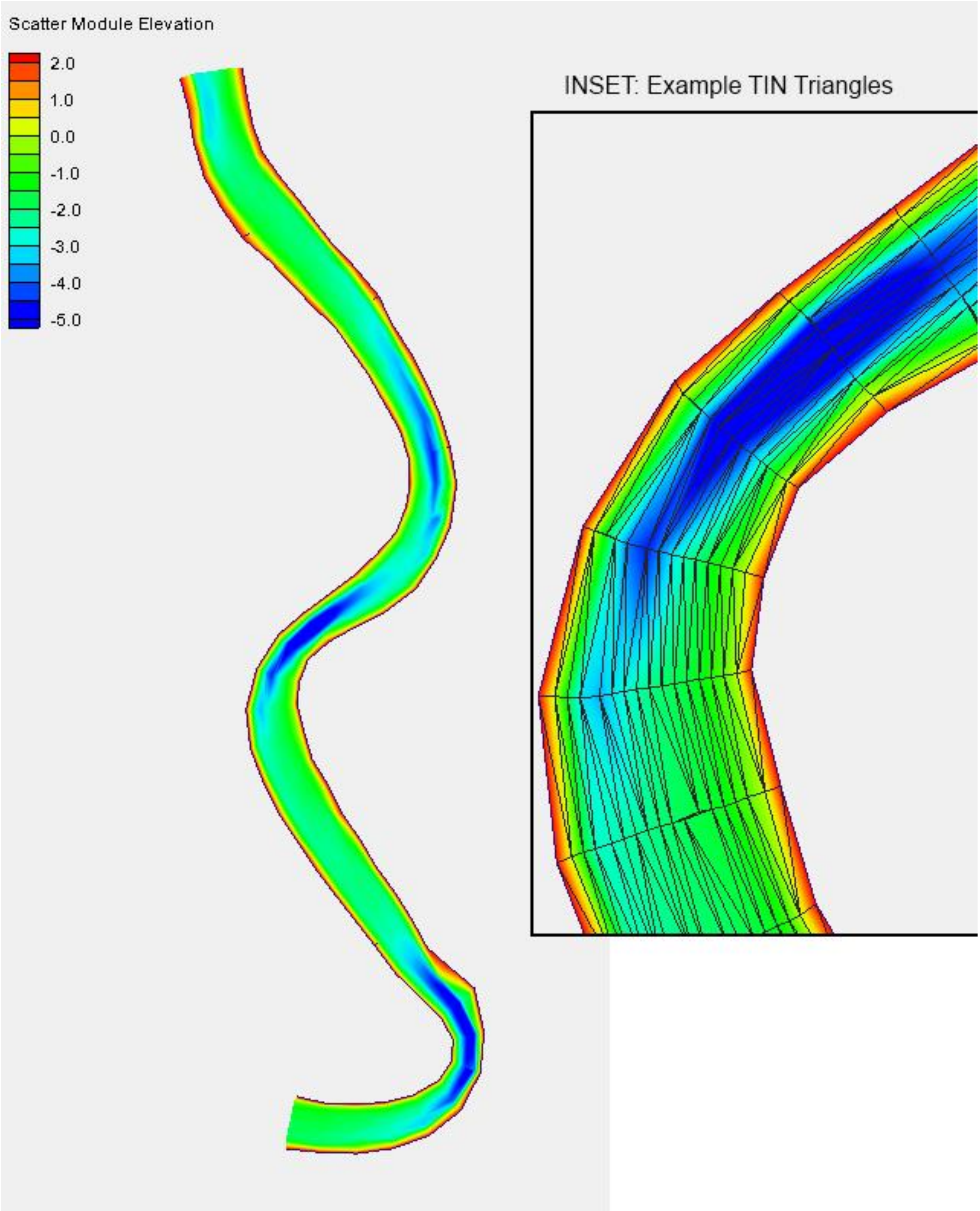


Figure 6-1 River Bend Tutorial: Bathymetry Data

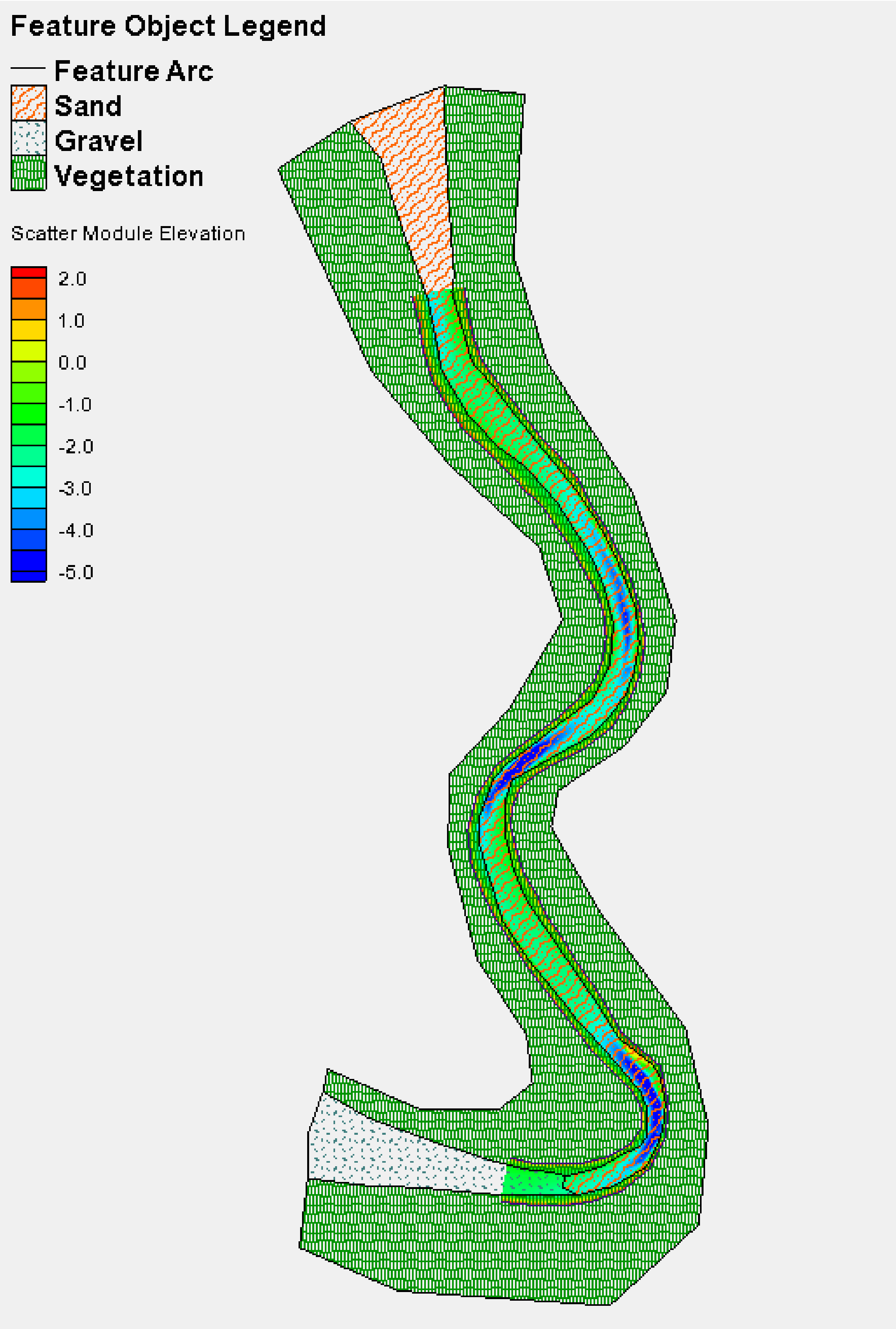


Figure 6-2 River Bend Tutorial: Land Use Data

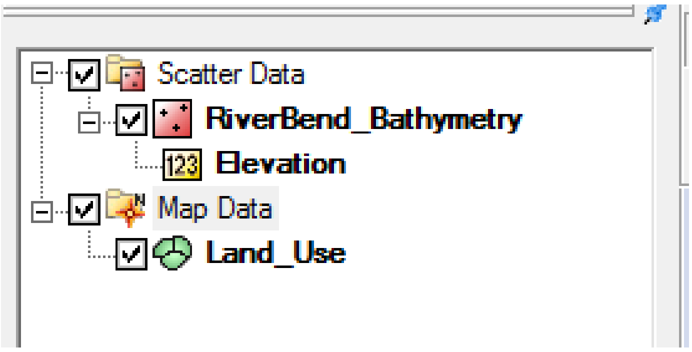
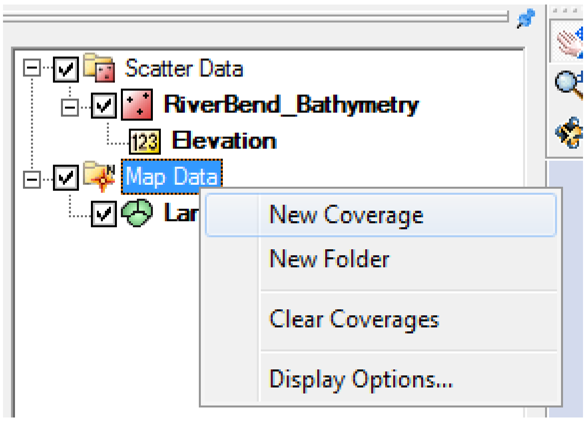


Figure 6-3 River Bend Tutorial: Table of Contents in SMS

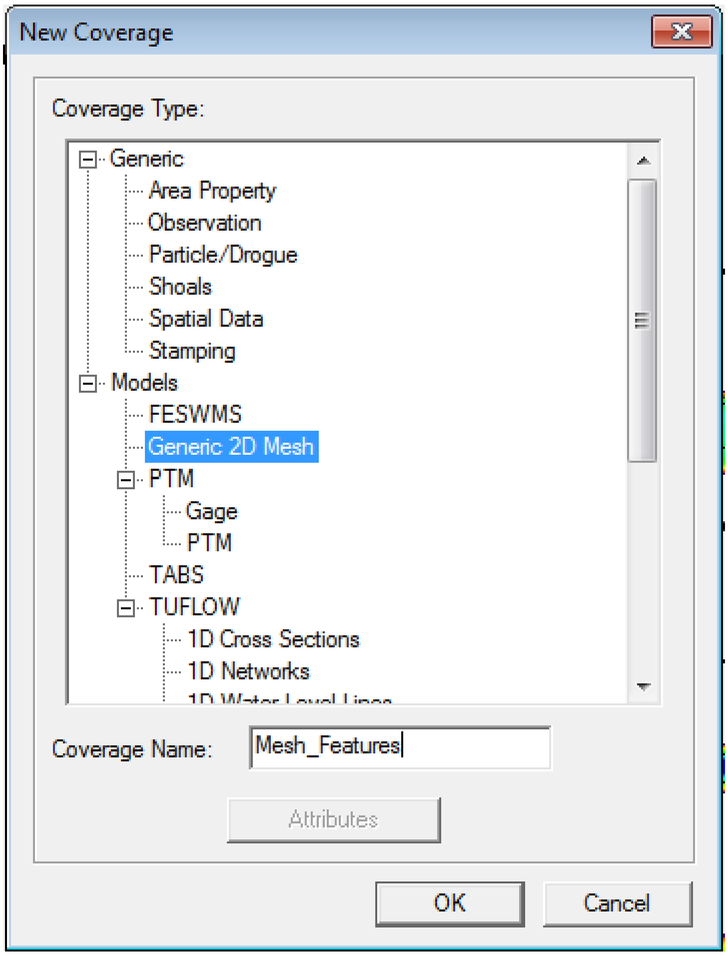
6.2.2 Mesh Creation

Before starting it is a good idea to save a project in SMS, when loading the project all the base data will be loaded. To save a project, select File >> Save as… Save the project as RiverBend_Mesh001.sms and ensure the file is being saved as a project file (.sms).

Now that the required datasets are loaded we can begin to create the model mesh. We need to create a new coverage in the map module. Right click on the "Map Data" heading and select "New Coverage" as shown below.

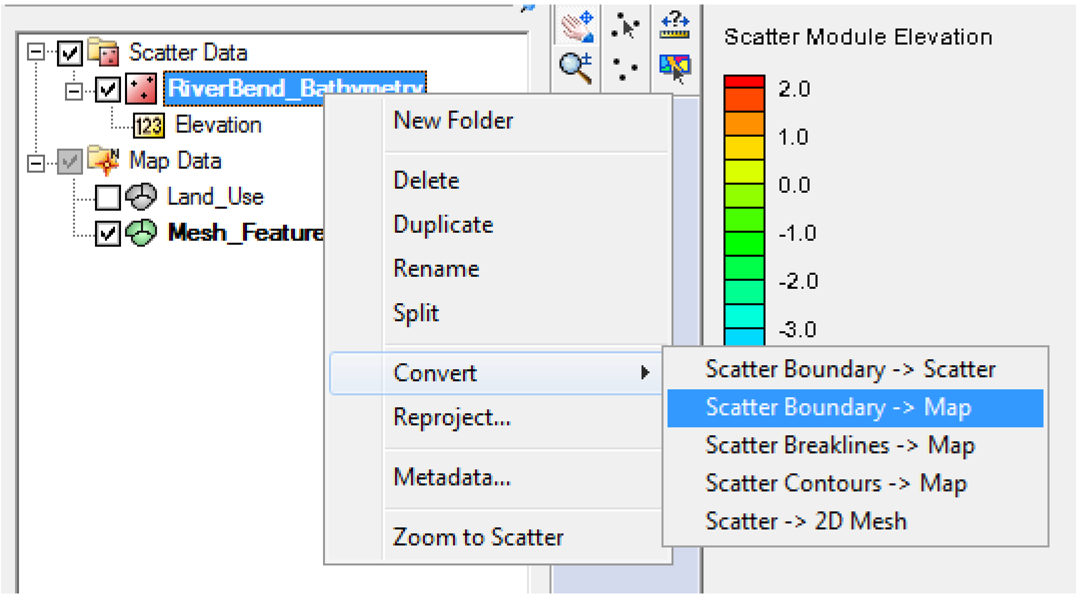


The coverage type should be set to Models >> Generic 2D Mesh. Name the coverage "Mesh_Features" and then select OK.

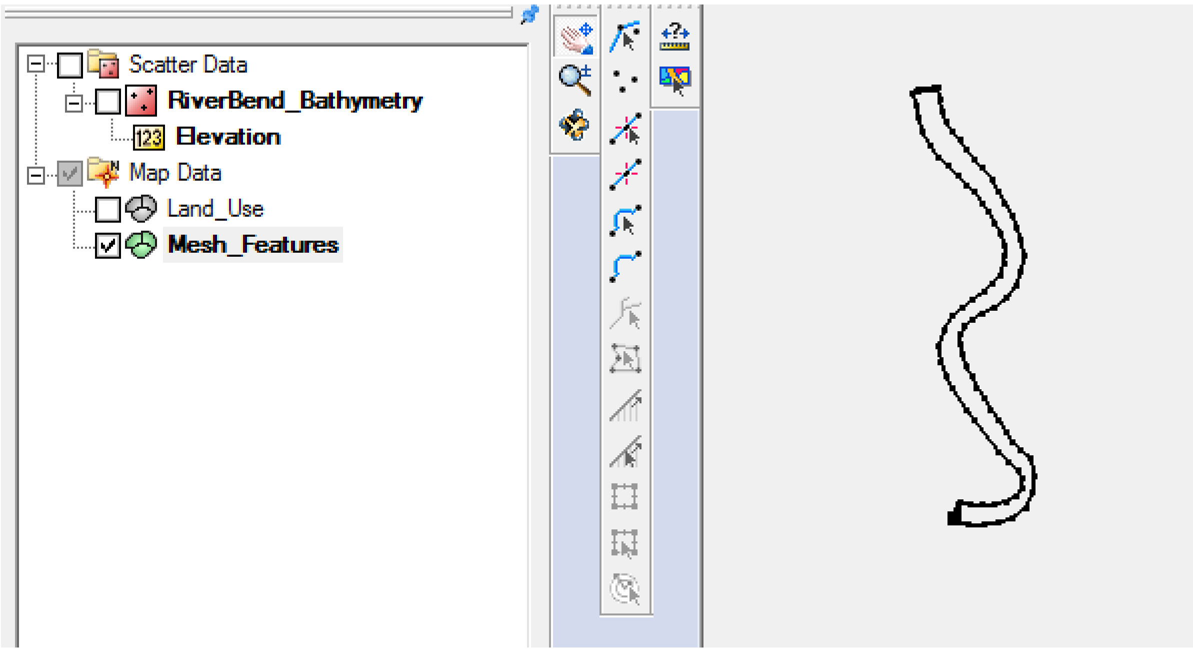


Make sure the newly create "Mesh_Features" layers is selected in the table of contents as per the image below.

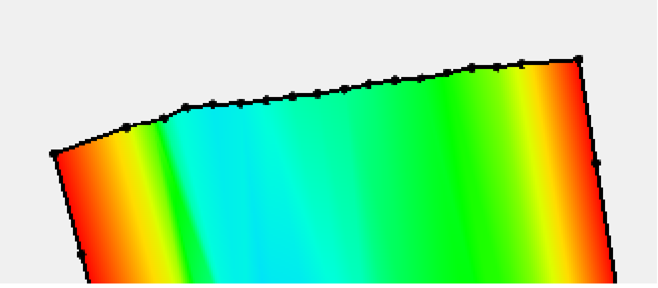
In this layer we need to create a feature arc (polyline) in SMS to define our model extent. This can be done using the create feature arc button (). However, in this case the model we are going to create covers the full extent of our bathymetry set, so we can use the extent of the bathymetry in defining the model extent, this needs to be converted into an object in our "Mesh_Features" layer. To do this right click on the RiverBend_Bathymetry scatter dataset and the select convert >> Scatter Boundary >> Map (this step is shown below).



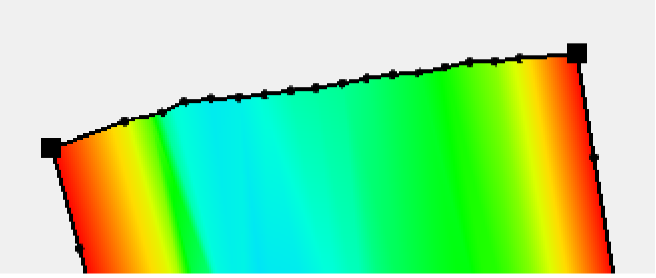
After the conversion, the scatter dataset boundary should be in the Mesh_Features layer (this is easier to see with the scatter set turned off). This is shown in the image below.



Zoom in to the northern boundary of the model, and select the two corner vertices () and convert these to nodes.

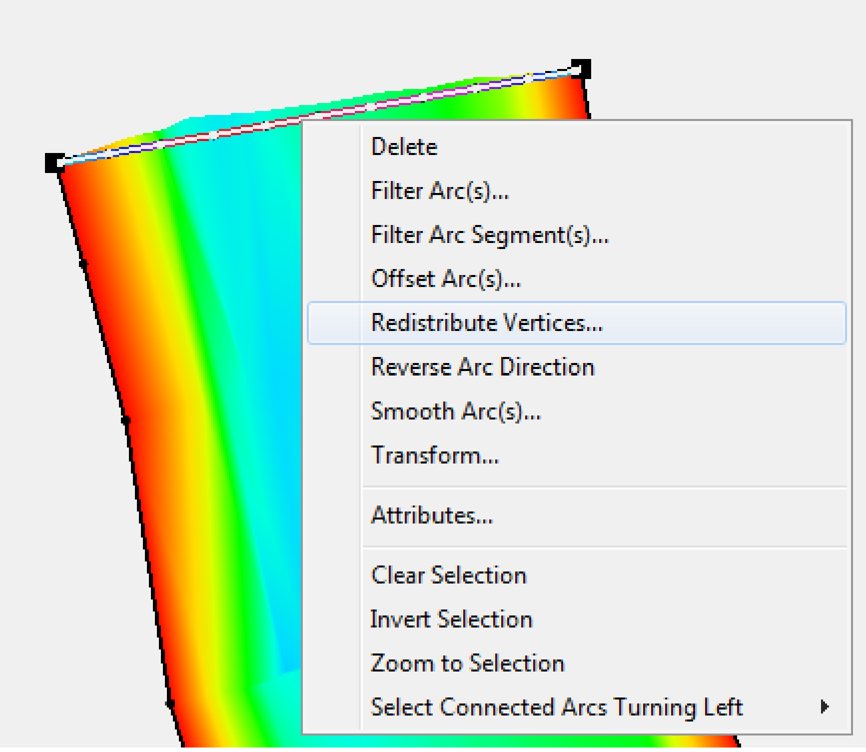


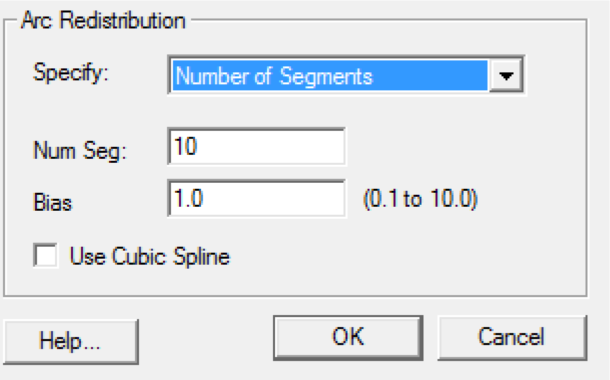
Before



After

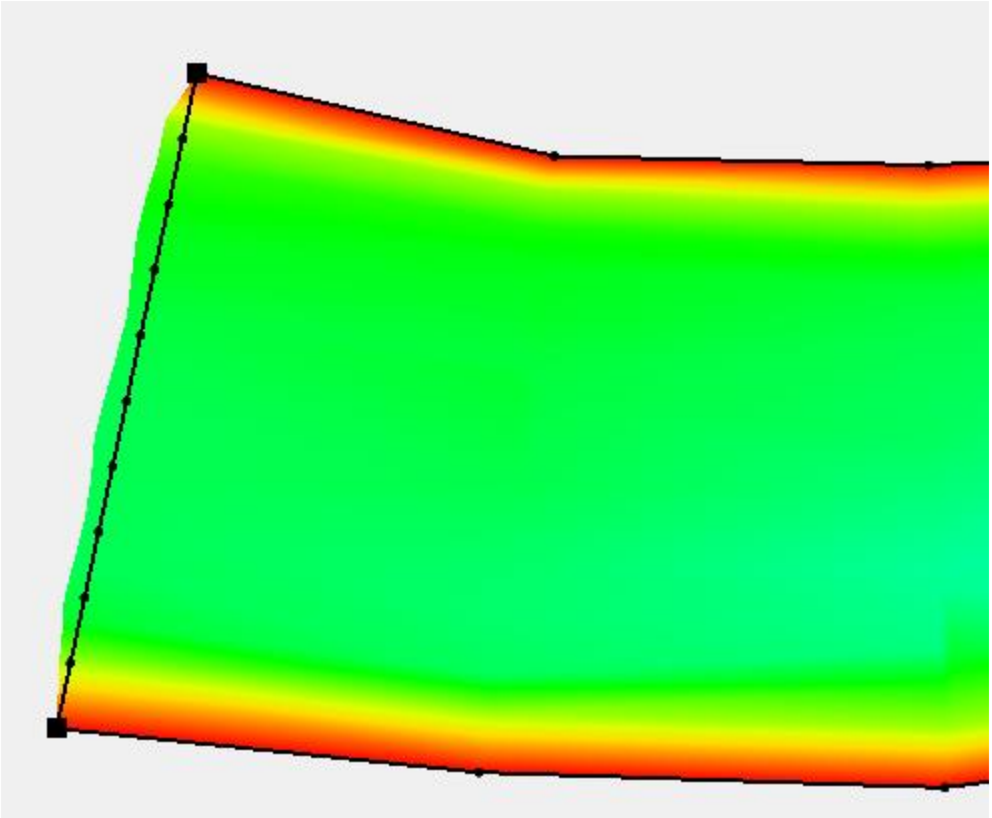
Select the feature arc () and then right click and use the redistribute vertices to redistribute 10 vertices along the line.





Redistribute 10 vertices across the channel.

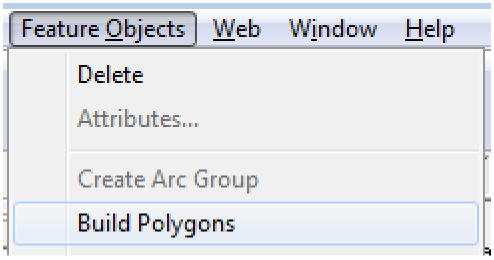
Repeat the process at the southern edge of the model.



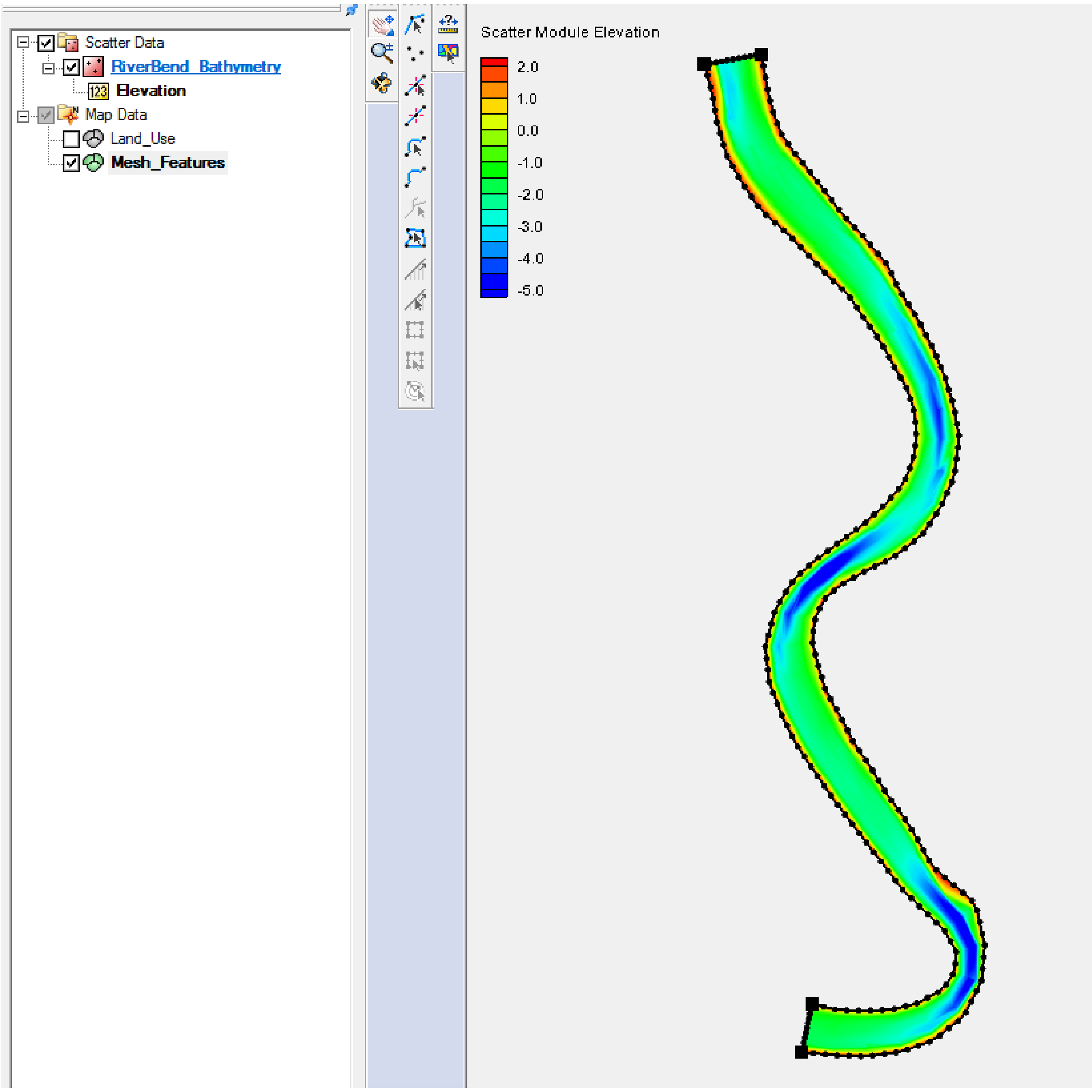
Redistributed vertices at Southern Edge of model.

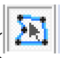
Select the two feature arcs along the banks of the river and redistribute with a specified spacing of 20 (metres).

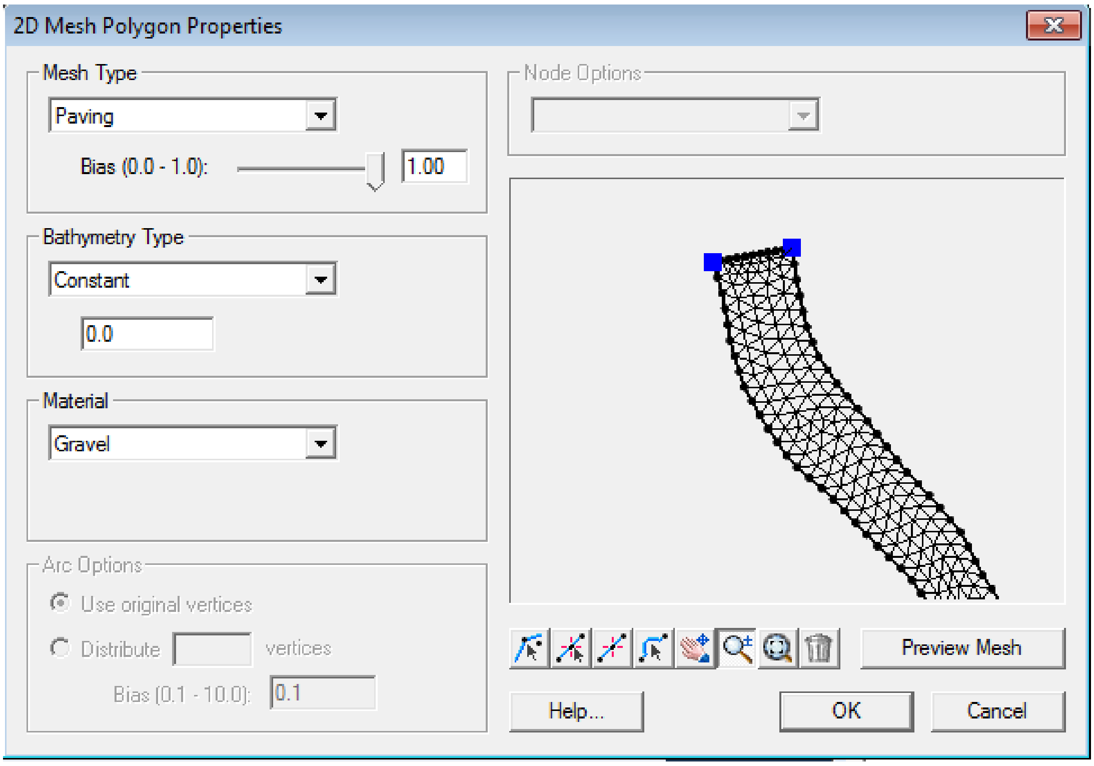
In order to build a mesh, we need to create a polygon from the feature arcs. To do this, select Feature Objects >> Build Polygons.



The SMS window should now appear as below.

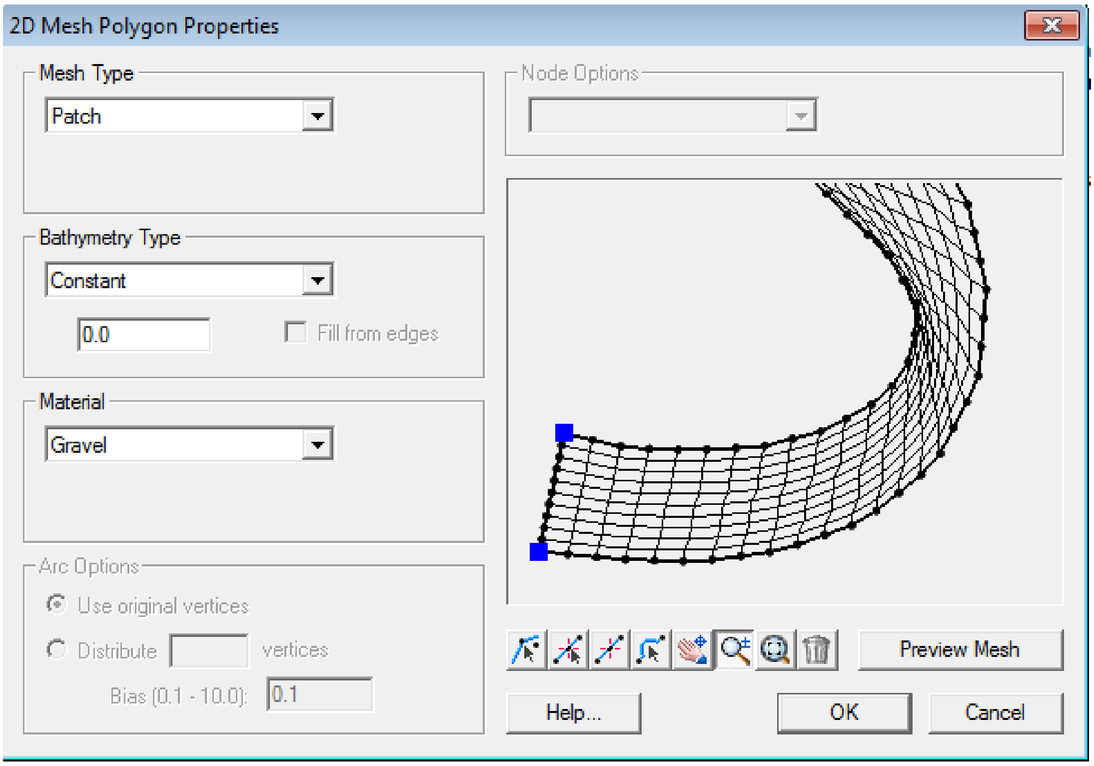


Using the polygon select tool (), select the polygon. The default mesh type is paving, using this mesh type, the default elements are triangles as per the image below



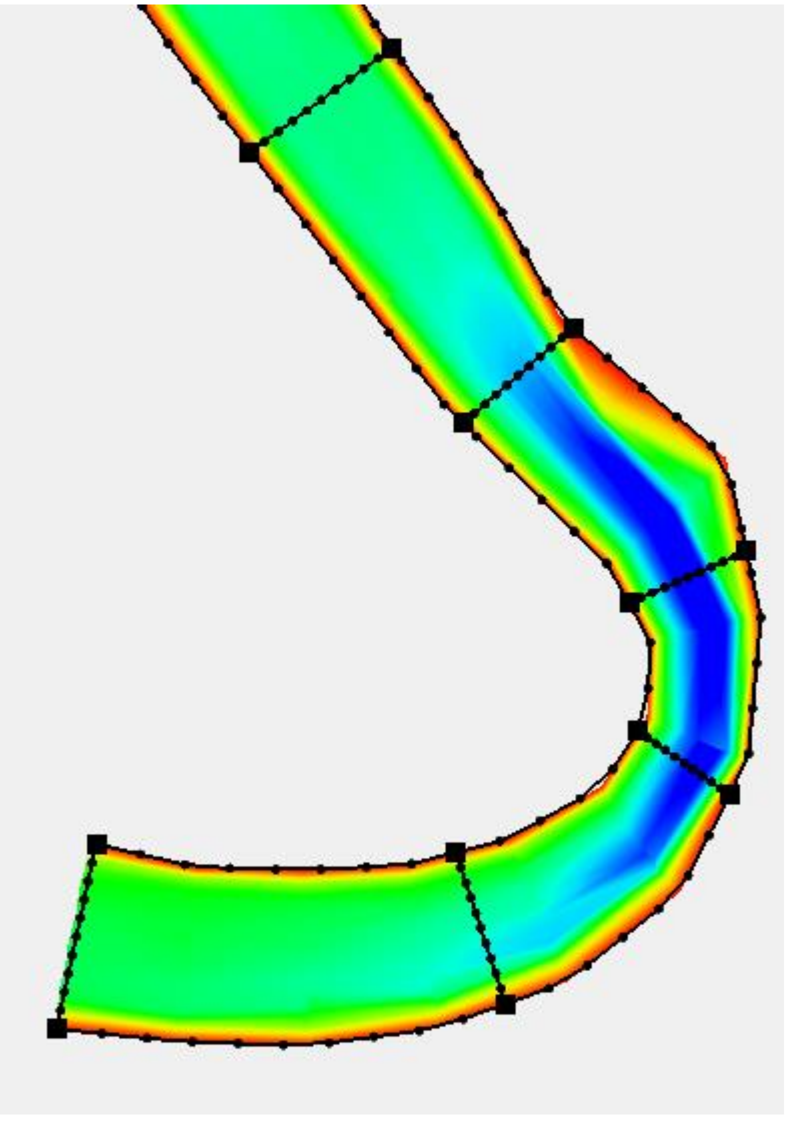
The mesh type patch uses quadrilateral elements preferentially. Change the mesh type to "Patch" and select Preview Mesh. With the entire section of river as a single patch mesh, the quadrilateral elements get wrapped around the bend as shown in the image below:

Note: You may receive an error about overlapping elements



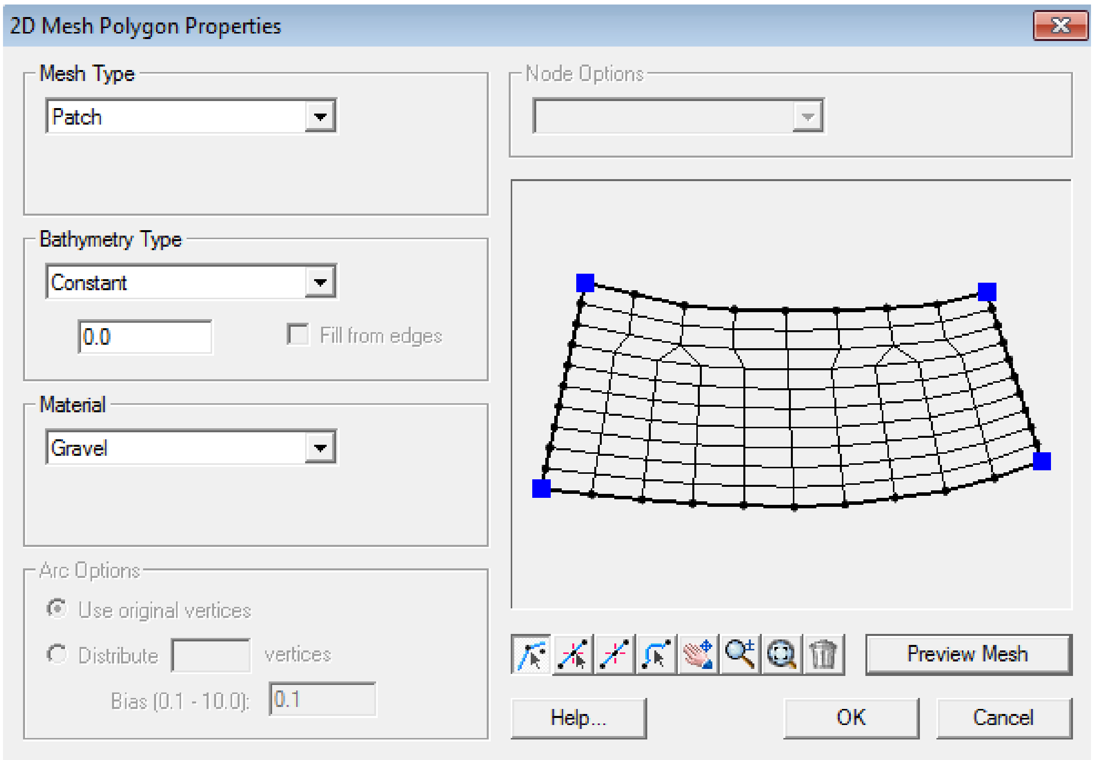
To avoid this, it is best to include sections across the channel (perpendicular to flow) at regular spacing along the channel, and in particularly around the bends.

To do this use the create feature arc button () to create the lines across the river. These do not need to snap to existing vertices, new ones will be created if required. An example is shown below, once the lines are drawn, 10 vertices should be distributed along each arc. Using the select tool, multiple arcs can be selected holding down shift.



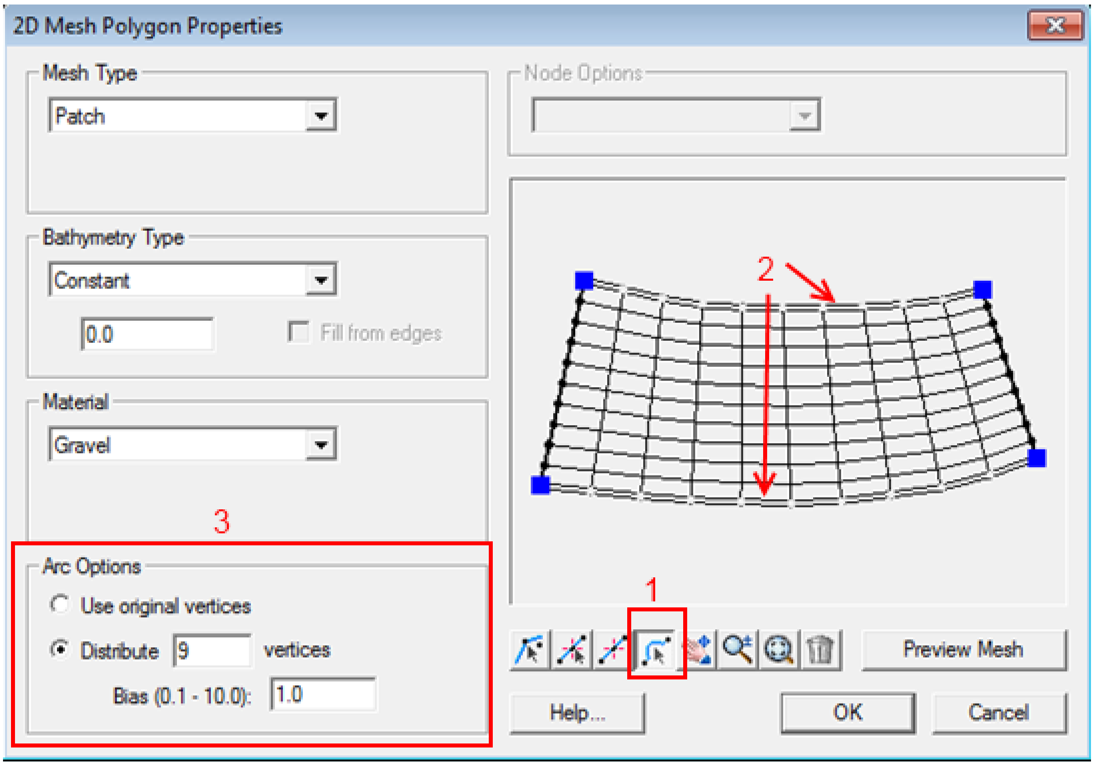
Once the additional arcs have been created, the polygons need to be rebuilt. To do this select Features Objects >> Build Polygons from the menu. Once rebuilt, the individual polygons can be selected and have different mesh types applied.

Select the southernmost polygon and select attributes (or double click on the polygon). Ensure the Mesh Type is set to patch and then hit preview mesh.

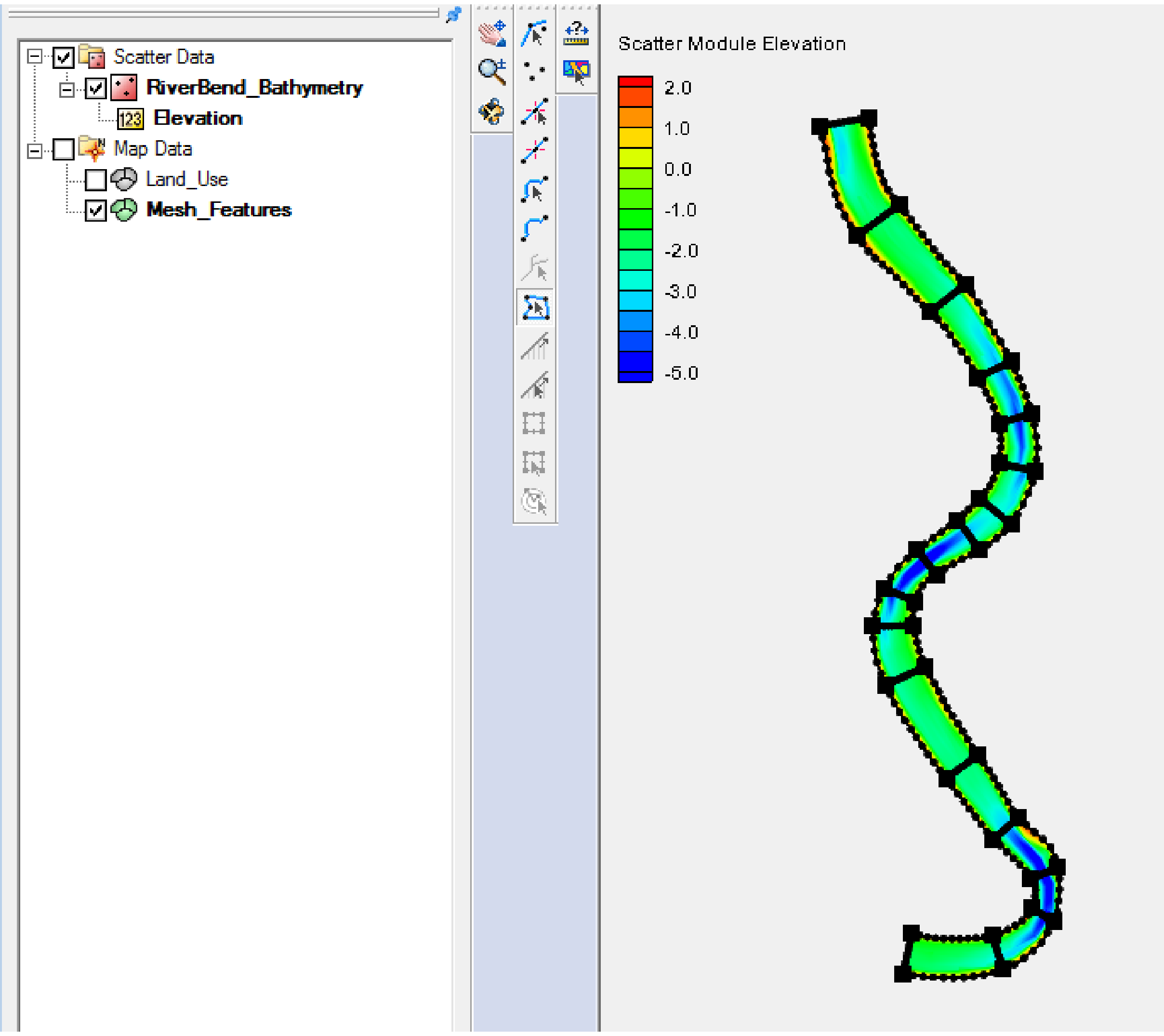


In the mesh preview window, the mesh is now much better aligned with the predominant flow direction than previously. However, along one bank there are more vertices than the other; TUFLOW FV can handle both triangles and quadrilateral elements, so this is not a major issue. However, to align the elements with the flow direction, quadrilaterals are preferred over triangles (small elements, such as triangles in deep water can also reduce the timestep).

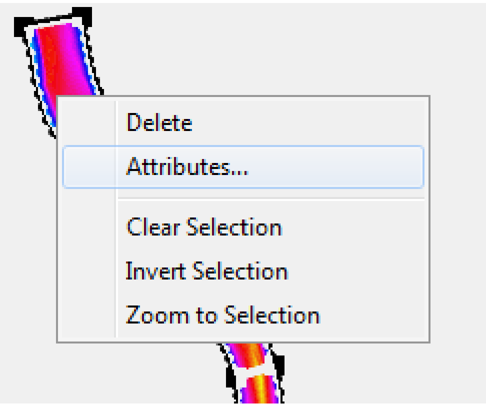
In this scenario, the river width remains relatively constant and we will use quadrilateral elements throughout the mesh. To achieve this, in the mesh properties dialogue, select the two bank lines and then in the Arc options the number of vertices can be redistributed. If both are selected, each bank will have the same number of vertices and quadrilateral elements will be created.



Repeat the process along the channel, an example is shown below.

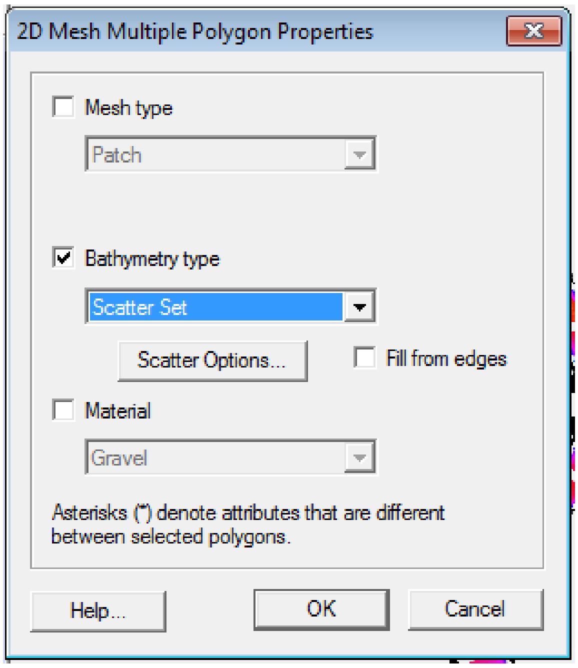


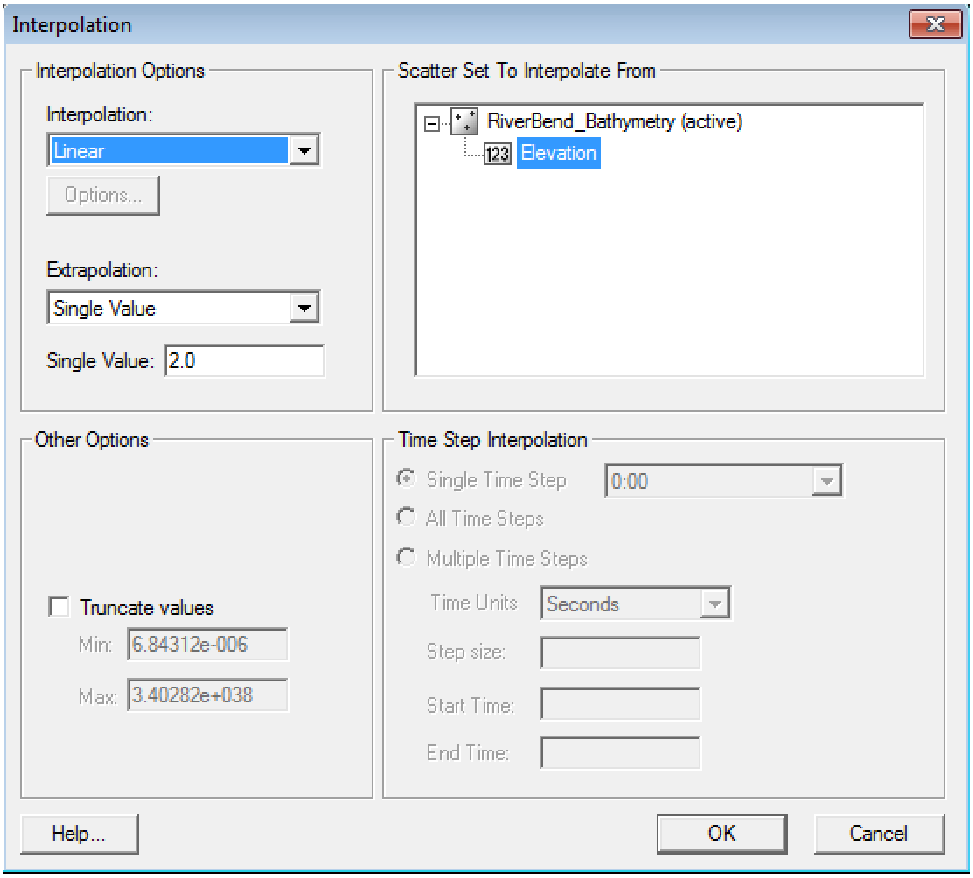
When you have finished creating you mesh areas, we need to specify an elevation data source. This can be done individually for each polygon, however, as the bathymetry source is consistent, we can select all polygons using the polygon select tool () you can drag and drop a box around all polygons. With all the polygons selected, right click and select "Attributes".



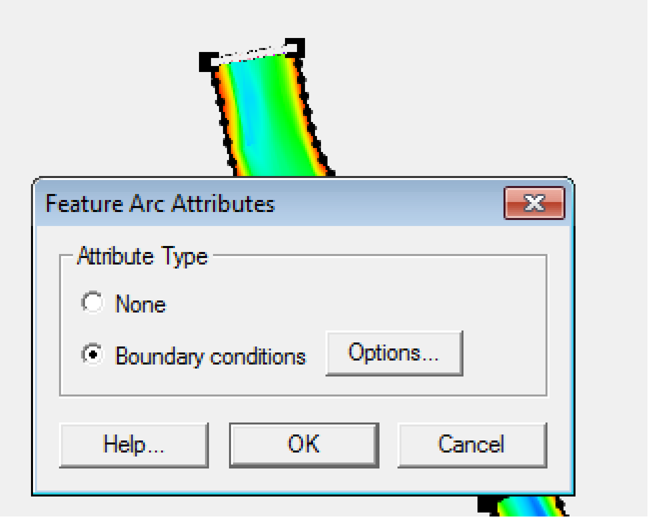
In the prompt, tick the check box next to Bathymetry type and the select “Scatter Set”. Once selected, hit the “Scatter Options” button. In the Scatter options set the Interpolation method to Linear and the extrapolation to “Single Value” and enter a value of 2.

When SMS has reshaped the vertices along the edge of the model it is possible that some are just outside the bathymetry dataset. The extrapolation method defines how these are set, we have used a high elevation, and alternative option is to use the Inverse Distance Weighting option.

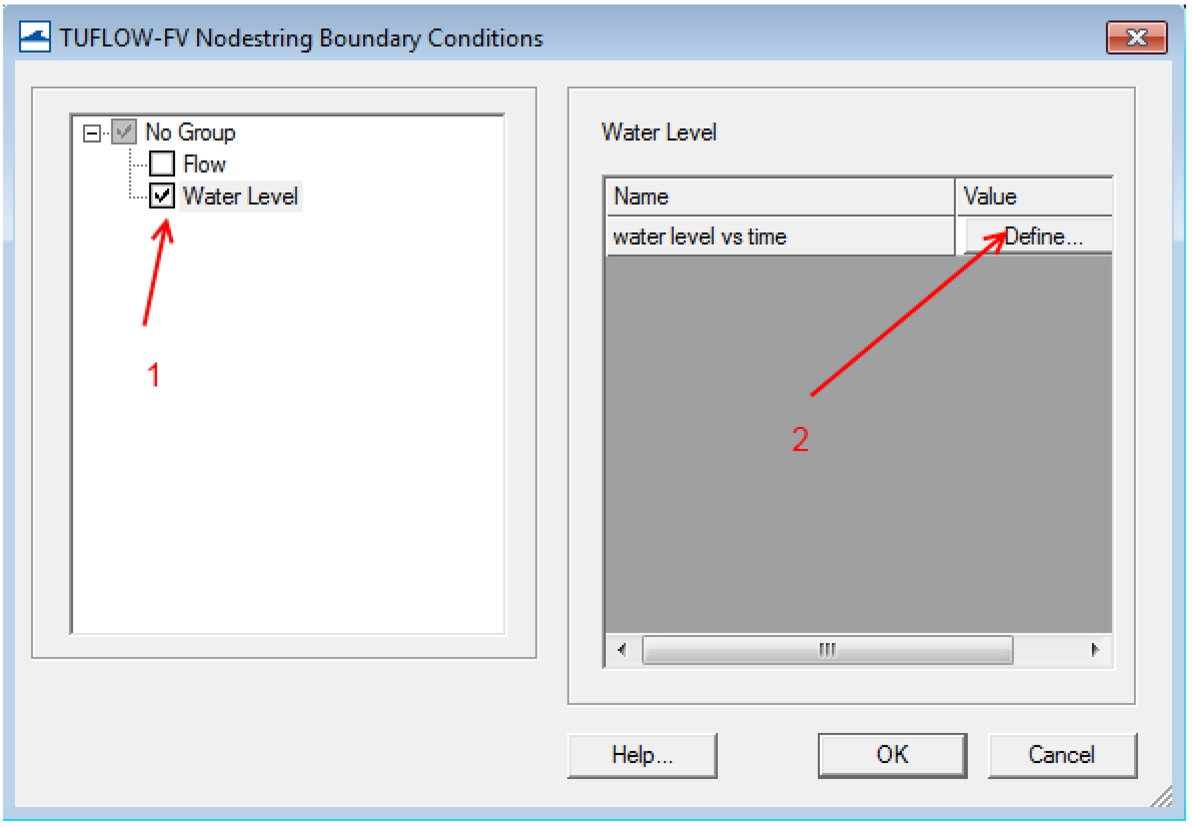




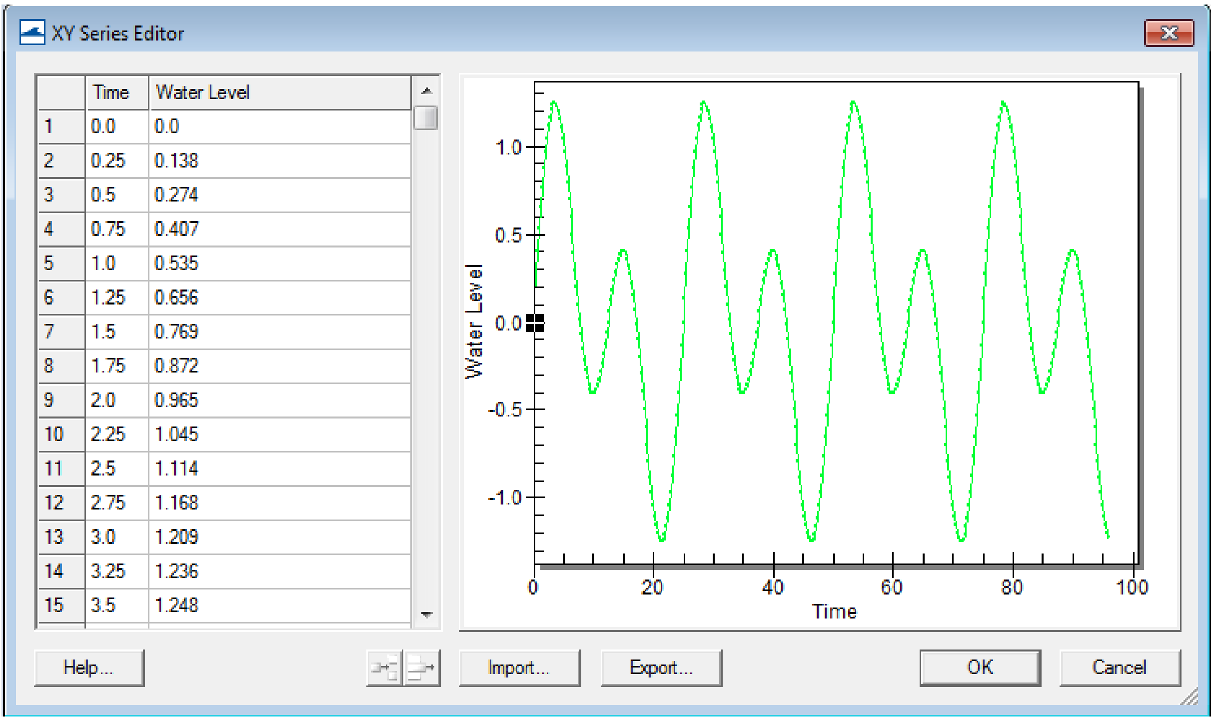
We next need to define the locations for our boundary conditions. Select the feature arc at the northern edge of the model and in the "attributes" dialogue, set this type to Boundary Condition, as per the image below. Then select Options...



In the boundary condition dialogue, set the boundary type to "Water Level" and the select "Define" (step 2 in the image below).

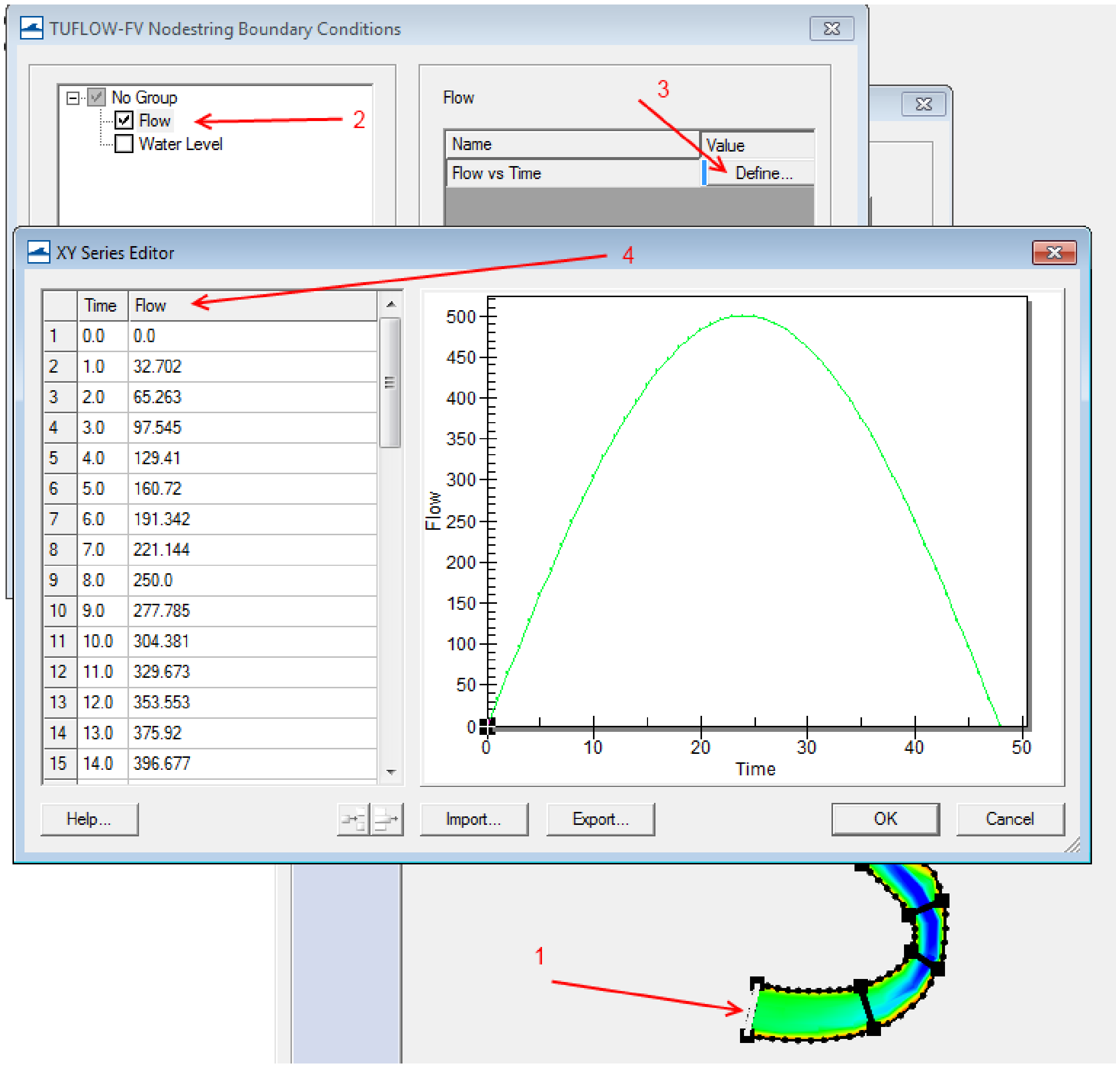


Open the Tide.xlsx or Tide.csv in Excel (these are in the provided data), copy and paste the data into the series editor. All data can be copied at the same time; this does not need to be done one column at a time. The dialogue should look like the below.



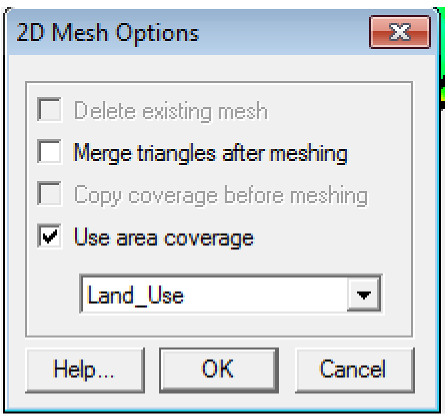
At the southern end of the model select the feature arc and apply a flow boundary. The flow is in the "flows.xlsx". See also the image below.

The boundary data and locations can be changed after the mesh has been created. This is described further below.

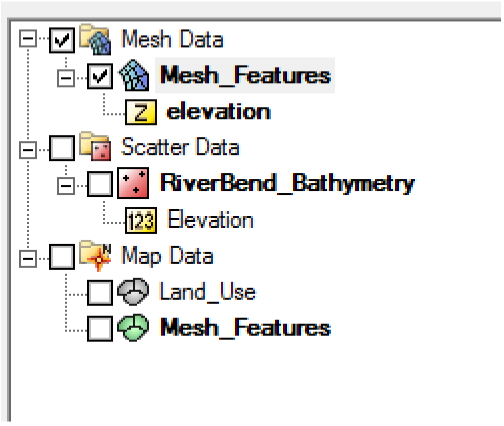


We are now ready to build the mesh from the map data. To do this select Feature Objects >> Map - > 2D Mesh.

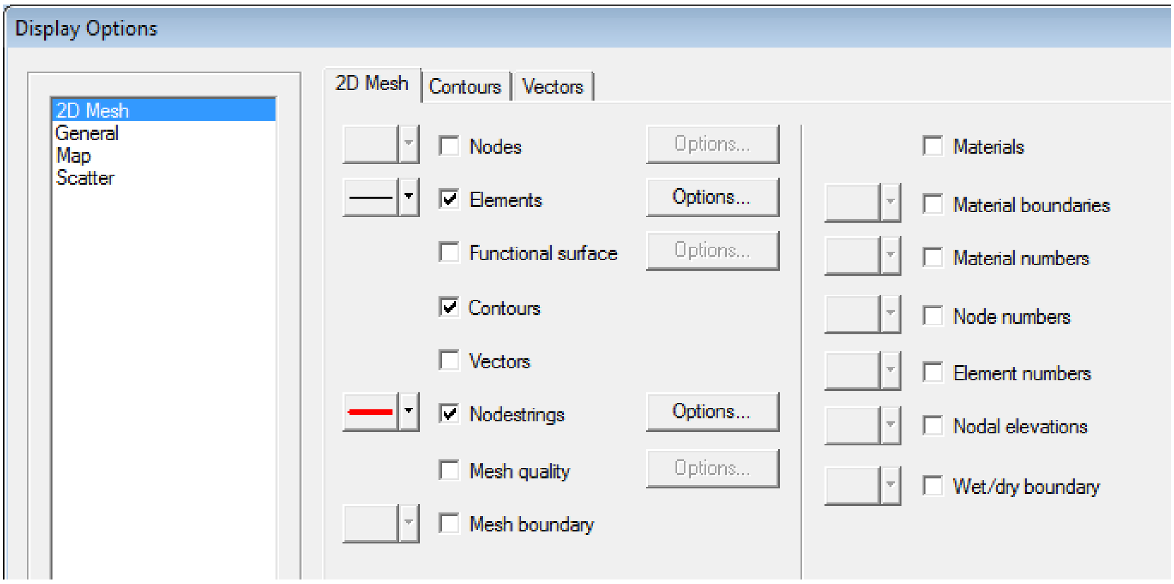
In the 2D Mesh Options check the “Use area coverage” option. In the drop box ensure that the Land_Use layer is selected. When the mesh is created, this uses the land use layer for setting the material definition in the elements created. This can also be set individually on the polygons in the “Mesh_Features” layer.



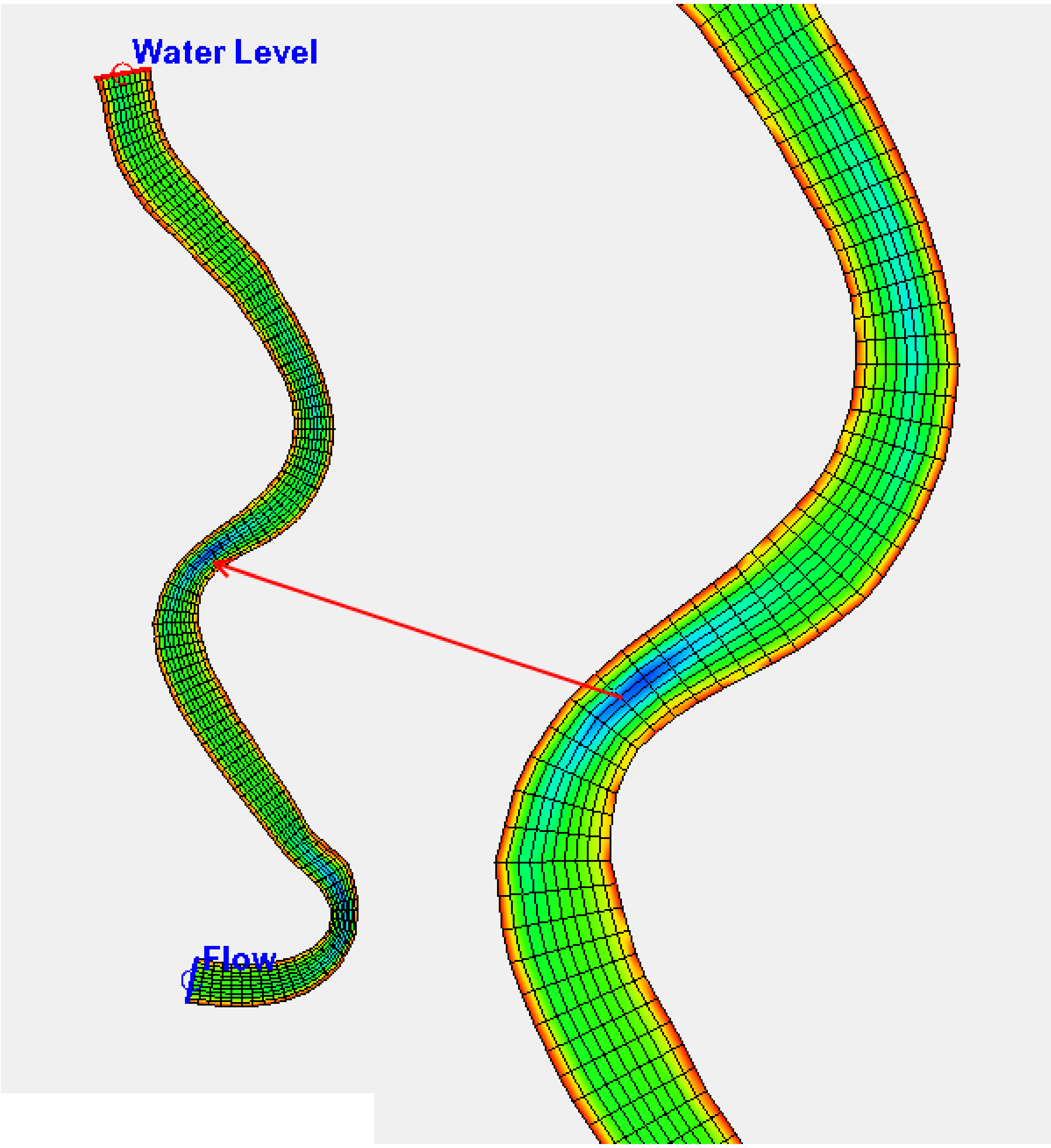
After the meshing is completed, turn off the scatter dataset and the map data:



In the display options, turn on the elements, contours and nodestrings:



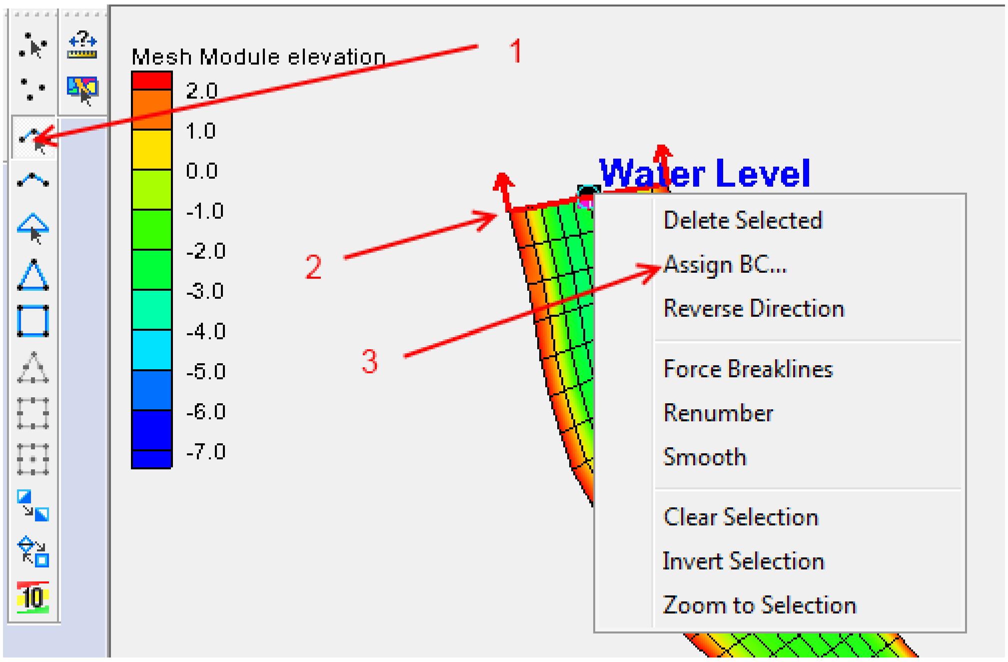
The mesh elements and boundaries should now be visible. The image below shows the entire mesh on the left and a zoomed in inset on the right. In the inset it can be seen that the mesh elements align with the flow direction. Whilst not mandatory, this is the preferred mesh alignment.



Screen Grab: Final Mesh

To modify the boundary data (we do not need to do this just yet, but it is useful to know), ensure you are in the mesh module (by clicking in the mesh in the table of contents), use the select nodestring button to select the nodestring, and then choose “Assign BC…” This is shown in the ge below.

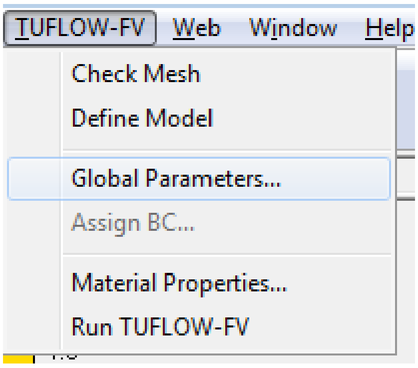
To create a new boundary after the mesh has been created, the create nodestring tool ( ) can be used. After a nodestring is created a boundary can be applied to it.



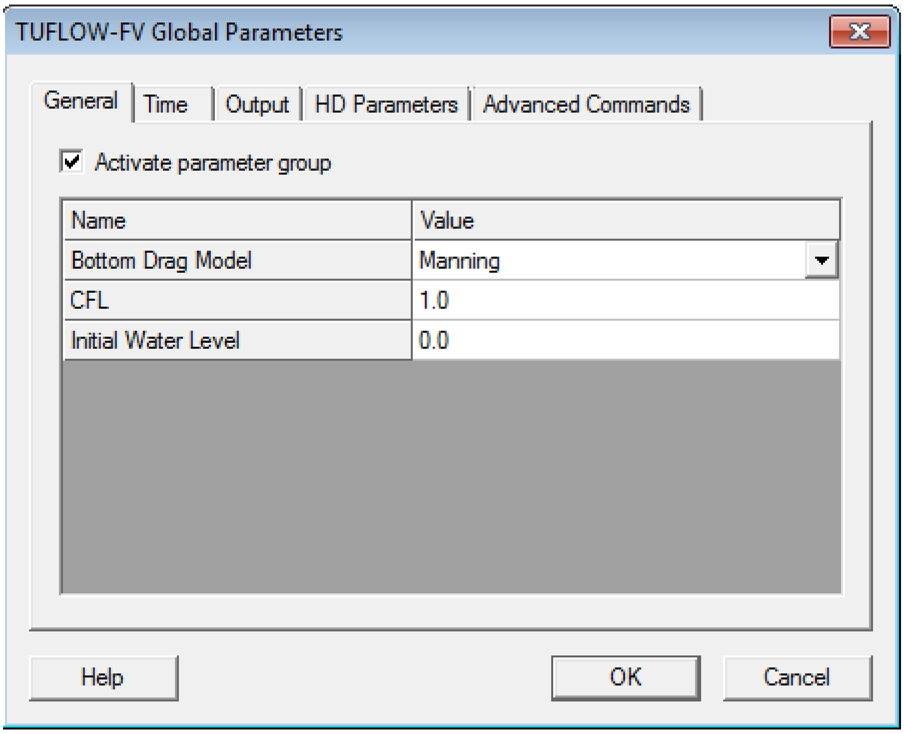
The next step in the modelling process is to assign the model parameters.

6.2.3 Model Parameters

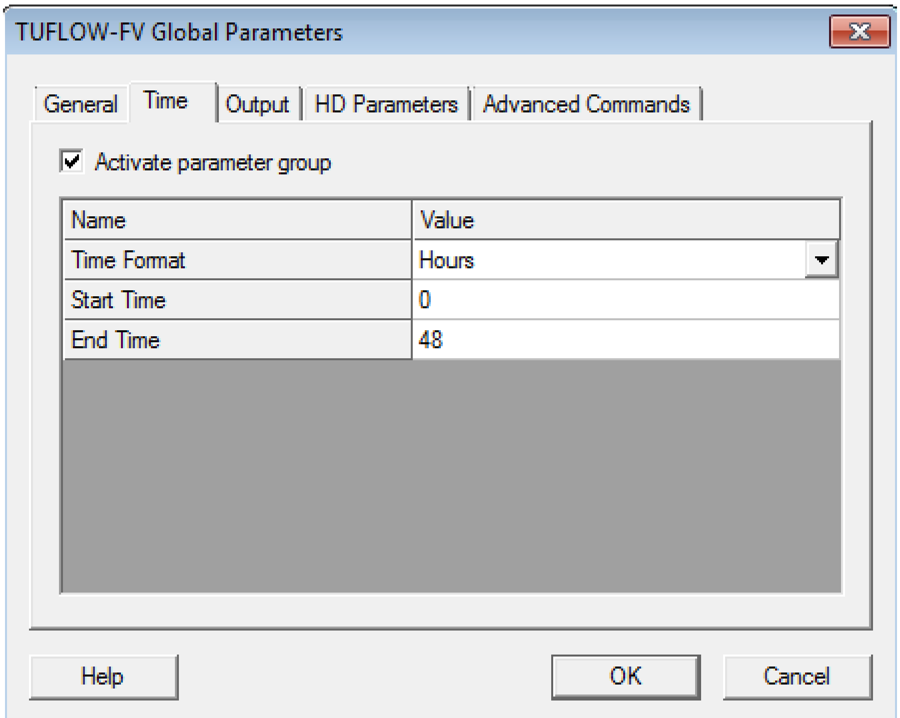
To assign the model parameters access the “Global Parameters” from the TUFLOW FV menu item:



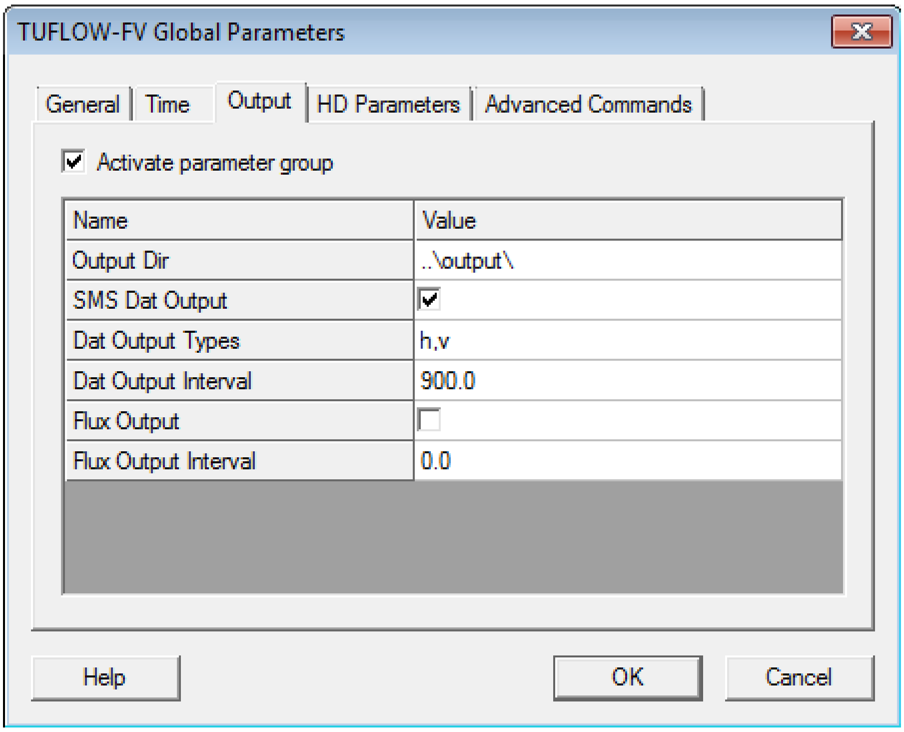
In the general tab, the default options are ok; these should be as per the image below:



In the “Time” tab, set the end time to 48, the model will be 48 hours.

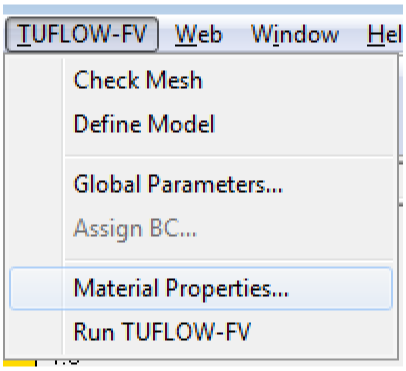


In the output options, set the following parameters



The HD and Advanced commands can be left unchanged. Hit OK to apply the changes.

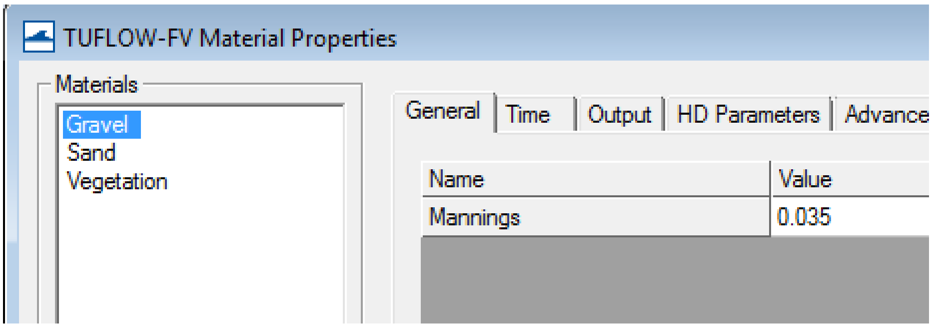
Once the global parameters are set, we need to set the Manning's value to be used for the three land use types (sand, gravel and vegetated). To do this select TUFLOW FV >> Material Properties.



For each of the three items listed on the left of the screen set a Manning's value. Suggested values are in the table below.

Table 6-1 River bend Tutorial: Suggested Manning's Values

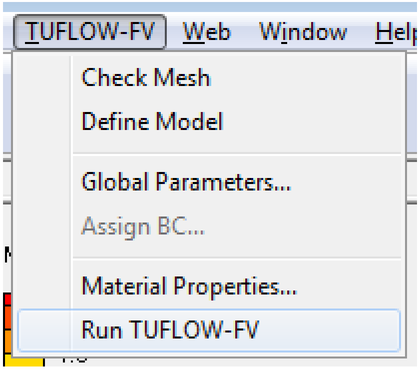
Land Use	Suggested Manning's n
Gravel	0.035
Sand	0.028
Vegetation	0.06



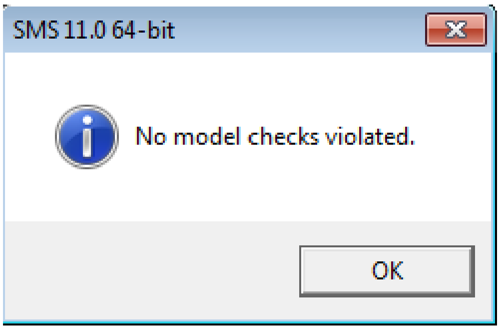
The model is now ready to run!

6.2.4 Running the Model

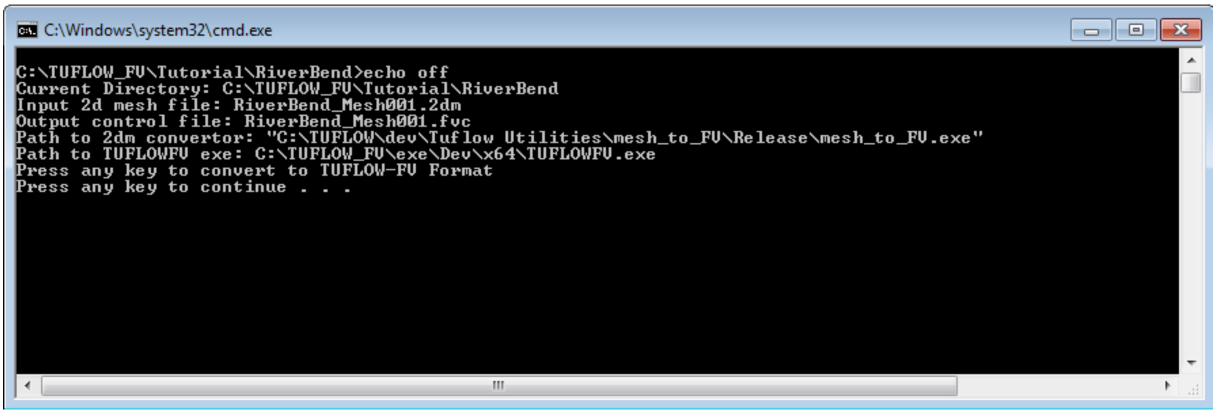
Ensure that the SMS project has been saved. The TUFLOW FV files will be created in a sub directory in the same location as the SMS project (.sms). To run the model select TUFLOW FV >> Run TUFLOW FV



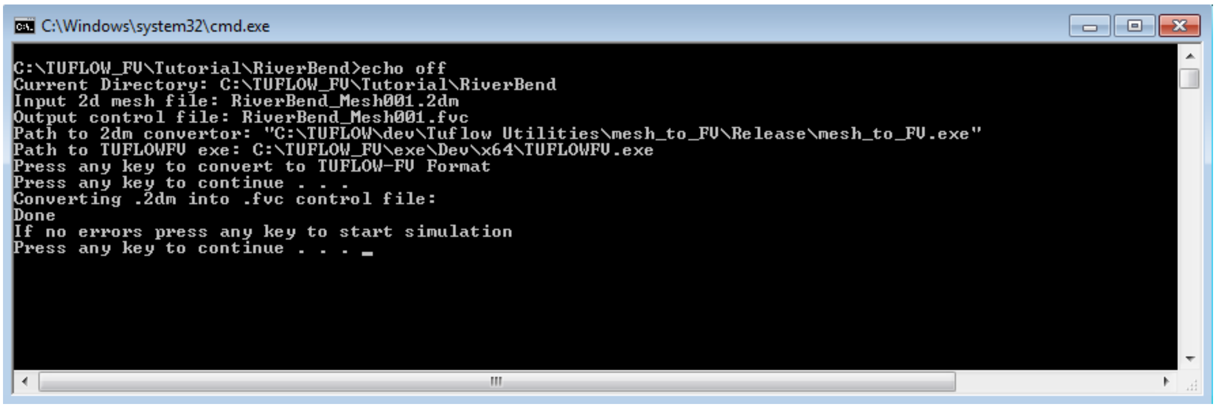
The following dialogue should be displayed, stating that no model checks have been violated.



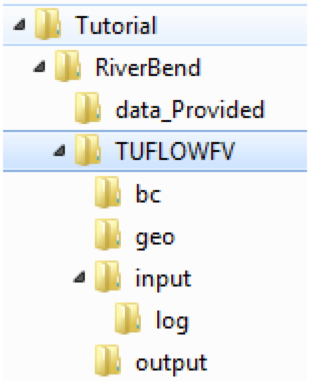
Select OK and a console window should be displayed; this will display the location of the inputs and outputs. This step is shown below.



Press any key to begin the conversion press any key. Once the conversion is completed, the console will pause.



In the same directory as the SMS project is saved a TUFLOWFV folder has been created:



In the TUFLOWFV\input\ directory which contains the control file (.fvc). This can be opened in a text editor. The first part of the control file is displayed below.

WBM_002.tcf | WBM_003_2012.tcf | WBM_002_2012.tlf | WBM_003_2012.tlf | RiverBend_Mesh001.fvc

```
!TUFLOW-FV control file (.fvc) generated from SMS mesh file (.2dm)
!.2dm file name: RiverBend_Mesh001.2dm
!Mesh_to_FV build: 2012-05-AB
!______________________________________________________________________________
!General Parameters
Bottom Drag Model == Manning
CFL == 0.9
Initial Water Level == 0.0
!______________________________________________________________________________
!Geometry Commands
Geometry 2D == ..\geo\RiverBend_Mesh001.2dm
!______________________________________________________________________________
!Time Commands
Time Format == Hours
Start Time == 0
End Time == 48
```

In the console window, press any key to start the model. If the model starts successfully, the console window should appear as below:

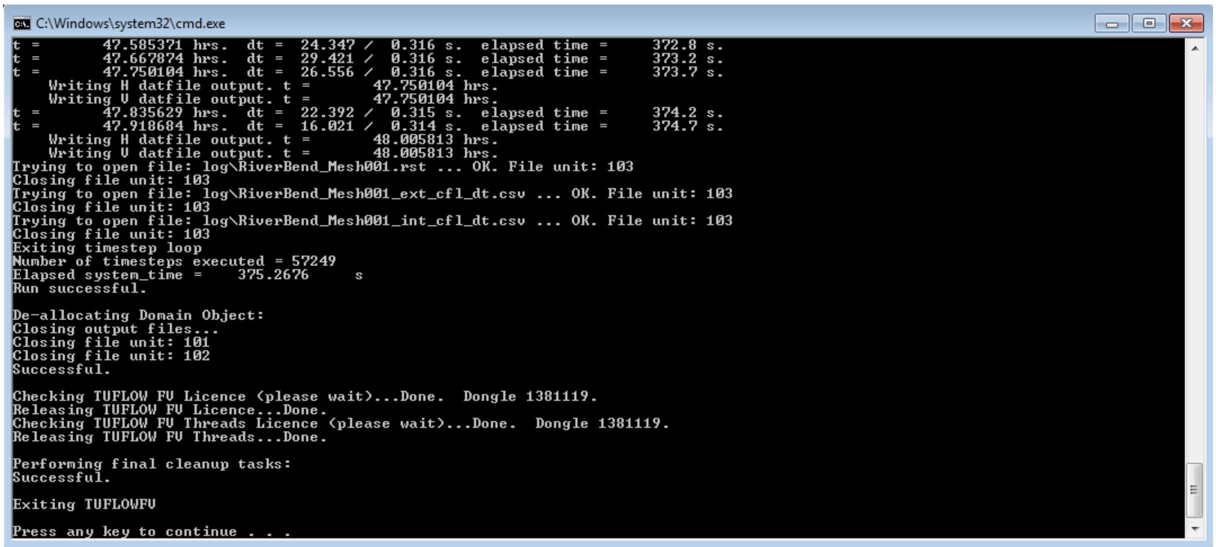
UltraEdit DOS Command Window

```
Running simulation
Number of OpenMP threads=8
Thread stacksize=4194304

Entering timestep loop
t =        0.000000 hrs.  dt =   0.000 /  0.000 s.  elapsed time =       0.0 s.
    Writing H datfile output. t =         0.000000 hrs.
    Writing V datfile output. t =         0.000000 hrs.
t =        0.111031 hrs.  dt =  99.886 /  0.282 s.  elapsed time =       0.8 s.
t =        0.169072 hrs.  dt =  26.203 /  0.282 s.  elapsed time =       1.1 s.
t =        0.250082 hrs.  dt =   8.095 /  0.279 s.  elapsed time =       1.6 s.
    Writing H datfile output. t =         0.250082 hrs.
    Writing V datfile output. t =         0.250082 hrs.
t =        0.334826 hrs.  dt =   6.911 /  0.276 s.  elapsed time =       2.1 s.
t =        0.418399 hrs.  dt =  10.168 /  0.275 s.  elapsed time =       2.5 s.
t =        0.500790 hrs.  dt =  18.965 /  0.275 s.  elapsed time =       3.0 s.
    Writing H datfile output. t =         0.500790 hrs.
    Writing V datfile output. t =         0.500790 hrs.
t =        0.584082 hrs.  dt =  17.878 /  0.275 s.  elapsed time =       3.5 s.
t =        0.671601 hrs.  dt =  23.310 /  0.274 s.  elapsed time =       4.0 s.
t =        0.750065 hrs.  dt =   5.168 /  0.272 s.  elapsed time =       4.5 s.
    Writing H datfile output. t =         0.750065 hrs.
    Writing V datfile output. t =         0.750065 hrs.
t =        0.835696 hrs.  dt =   9.451 /  0.270 s.  elapsed time =       5.1 s.
t =        0.919506 hrs.  dt =  16.948 /  0.269 s.  elapsed time =       5.6 s.
t =        1.002837 hrs.  dt =  12.650 /  0.269 s.  elapsed time =       6.1 s.
    Writing H datfile output. t =         1.002837 hrs.
    Writing V datfile output. t =         1.002837 hrs.
t =        1.085845 hrs.  dt =  12.897 /  0.269 s.  elapsed time =       6.6 s.
t =        1.167180 hrs.  dt =  14.981 /  0.267 s.  elapsed time =       7.2 s.
t =        1.251386 hrs.  dt =  14.887 /  0.266 s.  elapsed time =       7.8 s.
    Writing H datfile output. t =         1.251386 hrs.
    Writing V datfile output. t =         1.251386 hrs.
t =        1.333524 hrs.  dt =   8.470 /  0.265 s.  elapsed time =       8.3 s.
t =        1.418120 hrs.  dt =   6.339 /  0.264 s.  elapsed time =       8.9 s.
t =        1.501124 hrs.  dt =   4.484 /  0.264 s.  elapsed time =       9.5 s.
    Writing H datfile output. t =         1.501124 hrs.
    Writing V datfile output. t =         1.501124 hrs.
t =        1.583508 hrs.  dt =  13.414 /  0.263 s.  elapsed time =      10.0 s.
t =        1.667068 hrs.  dt =   6.025 /  0.262 s.  elapsed time =      10.5 s.
t =        1.751611 hrs.  dt =  11.486 /  0.261 s.  elapsed time =      11.2 s.
    Writing H datfile output. t =         1.751611 hrs.
    Writing V datfile output. t =         1.751611 hrs.
```

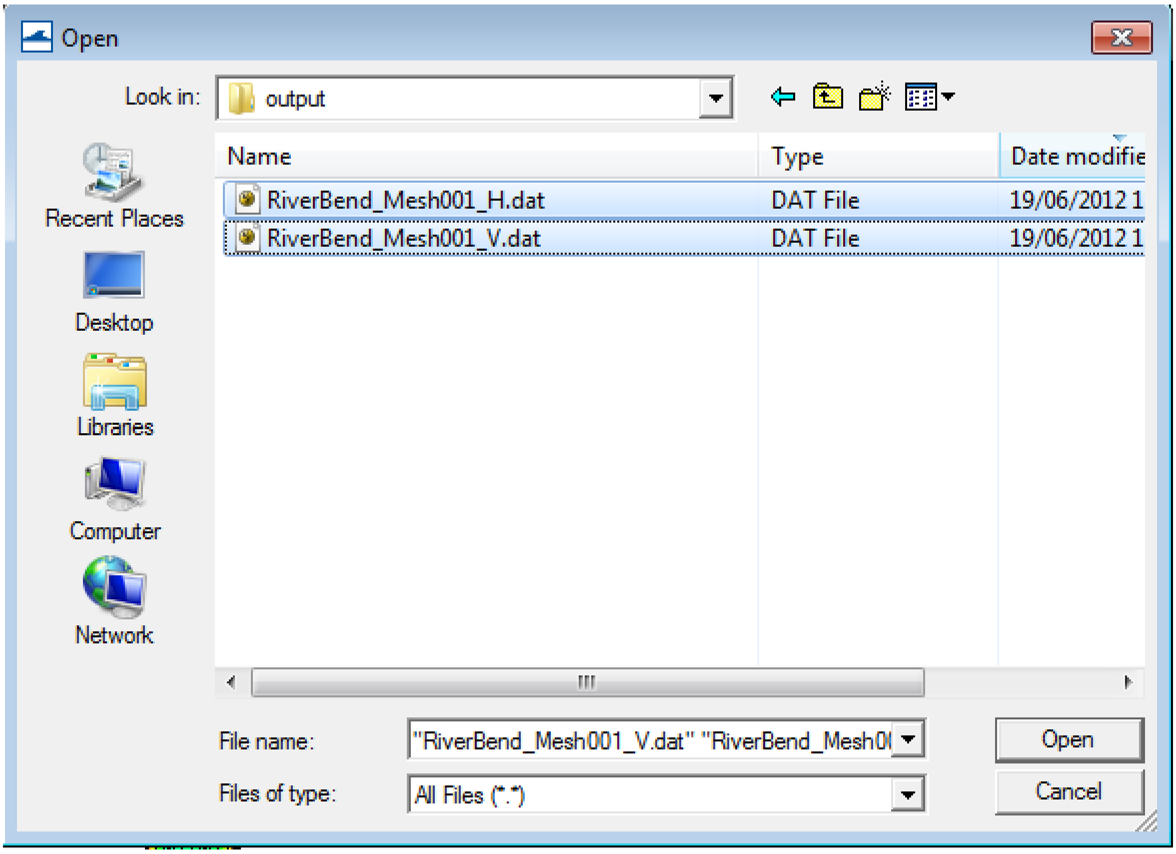
If the model fails to start successfully please see the troubleshooting section below.

Depending on computer speed and number of processors available, the model may take a few minutes to finish. On an i7 laptop (2 years old), the model runs in approximately 5 minutes. Once finished the console should appear as below:

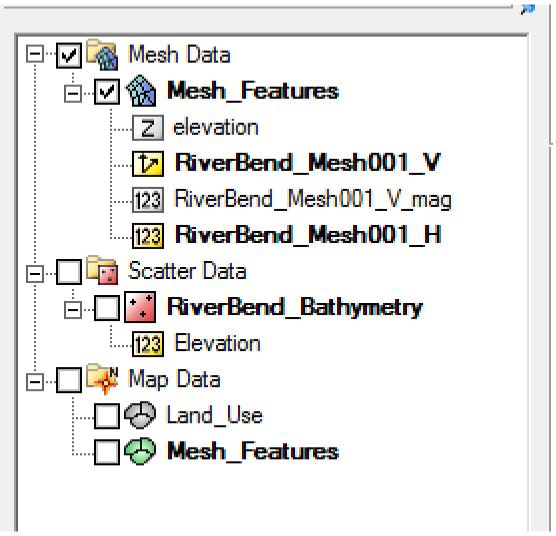


6.2.5 Reviewing Results

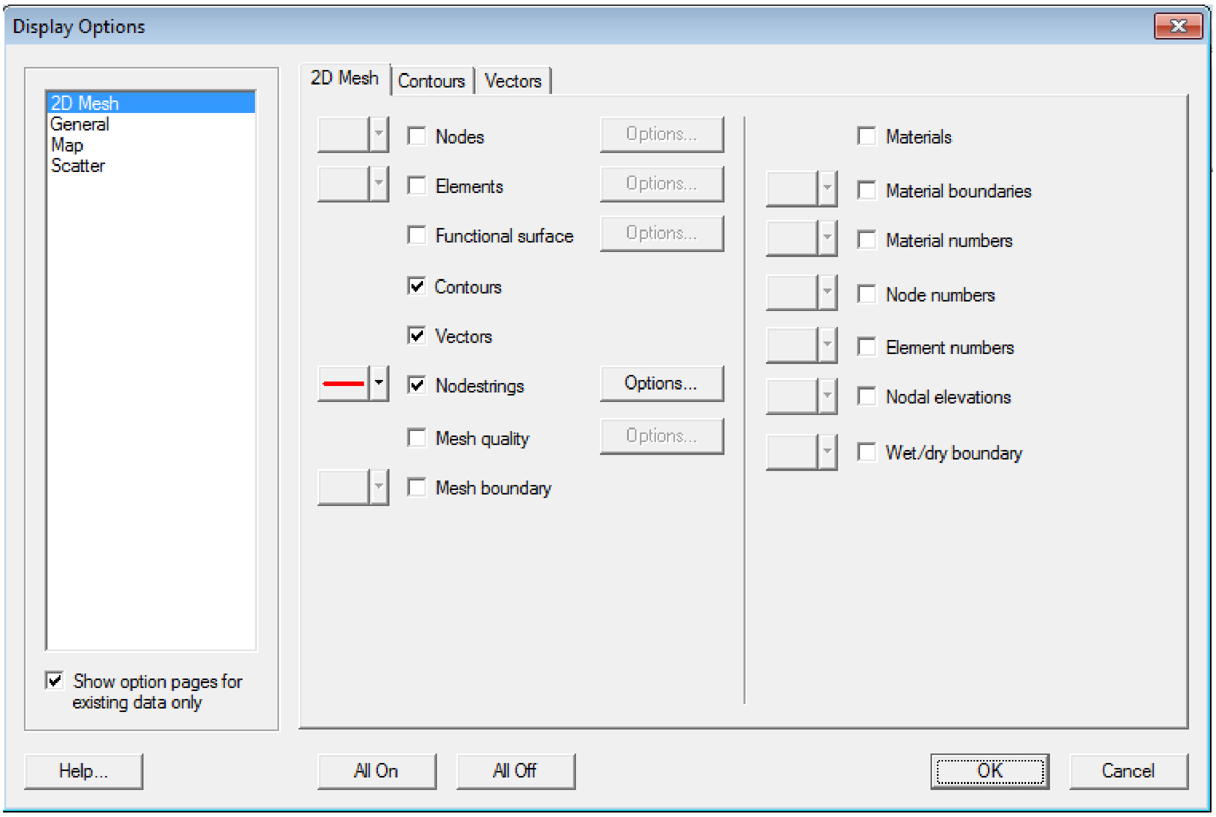
In the TUFLOWFV\output\ directory should be the results files, in the input we asked for two output h (level) and v (velocity). These files can be loaded in SMS either using the File >> Open interface or by dragging and dropping the files from Windows explorer.

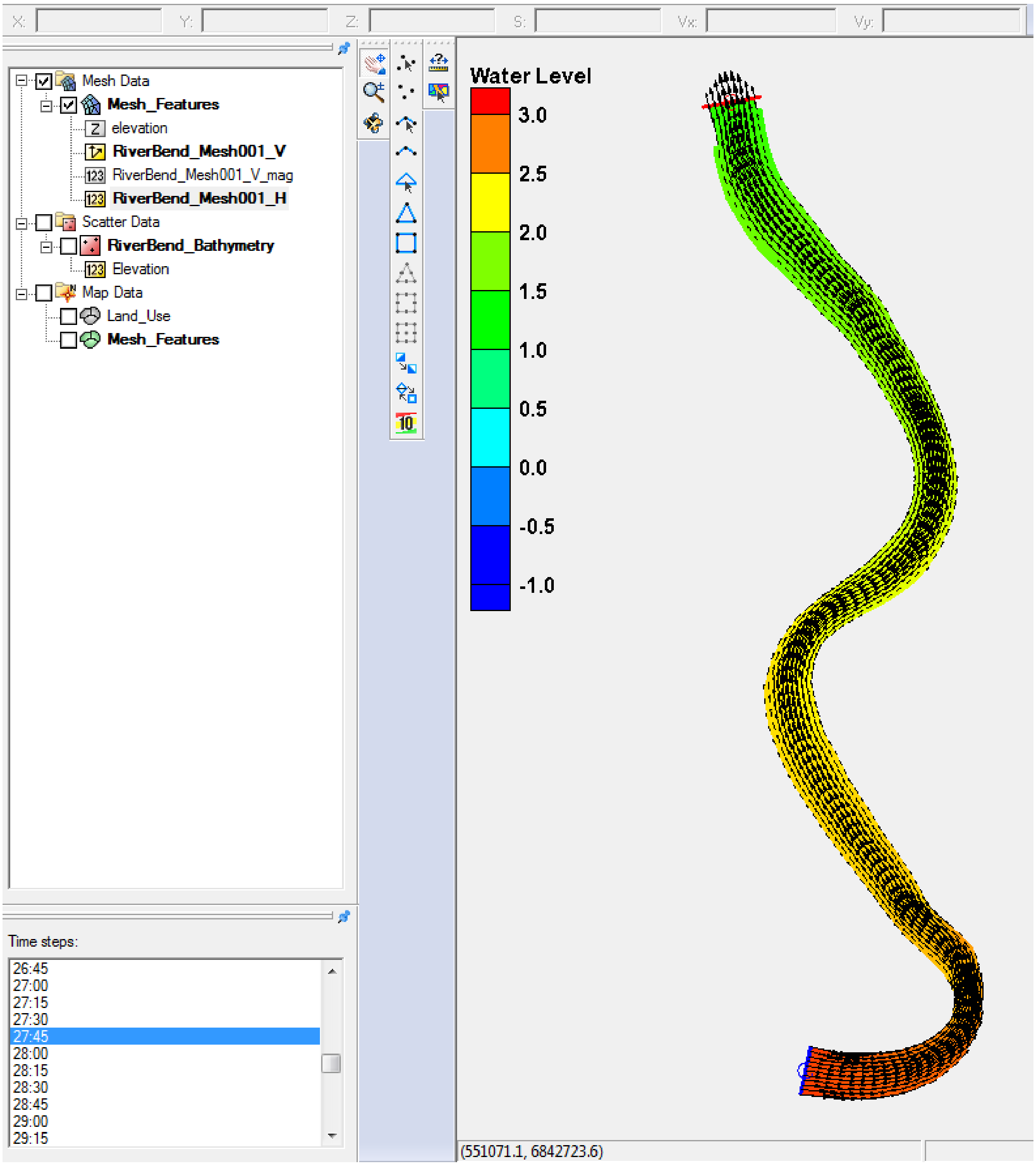


These results will display in the table of contents in the Mesh Data.



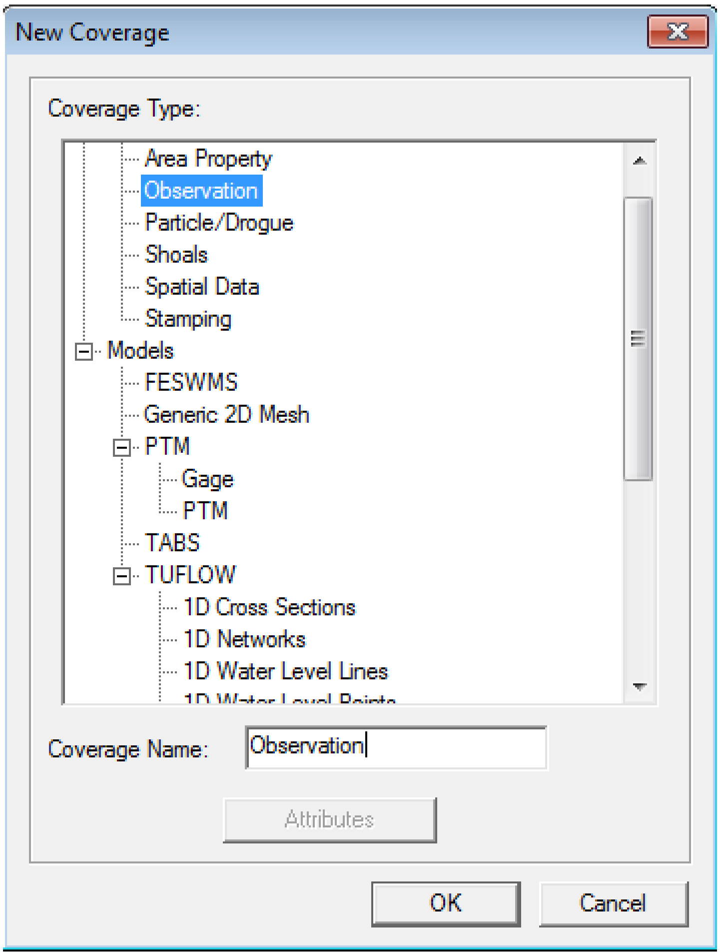
Turn off the Scatter and Map Data layers and in the display options:



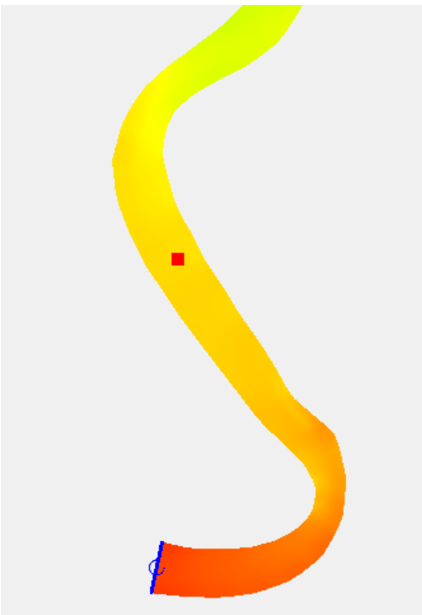


Step through the results in the "Time Steps" window. The contour increment can be changed in the display options. In the Mesh Data there are three scalar dataset available for viewing (elevation, RiverBend_Mesh001_H and RiverBend_Mesh001_V_mag). The elevation dataset does not change over time. There is one vector set (velocity) available.

To extract time series at a point, create a new Map Data coverage (by right clicking on Map Data). This should be set to an "Observation" type. This is shown in the image below.

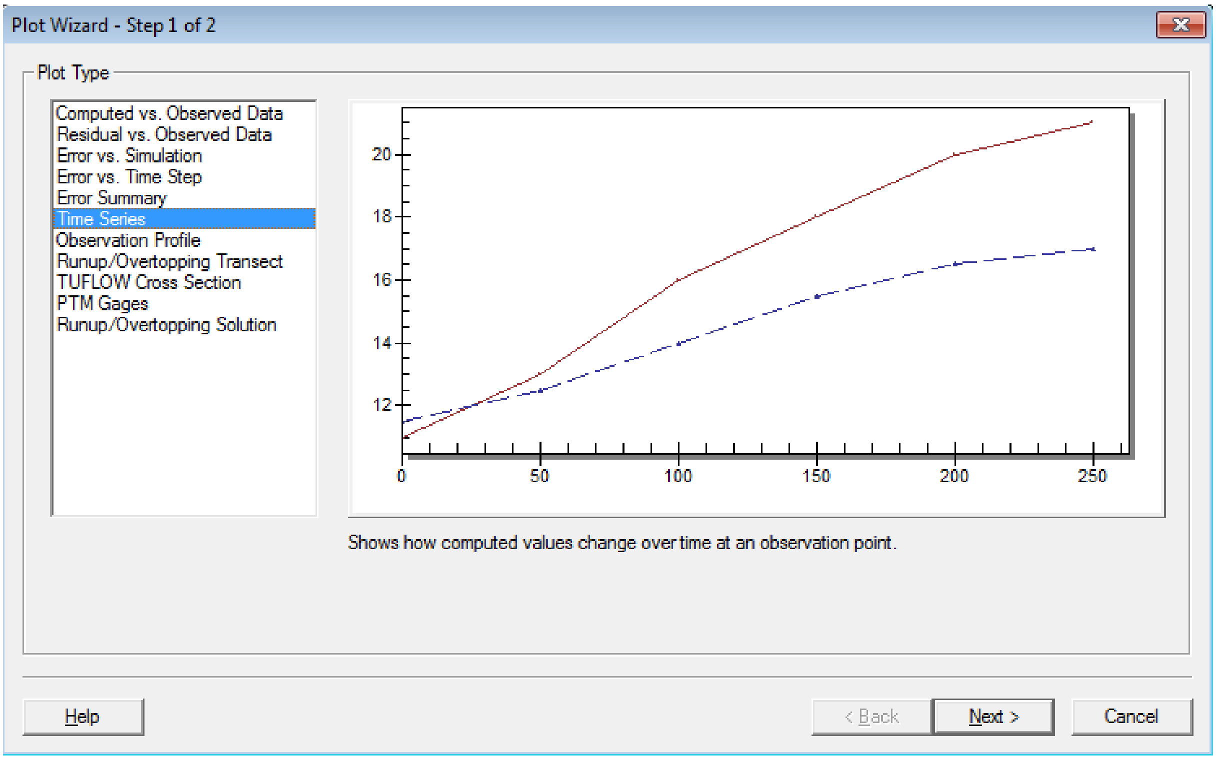


Once created, highlight the dataset and select the create feature point button (). Create a feature point in the location you would like to extract results. Multiple points can be extracted at the same time.

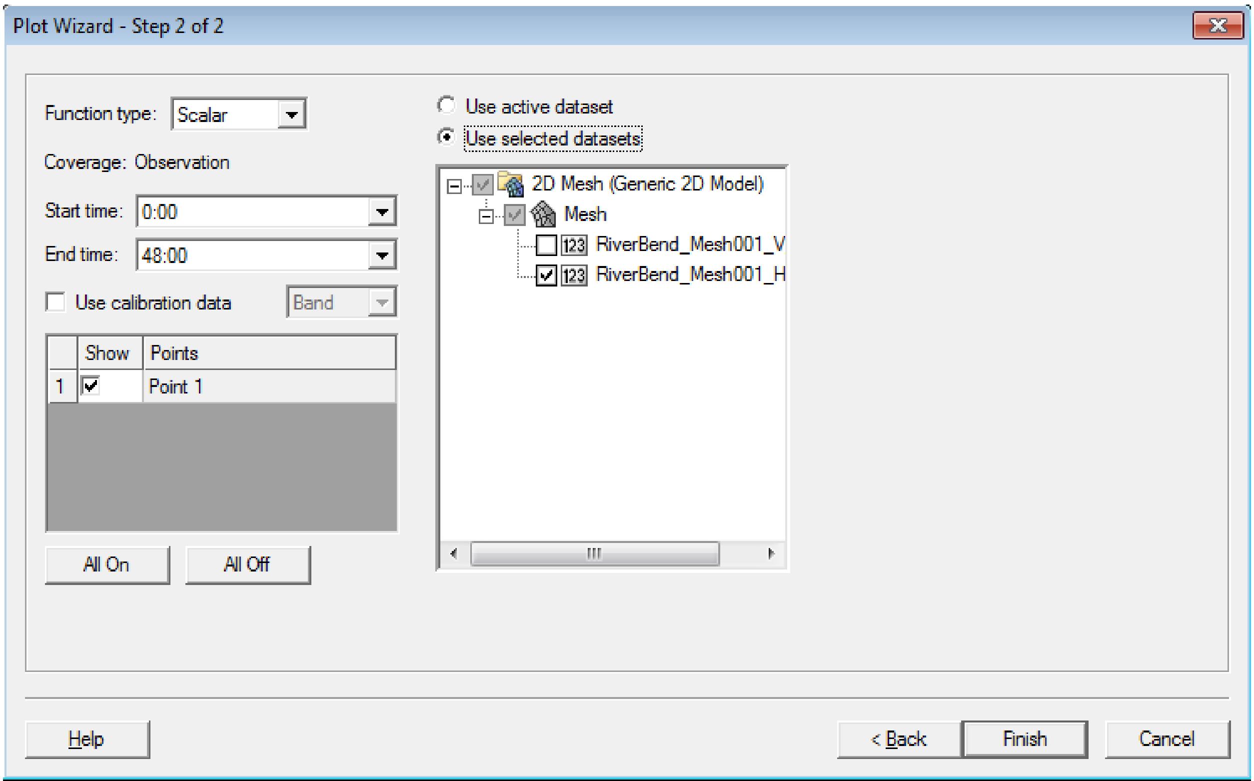


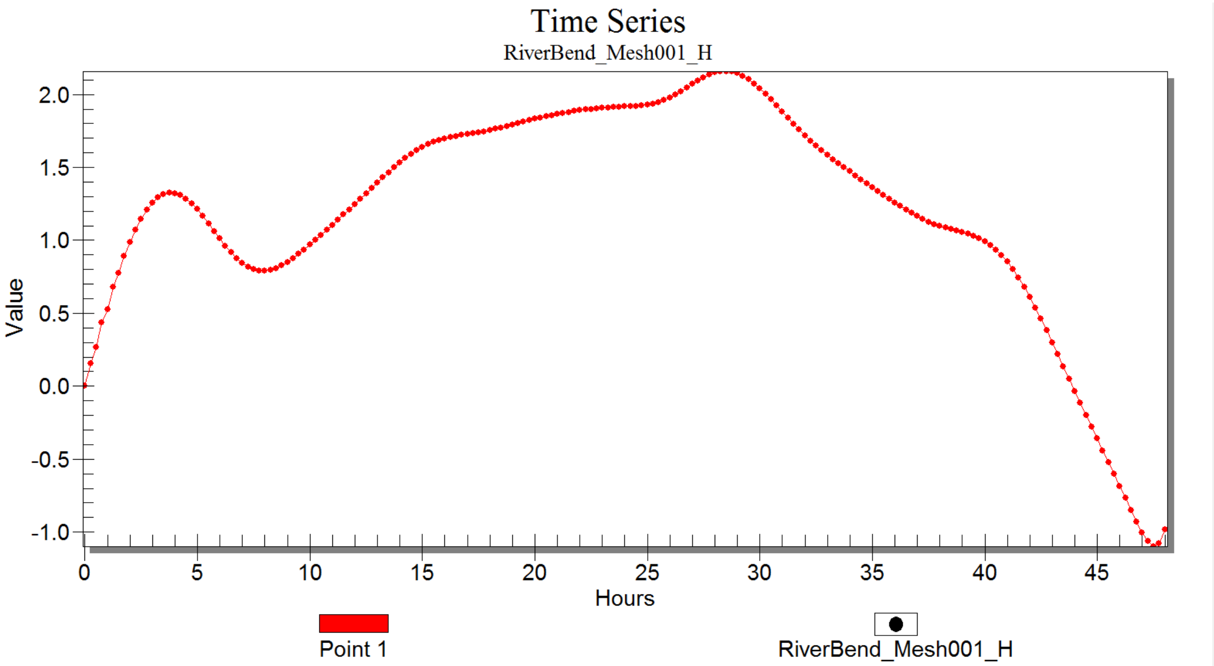
Once the points have been created select Display >> Plot Wizard

Select the Time Series plot type.



Choose the dataset and time period to extract the results, as per the image below and then slect Finish, the plot will be displayed.



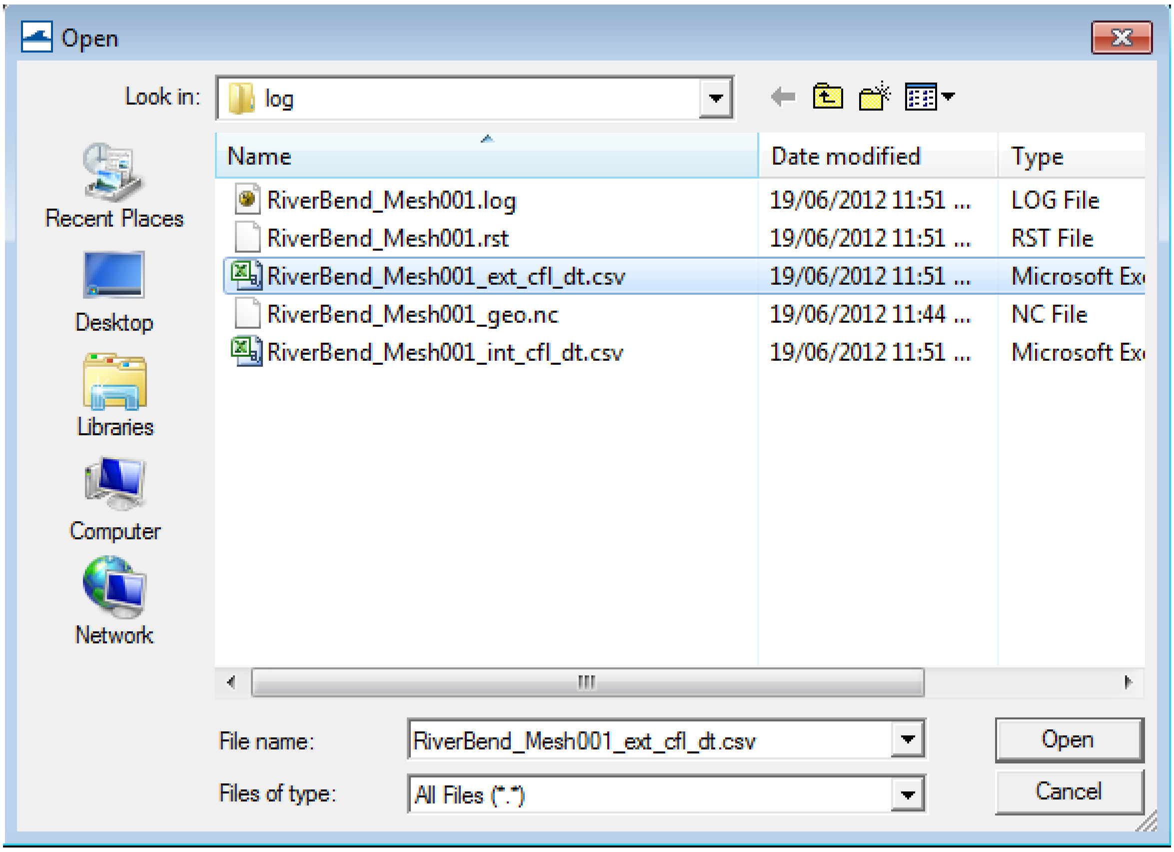


In later modules we cover using the point output in TUFLOW FV to output results directly in .csv format; this allows higher frequency results to be extracted than the map output.

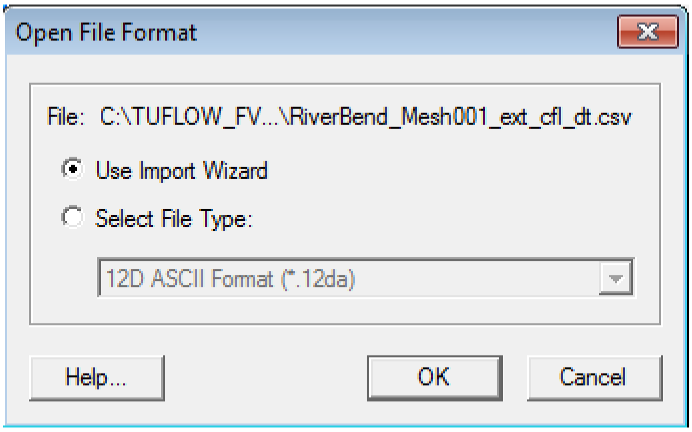
6.2.6 Reviewing Mesh Performance

In this section we will look at the performance of the mesh in terms of timesteps required. The TUFLOW FV model uses and adaptive timestep which is based on the specified Courant-Friedrichs-Lewy condition (CFL parameter). The model timestep is calculated based on the cell size and depth. A poorly configured mesh with a single small cell in deep water can limit the timestep of the model. Therefore, after running the model it is beneficial to review the timestep required to run the model.

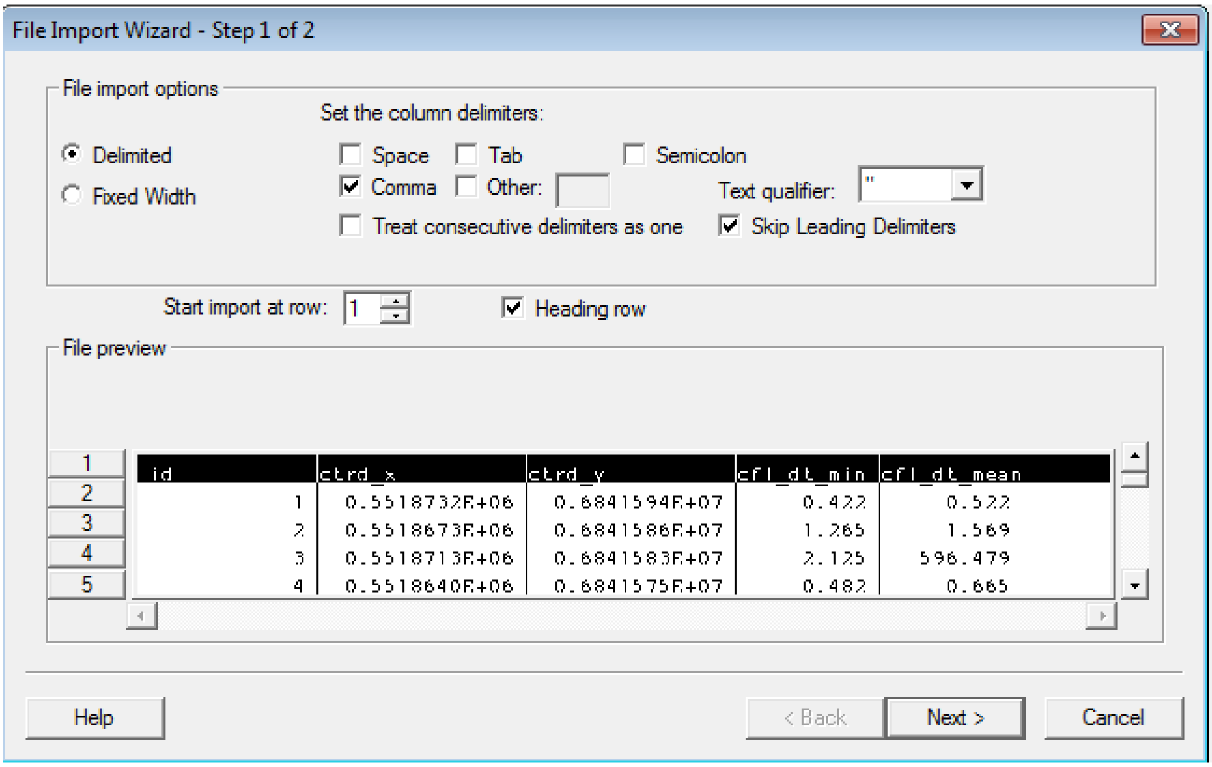
We will review the timestep information in SMS, using an output file created in the TUFLOWFV\input\log\ directory: In SMS open the RiverBend_Mesh001_ext_cfl_dt.csv file.



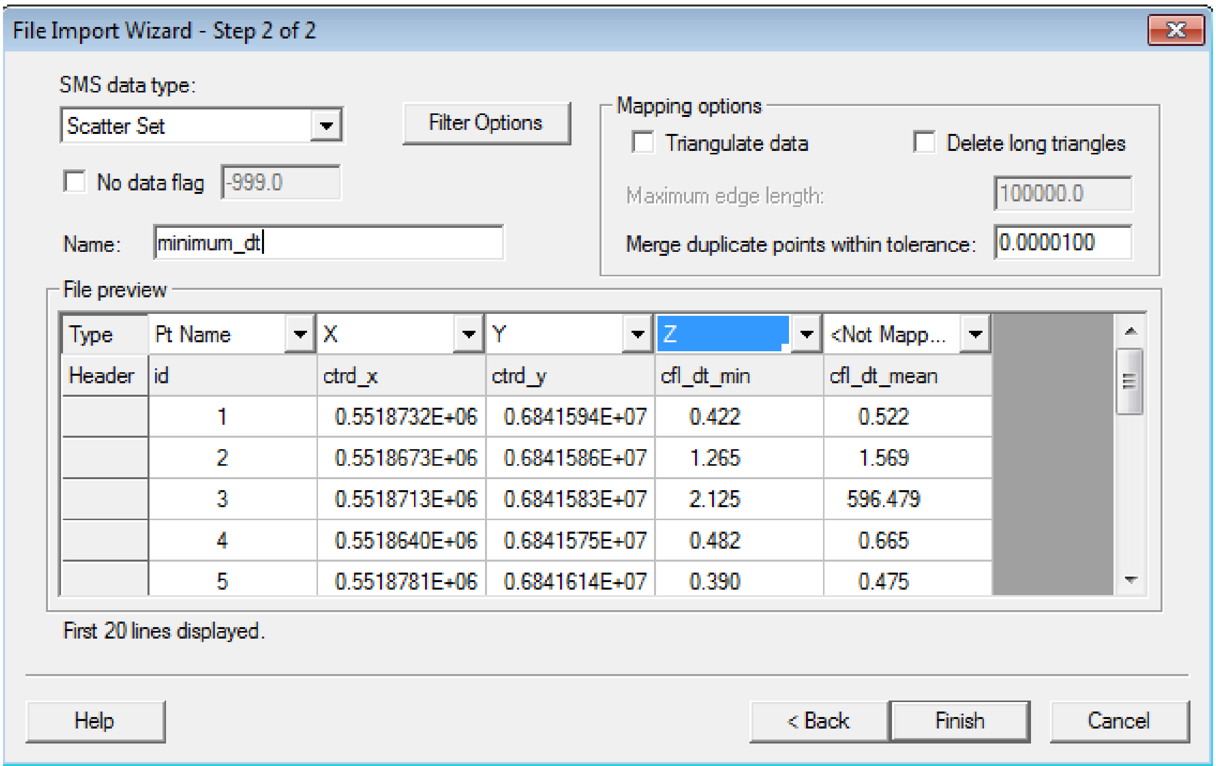
When prompted for a format to open the file, select “use import wizard”.



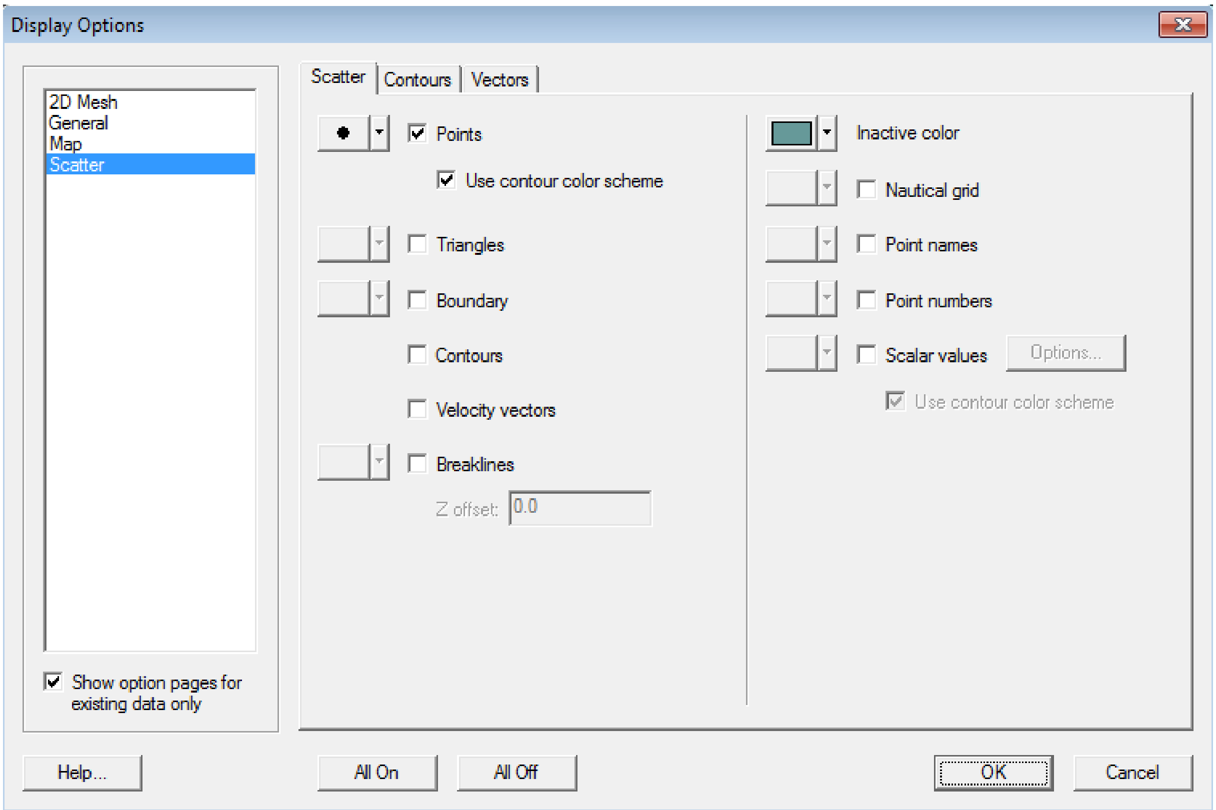
This wizard can be used to import a large variety of data into SMS. In this case the file is in a comma separated value (.csv) format. In the file import options select “Delimited” and select comma as the delimiter.



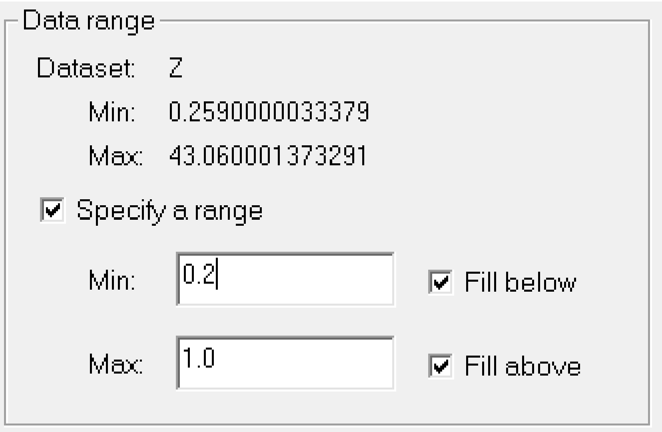
At the next prompt, turn off the triangulate data, and using the dropboxes, set the ctrd_x data to be mapped as X, the ctrd_Y to be mapped as Y and the dt_min (minimum timestep) to be mapped as Z. This is shown in the dialogue below:



Select “Finish” to open the data. There will be a new scatter dataset created, in the display options set the points to be visible, and select “Use contour colour scheme”.

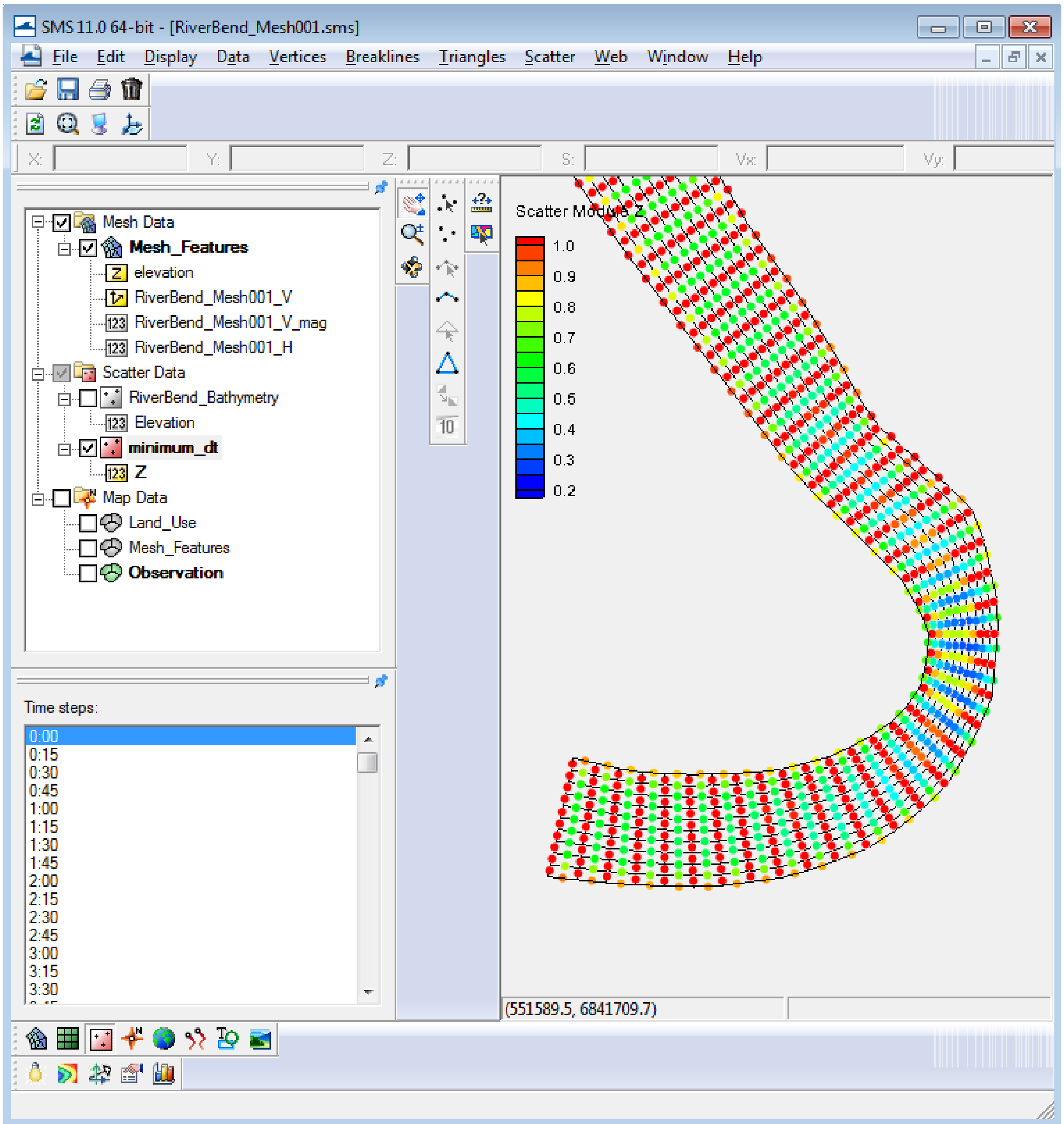


In the contour options, set the contour range to highlight the cells with small timesteps:



The timesteps should now appear as a series of points, as per the image below. This can be used to identify the cells that are limiting the timestep of the model. In this case the limiting cells are in the deep water around the bends in the model. To increase the speed of the model we would need to relax the mesh definition in these areas.

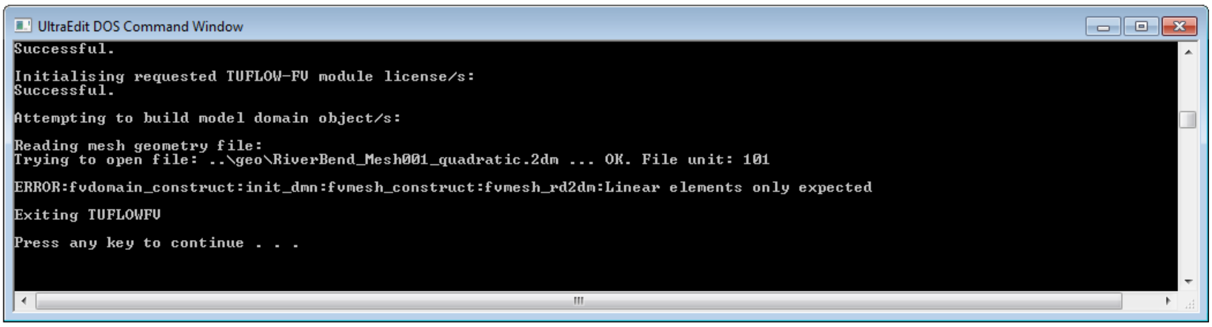
In Section 6.2.8, we look at refining the mesh in a shallow area, to see how this impacts on model runtime.



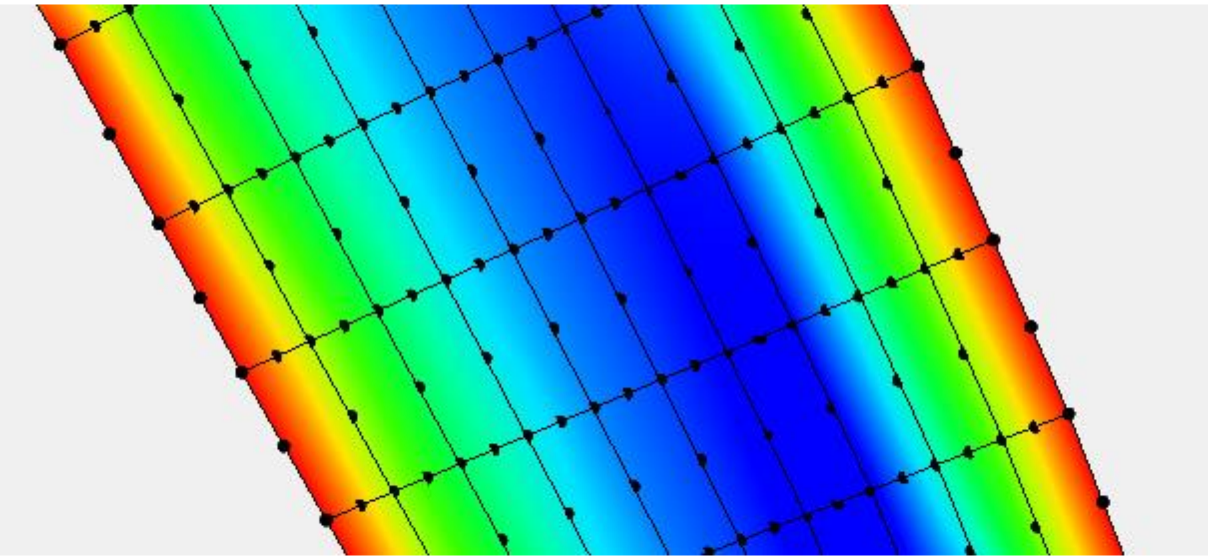
6.2.7 Troubleshooting

This section contains a list of the issues that may be encountered in the tutorial. If you are encountering a different problem, please email the log file, which can be found under TUFLOWFV\input\log directory to support@tuflow.com.

Error: fvdomain_constuct:init_dmn:fvmesh_construct:fvmesh_rd2dm:Linear elements only expected

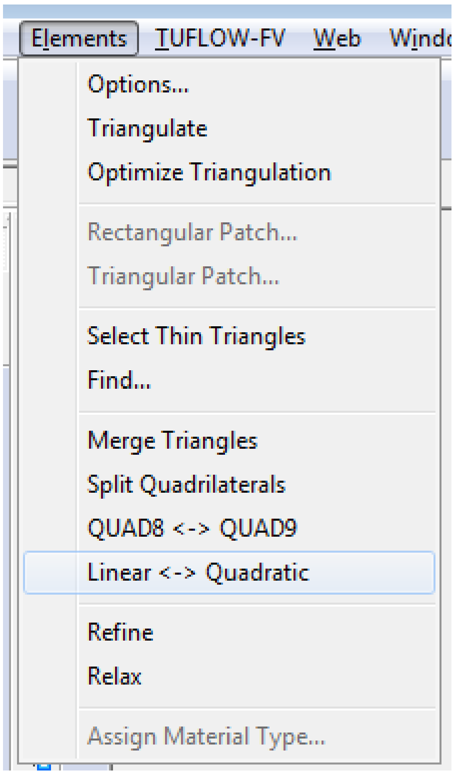


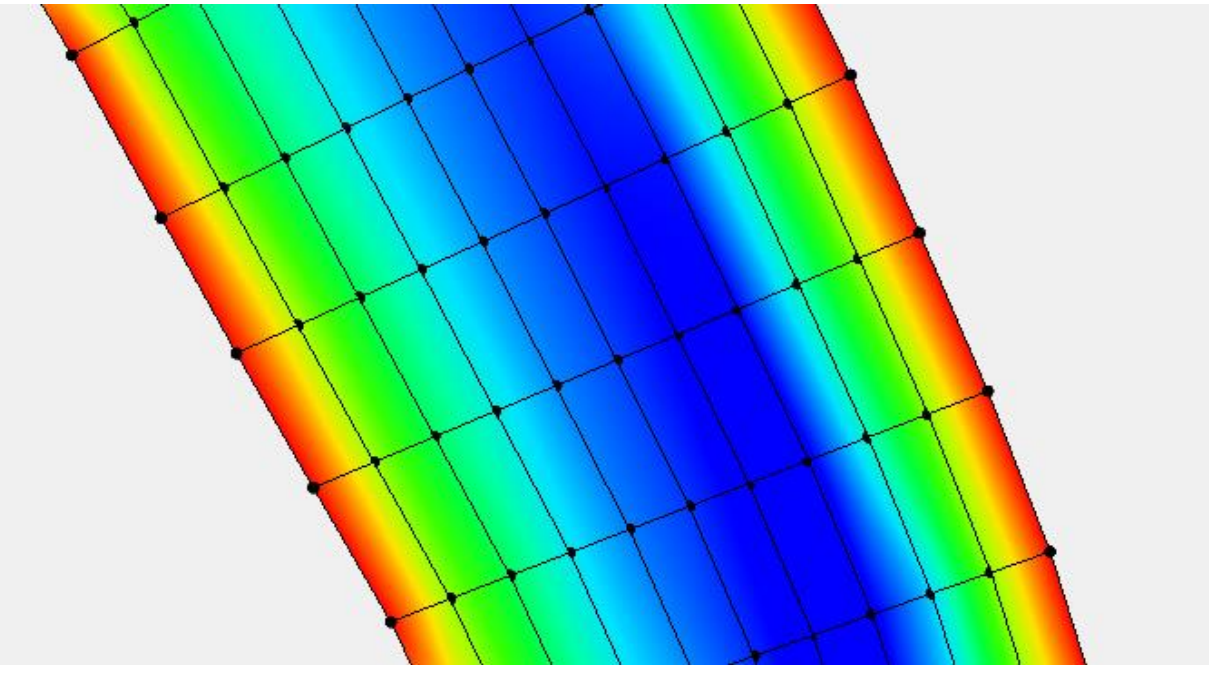
This error indicates that SMS has quadratic elements enabled. This means that the cell sides have nodes; this is not yet supported by TUFLOW FV. The mid side nodes can be seen by making the nodes visible in the display options (increase the size to make these easier to see).



Mid Side Nodes Enables (Quadratic)

In order to convert from quadratic (mid side nodes) to linear (cell corner nodes only) select Elements >> Linear < - > Quadratic. This switches between the two options.



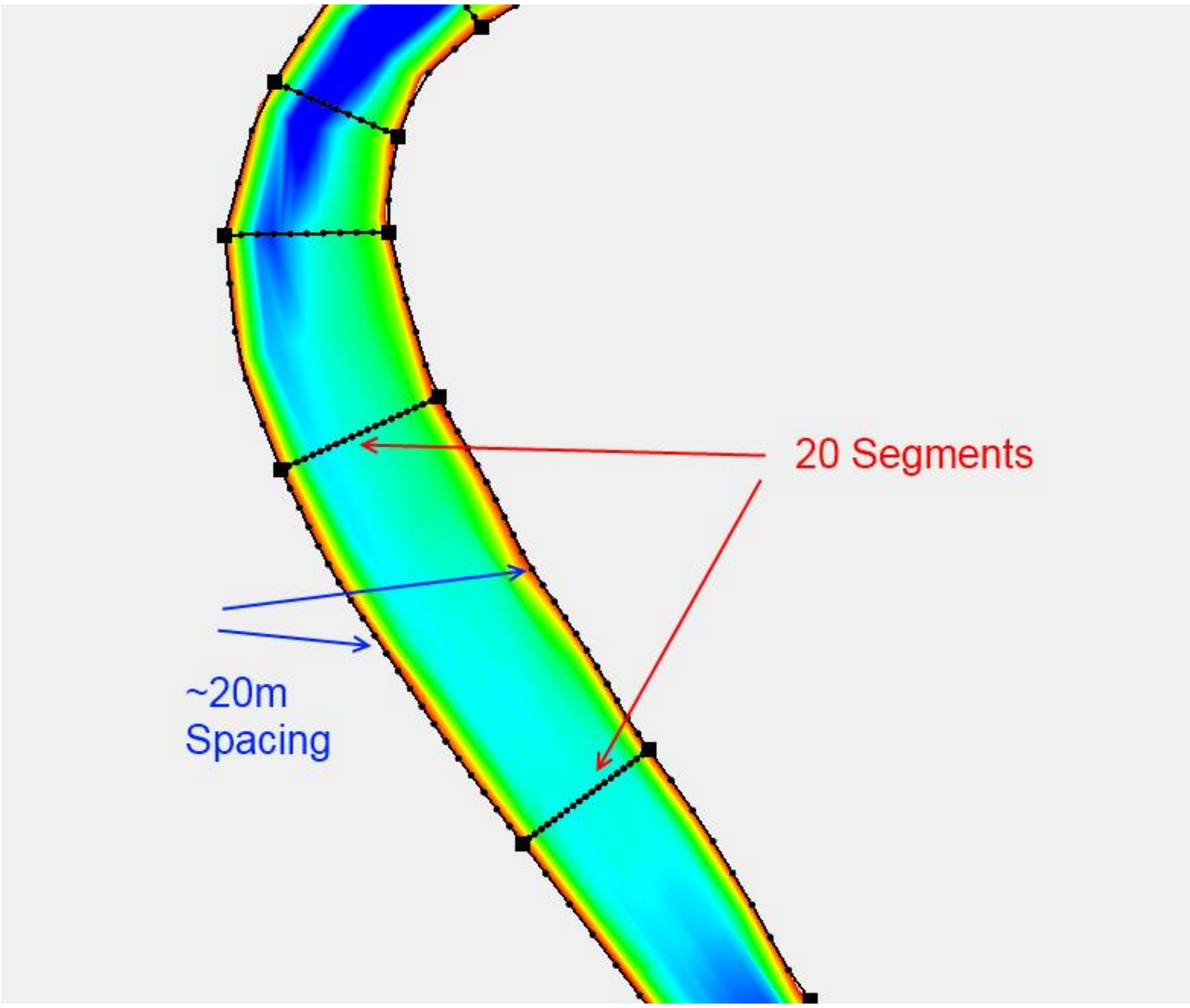


Cell Corner Nodes Only (Linear)

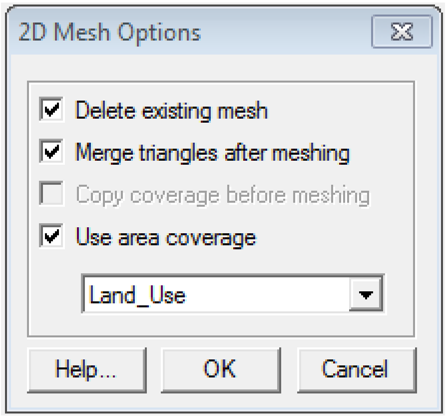
6.2.8 Optional Exercise: Refining the Mesh

In this section we will increase the resolution in the mesh to see how the impacts on the results and runtime of the model.

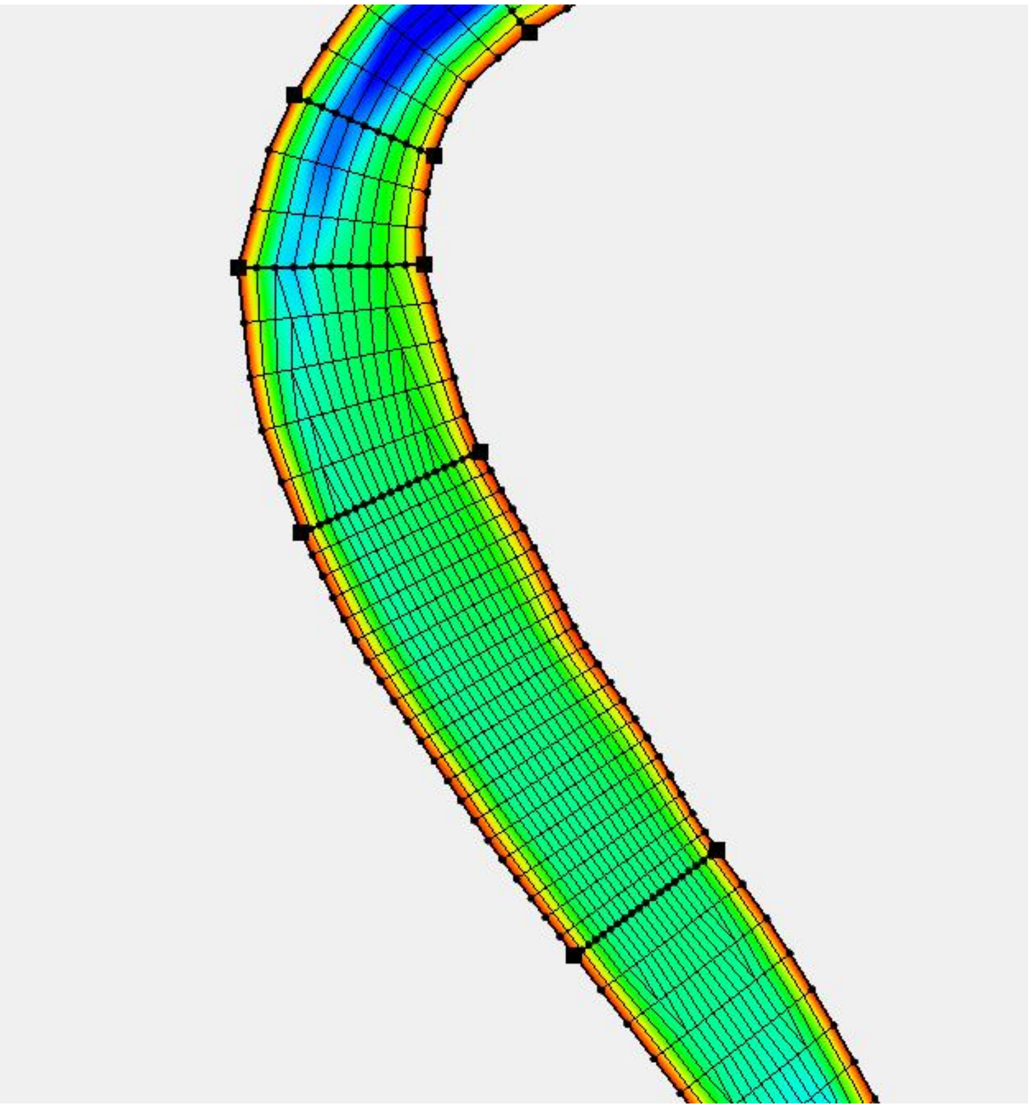
Save the project as RiverBend_Mesh002.sms to avoid overwriting the previous version of the model. Once a new project has been saved, in a shallow area of the model double to resolution in the model, see the image below for a suggested location.



Once the changes have been made to the feature arcs, rebuild the mesh (Feature Objects >> Map to Mesh). Make sure to tick the delete the existing mesh.



An example refined mesh is presented below:



If you are happy with the refined mesh, save the project and run the model again.

- How much did the model runtime increase?
- Did the timestep in the model change?

7 Tips, Tricks and Troubleshooting

7.1 Mesh generation tips

7.1.1 Primary goal

The primary goal when designing a flexible mesh is to **describe the key bathymetric and hydrodynamic features using the least, largest element sizes possible**. This is why flexible meshes are used; to optimise computational efficiency whilst achieving desired modelling accuracy.

7.1.2 Combine manual and automated mesh generation techniques

As shown in Section 5.1, creating a mesh is a combination of manual and automated steps. Keep it this way; 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.

7.1.3 Follow the contours

Water typically flows along contour lines. Ensuring that elements also follow the contours (for example, by using contours as arcs that define polygons and mesh regions) will in general produce the most efficient meshes. Remember to also include top-of-bank lines, thalwegs of channels, etc.

7.1.4 Build piece by piece

The map module of SMS allows you to construct pieces of your model, each defined as a polygon which in turn is defined by a series of arcs. Then, each polygon can contain specific mesh properties. Developing a mesh framework in this stepwise manner is recommended; the approach allows the flexibility to adjust components of the mesh design relatively easily and provides the balance of manual and automated.

7.1.5 Courant limits

TUFLOW FV is an explicit model. This means that the **timestep of the model is dependent upon the element which has the highest Courant number**. The Courant number (or CFL condition) limits each timestep in a model simulation as follows:

$$\Delta t < \Delta x / (\sqrt{(gd)} + v)$$

Where Δt = timestep, Δx is a nominal cell length, g is gravity, d is water depth and v is velocity.

This means that a small element in deep water and/or with a high velocity will likely become the limit for the timestep and hence the overall simulation time. It is important to **make sure that the element responsible for limiting the CFL condition has to be the size and shape it is**.

7.1.6 Which mesh type? Pave or Patch?

Pave is a series of triangles, Patch is a more uniform patch of quadrilaterals.

When considering mesh types it is important to reflect upon TUFLOW FV and how its computational scheme performs. As described above, a model geometry is best when it describes the physical features in the most computationally efficient manner. Generally speaking a patch mesh type is the most efficient mesh type and should be applied where possible. Also, a patch is easier to control (ie it is easier to keep element size and shapes regular and according to what you intend).

A patch of elements can only be done if there are 4 arcs defining the patch. In many instances this is not possible; natural features are often more irregular. Under such circumstances a pave mesh area should be used. Often, a mesh consists of a series of patches with paving connecting them together.

7.1.7 Interaction between DEM generation and mesh generation

As a general rule a flexible mesh design aligns with bathymetric / topographic contours and features. The mesh design is therefore intrinsically linked to the bathymetry and the data used to define it. It is important to be aware of this and interlink the processes of DEM generation and mesh generation.

Table 7-1 describes the process typically followed to create a DEM using GIS techniques. Also shown are the corresponding interactions with the mesh generation that can occur at each step.

Table 7-1 Interaction between DEM generation and mesh generation

Step	DEM Generation in a GIS	Interactions with mesh generation
1	Data is imported and quality checked.	Elevations in the mesh can be exactly those elevations measured if the mesh is snapped directly onto a data point.
2	Breaklines are defined to ensure consistency of levels between data points[1].	Breaklines specified in a GIS can be applied as arcs in the mesh generator, ensuring that the mesh alignment lies precisely along each breakline.
3	A TIN is generated.	Extracting elevations for a mesh using a TIN is more accurate than a DEM, especially if the

[1] Bathymetric data often requires some interpretation and adjustments when creating a DEM or model geometry. In particular, it is important that key topographic / bathymetric features are consistent and persist along their length. Examples include:

- A raised levee (or elevated road) is a key hydraulic feature for a flood simulation; it is important therefore to ensure that elevations between successive points along the levee are preserved.
- Similarly, if the thalweg of a natural flow channel is not preserved then a blockage to flows can occur.
- If cross-section surveys of river channels are conducted there is often some interpretation required to define the bathymetry between each cross-section. This is particularly the case around river bends and if linear interpolation between successive cross-sections is performed.

To address these issues within a GIS, breaklines are created.

Step	DEM Generation in a GIS	Interactions with mesh generation
		mesh alignment follows breaklines and is snapped to data points.
4	From the TIN, or via an alternative interpolation technique, a DEM is generated with a given resolution.	To avoid smoothing errors created by interpolating DEM elevations onto the mesh, a DEM resolution that is finer than the smallest element size is recommended.

7.1.8 The number of nodes and elements in a mesh

TUFLOW FV requires that there is consecutive numbering for nodes and elements in the input mesh file (the *.2dm file). In other words, if you have 100 nodes in your mesh then the highest node ID will be 100.

Mesh generation tools may allow you to have gaps in the ID lists of both nodes and elements, and this situation often occurs when you are adding or removing elements, etc during the mesh design process.

As a final step in the mesh generation process it is recommended that you **renumber** the mesh. To do this in SMS, follow the steps:

- Select a nodestring (any will do, however a boundary nodestring is preferred).
- Click on the menu command “Nodestrings -> Renumber” (or right click, and press “renumber”).

This will renumber all the elements and nodes. Note that all TUFLOW FV inputs are input via x/y coordinates or nodestring IDs, so renumbering should not influence your model runs (although it may be pertinent to check this, especially if you have identified a particular element or node ID to extract results).

7.1.9 Does node and element numbering influence computational performance?

No, not really.

Renumbering a mesh does have a small influence on the computational performance of TUFLOW FV; a “better numbered” mesh will have smaller memory allocation.

This is different to other flexible mesh models (implicit finite element models such as RMA for example), where mesh design and numbering have significant impacts upon computational performance.

7.2 How do I design a mesh for a river bend?

Check out the tutorial exercise in Section 6.2.

7.3 My model runs too slow

Don’t immediately go and request a bigger computer.

Remember that TUFLOW FV uses a flexible mesh and is limited by Courant criteria.

In other words, if you double model resolution then expect an 8 fold increase in simulation time (i.e. 4 times more cells and ½ the timestep). Watch out for small elements in deep water or in high velocity situations.

Use the flexible mesh to your advantage by adjusting the mesh to match your requirements and computational capacity.

A flexible mesh can be nested. This is advantageous when model simulations are becoming excessively long, or when (say) a regional model is performed over a long period and local models are run for sub-periods from it. In such circumstances, select specific model outputs at sufficient resolution from which boundary conditions for the nested models can be extracted.

7.4 Common reasons why a model crashes or won't start

7.4.1 You made a simple error

You may find that your simulation has crashed, or some other syntax error in the inputs has caused it to stop. If this happens, open the log file to see what may have gone wrong. Be logical and thoughtful in your model preparation; often it's a simple mistake that causes the most frustration.

7.4.2 Nodestrings and boundary conditions don't match

Check the 2dm file and make sure that the nodestring you are assigning a boundary condition to is the correct one.

Two ways to do this:

1 Within SMS, use the "Select Nodestring" tool in the Mesh module and click on the nodestring that you intend to be the boundary condition. On the display bar at the bottom of the SMS window the nodestring ID will be displayed; this is the nodestring ID to use in the TUFLOW FV .fvc file.

2 Open the 2dm file in a text editor and search for "NS" at the start of the line. The NS lines provide a list of nodes that define the specific nodestring. The final node on a nodestring has a "-" prefix, then the following number is the nodestring ID. It is this last number, the nodestring ID, that TUFLOW FV uses to identify the boundary condition.

Nodestring order does not influence open boundaries; inflow (into the model domain) is assigned positive and outflow (out of the model domain) is negative irrespective of the node order in the nodestring.

7.4.3 Initial condition / boundary condition mismatch

It's a common situation. Modellers always try to avoid warming up a model and hope that putting a 10,000 m^3/s inflow into an otherwise still model will run smoothly. TUFLOW FV is a relatively resilient model, but it has its limits.

As a quick fix, increasing the Stability Limits can assist. Otherwise a warmup of the boundary condition, transitioning from the initial state to the preferred boundary condition, should be considered.

7.5 Using multiple column csv files in a BC boundary

The BC command line (see Section 8.4.9) defines the csv file and column header associated with a particular boundary condition. Thus, the BC command line should have the column header of the time column and the boundary value column.

To illustrate, the following BC commands (extracted from an fvc file) define a series of 4 boundary conditions, each of which is a column in a multi-column csv file. BC 1 and 3 are nodestring flow (Q) boundaries, BC 2 is a cell inflow boundary (QC), and BC 4 is a nodestring water level boundary:

```
bc == Q, 1, ..\bcs\testbc.csv
  bc header == Time,QYB1
end bc
bc == QC, 216.5, 956.4, ..\bcs\testbc.csv
  bc header == Time,QYB2
end bc
bc == Q, 3, ..\bcs\testbc.csv
  bc header == Time,QX1
end bc
bc == WL, 4, ..\bcs\testbc.csv
  bc header == time,WSE
end bc
```

The first 5 lines of the corresponding csv file, which defines the values assigned to the boundaries, is as follows:

```
Time,QYB1,QYB2,QX1,WSE
0,28.31684659,28.31684659,28.31684659,9.35736
1,794.8538838,62.92003313,957.3421793,10.024872
2,802.7826009,60.56973486,981.4978653,16.965168
3,844.1251969,58.27607029,943.3474274,17.212056
4,868.477685,56.01072256,968.9588824,17.394936
5,901.9198808,53.80200852,997.8548085,17.535144
………
```

As shown, the BC command line defines the column headers which correspond to the first line in the csv file.

7.6 Structures

7.6.1 Overview

TUFLOW FV has a series of structure options, see Section 8.5 for a description of syntax.

Some important notes associated with the structures:

- Specification of an hQh relationship allows the user to insert practically any structure type (bridges, culverts, multiple structures, etc). This specification does however require some careful preparation of the hQh relationships.
- There are a number of alternative specifications of weirs, including weirs that are a given level above the existing ground level, or weirs that change elevation over time.
- "cell" type structures allow changes to be made to cell elevations (rather than cell faces). This is appropriate for simulating changing bed elevations over time.
- Flow can be in both directions. The "direction" of the structure is the same as the nodestring.
- For weirs, each cell face along the nodestring is considered as an individual weir, and flow is distributed accordingly. For hQh structures, flow is distributed uniformly across the nodestring according to the width of each cell face, and no adjustments are made to account for differences in water depths across the nodestring.

7.6.2 Using the hQh rating matrix

The hQh structure option in TUFLOW FV allows a flow relationship to be specified along a cell face, or several cell faces defined by a nodestring. 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 defined in a matrix of values (the hQh table). The following figure illustrates this.

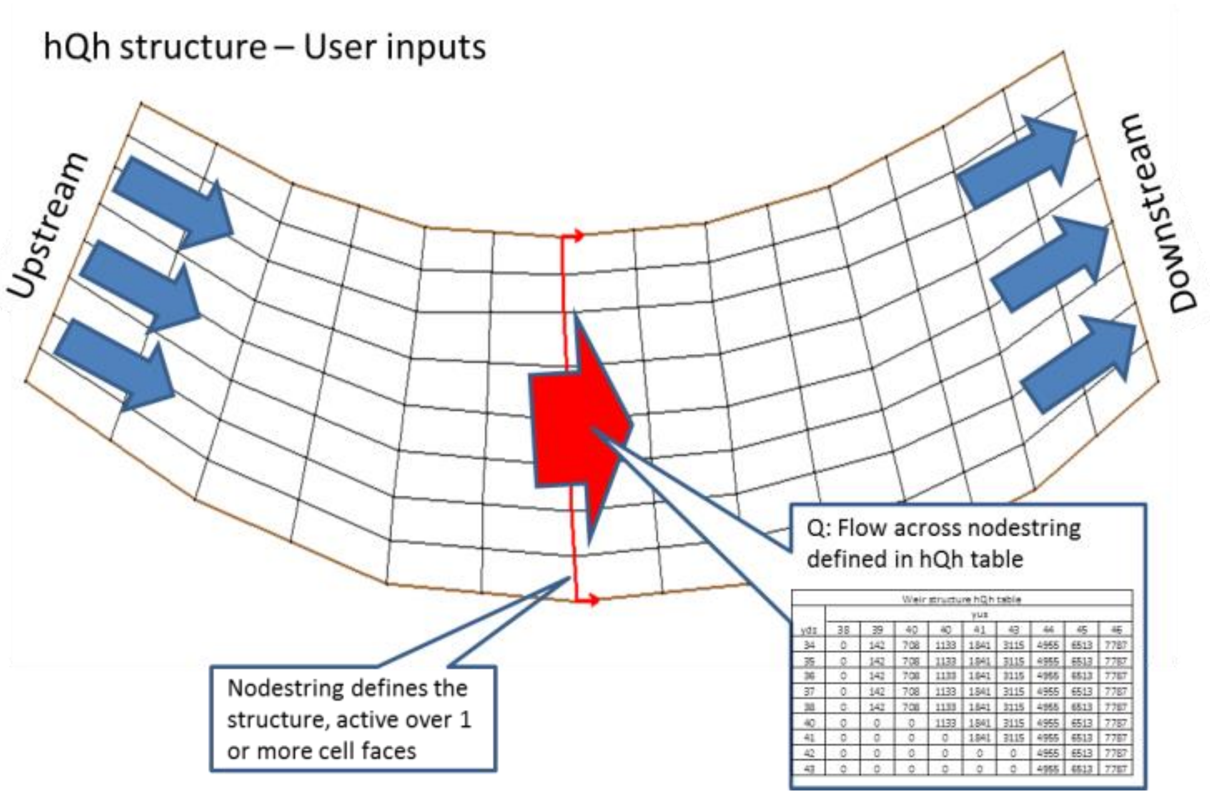


Figure 7-1 Illustration of the user inputs for an hQh structure

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[2] is performed between the tabulated flow (Q_{hqh}) and that calculated using the Shallow Water Equation (Q_{SWE}), where:

- IF $Q_{hqh} < Q_{SWE}$ THEN
  - Apply Q_{hqh} to cell face
- ELSE
  - Apply Q_{SWE} to cell face

4 Two momentum transfer options are available:

(a) Momentum is actively transferred through the structure based on Q_{hqh} and upstream velocity. This approach is recommended, especially for structures with relatively low energy losses (for example a bridge crossing where flow remains below the bridge deck). This is the default (and recommended) option[3].

(b) The structure is set to be a reflecting wall (and the "source-sink" transfer of Q_{hqh} is applied with no momentum). This approach can be considered for structures that represent a significant obstruction to flow, such as a dam. This approach is not generally recommended.

The following figure provides an illustration of the computation of the hQh structure.

[2] This check means that the hQh structure should represent a constriction to flow.

[3] This option has actually been set as default in the latest TUFLOW FV release - the reflecting wall momentum transfer type is no longer available.

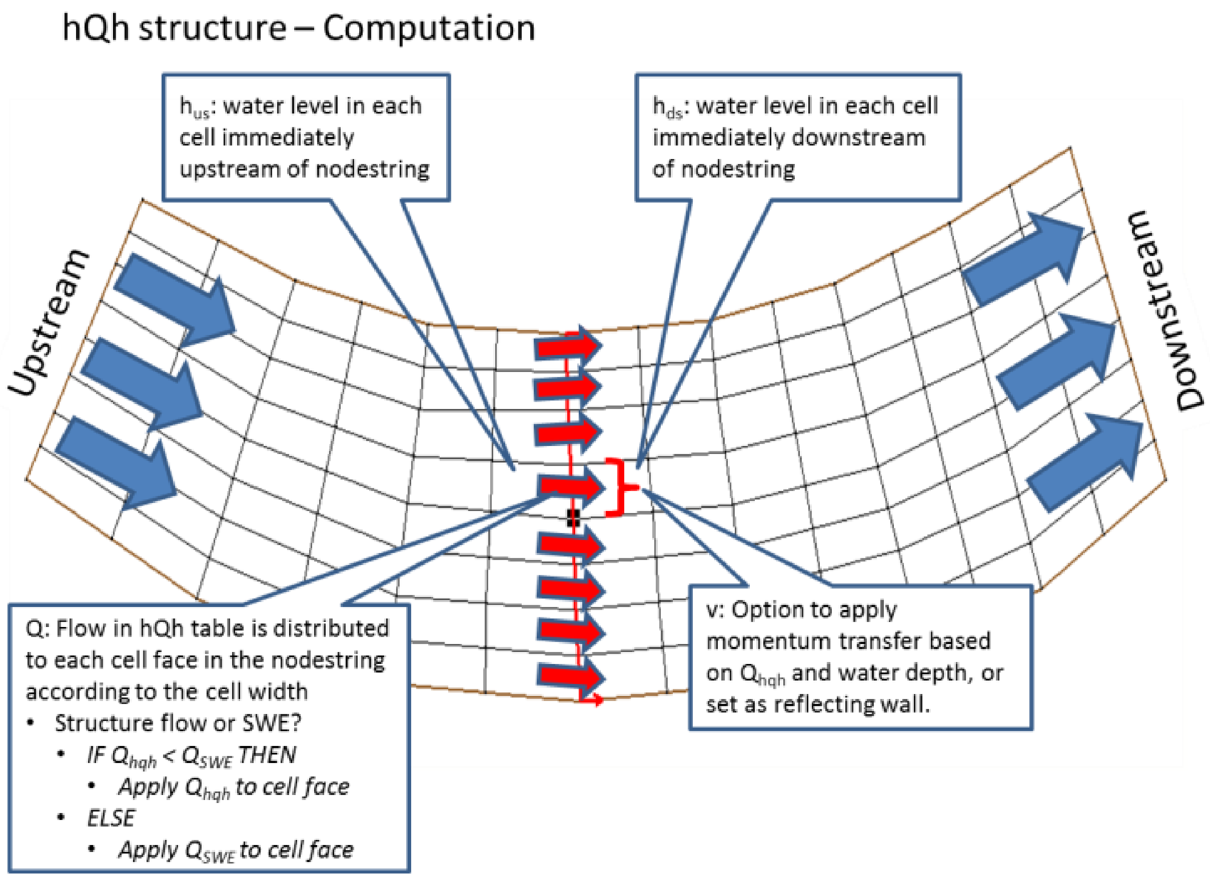


Figure 7-2 Illustration of the computational logic for an hQh structure

7.6.3 Calculating an hQh relationship

The TUFLOW FV hQh structure leaves the calculation of flow through the structure to the user. This makes the hQh structure flexible in its application (any structure, be it weir, culvert, pipe, etc can be applied) but also means that the user needs to create the hQh relationship. Options for doing this include:

- Calculation from first principles: This is relatively easy for simple structures.
- Use of other models: In particular, HEC-RAS is commonly used to establish flow conditions through structures. The calculated Q values from HEC-RAS simulations for a range of upstream and downstream water levels can provide a relatively straightforward means of creating a hQh matrix.

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 of the losses that occur in a structure or only internal losses. The following figure provides an illustration of this concept.

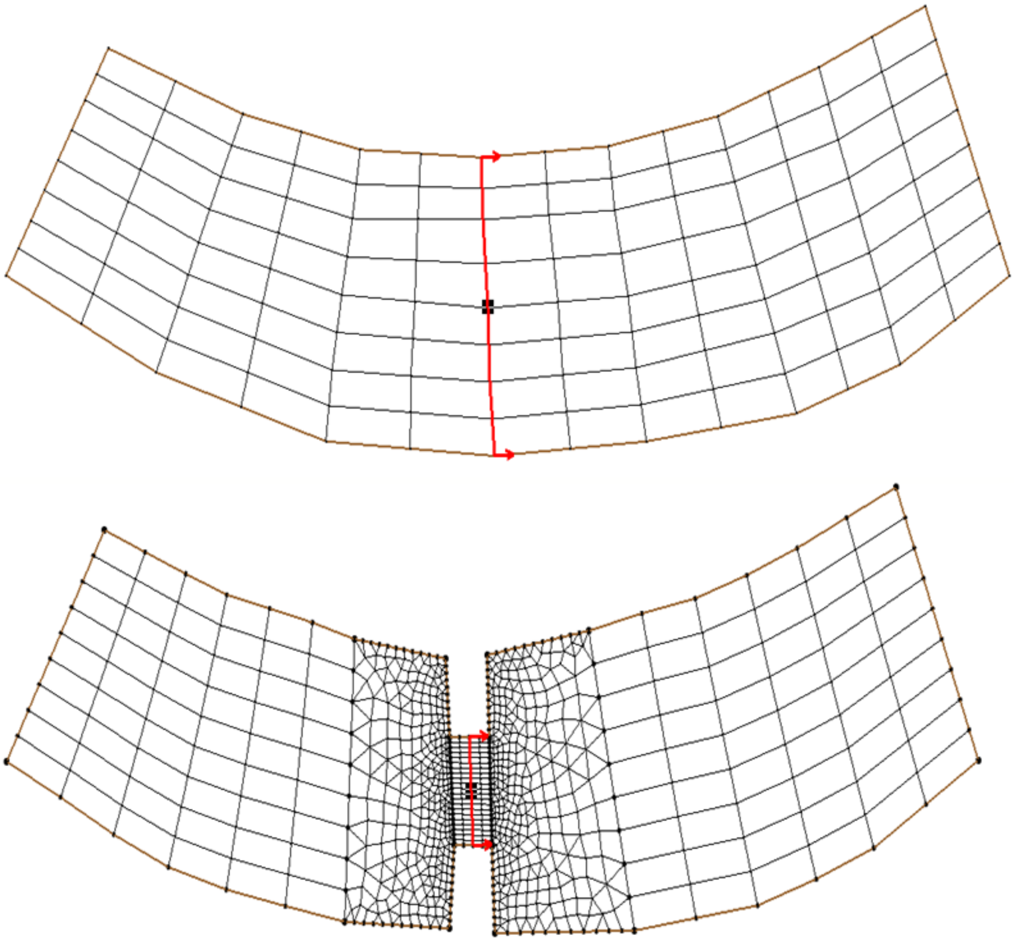


Figure 7-3 Examples of different approaches to defining a structure[4]

7.6.4 Logic controls

Logic controls adjust flow conditions through the 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, etc.

Note that adjustable the adjustable weir options are suitable for simulation of levee breach and failures, etc.

Contact support@tuflow.com for more information.

[4] The top image assumes that all losses (including entry and exit losses) are fully incorporated into the hQh relationship. The bottom image assumes that only internal losses (for example, friction losses through a culvert or pier losses through a bridge) are included in the hQh relationship, while entry and exit losses are simulated in the TUFLOW FV mesh.

7.7 TUFLOW FV is cell centred

The cells (or elements) are the computational blocks of the finite volume approach used by TUFLOW FV. This means that TUFLOW FV uses a single bed level value assigned to each cell in its calculations, then produces output that is applicable for each cell (cell velocities are derived from the values across each cell face).

This in itself is not a problem. However, at present SMS only permits values to be assigned at nodes; the corners of the cells. Thus, when TUFLOW FV reads the 2dm file during a model simulation the cell centred bed levels are interpolated from the corner node values. Then, when writing output via the "datv" output format, TUFLOW FV interpolates cell centred results back onto the corner nodes.

In many instances, this interpolation of both input bed levels and output results is not an issue. However, there may be instances where it is an issue. It is important to be aware of this constraint. The following sections provide some insight in this regard.

7.8 How do I get cell centred outputs?

SMS will not open a file with cell centred results and overlay it with the 2dm mesh file. If cell centred results are desired then there are some workarounds:

1 Save the results as a netcdf file using the output command "output == netcdf". Several MatLab scripts (and corresponding executable files that can be used in the absence of MatLab) are then available to export results from this format file. Contact support@tuflow.com for more information on the scripts.

2 Open a scatter dataset in SMS. The output command "output == dat" will produce an output file to do this.

7.9 Specific insertions into the model geometry: the "Cell elevation" command

The command line "cell elevation" provides an option to insert elevations at some or all cells (or elements) in the model domain. As outlined in the command reference, this is done by providing a csv file that lists the x and y coordinates of the specific cells, then the z values. xy coordinates (instead of the element IDs) are used in case element renumbering is performed as part of the mesh design, however if preferred cell IDs can be used.

More than one "cell elevation" command line can be entered, 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.

This option can be used to address the issues described in Section 7.7. For example, the invert along a drain could be specified using a single csv file entry (perhaps called "drain 01.csv"). This csv file could be directly extracted from a GIS polyline.

Insertion of cell elevation files allows the user to build a number of specific features into the model geometry in a systematic, structured manner, starting from the underlying geometry in the 2dm file and adding specific features (roads for example).

Note that 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.

7.10 Output of discharge along nodestrings

The command entry "output == flux" will output fluxes (discharge) and other relevant parameters from defined nodestrings. Specifying this command line will output values for ALL nodestrings listed in the 2dm file.

Extraction of fluxes from the model simulation using this command is recommended as opposed to post processed extraction via SMS. The interpolation from cell centres to corner nodes can create discrepancies in the flux extraction in SMS (see Section 7.7 for more information).

7.11 Mass balance in TUFLOW FV

TUFLOW FV applies the finite volume numerical method for its computational scheme. A feature of the finite volume method is that it conserves mass to numerical precision for all cells and for the entire computational domain. This is valid down to single precision accuracy for the TUFLOW FV engine build used in the study.

Mass balance can be checked via flux outputs along nodestrings and also by specification of the "volume" output parameter specification (See Section 8).

7.12 Distribution of flows across a nodestring "Q" boundary condition

There are two[5] ways to apply a flow boundary condition to a TUFLOW FV model:

1 Flow is distributed according to the width of each individual cell face along the nodestring (by setting "sub-type == 1" in the fvc input control file).

- If sub-type == 1, then the flow (Q_i) entering each of the (i = 1,…, n) cells along the boundary is distributed from the total flow (Q_{tot}) according to the width (w_i) of each cell face:

$$\mathbf{Q_i = Q_{tot} \frac{w_i}{\sum_{i=1}^{n} w_i}}$$

2 Flow is distributed according to the width and depth of each individual cell face along the nodestring (by setting "sub-type == 3" in the fvc input control file).

[5] Actually, there's 4 ways! But, sub-types 2 and 4 relate to a more specific boundary type applicable to 3D applications. Contact support@tuflow.com for more information.

- If sub-type == 3, then the flow (Q_i) entering each of the (i = 1,…, n) cells along the boundary is distributed from the total flow (Q_{tot}) according to the width (w_i) of each cell face and also the depth (h_i) in each cell:

$$\mathbf{Q_i} = \mathbf{Q_{tot}} \frac{\mathbf{w_i} \boldsymbol{h_i^{1.5}}}{\sum_{i=1}^{n} \mathbf{w_i} \boldsymbol{h_i^{1.5}}}$$

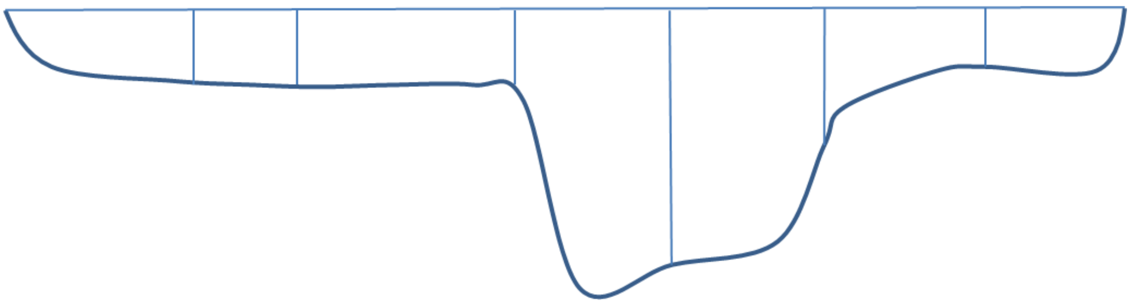
The logic for this formulation is derived from the Chezy equation describing friction flow;

$$\mathbf{Q = AC(RS)^{0.5}}$$

where Q is flow, A is area (width w * depth h), C is the Chezy coefficient, R is hydraulic radius (approximately equal to depth h) and S is slope. From this is a proportionality between flow Q and water depth h:

$$\mathbf{Q \approx h^{1.5}}$$

What does this mean for a model simulation? It is important to consider the flow distribution along inflow boundaries that have a significant variation in bed levels across the nodestring; a common example is a boundary condition representing a floodplain and main channel, illustrated as follows.



For this boundary condition, application of a sub-type == 1 will result in significantly higher velocities on the floodplains compared to the main channel.

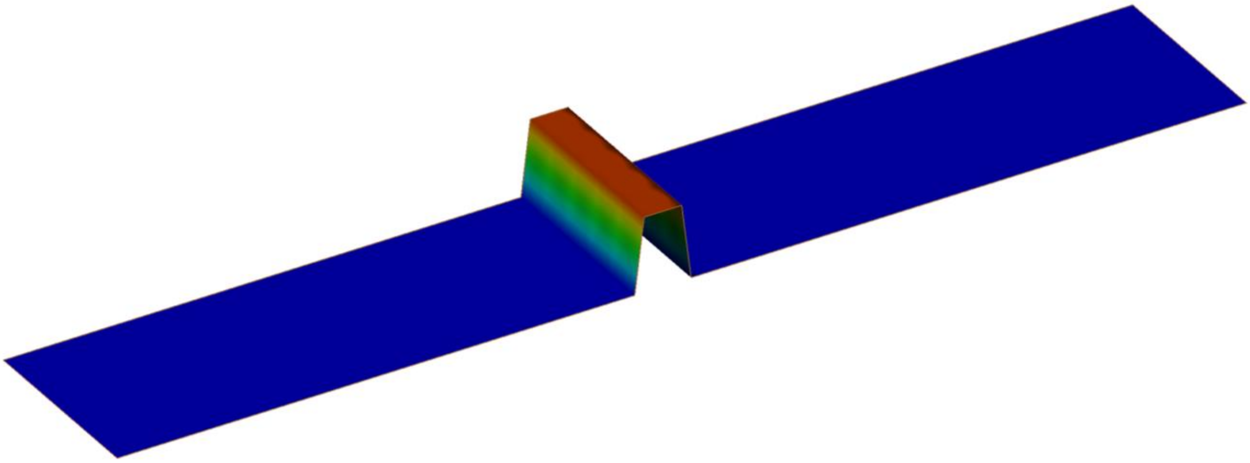
In comparison, application of a sub-type == 3 will distribute the flows so that there is more flow in deeper water, less flow in shallower water, and a generally uniform velocity distribution. This specification is recommended for the majority of inflow boundary conditions in overland and riverine situations.

7.13 How accurately does TUFLOW FV simulate weir flow when not applying a weir structure?

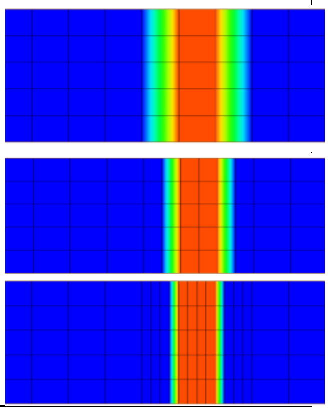
As a precursor to the following discussion, TUFLOW FV allows specification of the weir equation ($Q = CL(P+H)^{(3/2)}$, where coefficient C and crest level P are user inputs, see Section 8.5.1). Specifying a weir using this method provides an exact solution to the weir equation.

But, TUFLOW FV will simulate weir flow using the standard shallow water equations (SWE). How accurate is this approach?

Consider the following test with a broad crested weir which is 5 m high and 250 m across with a 1000 m^3/s discharge applied. Using the standard weir equation, the water level upstream of the weir is 6.765 m. Running a simulation without applying the weir equation (and using a Manning's n = 0.018), upstream water levels can differ.



A series of tests were applied with various cell resolutions across the weir and compared to the "exact" solution from the weir equation. The tests and the resulting elevation upstream are shown in the following table.

ID	Test	Elevation above weir crest level (m)	Images showing mesh resolution across weir (1, 2 and 4 cells)
1	SWE, 1 cell width	1.12	
2	SWE, 2 cell width	1.55	
3	SWE, 4 cell width	1.64	
4	Application of a weir nodestring structure, C = 1.7	1.77	
5	SWE, 1 cell width, but with the Manning's friction increased across the weir (n = 0.035)	1.30	
6	Standard Weir Equation	1.77	

As shown, the nodestring structure (test 4) perfectly matches the weir equation (test 6). The solution using the shallow water equations (SWE) tends to underpredict head loss across the structure. If the number of cells defined across the weir crest is increased, a more accurate solution is obtained. Note also that the solution is dependent upon friction (which is one of the dominant physical processes being simulated by the SWE, as shown in test 5).

Concluding, it is recommended that in situations where an accurate representation of weir flow is required a nodestring structure that inserts the weir equation into the solution scheme is used. In other situations, using the SWE (in other words, just letting TUFLOW FV simulate weir flow without any direction specification of weir structures) may be entirely acceptable. Importantly, the modeller should be aware that differences do exist between these two approaches.

8 Command File (FVC) Reference

8.1 List of Available Commands

Bottom roughness	Cell elevation	cell wet/dry depths
CFL	Decay Rate	Density air
Density water	Display dt	Echo geometry
Eddy viscosity	End material	End output
End scalar	End Time	g
Geometry 2d	global eddy viscosity	global scalar diffusivity
ic		
Include	Include mslp	Include salinity
Include sediment	Include temperature	Include wind
Include wavestress	Initial Water Level	Latitude
Limiter	Material	Mode split
Momentum mixing model		
Nscalar	Output	Output dir
Output interval	Output parameters	Output points file
Reference MSLP	Restart	Reset time
Scalar	Scalar diffusivity	Scalar mixing model
Sediment Control File	Settling Velocity	Spatial order
Spherical	Stability Limits	Start time
Timestep	Time format	Timestep limits
Write restart	Wind Stress Params	

8.2 Command line syntax

Each command line entry is defined by a descriptor, followed by a "==", followed by the specified value (or values) for the particular command line. The syntax in the tables that follow use a triangular bracket to specify a value that requires user specification.

- As an example, for the command line "***Include == <file name>***" the syntax inserted into the fvc would be (for example) "***Include == includefile.inc***".

For command lines that have an option of several values, a "**;**" separator is specified in the syntax.

- For example, the command line "***Time format == <Hours;ISODate>***" requires a choice of two options, so that the command line will be either "***Time format == Hours*** " or "***Time format == ISODate***".

For command lines that have a series of values to specify, a "," separator is specified in the syntax.

- For example, the command line "***Timestep Limits == <min timestep (s), max timestep (s)>***" requires two entries in the command line, such as "***Timestep Limits == 0.1, 10.0***".

Some command lines specify an "on or off" switch for a particular parameter.

- In such cases a "1" means "on" (or TRUE) and a "0" means "off" (or FALSE).

When specifying file names in the fvc file it is recommended that relative file paths are specified. This will make the TUFLOW FV simulation files more portable (it's easier to move an entire folder structure in this way). However, a full path name can also be inserted if preferred (a common example is when output files are written to a separate folder on another disk drive).

Strictly speaking, TUFLOW FV inputs are entered as integers (whole numbers), reals (float, or decimal numbers) and characters (text). The command line entries in the following tables adhere to this syntax, although real numbers can be inserted as integers.

- For example, the default "***CFL == 1.0***" can also be entered as "***CFL==1***".

Finally, take note that ***not all command lines have to be included in an fvc file***! A simple model setup often requires only a small list of command lines, while the remaining model parameters etc are either unused or remain as the default value.

8.3 Control File Layout

See Section 3.9 for further discussion on the layout of the fvc file.

Table 8-1 Recommended TUFLOW FV Control File Sections

Section	Command Categories
Definition	General definitions for the simulation, what modules are included, locations of files, etc
Time	Time Format and Reference Time
	Start / End Times
Geometry	Mesh file
	3D geometry definitions
Solution Scheme	Wetting / drying, CFL limits, etc
Turbulence	
Physical Parameters	
Materials	Material properties (roughness, mixing parameters, etc)
Initial Conditions	Initial model state (initial parameters, restart files, etc)
Boundary Conditions	Global (winds, waves, rainfall, etc)
	Nodestring (external boundaries, water levels, flows, etc)
	Cell (source)
	Node (point source)
Output	Output directory
	Prescribe model output
Additional Modules	Depending upon your preference, these commands can be included in the above structure (for example, your Definition category may include specification to include the advanced modules) or as separate entries (for example, if you started with a HD model then added salinity as a subsequent step).
	Structures
	3D
	Salinity, temperature, density
	Heat exchange
	Sediments
	Water Quality

8.4 Control File Structure (General)

8.4.1 Definition

Command Line	Default	Description
`Display Depth == <minimum display depth (m)>`	0.01	Allows the user to specify the minimum cell depth, for which a cell will be displayed as “wet” in SMS.
`Logdir == <path>`	Same location as fvc file.	Specifies the directory for writing the log file output. If not specified the Logdir defaults to the control file directory or to a /log subdirectory where this has been first created by the user.
`Include == <file name>`	No default.	At any location in the fvc file an include file can be included. At this location, all commands contained in the “include file” will be read as if they are listed in the fvc file.
`Debug == <0;1>`	0	Detailed program echo if set = 1.
`Units == <metric; English; Imperial; US Customary>`	Metric	Option to apply “US Customary” (or “English” or “Imperial”) units. Input and output units are as follows: • Elevation, distance = feet • Discharge = cfs • Manning’s friction is adjusted accordingly • Constant eddy viscosity value = ft2/s Note that currently the units are valid only for 2D hydrodynamics; please contact support@tuflow.com if considering using customary units for additional modules.

8.4.2 Time

Command Line	Default	Description
`Time Format == <Hours;ISODate>`	Hours	Specifies the format for time specification both in the control file and any boundary condition files. ‘Hours’ is the default and requires a decimal hour specification e.g. Start Time == 3.0.

Command Line	Default	Description
		'ISODate' requires a date specification in the form *dd/mm/yyyy HH:MM:SS* (or some truncation thereof), e.g. Start Time == 03/01/2009 03:00. Where inputs or output times are in decimal time, this will be relative to the reference time.
Reference Time == <Input/Output reference time>	For Time Format == Hours, the default is 0. For Time Format == ISODate, the default is 01/01/1990 00:00:00.	Sets the model reference time.
Start Time == <simulation start time>	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* (or some truncation thereof).
End Time == <simulation end time>	No default.	Specifies the end time for the simulation. See the Time Format command for input formats.
Timestep == <constant timestep (s)>	If not entered then variable timestep applied, see timestep limits.	Specifies the value of a constant timestep that is to be used during the simulation.
Timestep Limits == <min timestep (s), max timestep (s)>	No default.	Specifies the maximum and minimum timestep that are allowed when timestep is allowed to vary according to the Courant-Frederic-Lewy stability criterion. See also CFL.
Display dt == <display timestep (s)>	300 s (5 minutes).	Allows the user to specify the simulation interval between displaying timestep information.
CFL == <global maximum courant number>	1.0	Sets the courant Courant–Friedrichs–Lewy condition used in internal and external mode timestep calculation. The default value is 1., which is the theoretical stability limit.

Command Line	Default	Description
		Sometimes models can be successfully 'overclocked' with CFL>1.
CFL internal == <global maximum courant number>	1.0	Sets the courant Courant–Friedrichs–Lewy condition used in timestep calculation for the advective terms. The default value is 1., which is the theoretical stability limit.
CFL external == <global maximum courant number>	1.0	Sets the courant Courant–Friedrichs–Lewy condition used in timestep calculation for the free-surface gravity wave terms. The default value is 1., which is the theoretical stability limit.
CFL_dx min == <dx_min>	1.0	TBC

8.4.3 Geometry

Command Line	Default	Description
Geometry 2d == <mesh file (.2dm)>	No default.	Specifies the model 2D geometry input file. The input file should be an sms generic .2dm mesh file. Only linear triangular and quadratic elements are supported. Cell elevations can be set separately, see also cell elevation file command. Eg: geometry 2d == ..\geo\ mesh_name.2dm

Cell elevation file == <cell elevation file (.csv), xytype, ztype>	If not entered then geometry reverts to geometry 2d file.	This command can be used to set the cell bed elevations for some or all cells in the model domain.
		
		If xytype = cell_ID or is blank:
		• The csv file contains a first line header then the columns:
		    ○ cellID, Z
		
		If xytype = coordinate:
		• The csv file contains a first line header then the columns:
		    ○ X,Y, Z
		
		If ztype = average:

Command Line	Default	Description
		• all z values identified within a given cell will be averaged. If ztype = overwrite: • the cell bed level will be the last z value read. Note that more than one cell elevation file can be listed, each entry supersedes the previous.
Echo Geometry == <0;1>	1	Setting this to 0 stops the model from writing a .geo (geometry) output file.
Spherical == <0;1>	0	Specifies that the model is in spherical coordinates. • 0 = Cartesian where input coordinates are in metres. • 1 = Spherical where input coordinates are in degrees.
Partition == <val>		TBC

8.4.4 Solution Scheme

Command Line	Default	Description
Cell dry/wet depths == <cell dry depth (m), cell wet depth(m)>	<1.0 x 10^{-6}, 1.0 x 10^{-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). 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.
Cell wet/dry depths == <cell dry depth (m), cell wet depth(m)>	As above	In case you are wet before you are dry ;)
Stability Limits ==	No stability limits (i.e. the model does not	Specifies a maximum water level and maximum velocity, which define an unstable model if exceeded.

Command Line	Default	Description
<maximum WL, maximum velocity>	undertake stability checks).	The run will stop when these limits are exceeded.
Spatial Order == <1;2 (horizontal), 1;2 (vertical)>	<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. As a general rule of thumb, 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 TVD limiting schemes employed during the higher-order reconstructions.
Include Coriolis == <0;1>	1 (true)	Includes the Coriolis force source term from the momentum conservation equations. 0 for false, 1 for true.
Include invisc == <0;1>	1 (true)	Include the inviscid flux terms in the momentum and mass transport equations. 0 for false, 1 for true.
Include visc == <0;1>	1 (true)	Include the viscous flux terms in the momentum and mass transport equations. 0 for false, 1 for true.

Command Line	Default	Description
Include bed friction == <0;1>	1 (true)	Option to turn off bed friction.
Include parallel transport == <0;1>	1 (true), but only if a spherical coordinate system is applied.	Includes the parallel transport terms in the momentum flux equations (spherical coordinates only). 0 for false, 1 for true. These terms ensure that advective tendencies follow great circle paths on the sphere. This will be significant for very large domains (ocean scale) or at high latitudes but may be neglected for smaller domains.
Equation of state == <UNESCO; Direct>	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: the salinity tracer is assumed to be a direct proxy for density.
Horizontal gradient limiter == <LCD;MLG>	LCD	Sets the Total Variation Diminishing (TVD) limiting scheme for 2^{nd} order horizontal spatial integration scheme. The options are LCD (Limited Central Difference) and MLG (Maximum Limited Gradient). • LCD is the less compressive option and the least computationally intensive • MLG is the most compressive option and the most computationally intensive
Horizontal AlphaR == <alphaH (depth), alphaV (velocity), alphas (scalars)>	<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 higher-order simulation. Default is <1.0, 1.0, 1.0>, i.e. no gradient reduction, whereas <0.0, 0.0, 0.0> would revert to a first-order scheme.
Mode split == <0;1>	0	TBC
External mode 2D == <0;1>	0	TBC

8.4.5 Turbulence

Command Line	Default	Description
Turbulence update dt == <timestep (s)>	If not specified this will occur at every model timestep.	Specifies the timestep for vertical turbulence mixing eddy-viscosity and scalar-diffusivity term updating.
Scalar mixing model == <None; Constant; Smagorinsky; Elder; Warmup>	None	Sets the scalar mixing model. See also global scalar diffusivity. • None: no horizontal scalar mixing • Constant: specify a constant isotropic scalar diffusivity • Smagorinsky: specify the Smagorinsky coefficient –calculates an isotropic scalar diffusivity • Elder: specify longitudinal and transverse coefficients – calculates a non-isotropic diffusivity
Momentum mixing model == <None; Constant; Smagorinsky>	None	Sets the horizontal eddy viscosity calculation method. See also global eddy viscosity. • None: no horizontal momentum mixing • Constant: specify a constant eddy viscosity • Smagorinsky: specify the Smagorinsky coefficient –calculates a local eddy viscosity
Kinematic viscosity == <kinematic viscosity value (m^2/s)>	1.0e-6	Specifies the kinematic viscosity.
Global horizontal eddy viscosity == <eddy viscosity; coefficient/s (m^2/s;-)>	0.0	Globally sets the eddy viscosity coefficient. This is dependent on the turbulence model. • Constant: specify a constant eddy viscosity • Smagorinsky: specify the Smagorinsky coefficient. See momentum mixing model command to set momentum mixing turbulence model.
Global horizontal eddy viscosity limits == <v1>, <v2>		TBC
Global horizontal	0.0	Globally sets the diffusivity or diffusivity model coefficients. This is dependent on the

Command Line	Default	Description
scalar diffusivity == <diffusivity; coefficient/s (m^2/s;-)>		turbulence model used. • Constant: specify a constant isotropic scalar diffusivity • Smagorinsky: specify the Smagorinsky coefficient • Elder: specify longitudinal and transverse coefficients – calculates a non-isotropic diffusivity See scalar mixing model command to set scalar mixing turbulence model.
Global horizontal scalar diffusivity limits == <v1>, <v2>		TBC
Diffusivity limiter dt == <v1>		TBC
External turbulence model dir == <dir>		TBC

8.4.6 Physical Parameters

Command Line	Default	Description
g == <gravitational acceleration m/s^2>	9.81	Gravitational acceleration.
Latitude == <latitude in degrees (-ve for Southern Hemisphere)>	0.0	Sets the latitude for Coriolis calculations. Not required when a spherical coordinate system is used (see also Spherical).

8.4.7 Materials

Command Line	Default	Description

Command Line	Default	Description
Bottom drag model == <'Manning' ; 'Ks'>	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 should be specified. An alternative model, assumes a log-law velocity profile and requires specification of a surface roughness length-scale, "ks". The global bottom roughness and material bottom roughness commands can be used to specify the bottom roughness value/s to be used in the model.
Global bottom roughness == <bottom roughness)>	No default.	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).

8.4.7.1 Description of Material Block Commands

Command Line	Default	Description
Material == <material id #> … … … **End Material**	One block required for each material type specified in geometry file (see also .2dm mesh file).	This command indicates the beginning of a material block, specifying properties for cells with material id #. Material properties are listed in the following rows. Example material block: material == 1 bottom drag == 0.020 eddy viscosity == 0.20 scalar diffusivity == 60.0, 6.0 end material Note that several material types can be grouped into a single material block: material == 4,6,9,11,12 !Forest, etc bottom roughness == 0.1 end material

Command Line	Default	Description
Material group == <group>		TBC
Inactive == <0;1>	0	If 1 (true) then cells with material ID are excluded from the computational domain.
Bottom roughness == <roughness value>	Default to global value.	Sets the bottom roughness value. The bottom roughness specification depends on the Bottom drag model, and may be a Manning's "n" coefficient or an equivalent Nikuradse roughness, "ks" (m).
Surface roughness == <roughness value>	0.0	Sets the surface roughness value (for example, ice cover).
Horizontal eddy viscosity == <eddy viscosity; coefficient (m^2/s;-)>	Default to global value.	This command defines the eddy viscosity value/model-coefficient for a given material type (overwriting any default or globally defined values). This is dependent on the turbulence model used (constant or Smagorinsky). See momentum mixing model command to set momentum mixing turbulence model.
Horizontal eddy viscosity limits == <dv_limit1, dv_limit2>	Default to global value.	
Horizontal scalar diffusivity == <diffusivity; coefficient (m^2/s;-)>	Default to global value.	This command defines the scalar diffusivity value/model-coefficient/s for a given material type (overwriting any default or globally defined values). This is dependent on the turbulence model used (constant, Smagorinsky or Elder). See scalar mixing model command to set scalar mixing turbulence model.
Horizontal scalar diffusivity limits == <ds_limit1, ds_limit2>	Default to global value.	TBC
Vertical eddy viscosity limits == <dv_limit1,	Default to global value.	TBC

Command Line	Default	Description
dv_limit2>		
Vertical scalar diffusivity limits == <ds_limit1, ds_limit2>	Default to global value.	TBC
End Material		This command indicates the end of a material block.

8.4.8 Initial Conditions

Command Line	Default	Description
Initial Water Level == <water level (m)>	No default.	Globally sets the initial water level. Alternative options for setting Initial Conditions are the IC or Restart commands.
Initial Condition 2d == <initial condition file (.csv)>	Not used if not entered.	Reads in a comma separated variable file of initial conditions. This csv file contains initial conditions for each cell of the mesh. The following column headers are required in this file: ID, WL, U, V, [Sal], [Temp], [Sed_1,…], [Scal_1,…] An example of the command usage and corresponding CSV file is given below: ic == ..\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 ……………
Initial scalar profile == <initial condition file (.csv)>	Not used if not entered.	TBC

Command Line	Default	Description
Initial condition OGCM == <IC>	Not used if not entered.	TBC
Restart == <restart file name>	Not used if not entered.	Loads model initial conditions from a restart file. Unless the reset time command is used the simulation start time will be set to the timestamp in the restart file. See also write restart command.
Reset Time == <1;0>	0 (false).	This command resets the model start time to be equal to the value specified using Start Time when a restart file is used (see also Restart). Without this command or when set to 0 (false), the start time is set equal to the restart file timestamp.

8.4.9 Boundary Conditions

Command Line	Default	Description
Grid definition file == <netcdf file defining grid coords (.nc)>		Specifies a netcdf filename that defines grid coordinates to be used in mapping input/output files to the model mesh. This command should be used in conjunction with the W10_grid, MSLP_grid and Wave_grid BC types to establish the grid to mesh mapping. Multiple BCs can point to the same grid definition.
Grid definition variables == <v1, v2, v3>		TBC
Grid definition == <x0, y0, alp, mx, my, dx, dy, typ>		Geometry definition parameters for grid definition file, including • Origin (x0, y0) • Grid size (dx, dy) • Angle (alp) • Number of cells (mx, my) • Typ (TBC)

8.4.10 Description of BC Block Commands

Command Line	Default	Description

BC == <bc type, [id], [input file]>
…
…
…
…
End bc
	One block required for each boundary type.	This command indicates the beginning of a Boundary Condition (BC) Block.
		
		See Table 8-1 for list of boundary types.
		
		Boundary conditions can be:
		• Global (winds, waves, rainfall, etc)
		• Nodestring (external boundaries, water levels, flows, etc)
		    o The [id] value is the nodestring identifier from SMS (or, if using SMS versions earlier than 11.0, the sequential order of the nodestrings in the 2dm file).
		• Cell (source)
		    o The [id] value is the cell ID from the geometry.
		
		Possible commands that can be used to specify a BC block are:
		• BC header
		• BC offset
		• BC scale factor
		• BC update dt
		• Includes MSLP

BC Header == <Header1,Header2,…>		Allows the user to specify the CSV input file column headers or NETCDF file variable names (overwriting the defaults in Table 8-2). This command should immediately follow a BC command. For example, the following lines apply a cell inflow at the cell which lies at the xy coordinate 1025.5, 950.5. It looks in the specified csv file for columns: Time,Tailwater_Flow,Turbidity: BC == QC, 1025.5, 950.5, ..\bc\tailwater_discharge.csv BC header == Time,Tailwater_Flow,Turbidity End BC Another example shows a nodestring flow boundary applied to nodestring 1, which looks in the specified csv file for columns: Time,INFL1A:

Command Line	Default	Description
		BC == Q, 1, ..\bc\ flowbc.csv BC header == Time, INFL1A End BC

Sub-type == <subtype>	1	This command is only applicable for a [“Q” type](#) nodestring flow boundary condition.
		• If subtype = 1 (default)
		  ○ The flow boundary condition is distributed across a nodestring by cell width.
		• If subtype = 3
		  ○ The flow boundary condition is distributed across a nodestring by cell width and depth.
		Specific to 3D applications:
		• If subtype = 2
		  ○ The flow boundary condition is a source inflow into the first string of cells inside the nodestring boundary condition, with flow distributed as per subtype =1.
		• If subtype = 3
		  ○ The flow boundary condition is a source inflow into the first string of cells inside the nodestring boundary condition, with flow distributed as per subtype =3.

BC offset == <Var1_Offset, [Var2_Offset],...>		Specify offset/s to be applied to boundary condition values.
BC time offset == <timeoffset>		TBC
BC scale == <Var1_Scale_Factor, [Var2_Scale_Factor] ,...>		Specify scale factors to be applied to boundary condition values.
BC flag == <1;0>		0: False 1: True

Command Line	Default	Description
BC time scale == <timescale>		TBC
BC update dt == <Update timestep>	Updated at every model timestep.	Allows the user to specify the update timestep for a boundary condition. This is especially useful for gridded boundaries. If not specified the BC is updated at every model timestep.
Includes MSLP == <1;0>	1	Allows the user to specify whether the various water level boundary conditions include an inverse barometer offset. The default assumption (1) is that the boundary does already include an inverse barometer component. If Includes MSLP == 0 then an offset determined by the local MSLP difference from the reference MSLP is applied at the boundary.
Layer == <layer>		Vertical layer to apply boundary condition.
Nlayers == <nlayers>		Number of vertical layers in boundary condition.
Vertical distribution == <vdfil>		Csv file containing the vertical distribution of layers.
Vertical coordinate type == <ztyp>		Options for coordinate type: • Elevation • Depth • Sigma • Height
BC nodestrings == <id1,....,idn>		TBC
Sub type == <wsm>		Applicable for wave inputs: • 1 = cell-centred radiation-stress gradient area integration • 2 = face-centred radiation-stress boundary integration

Command Line	Default	Description
End BC		This command indicates the end of a BC block.

Table 8-2 BC types

BC Type	BC Description	ID	Input File	Default Columns Header[6]
AIR_TEMP	TBC			
AIR_TEMP_GRID	TBC			
CLOUD	TBC			
CLOUD_GRID	TBC			
CP	TBC			
CYC_HOLLAND	Parametric cyclone wind and pressure field	N/A	CSV	TIME, X, Y, PO, PA, RMAX, B, RHOA, KM, THETMAX, DELTAFM, WBGX, WBGY
FB	Sediment bed flux	Cell	CSV	TIME, FLUX_SED_1,…
FBM	TBC			
FC	Cell scalar flux	Cell	CSV	TIME, [FLUX_SAL]. [FLUX_HEAT], [FLUX_SED_1,…], [FLUX_SCAL_1,…]
FCM	TBC			
LW_NET	TBC			
LW_NET_GRID	TBC			
LW_RAD	TBC			
LW_RAD_GRID	TBC			
MSLP_Grid	Mean sea level pressure field	Grid	NETCDF	TIME, MSLP
OBC	Fully specified boundary condition	External nodestring	CSV	TIME, WL, U, V, [SAL], [TEMP], [SED_1,…], [SCAL_1,…]
OBC_PROF	TBC			
OBC_CURT	TBC			
OBC_GRID	TBC			
OP	Zero-gradient	External nodestring	N/A	Not Required

[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.

BC Type	BC Description	ID	Input File	Default Columns Header[6]
PRECIP	TBC			
PRECIP_GRID	Precipitation grid	Grid	NETCDF	
Q	Nodestring flow	External nodestring	CSV	TIME, Q, [SAL], [TEMP], [SED_1,…], [SCAL_1,…]
QC	Cell inflow (m^3/s) - uses internal concentration during outflow.	Cell	CSV	TIME, Q, [SAL], [TEMP], [SED_1,…], [SCAL_1,…]
QCA	Cell inflow (m^3/s) - uses specified concentration during outflow.	Cell	CSV	TIME, Q, [SAL], [TEMP], [SED_1,…], [SCAL_1,…]
QCM	TBC			
QG	Global Cell Inflow (m/s) - uses internal concentration during outflow.	N/A	CSV	TIME, Q/A, [SAL], [TEMP], [SED_1,…], [TRACE_1,…]
QGA	Global Cell Inflow (m/s) - uses specified concentration during outflow.	N/A	CSV	TIME, Q/A, [SAL], [TEMP], [SED_1,…], [SCAL_1,…]
REL_HUM	TBC			
REL_HUM_GRID	TBC			
RNS	Reflective, no Slip	External nodestring	N/A	N/A
RS	Reflective, free slip.	External nodestring	N/A	N/A
SCALAR	TBC			
SCALAR_PROF	TBC			
SCALAR_CURT	TBC			
SW_RAD	TBC			
SW_RAD_GRID	TBC			
SURF_TEMP	TBC			
SURF_TEMP_GRID	TBC			
TRANSPORT	TBC			
W10	Global 10m Wind	N/A	CSV	TIME, W10_X, W10_Y
W10_Grid	10m Wind field	Grid	NETCDF	TIME, W10_X, W10_Y
Wave	Wave parameter field	Grid	NETCDF	TIME, HSIGN, TPS, DIR, FORCE_X, FORCE_Y
Wave_coupled	TBC			
WL	Water level	External nodestring	CSV	TIME, WL, , [SAL], [TEMP], [SED_1,…], [SCAL_1,…]
WL_CURT	TBC			
WLNR	TBC	External	CSV	TBC

BC Type	BC Description	ID	Input File	Default Columns Header[6]
		nodestring		
WLS	Sloping Water Level	External nodestring	CSV	Time, WL_A, WL_B, , [SAL_A, SAL_B], [TEMP_A, TEMP_B], [SED_1_A, SED_1_B,…], [SCAL_1_A, SCAL_1_B,…]
ZG	Zero gradient boundary	External nodestring	N/A	N/A
ZVAR	TBC			

8.4.11 Output

Command Line	Default	Description
Output dir == <filepath>	Same location as FVC file.	Specify the output directory for results files. E.g. output dir == D:\FVWBM\Output\ or output dir == ..\Output\
Write restart dt == <time (hours)>		Writes a restart file (to the directory where the .fvc file sits) at the time specified. The restart file is a binary file. The restart file is read in using the restart command.
Restart overwrite == <0;1>	1	Overwrite restart file at each restart dt step or create a series of restart files (each file has a counter included in the name). • 1 means the restart file will be overwritten • 0 means the restart file will not be overwritten

8.4.12 Description of Output Block Commands

Command Line	Default	Description
Output == <output format> … … …		This command indicates the beginning of an output block, and specifies the type of output. Table 8-3 presents the output types available. Output block properties include: • Output Interval • Output Parameters

Command Line	Default	Description
`end output`		• Output Points File Example output block: output == datv output parameters == h,v,scal_1,scal_2 output interval == 900 end output
`Suffix == <suffix>`		Output file name suffix option.
`Output Parameters == <many>`		Specify the required output parameters; see Table 8-4. Note that not all parameters are supported depending on output type (see the output command and Table 8-3 for details).
`Output points file == <file name (.csv)>`		This provides the name of a file with the coordinates of output points, required for a points output type. This file is a CSV format containing x and y coordinates of the desired output locations, additional columns are ignored. E.g. Output points file == ..\Points.csv Points.csv contents: X, Y ID (not used) 314000., 7368000., Point 1 300000., 7350000., Point 2 …….
`Start Output == <time>`	If not specified this will default to the simulation start time.	Specify the start time for an output request. The time format must be consistent with the simulation time format.
`Final Output == <time>`	If not specified this will default to the simulation end time.	Specify the final time for an output request. The time format must be consistent with the simulation time format.
`Output Interval == <timestep (s)>`	0, resulting in output at every timestep!	Output interval in seconds.
`Output compression`	0	If = 1 then output compression is activated.

Command Line	Default	Description
== <0;1>		
End Output		This command indicates the end of an output block.

Table 8-3 Output Types

Output Format	Description	Parameters	Relevant Commands
dat	Sheet output at cell centroids in SMS .dat format. Note that this output format can be read in as a scatter dataset and not as a data file attached to a 2dm geometry file (see "datv" below).	H, V, D, Z, Sal, Temp, Sed_1,…, Scal_1,…, W10, MSLP, Hsig, T[, Wvdir, Wvstr	Output Parameters Output Interval
datv	Sheet output at cell vertices (nodes) in SMS .dat format. This is the required format to view results in SMS.	H, V, D, Z, Sal, Temp, Sed_1,…, Scal_1,…, W10, MSLP, Hsig, T[, Wvdir, Wvstr	Output Parameters Output Interval
flux	Flux across nodestrings specified in .2dm file. Note that entering a flux output type will provide outputs at ALL nodestrings listed in the input 2dm geometry file.	N/A. This will output flow, and salinity, temperature, sediment and scalar fluxes as required.	Output Interval
mass	Outputs a mass comma separated variable file with mass output.	N/A. This will output flow, and salinity, temperature, sediment and scalar "mass" as required.	Output Interval
netcdf			
netcdfv			
points	Outputs result timeseries at specific locations as a csv file. A points file needs to be read in using Output Points File command.	H, V, D, Z, Sal, Temp, Sed_1,…, Scal_1,…, W10, MSLP, Hsig, Tp, Wvdir, Wvstr	Output Points File Output Parameters Output Interval
Transport	TBC		

Table 8-4 Output Parameters

Parameter	Description
Air_temp	Air temperature (degrees Celsius)
Bed_mass_total	Total bed mass (kg/m^2)

Parameter	Description
Bed_mass_Layer_#	Bed mass in layer # (kg/m^2)
Bedload	Bed load (g/m/s)
Bedload_TOTAL	Total Bed load (g/m/s)
D	Water depth (m)
Deposition_total	(g/m^2/s)
DZB	Bed elevation change (m)
FLOW	(m^3/s)
H	Water surface elevation (m)
Heat_content	(Degrees Celsius m^3)
LW_rad	Downward long wave radiation flux (W/m^2)
MSLP	Mean sea level pressure (hPa)
Netsedrate_total	(g/m^2/s)
Pickup_total	(g/m^2/s)
PRECIP	Precipitation rate (m/day)
Rel_hum	Relative humidity (%)
Rhow	Water density (kg/m3)
Sal	Salinity concentration (TBC)
Salt_flux	(psu m^3/s)
Salt_mass	(psu m^3)
Sed_#	Suspended concentration of sediment fraction # (mg/L)
Sed_#_BED_MASS	Sediment bed mass of fraction # (kg)
Sed_#_FLUX	Suspended sediment flux of fraction # (10^{-3} kg/s)
Sed_#_MASS	Suspended sediment mass of fraction # (10^{-3} kg)
Sedload	Sediment load (g/m/s)
Sedload_TOTAL	Total Sediment load (g/m/s)
Suspload	Suspended load (g/m/s)
Suspload_TOTAL	Total Suspended load (g/m/s)
SW_rad	Downward short wave radiation flux (W/m^2)
Taub	Bed shear stress (N/m^2) (Hydrodynamic module)
Taus	Surface shear stress (N/m^2) (Hydrodynamic module)
Tauc	Current related effective bed shear stress component (N/m^2) (Sediment transport module)
Tauw	Wave related effective bed shear stress component (N/m^2) (Sediment transport module)
Taucw	Combined effective current/wave bed shear stress (N/m^2) (Sediment transport module)

Parameter	Description
Temp	Temperature (degrees Celsius)
Temp_flux	(degrees Celsius m^3/s)
THICK	Total bed thickness (m)
Trace_#	Tracer concentration (units/m^3)
Trace_#_FLUX	Tracer flux (units m^3/s)
Trace_#_MASS	Tracer mass (units)
TSS	Total suspended solids concentration (mg/L)
TURBZ	Output of vertical turbulence parameters, which includes: • TURBZ_TKE (m^2/s^2) • TURBZ_EPS (m^2/s^3) • TURBZ_L (m) • TURBZ_SPFSQ ($/s^2$) • TURBZ_BVFSQ ($/s^2$) • TURBZ_NUM (m^2/s) • TURBZ_NUH (m^2/s) • TURBZ_NUS (m^2/s)
V	Velocity vector (m/s)
Volume	(m^3)
W	Vertical velocity (m/s)
W10	10 m wind speed vector (m/s)
WQ_ALL	Output of water quality parameters
WQ_DIAG_ALL	Output of water quality parameters
Wvht	Wave height (m) - typically significant wave height
Wvper	Wave period (s) - typically peak wave period
Wvdir	Wave direction (degrees true coming from)
Wvstr	Wave stress vector (N/m^2)
ZB	Bed elevation (m)

8.5 Control File Structure (Advanced)

The following command line entries are required to include additional features and modules of TUFLOW FV beyond the standard 2D hydrodynamic.

8.5.1 Structures

Command Line	Default	Description
`Structure == <Structype, ID>` …	No default.	Marks the beginning of a structure block. Structype can be: • Nodestring

Command Line	Default	Description
… … **End Structure**		○ The structure is situated between one or more elements (ie – along the cell faces, defined by a nodestring) ○ The [id] value is the nodestring identifier from SMS that represents the structure in the model geometry. • Cell ○ The structure is a series of cells, defined by a polygon. Presently, this defines a series of cells with an adjustable bed elevation (although other cell definition structures will come online in future). ○ No [id] value is required.
Name == <sname>		Name of structure
Flux function == <fluxtype>	None	If structype = nodestring then the flux function type is required. Flux function type can be: • Wall: ○ a solid wall (Q=0) • Matrix: ○ an hQh relationship defines the structure (contained in the flux file) • Weir: ○ A broad crested weir structure with a fixed crest level • Weir_dz: ○ A broad crested weir structure with a crest level dz above existing bed levels • Weir_adjust: ○ A broad crested weir structure with an adjustable crest level • Weir_dz_adjust: ○ A broad crested weir structure with an adjustable crest level dz above existing bed levels • Porous: ○ A porous structure (Darcy flow conditions) • Timeseries: ○ A specified timeseries of flow

Command Line	Default	Description

Cell function == <celltype>	None	If structype = cell then the cell function type is required.
		
		Cell 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

Flux file == <hQh file>		If fluxtype = matrix then a flux file is required. 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 given below: Weir Structure example Yds, yus 1., 0., 2., 3., 5. 0., 0., 10., 100., 125. 1., 10., 30., 100., 125. 2., 20., 50., 100., 125. 4., 30., 10., 100., 125.
Polygon file == <polyfile>		Reads in a comma separated variable file with a polygon. The file contains a header line with column labels "x" and "y", which define the points

Command Line	Default	Description
		describing the perimeter of the polyfile. 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.

Properties == <p1,....,pn>		If fluxtype = "Weir" or "Weir_dz", then
		• P1 = weir crest level (for a weir) or level above existing bed levels (for a weir_dz)
		• P2 = weir coefficient (default = 1.6)
		
		If function type = "Porous" then
		• P1 = Porous structure hydraulic conductivity
		• P2 = Porous structure width

Control == <controltype>		Specification of structure logic definition.
		
		If the structure fluxtype= weir_adjust or weir_dz_adjust, or the structure celltype = zb_adjust or dzb_adjust, then options available are:
		• Control == Trigger
		  ○ the change in levels will commence upon the exceedence of a specific trigger value.
		• Control == Time series
		  ○ The change in levels will commence according to a defined time series.
		
		Other options are available (see support@tuflow.com for more information):
		• Fully_open
		• Timeseries
		• Sample_rule
		• Target_rule

Sample point == <spx, spy>		If control == trigger then • spx, spy defines the location that controls the variable z value structure (ie the "control" point)
Sample dt == <sdt (hours)>		The frequency of updating the variable structure (hours).
Trigger value ==		The value of the specified model parameter at the sample point spx, spy that, when exceeded,

Command Line	Default	Description
<tv>		will trigger a change in structure elevations. Note that currently the trigger value can only be an absolute water level.

Control file == <cfile>		A comma separated file with structure controls.
		
		The file contains a header line with specific column labels required for specific structure types:
		
		If flux function == weir_adjust
		• Column headers = "Time, weir_crest"
		
		If cell function == zb_adjust
		• Column headers = "Time, zb"
		
		If flux function == weir_dz_adjust or cell function == dzb_adjust
		• Column headers = "Time, dzb"
		
		The "Time" values are specified as:
		• If control == trigger
		  ○ Time is in hours from the moment that the structure adjustment commences.
		• If control == time series
		  ○ Time is in hours from the start of the model simulation.

Structure logging == <0;1>	0	If active a structural log file will be created (logfilename.slf) containing the operational behaviour of the structure through time.

8.5.2 Wind, Atmospheric Pressure and waves

Command Line	Default	Description
Include wind == <0;1>	True (1) if wind boundary condition specified through a BC command, otherwise false (0).	Includes wind forcing in the calculations. 0 for false, 1 for true. Wind forcing will be automatically activated if a wind boundary condition has been specified through a BC command. In this case include wind == 0 can be used to de-activate the forcing.

Command Line	Default	Description
Include wavestress == <0;1>	True (1) if wave boundary condition specified through a BC command, otherwise false (0).	Includes wave radiation stress forcing in the calculations. 0 for false, 1 for true. Wave stress forcing will be automatically activated if a wave model boundary condition has been specified through a BC command. In this case include wavestress == 0 can be used to de-activate the forcing.
Wave model == <wvmod>	'SWAN"	Presently, SWAN is the only wave model option.
Include MSLP == <0;1>	True (1) if an MSLP boundary condition specified through a BC command, otherwise false (0).	Includes atmospheric pressure forcing in the calculations. 0 for false, 1 for true. Atmospheric pressure forcing will be automatically activated if a MSLP field has been specified through a BC command. In this case include mslp == 0 can be used to de-activate the forcing.
Include stokes drift == <0;1>	0	Incorporate Stokes drift velocity. 0 for false, 1 for true.
Atmos update dt == <timestep (s)>	If not specified this will occur at every model timestep.	Specifies the timestep for performing atmospheric forcing.
Density Air == <air density (kg/m³)>	1.2	Allows the user to specify the density of air.
Density Water == <water density (kg/m³)>	1000	Allows the user to specify the reference density of water.
Theta baroclinic == <theta_b>		TBC
Reference Density == <Density (kg/m³)>	1000.	Sets the reference density value used in calculation of the baroclinic pressure term.
Reference MSLP == <Mean Sea Level	1013.25	Sets the reference mean sea level pressure value.

Command Line	Default	Description
Pressure (hPa)>		
Wind stress params == <W_a(m/s), C_a(-), W_b(m/s), C_b(-)>	The default parameters are <0., 0.8E-03, 50., 4.05E-03> corresponding to the Wu parameterisation (with a 50 m/s upper limit).	Specifies the parameter values in the following wind stress drag model: $C_d = C_a$; $[W_{10}<W_a]$ $C_d = C_a + (W_{10}-W_a)/(W_b-W_a)*(C_b-C_a)$; $[W_a<=W_{10}<=W_b]$ $C_d = C_b$; $[W_{10}>W_b]$
Wave stress params == <gamma, Hmin>		TBC

8.5.3 3D

Command Line	Default	Description
Vertical mesh type == <Sigma;Z>	Sigma.	Specifies the type of discretisation applied to the 3D layer structure. Can be either Sigma coordinates or fixed Z-level coordinates.
Min bottom layer thickness == <dzmin>		Specify the minimum thickness of the lowest layer (ie at the bed).
Sigma layers == <Nsigma>		Number of sigma layers.
Cell 3D depth == <3d_dep>		If the water depth is less than this value then the 3D cells essentially revert to a 2D representation.
Layer faces == <file specifying layer interface levels (.csv)>	Specifying a layer face file will result in a 3D simulation. A 3D module license must be available for this to proceed.	Specifies the name of the file containing the 3D layer face information. The coordinate type depends on the Vertical mesh type. In the case of sigma-coordinates the layer elevations need to be specified in a 'SIGMA' column. In the case of z-coordinates the layer elevations should be specified in a 'Z' column.
Vertical gradient limiter ==	MC	Sets the Total Variation Diminishing (TVD) limiting scheme for 2nd order vertical spatial integration scheme.

Command Line	Default	Description
<MINMOD;MC;SUPERBEE>		The options are MINMOD, MC (Monotized Central) and SUPERBEE (ranging from least compressive to most compressive).
Vertical AlphaR == <alphaV (velocity), alphas (scalars)>	<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>, whereas <0.0, 0.0> would revert to a first-order scheme.
Vertical mixing model == <Constant; Parametric; External>	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 fvwbm_external_turb.dll.
Vertical mixing parameters == <v1>, <v2>		TBC
Global vertical eddy viscosity limits == <v1>, <v2>		TBC
Global vertical scalar diffusivity limits == <v1>, <v2>		TBC
Initial Condition 3d == <initial condition file (.csv)>	Not used if not entered.	Reads in a comma separated variable file of initial conditions. This csv file contains initial conditions for each cell of the mesh. The following column headers are required in this file:

Command Line	Default	Description
		ID, WL, U, V, [Sal], [Temp], [Sed_1,…], [Scal_1,…] An example of the command usage and corresponding CSV file is given below: ic == ..\bc\initial_conditions_001.csv 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 ……………

8.5.4 Salinity, Temperature, Density

Command Line	Default	Description
Include salinity == <0;1, 0;1>	<0, 0> (no salinity)	Include salinity as a modelled parameter. 0 for false, 1 for true. The second flag specifies whether density is a function of the modelled salinity. 0 for false, 1 for true.
Include temperature == <0;1, 0;1>	<0, 0> (no temperature)	Include temperature as a modelled parameter. 0 for false, 1 for true. The second flag specifies whether density is a function of the modelled temperature. 0 for false, 1 for true.
Reference Salinity == <Salinity (PSU)>	0.	Sets the model reference salinity.
Reference Temperature == <Temperature (degrees celsius)>	20.	Sets the model reference temperature.
Initial Salinity == <salinity (psu)>	0.	Globally sets the initial scalar concentration fields.

Command Line	Default	Description
Initial Temperature == <temperature (degrees Celsius)>	0.	Globally sets the initial scalar concentration fields.

8.5.5 Sediments

Command Line	Default	Description
Include sediment == <0;1, 0;1>	<0, 0> (no sediment).	Include suspended sediment fraction/s as modelled parameter/s. 0 for false, 1 for true. The second flag specifies whether density is a function of the modelled sediment fractions (0 for false, 1 for true). Additional information pertaining to sediment modelling is specified through the Sediment control file.
Sediment Control File == <file name (.fvm)>	No default.	Specifies the sediment control file, which is required if the include sediment flag is set to 1. The sediment control file commands are described in Section 0.
Initial Sediment Concentration == (sed_1,...,sed_Nsed (mg/L)>		Globally sets the initial scalar concentration fields.

8.5.6 Heat Exchange

Command Line	Default	Description
Include heat == <0;1>	0	Include heat exchange in the model solution. 0 for false, 1 for true.
Heat cp == <cp>		Specific heat of water

Command Line	Default	Description
Heat cpa == <cpa>		Specific heat of air
Heat cln == <cln>		Latent heat transfer coefficient
Heat csn == <csn>		Sensible heat transfer coefficient
Heat albedo lw == <alb_lw>		Long wave radiation albedo
Heat water emissivity == <EPS_w>		Heat water emissivity
Heat PAR fraction == <PAR_frac>		Fraction of PAR SW radiation
Heat NIR fraction == <NIR_frac>		Fraction of NIR SW radiation
Heat UVA fraction == <UVA_frac>		Fraction of UVA SW radiation
Heat UVB fraction == <UVB_frac>		Fraction of UVB SW radiation
Heat PAR extinction == <PAR_eta>		Extinction coefficient of PAR SW Radiation
Heat NIR extinction == <NIR_eta>		Extinction coefficient of NIR SW Radiation
Heat UVA extinction == <UVA_eta>		Extinction coefficient of UVA SW Radiation
Heat UVB extinction == <UVB_eta>		Extinction coefficient of UVB SW Radiation

Command Line	Default	Description
Heat SED absorption == <Sed_abs>		Rate of light absorption by sediments
Heat ref height == <zrefa>		Meteorological sensor height
Heat albedo SW == <alb_swo>		Mean SW radiation albedo at equator
Heat relax dt == <heat_relax_dt>		Heat module relaxation timestep
Heat lh model == <LHmodel>		Latent heat transfer model
Heat lw model == <LWinput>		Long wave radiation heat transfer model
Heat sw model == <SWinput>		Short wave radiation heat transfer model

8.5.7 Water Quality

Command Line	Default	Description
Water quality model == <external>	None	Must be external.
WQ update dt == <timestep (s)>	If not specified this will occur at every model timestep.	Specifies the timestep for performing water quality parameter updating.
Initial WQ Concentration == <wq_1, …., wq_Nwq (mg/L)>		Globally sets the initial scalar concentration fields for water quality.

8.5.8 Tracer

Command Line	Default	Description
Ntracer == <number of passive tracers>	0. If Ntracer > 0 then Tracer Block commands are required.	Sets the number of passive tracers to be modelled.
Initial Tracer Concentration == (tracer_1,tracer_Ntrace (units/m^3)>		Globally sets the initial scalar concentration fields.

8.5.8.1 Description of Tracer Block Commands

Command Line	Default	Description
Tracer == <tracer id #> … … … **end tracer**		This command indicates the beginning of a tracer properties block, specifying the tracer id # that the properties should be applied to. Tracer properties include: • Settling Velocity • Decay Rate Example Tracer Block: tracer == 2 settling velocity == 1.0e-5 decay rate == 0.05 end tracer
Settling Velocity == <w$_{s0}$ (m/s)>		Specifies 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.
Decay Rate == <Kd (units/day)>		Specifies the scalar decay rate in concentration units/day. This results in a sink term flux, S: $S = K_dCh$

Command Line	Default	Description
		where *C* is the scalar concentration and *h* is the flow depth.
`End Tracer`		This command indicates the end of a tracer block.

9 Sediment Module Control File (fvm) Reference

9.1 List of Available Commands

Concentration profile model	Consolidation model	Nlayer
Flocculation settling model	Nline	Layer
Hindered settling model	Nsed	rhodry
Erosion model	ws0	Mass
End layer	taucd	tauce
End material	rhos	Erosion rate params
End output	Material	Output dir
End sed frac	ks	Output
Output parameters	Output interval	
Write restart	Flocculation settling params	
Sed frac	Hindered settling params	
Concentration profile params	Consolidation model params	

9.2 Description of General Commands

Concentration Profile Model == <Uniform;Rouse>

Sets the concentration profile model. Uniform = concentration uniform with depth. Rouse = the con

Concentration Profile Params == <TBC>

TBC

Flocculation Settling Model == <Constant;Concentration;Concentration&Salinity>

Sets the flocculation settling model. Constant =

Flocculation Settling Params == <TBC>

Hindered Settling Model == <None;RZ>

Sets the hindered settling model. None = hindered settling neglected. RZ = hindered settling is calculated according to Richardson and Zaki (1954).

Hindered Settling Params == <TBC>

TBC

Erosion Model == <Metha>

Sets the erosion model.

Consolidation Model == <None;Constant>

TBC

NSed == <number of sediment fractions (between 1 and 100)>

This command specifies the number of sediment fractions in the simulation. Each sediment fraction has specific properties defined in the Sed frac block. The maximum number of fractions is 100.

Nlayer == <number of bed layers (between 1 and 10)>

This command specifies the number of bed layers in the simulation. Each bed layer has specific properties defined in the layer block. The maximum number of bed layers is 10.

9.3 Description of Sediment Block Commands

Sed frac == <Nsed id #>

This command indicates the beginning of a sediment fraction block, specifying properties for the sediment fraction with Nsed id #. Sediment fraction properties include:

ws0

taucd

rhos

Example sediment fraction block:

```
sed frac == 1
   ws0 == 0.001
   taucd == 0.10
   rhos == 2650
end material
```

ws0 == <settling velocity (m/s)>

This command specifies the sediment fraction settling velocity in m/s.

If the flocculation settling model is “constant” and hindered settling is neglected, the sediment settling velocity is not influenced by the concentration in the water column and a constant sediment settling velocity is applied.

If the flocculation settling model is “concentration” or “concentration&salinity” the sediment settling velocity is influenced the water column parameters.

If the hindered settling model is “RZ” the sediment settling velocity is determined according to Richardson and Zaki (1954).

taucd == <deposition critical shear stress (N/m^2)>

TBC

rhos == <sediment density (kg/m^3)>

TBC

9.4 Description of Material Block Commands

Material == <material id #>

This command indicates the beginning of a material block, specifying properties for cells with material id #. Material properties are specified for each bed layer and include:

ks

rhodry

mass

tauce

erosion rate params

Example material block:

```
material == 1
   ks == 0.001
  layer == 1
      rhodry == 450
      mass == TBC
      tauce == 0.2
      erosion rate params == 0.0,1.0
  end layer
end material
```

ks == <Nikuradse roughness length (-)>

TBC

Layer == <layer id #>

This command indicates the beginning of a layer block, specifying properties for the bed layer with Nlayer id #.

Example layer block:

```
layer == 1
  rhodry == 450
  mass == TBC
  tauce == 0.2
  erosion rate params == 0.0,1.0
end layer
```

rhodry == <bed layer dry density (kg/m^3)>

TBC

Mass == <bed layer mass (kg/m^2)>

TBC

tauce == <erosion critical shear stress (N/m^2)>

TBC

Erosion Rate Params == <erosion rate (g/m^2/s), alpha (-)>

TBC

Consolidation Model Params == <TBC>

TBC

End Layer

This command indicates the end of the layer block

End Material

The command indicates the end of the material block

Write Restart

TBC

10 2dm Mesh File Format Reference

This section provides a reference to the various components of the mesh file and provides further insight into how TUFLOW FV uses it.

Unstructured mesh geometries can be created using any suitable mesh generation tool. As a preference, BMT uses the SMS Generic Mesh Module (www.aquaveo.com/sms) for building meshes. As a result the TUFLOW FV mesh file format is the SMS mesh file format.

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 another mesh generator) provides a graphical interface to do this instead. Nevertheless, a modeller may find it necessary at times to interrogate the 2dm file in detail.

The mesh file is an ASCII format that can be viewed and manipulated using text editing software. The mesh has an extension ".2dm". The 2dm file has a series of lines and blocks that define the various properties of the mesh file and associated structures. There are 3 specific line / block types that TUFLOW FV reads and uses:

1 The element (or cell) definitions - lines defining elements begin with the characters "E4Q" or "ET3"

2 The node definitions - lines defining nodes begin with the characters "ND"

3 The nodestring definitions – lines defining nodestrings begin with the characters "NS"

Note that if created using SMS, the 2dm file contains additional blocks which TUFLOW FV ignores that are not described below.

10.1 Element definitions – E4Q and E3T

The 2dm file begins with a header line followed by a list of element definitions. Figure 10-1 shows the first 20 lines of a mesh file (using the UltraEdit text editor).

Mesh elements can be either quadrilateral or triangular.

A line describing a **quadrilateral** element begins with "E4Q" and is followed by a number corresponding to the element (or cell) id. The next 4 numbers are the IDs of the 4 nodes that define the quadrilateral element corners in a counter-clockwise direction.

The final number is the element "material id" that is used to define areas with the same bed roughness. The SMS screen shot in Figure 10-2 shows a quadrilateral element highlighted in red with the panel below describing how the element is defined in the mesh file.

A line in the mesh file that describes a **triangular** element begins with "E3T" and is followed by a number corresponding to the element (or cell) id. The next 3 numbers are the ids of the three nodes that connect to create the triangular element.

The final number is the element "material id" that is used to define areas with the same bed roughness. The SMS screen shot in Figure 10-3 shows a triangle element highlighted in red with the panel below describing how the element is defined in the mesh file.

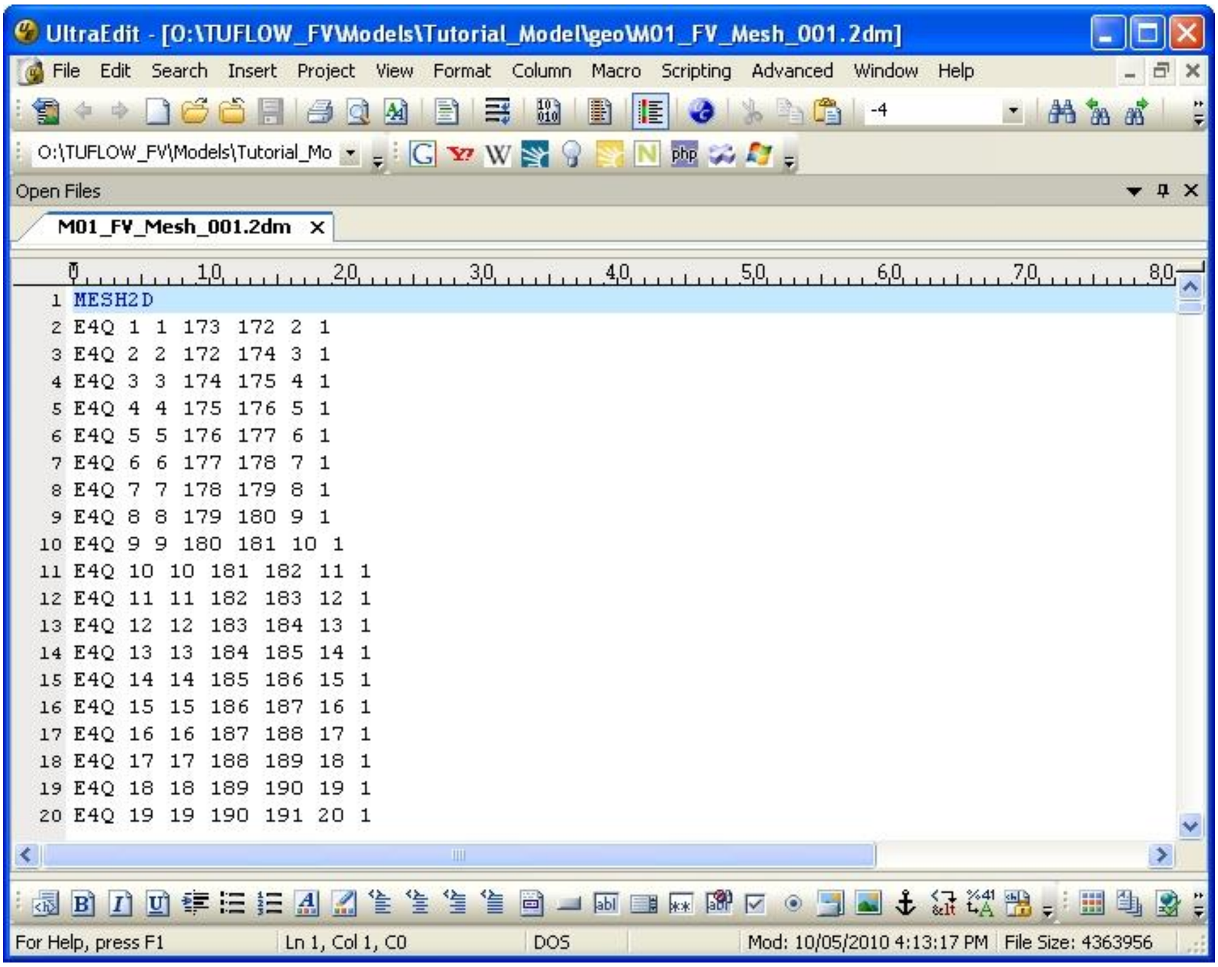
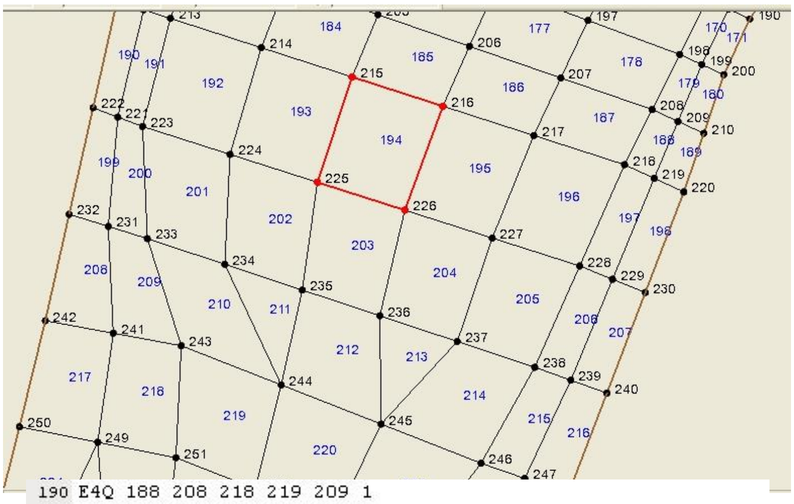
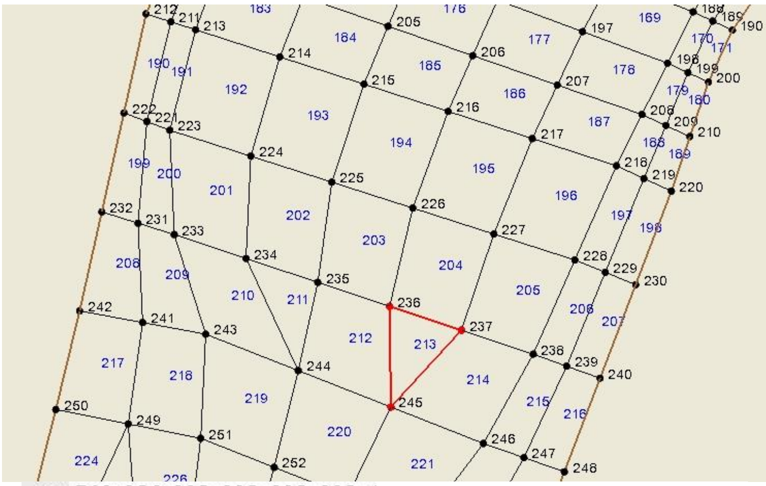


Figure 10-1 TUFLOW FV Mesh File Viewed Using UltraEdit Text Editor



```
190 E4Q 188 208 218 219 209 1
191 E4Q 189 209 219 220 210 2
192 E4Q 190 212 222 221 211 2
193 E4Q 191 211 221 223 213 1
194 E4Q 192 223 224 214 213 1
195 E4Q 193 224 225 215 214 1
196 E4Q 194 225 226 216 215 1
197 E4Q 195 226 227 217 216 1
198 E4Q 196 227 228 218 217 1
199 E4Q 197 218 228 229 219 1
200 E4Q 198 219 229 230 220 2
201 E4Q 199 222 232 231 221 2
202 E4Q 200 221 231 233 223 1
```

Figure 10-2 Example Quadrilateral Element Definition



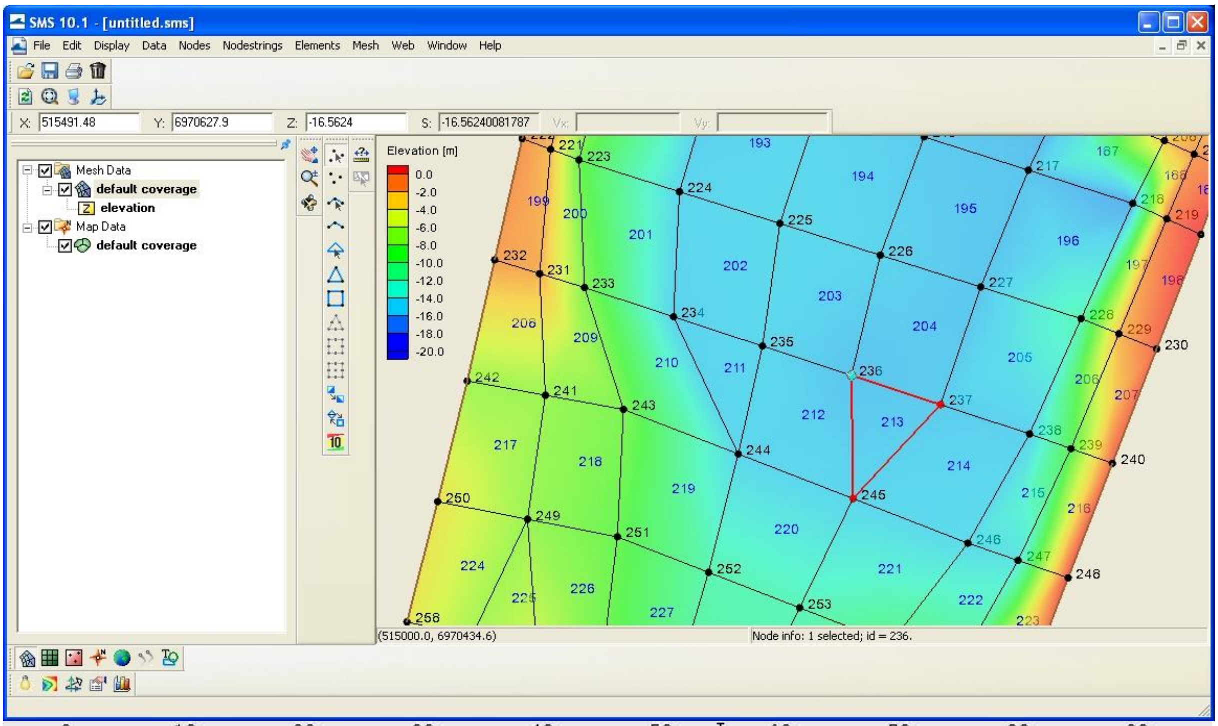
```
208 E4Q 206 228 238 239 229 1
209 E4Q 207 229 239 240 230 2
210 E4Q 208 232 242 241 231 2
211 E4Q 209 231 241 243 233 1
212 E4Q 210 243 244 234 233 1
213 E3T 211 244 235 234 1
214 E4Q 212 244 245 236 235 1
215 E3T 213 245 237 236 1
216 E4Q 214 245 246 238 237 1
217 E4Q 215 238 246 247 239 1
218 E4Q 216 239 247 248 240 2
219 E4Q 217 242 250 249 241 2
220 E4Q 218 241 249 251 243 1
```

Figure 10-3 Example Triangular Element Definition

10.2 Node definitions – ND

A line in the mesh file that defines a node begins with “ND” and is followed by a number corresponding to the node id. The final three numbers in a node line define the X,Y spatial position of the node (in either Cartesian or Spherical coordinates) and the elevation in meters at that location.

The screen shot in Figure 10-4 shows node 236 selected and its corresponding position and elevation displayed in the X,Y,Z dialog boxes. The panel below describes how the node is defined in the mesh file.



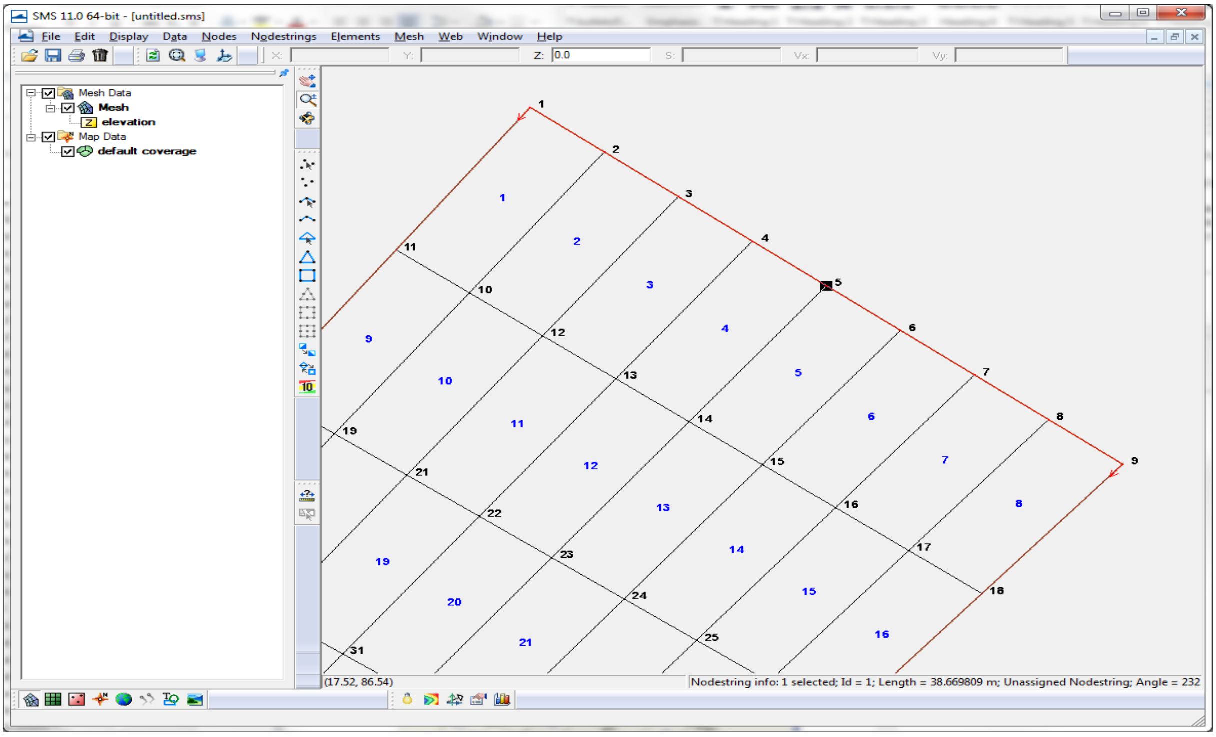

```
ND 230 5.15807750e+005 6.97065546e+006 -5.27855000e-001
ND 231 5.15168130e+005 6.97073383e+006 -2.68195000e+000
ND 232 5.15121600e+005 6.97074826e+006 -2.17900000e+000
ND 233 5.15214660e+005 6.97071940e+006 -8.64582000e+000
ND 234 5.15306940e+005 6.97068890e+006 -1.61260000e+001
ND 235 5.15399210e+005 6.97065840e+006 -1.59007000e+001
ND 236 5.15491480e+005 6.97062790e+006 -1.65624000e+001
ND 237 5.15583760e+005 6.97059739e+006 -1.63117000e+001
ND 238 5.15676030e+005 6.97056689e+006 -1.61460000e+001
ND 239 5.15718970e+005 6.97055155e+006 -1.01942000e+001
ND 240 5.15761900e+005 6.97053622e+006 -3.72059000e-001
ND 241 5.15174110e+005 6.97060721e+006 -7.99626000e+000
ND 242 5.15093150e+005 6.97062203e+006 -7.75618000e+000
```

Figure 10-4 Example Node Definition

10.3 Nodestring definitions – NS

TUFLOW FV uses nodestrings to define open boundary locations. A line in the mesh file that defines a nodestring begins with “NS” and is followed by a list of node ids that connect to create the nodestring. Figure 10-5 shows a nodestring across an open boundary of a mesh. The panel below describes how the nodestring is defined in the mesh file. Note that the final node in the nodestring is indicated by a "-" symbol (i.e. node 9 in Figure 10-5), followed by the nodstring ID (i.e. nodestring 1 is highlighted in Figure 10-5).

TUFLOW FV reads nodestrings according to their ID.




```
ND 75 -3.11900000e+001 4.75250000e+000 0.00000000e+000
ND 76 -2.76850000e+001 2.67875000e+000 0.00000000e+000
ND 77 -2.41800000e+001 6.05000000e-001 0.00000000e+000
ND 78 -2.06750000e+001 -1.46875000e+000 0.00000000e+000
ND 79 -1.71700000e+001 -3.54250000e+000 0.00000000e+000
ND 80 -1.36650000e+001 -5.61625000e+000 0.00000000e+000
ND 81 -1.01600000e+001 -7.69000000e+000 0.00000000e+000
NS  1 2 3 4 5 6 7 8 -9 1
NS  74 73 75 76 77 78 79 80 -81 3
BEGPARAMDEF
GM  "Mesh"
ST  1
```

Figure 10-5 Example Nodestring Definition

11 References

11.1 References in document

11.2 Additional references to TUFLOW FV in literature

12 Index