XP 2D

FOR INTEGRATED 1D/2D MODELING

USER MANUAL

Contents

1 Introduction

xp2D is a computer program for simulating depth-averaged, two and one-dimensional free-surface flows such as occurs from floods and tides. xp2D is based on the computational engine TUFLOW which was originally developed for modelling two-dimensional (2D) flows, and stands for Two-dimensional Unsteady FLOW. xp2D has been dynamically linked (fully integrated) with the xpswmm and xpstorm 1D solution engine.

The xp2D solution algorithm is based on Stelling 1984, and is documented in Syme 1991. It solves the full two-dimensional, depth averaged, momentum and continuity equations for free-surface flow. The scheme includes the viscosity or sub-grid-scale turbulence term that other mainstream softwares omit. The initial development was carried out as a joint research and development project between WBM Oceanics Australia and The University of Queensland in 1990. The project successfully developed a 2D/1D dynamically linked modelling system (Syme 1991). Latter improvements from 1998 to today focus on hydraulic structures, flood modelling, advanced 2D/1D linking and using GIS for data management (Syme 2001a, Syme 2001b). xp2D has also been the subject of extensive testing and validation by WBM Pty Ltd and others (Barton 2001, Huxley, 2004).

xp2D is specifically orientated towards establishing flow patterns in coastal waters, estuaries, rivers, floodplains and urban areas where the flow patterns are essentially 2D in nature and cannot or would be awkward to represent using a 1D network model.

A powerful feature of xp2D is its ability to dynamically link to the 1D network hydrodynamic solutions of XP 1D. The user sets up a model as a combination of 1D network domains linked to 2D domains, ie. the 2D and 1D domains are linked to form one overall model.

xp2D solves the depth averaged 2D shallow water equations (SWE). The SWE are the equations of fluid motion used for modelling long waves such as floods, ocean tides and storm surges. They are derived using the hypotheses of vertically uniform horizontal velocity and negligible vertical acceleration (ie. a hydrostatic pressure distribution). These assumptions are valid where the wave length is much greater than the depth of water. In the case of the ocean tide the SWE are applicable everywhere.

The 2-D SWE in the horizontal plane are described by the following partial differential equations of mass continuity and momentum conservation in the X and Y directions for an in-plan cartesian coordinate frame of reference. The equations are:

$$\frac{\partial\zeta}{\partial t}+\frac{\partial(Hu)}{\partial x}+\frac{\partial(Hv)}{\partial y}=0$$

(2D Continuity)

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}-c_f v+g\frac{\partial\zeta}{\partial x}+g\,u\left(\frac{n^2}{H^{4/3}}+\frac{f_l}{2g\partial x}\right)\sqrt{u^2+v^2}-\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\frac{1}{\rho}\frac{\partial p}{\partial x}=F_x$$

(X Momentum)

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+c_f u+g\frac{\partial\zeta}{\partial y}+g\,v\left(\frac{n^2}{H^{4/3}}+\frac{f_l}{2g\partial y}\right)\sqrt{u^2+v^2}-\mu\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)+\frac{1}{\rho}\frac{\partial p}{\partial y}=F_y$$

(Y Momentum)

where

$\zeta = \text{Water surface elevation}$

$u \text{ and } v = \text{Depth averaged velocity components in } X \text{ and } Y \text{ directions}$

$H = \text{Depth of water}$

$t = \text{Time}$

$x \text{ and } y = \text{Distance in } X \text{ and } Y \text{ directions}$

$c_f = \text{Coriolis force coefficient}$

$n = \text{Manning's } n$

$f_l = \text{Form (Energy) Loss coefficient}$

$\mu = \text{Horizontal diffusion of momentum coefficient}$

$p = \text{Atmospheric pressure}$

$\rho = \text{Density of water}$

$F_x \text{ and } F_y = \text{Sum of components of external forces (eg. wind) in } X \text{ and } Y \text{ directions}$

The terms of the SWE can be attributed to different physical phenomena. These are propagation of the wave due to gravitational forces, the transport of momentum by advection, the horizontal diffusion of momentum or sub-grid scale turbulence and external forces such as bed friction, rotation of the earth, wind, wave radiation stresses, and barometric pressure.

The computational procedure used is an alternating direction implicit (ADI) finite difference method based on the work of Stelling, 1984. The method involves two stages each having two steps, giving four steps overall. Each step involves solving a tri-diagonal matrix.

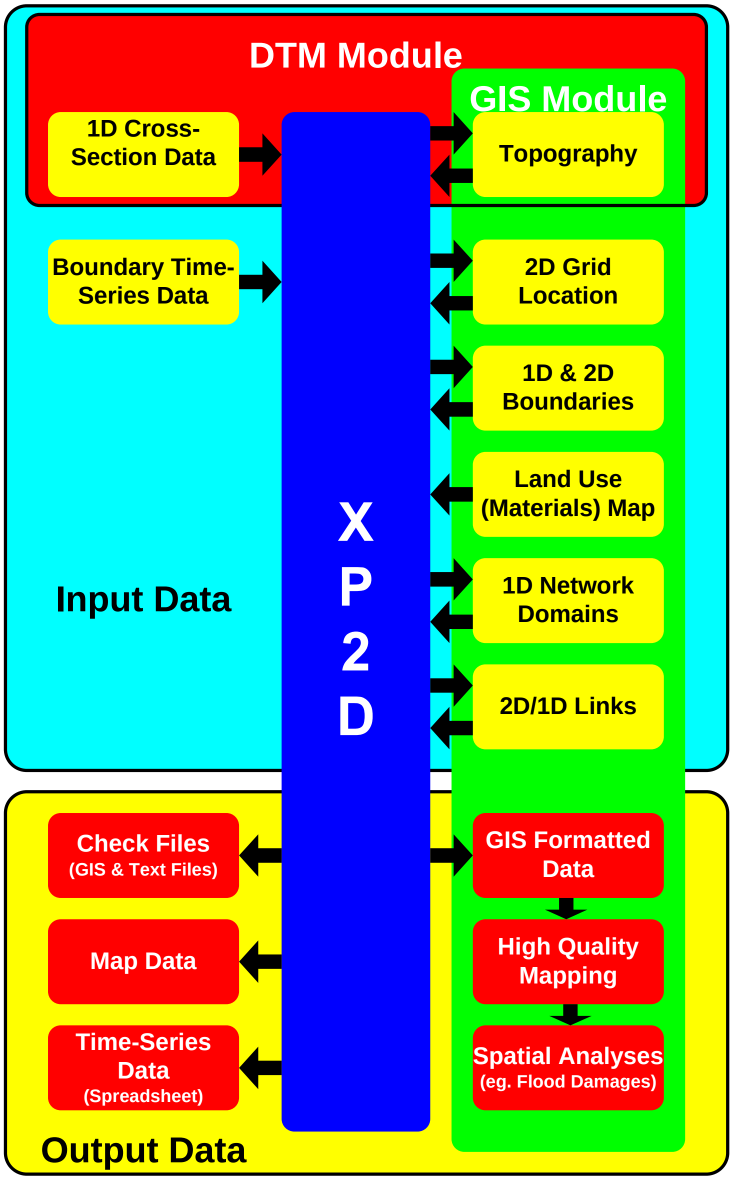
Stage 1, step 1 solves the momentum equation in the Y-direction for the Y-velocities. The equation is solved using a predictor/corrector method, which involves two sweeps. For the first sweep, the calculation proceeds column by column in the Y-direction. If the signs of all velocities in the X-direction are the same the second sweep is not necessary, otherwise the calculation is repeated sweeping in the opposite direction.

The second step of Stage 1 solves for the water levels and X-direction velocities by solving the equations of mass continuity and of momentum in the X-direction. A tri-diagonal equation is obtained by substituting the momentum equation into the mass equation and eliminating the X-velocity. The water levels are calculated and back substituted into the momentum equation to calculate the X-velocities. This process is repeated for a recommended two iterations. Testing on a number of models showed there to be little benefit in using more than two iterations.

Stage 2 proceeds in a similar manner to Stage 1 with the first step using the X-direction momentum equation and the second step using the mass equation and the Y-direction momentum equation.

The solution as formulated by Stelling has been enhanced and improved to provide much more robust wetting and drying of elements, upstream controlled flow regimes (eg. supercritical flow and upstream controlled weir flow), modifications to cells to model structure obverts and additional energy losses due to fine-scale features.

2 Software Structure



3 File Types and Naming Conventions

The XP interface creates all the required files for running xp2D. Hence the user not to be concerned about these files and formats. Files created by XP interface are generally classified as:

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

Control files are used for directing inputs to the simulation and setting parameters. The style of input is very simple, free form commands, similar to writing down a series of instructions. Data input files are primarily GIS layers and comma-delimited files generated using spreadsheet software. Models may still use the original fixed field data input formats if desired.

Data output files are primarily map output in *.sup formats, GIS layers, text files and comma-delimited files. In addition to the above, an extensive range of check files are produced in GIS, text and comma-delimited formats to carry out quality control checks.

The most common file types and their extensions are:

File	Extension	Description	Format
Control Files			
xp2D Simulation Control File	.tcf	Controls the data input and output for a 2D or a 2D/1D simulation. The filename (without extension) is used for naming all 2D domain files. Mandatory.	Text
xp2D Boundary Conditions Control File	.tbc	Controls the 2D boundary condition data input. Is mandatory for a 2D or 2D/1D simulation.	Text
xp2D Geometry Control File	.tgc	Controls the 2D geometric or topographic data input. Is mandatory for a 2D or 2D/1D simulation.	Text
Read Files	.trd .erd .rdf	A file that is included inside another file using the Read File command in .tcf, .tgc files. Minimises repetitive specification of commands common to a group of files.	
Data Input			
Comma Delimited Files	.csv	These files are used for boundary condition databases, boundary condition tables, 1D cross-sections, 1D storage tables, etc. They are opened and saved using spreadsheet software such as Microsoft Excel.	Text
GIS MIF/MID Files	.mif .mid	MapInfo's industry standard GIS data exchange format. The .mif file contains the attribute data definitions and the geographic data of the objects. The .mid file contains the attribute data. Used for the majority of data inputs. The .mid files are of similar format to .csv files, so they can be opened by Excel or other spreadsheet software.	Text

File	Extension	Description	Format
xp2D Materials File	.tmf	Sets the Manning's n values for different bed material categories in 1D and 2D domains.	Text
Data Output			
Super File	.sup	Super file containing the various files and other commands that make up the output from a single simulation. Opening this file in SMS opens the .2dm file and the primary .dat files.	Text
Mesh File	.2dm	2D mesh file containing the 2D/1D model mesh and elevations. It also contains information on materials and 2D grid codes.	Text
Data File	.dat	Generic formatted simulation results file. xp2D output is written using the .dat format.	Binary
Comma Delimited Files	.csv	These files are used for 2D and 1D time-series data output. They are opened and saved using spreadsheet software such as Microsoft Excel.	Text
MIF/MID Files	.mif .mid	Used for GIS based output including graphing of 1D and 2D time-series output within a GIS.	Text
xp2D Restart File	.trf	2D domain computational results at an instant in time for restarting simulations.	Binary
Check Files (see Section 0)			
xp2D Log File	.tlf	A log file containing information about the 2D/1D data input process and a log of the 2D simulation.	Text
Comma Delimited Files	.csv	These files are used for outputting processed 1D and 2D domain time-series boundaries and other data for checking. They are opened and saved using spreadsheet software such as Microsoft Excel.	Text
MIF/MID Files	.mif .mid	A range of 1D and 2D domain check files are produced for checking processed input data within a GIS.	Text

4 GIS Input File Types and Naming Conventions

Please note that XP interface will create all these files when you click on the solve button:

GIS Data Type	File Prefix	Description
2D Domain GIS Layers		
2D Boundaries and 2D/1D Links	2d_bc_ 2d_hx_ 2d_sx_	Mandatory layer(s) defining the locations of 2D boundaries and 2D/1D dynamic links. For large models it may be wise to separate the boundary conditions from the 1D/2D links, in which case use the 2d_hx_ and 2d_sx_ prefixes. Cell code values may also be defined in this layer.
2D Cell Codes	2d_code_	Optional GIS layers containing objects, typically polygons, that define the cell codes. **Note:** The preferred approach is to define cell codes using the 2d_bc layer (see Read MI Code with the BC option).
2D Grid	2d_grd_	Optional layers used to define the 2D grid or mesh. Now primarily used as a quality control check file (in earlier versions was a mandatory input). Contains information on the 2D cell: reference, code, material and other information.
2D Initial Water Levels	2d_iwl_	Optional layer(s) defining the spatial variation in 2D domain initial water levels at the start of the model simulation.
2D Grid Location	2d_loc_	GIS layer defining the origin and orientation of the 2D grid. This layer is optional, however, is the preferred method for geographically locating 2D domains.
2D Longitudinal Profile Output Locations	2d_lp_	Optional layer(s) defining the locations longitudinal profile output from the 2D model domain
2D Land-Use (Materials) Categories	2d_mat_	Layers to define or change the land-use (material) types on a cell-by-cell basis.
2D Plot (Time-Series) Output Locations	2d_po_	Optional layer(s) defining the locations and types of time-series output from the 2D domains.

GIS Data Type	File Prefix	Description
Rainfall over Area	2d_rf_	Optional layer(s) defining the polygons of sub-catchment areas for applying rainfall directly onto 2D domains.
2D Source over Area	2d_sa_	Optional layer(s) defining the polygons of sub-catchment areas for applying a source (flow) directly onto 2D domains.
2D Elevations over an area	2d_za_	Optional layer(s) that define areas (polygons) of elevations at a constant height.
Elevation Lines (Breaklines) (Ridges and Gullies)	2d_zln_ 2d_zlr_ 2d_zlg_	Optional 2D or 3D breaklines defining the crest of ridges (eg. levees, embankments) or thalweg of gullies (eg. drains, creeks). Ridges and gullies can not occur in the same layer so 2d_zlr_ is often used for ridges and 2d_zlg_ for gullies.
2D Elevations as points	2d_zpt_	Layer(s) that define the elevations at the 2D cells mid-sides, corners and centres.
2D and 3D Shapes for (Points, Lines, Polygons)	2d_zsh_ 2d_ztin_	Points, lines and polygons defining 2D and 3D shapes for changing the elevations. Lines can be specified with a width (thickness), and polygons are used as the boundaries for creating TINs (triangulations).
Variable (over time) 2D and 3D Shapes	2d_vzsh_	Points, lines and polygons defining the final geometry of a breach or other variation in model topography over time. Also includes points defining trigger locations.

5 Data Output

xp2D produces a range of output as presented below. In addition, several post-processing utilities are used for transferring data to GIS and other software.

Output is structured into two categories:

- Check Files for checking and quality control of models.
- Result Files containing the 1D and 2D results.

Result Files: Result files contain the hydraulic results of the simulation in the 1D and 2D domains:

- Mesh and results files for viewing the 2D and 1D domains and their results.

- .CSV (comma delimited) text output of time series data for direct input into spreadsheet software such Microsoft Excel.
- .mif/.mid files for viewing 2D and 1D domain results in GIS.
- Text files that log the simulation.

Check Files: Check files are produced so that modellers and reviewers can readily check that the constructed model is as intended. Advanced models draw upon a wide variety of data sources. The check files represent the final data set after all data inputs, allowing the model construction to be viewed in its final form. The check files take the following forms:

- .mif/.mid GIS formats for viewing graphically any errors, warnings and checks, the 1D network, 2D grid, 2D topography, 2D/1D boundaries and connections, and other formats;
- Text files for checking parameter and tabular inputs.

6 Limitations and Recommendations

xp2D is designed to model free-surface flow in coastal waters, estuaries, rivers, creeks, floodplains and urban drainage systems. Flow regimes through structures are handled by adaptation of the 1D St Venant Equations and the 2D Shallow Water Equations using standard structure equations. Supercritical flow areas can be represented (see note below).

Limitations and recommendations to note are:

1 In areas of super-critical flow through the 2D and 1D domains, the results should be treated with caution, particularly if they are in key areas of interest. Hydraulic jumps and surcharging against obstructions may occur in reality – these highly 3D localised effects are not modelled by software such as xp2D.

2 The Smagorinsky viscosity formulation is preferred over the constant viscosity formulation to model sub-cell turbulence (Barton 2001). It is always good practice to carry out sensitivity tests to ascertain the importance of the viscosity coefficient(s) and formulation.

3 Caution should be used when using very small 2D cell sizes, particularly when the flow depth is significantly larger than the cell width (Barton 2001). Modelling on a very fine grid (<1m), especially in deep water, may start to violate the assumptions of 2D solutions based on the Shallow Water Equations. CFD (Computational Fluid Dynamics) codes that model the turbulence and other terms much more accurately may be needed in these situations. The influence of the viscosity (sub-grid scale turbulence) term can be particularly relevant in these situations.

4 There is no momentum transfer between 1D and 2D connections when using the sink/source connection approach (SX link). The HX link does preserve momentum in the sense that the velocity field is assumed to be undisturbed across the link, but the velocity direction is not influenced by the direction of the linked 1D channel. In most situations these assumptions are not of significant concern, however they may influence results where a large structure (relative to the 2D cell size) is modelled as a 1D element.

7 Is a 2D or 2D/1D Model Feasible?

With present day computers, there are few computer hardware constraints in setting up 1D models. However, for 2D models the first step is to decide whether it is feasible and practical to set up a model, given the limitations of your computer hardware. Experienced modellers can usually quickly determine an answer by considering the following:

1 Clearly understanding/defining the model's objectives, and if known, the modelling budget.

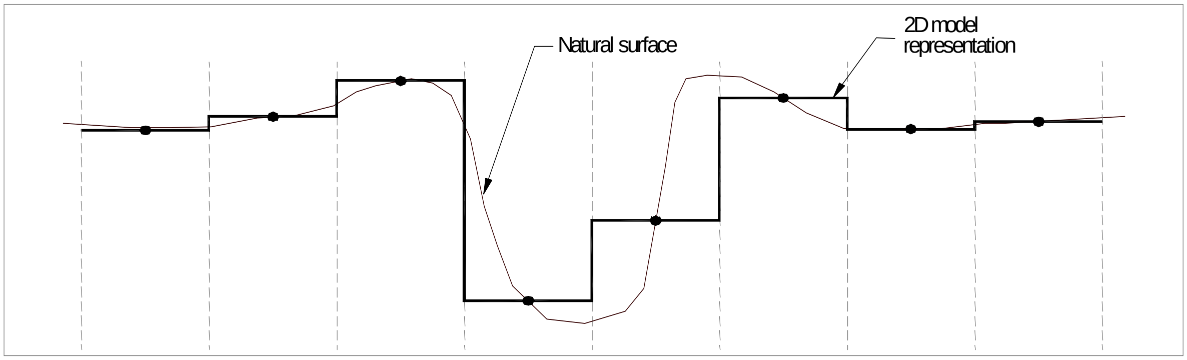
2 Determining the minimum cell size required to model the hydraulics accurately enough to meet the study's objectives. Preferably at least three to four cells across the major flowpaths (depending on the topography). Minor flowpaths may be more coarsely or not represented if they play no significant role hydraulically in regard to meeting the modelling objectives. For example, residual water drains over a floodplain may not affect peak flood levels; in which case, it may not be necessary to model them.

3 If it is not possible to model a major flowpath with a sufficient cell resolution, the flowpath can be modelled as a 1D branch cut through the 2D domain. This may allow a larger cell size to be used, and a greater area modelled in 2D, or a faster simulation time. For example, the river may be modelled in 1D and the floodplain in 2D.

4 Establish possible boundary locations for the model. These are influenced by locations that are well defined hydraulically, and any constraints on the extent of the topographic data (DTM). Dynamically linking with a 1D domain offers significant flexibility in locating the 2D domain.

5 Determine the number of rows and columns of the grid based on the overall dimensions of the 2D domain and the minimum cell size. Calculate the number of cells (rows by columns), and estimate the average number of cells that would be wet.

6 In 2001, using a P3 1GHz computer, overnight simulations of models varying in cell size from 5m to 60m for durations of 12 to 120 hours were achieved with several hundred thousand wet cells. In 2007, these models are running around 10 times faster, and models in excess of a million wet cells have been created! For large models, it may be beneficial to start with a coarser cell size to facilitate quick turnover of simulations before proceeding to a finer cell size. This is a relatively easy process as most input data is not cell size dependent. **Note that halving the 2D cell size typically corresponds to increasing the simulation time by a factor of eight (four times as many cells and half the timestep).**

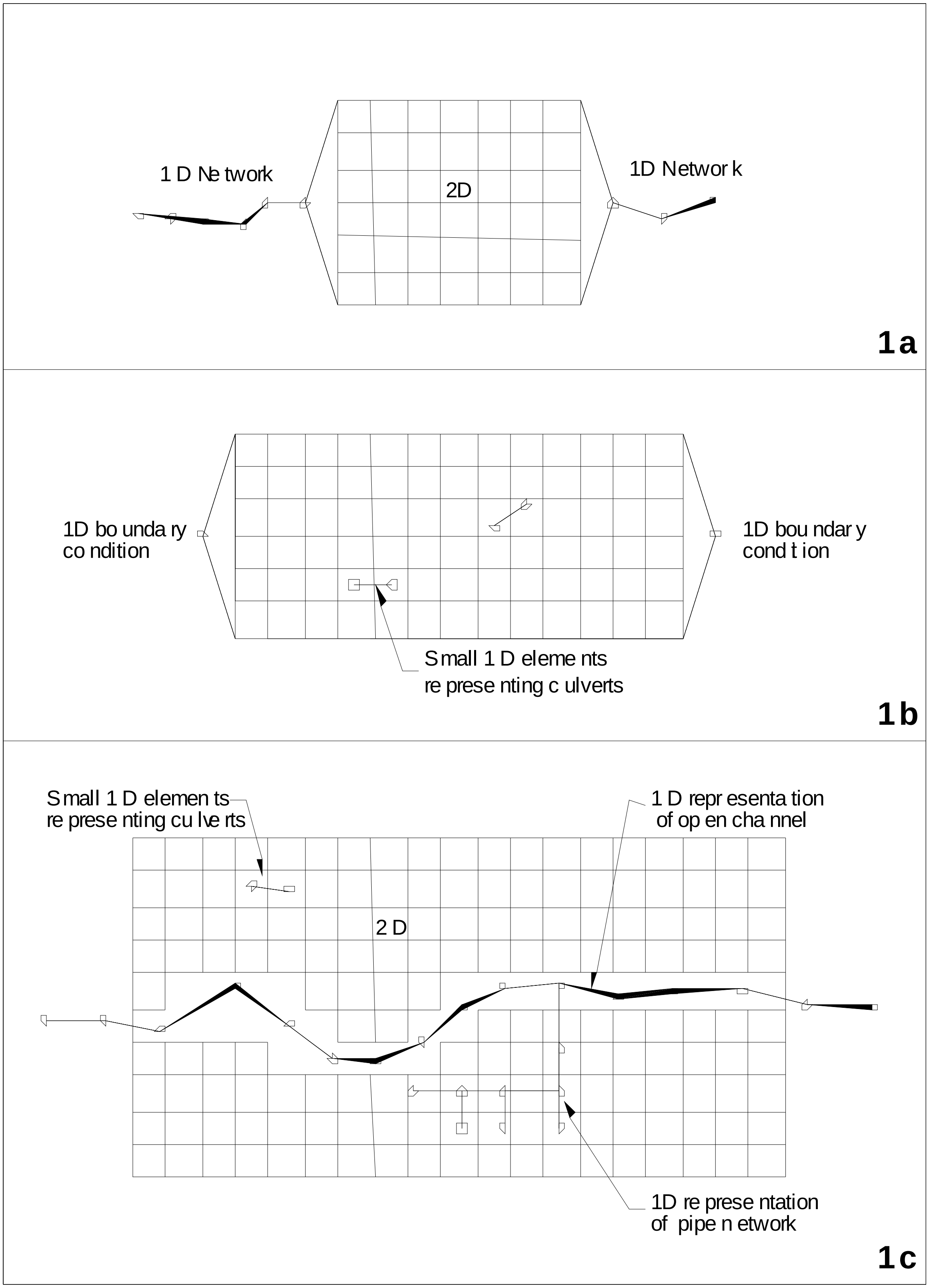


8 Linking 1D and 2D Domains

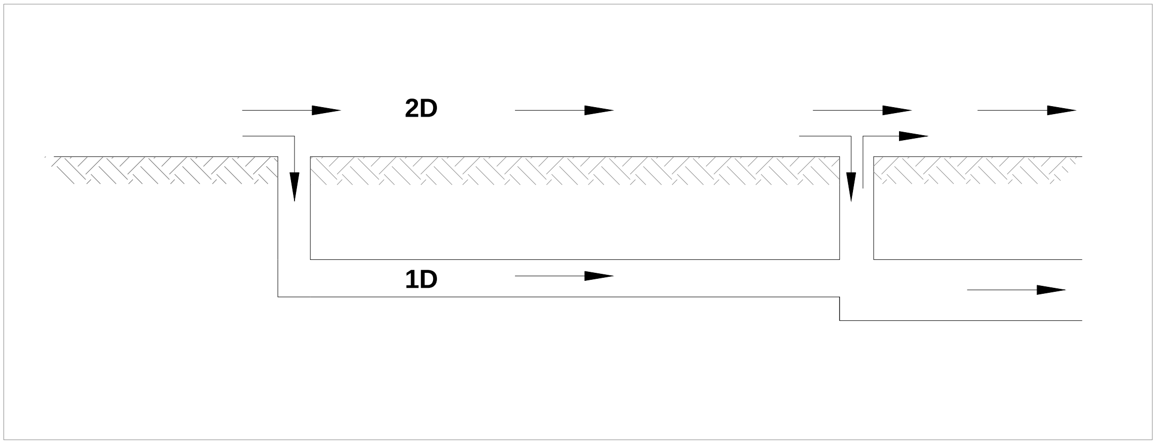
1D and 2D domains can be linked in a variety of ways. The simplest approach is to replace part of a 1D model by nesting a 2D domain inside the broader scale 1D model as shown in Sketch. This approach was developed by Syme (1991) and has been widely applied through various versions.

Further refinements to xp2D were incorporated:

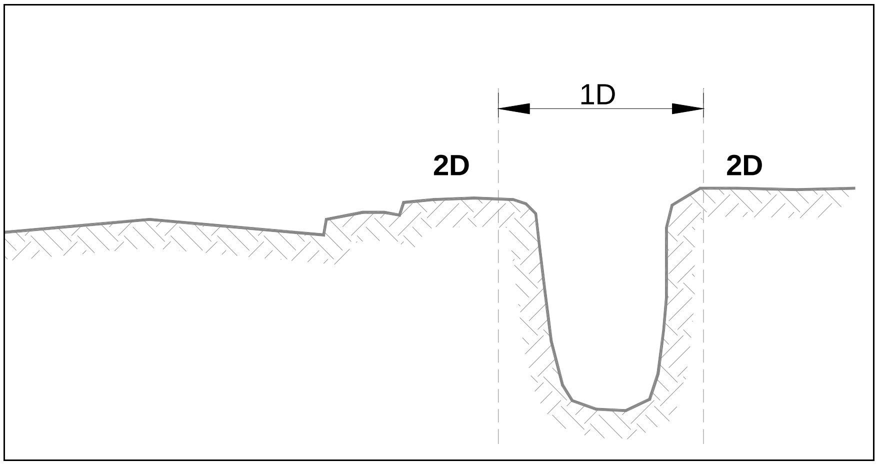
- Insert 1D networks “underneath” a 2D domain or through, for example, an embankment
- Replace or “carve” a 1D channel through a 2D domain.


1D Network
2D
1D Network
1a
1D boundary condition
1D boundary condition
Small 1D elements representing culverts
1b
Small 1D elements representing culverts
1D representation of open channel
2D
1D representation of pipe network
1c

1D/2D Linking Mechanisms



Modeling a Pipe System in 1D underneath a 2D Domain



Modeling a Channel in 1D and the Floodplain in 2D

9 Data Requirements

The minimum data requirements for setting up a 2D/1D hydraulic model are:

1 A DTM with sufficient resolution and accuracy to depict the topography of all flowpaths and storage areas in the 2D domain(s). The vertical accuracy depends on the modelling objectives and budget constraints, however, for large scale models ±0.2m is preferred, whilst for fine-scale urban models <±0.1m is recommended. The vertical accuracy is dependent on the typical depths of inundation in key areas.

2 Cross-sections for any 1D flowpaths.

3 If bed resistance varies over the model, geo-corrected aerial photography or other GIS layer from which material (land-use) zones are digitised for setting Manning's n values.

4 Boundary conditions (eg. ocean water levels, catchment inflows, rainfall, evaporation, etc).

5 Calibration data locations as points in a GIS layer. Peak levels should be attached as attributes to the calibration points.

6 Surveys of key hydraulic controls such as levees / embankments (3D breaklines), culverts, bridges, etc.

10 Calibration and Sensitivity

Models are usually calibrated against known flood or tidal conditions with the bed resistance coefficient (eg. Manning's n) adjusted until calculated water levels and flows are consistent with recorded field measurements. Where there is poor or insufficient topographic data the calibration procedure may also involve adjustments to the model topography to provide an adequate representation of the recorded flow behaviour. This is more common in 1D domains (where there is a choice of cross-sections to define a flowpath). There is usually little opportunity to adjust topography (from that surveyed) in 2D domains.

Ideally, the model would be calibrated for conditions similar to those under investigation although this is not always possible, particularly when major floods are being considered. In these situations, a sensitivity analyses maybe carried out by increasing and decreasing calibration factors such as Manning's n.

11 Model Resolution

2D Cell Size

The cell sizes of 2D domains need to be sufficiently small to reproduce the hydraulic behaviour. Refer to Section Is 1D/2D model possible.

1D Network Definition

The adequacy of the 1D domain is primarily dependent on the network representation adopted. In general, the finer the resolution the more accurate the model, but the longer the computing time. For stability reasons, the timestep for computation is normally controlled by the minimum channel length. The end result may require a compromise between the level of detail and the computational effort.

The first step in setting up a model is to define the flow patterns and to use each identified flow path as the basis for a channel of the network. Following this step the flow paths are linked at junctions, or nodes, and each node is considered as a storage element, which accepts the flow from the adjoining channels. In this way, the model is built up as a series of interconnected channels and nodes with the channels representing the flow resistance characteristics.

For compatibility with the mathematical assumptions, the channels would ideally have more or less uniform cross-sections with constant bottom slope and a minimum of longitudinal curvature. In practice this requirement cannot always be met, particularly where a fine resolution of detail is not required in a portion of the study area. In this case, a flow path is represented by an "equivalent" channel. Experience has indicated that in most cases an adequate calibration can be achieved by deriving a single channel equivalent to a number of series or parallel channels using the steady state Manning's relation for deriving the equivalent channel characteristics.

All nodes and channels are labelled with an ID. No two nodes or two channels can have the same ID. A node and a channel can have the same ID.

Computational Timestep

The selection of the timestep is critically important for the success of a model. The run time is directly proportional to the number of timesteps required to calculate model behaviour for the required time period, while the computations may become unstable and meaningless if the timestep is greater than a limiting value. This is known as the Courant stability criterion.

For the 2D scheme, the Courant Number generally needs to be less than 10 and is typically around 5 for most real-world applications (Syme 1991). The computation timestep in the .tcf file should be set in accordance with this criterion as given in the equation below.

$$C_r = \frac{\Delta t \sqrt{2gH}}{\Delta x} \qquad \text{2-D Square Grid}$$

$$\begin{aligned}
&\textit{where}\\
&\Delta t = \textit{timestep}, s\\
&\Delta x = \textit{length of model element}, m\\
&g = \textit{acceleration due to gravity}, ms^{-\frac{1}{2}}\\
&H = \textit{depth of water}, m
\end{aligned}$$

As a rule, the timestep is typically half the cell size. For steep models with high Froude numbers and supercritical flow, smaller timesteps may be required. It is strongly advised to not simply reduce the timestep if the model is unstable, but rather to establish why it is unstable and, in most instances, correct or adjust the model topography, initial conditions or boundary conditions to remove the instability.

If the model is operating at high Courant numbers (>10), sensitivity testing with smaller timesteps to demonstrate no measurable change in results should be carried out.

The occurrence of high mass errors is also an indicator of using too high a timestep.

Eddy Viscosity

Two options exist for specifying eddy viscosity for the 2D domains to approximate the effect of small-scale motions that cannot be modelled directly.

The first method (Viscosity Formulation = Constant) is to supply a constant value, which is used throughout the model. This is generally satisfactory when the cell size is much greater than the depth or when other terms are dominant (eg. high bed resistance). The recommended coefficient for the constant formulation is 1 m^2/s.

The second method (Viscosity Formulation = Smagorinsky) applies the Smagorinsky formulation as given by the equation below. This formulation is preferred over the Constant option.

Note, if the formulation is changed, the user must also reset the coefficient.

The default setting is to have a Smagorinsky coefficient of 0.2 and a constant component coefficient of 0.1. This seems to give improved flow patterns around bluff objects.

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

Smagorinsky Formulation used by xp2D

$$\begin{aligned}
&where\\
&u\ and\ v = Depth\ averaged\ velocity\ components\ in\ X\ and\ Y\ directions\\
&x\ and\ \ y = Distance\ in\ X\ and\ Y\ directions\\
&\mu = Horizontal\ diffusion\ of\ \ momentum\ coefficient\\
&A_c\ = Area\ of\ \ Cell\\
&C_s\ = Smagorinsky\ Coefficient\ (default = 0.2)
\end{aligned}$$

12 Control Files – Rules and Notation

Control files, such as the .tcf, .tbc and .tgc files, are command or keyword driven text files. The commands are entered free form, based on the rules described below. Comments may be entered at any line or after a command.

An example of a command is:

Start Time == 10. ! Simulation starts at 10:00am on 2/9/1962

Which sets the simulation start time to 10 hours. The text to the right of the "!" is treated as a comment and not used by xp2D when interpreting the line.

The style of input is totally flexible bar a few rules. Commands are not case sensitive and can be repeated as often as needed. This offers significant flexibility and effectiveness when modelling, particularly in building 1D and 2D model topography. Note that a repeat occurrence of a command may overwrite the effect of previous occurrences of the same command.

The rules are:

- A few characters are reserved for special purposes as described

- Only one command can occur on a single line.
- A few commands rely on another command being previously specified. These are documented where appropriate.

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

Reserved Characters – Text Files

Additional text can be placed before and/or after a command. For example, a line containing the command Start Time to set the start time of a simulation to 10 hours can be written as "Start Time == 10" or "Start Time (h) == 10". The "(h)" text is not a requirement, but is useful to indicate that the units are hours. Alternatively, "Start Time == 10 ! hours" would be acceptable, noting the use of the comment delimiter "!".

<table>
<tr><th>Documentation Notation</th><th>Description</th></tr>
<tr><td>< … ></td><td>Greater than and less than symbols are used to indicate a variable parameter. For example, the commonly used <file> example is described below.</td></tr>
<tr><td><file></td><td>Is a filename (can include an absolute or relative path, or a URL). Examples are:
boundaries.tbc (must be located in same folder as .tcf file)
..\model\boundaries.tbc (this is a relative path – the ".." indicates to move up a level)
L:\jb99\ xp2D \model\boundaries.tbc
(this is an absolute path)</td></tr>
<tr><td>[{Op1} | Op2]</td><td>The square brackets "[" and "]" surround parameter options.
The "|" symbol separates the options.
The "{" and "}" brackets indicate the default option. This option is applied if the command is not used.
For example, the options for the Store Maximums and Minimums command are:
[ON | ON MAXIMUMS ONLY | {OFF}]
where the default is OFF.</td></tr>
<tr><td>spaces</td><td>Spaces can occur in commands and parameter options. If a space occurs in a command, it is only one (1) space, not two or more spaces in succession.
Spaces can occur in file and path names.</td></tr>
</table>

Notation Used in Command Documentation –Text Files

13 Simulation Control Files

xp2D Control File (.tcf File)

The xp2D Control File or .tcf file sets simulation parameters and directs input from other data sources. It is the top of the tree, with all input files accessed via the .tcf file or files referred to from the .tcf file. An example of a simple .tcf file is shown further below.

The final .tcf file must reference:

- one .tgc file using Geometry Control File for each 2D domain;
- one .tbc file using BC Control File for each 2D domain;
- a .tmf file using Read Materials File if material (land-use) polygons are being used.

Other mandatory or most commonly used commands are: BC Database; End Time; Map Output Data Types; Map Output Interval; MI Projection; Output Folder; Start Time; Store Maximums and Minimums; Time Series Output Interval; Timestep; Write Check Files; Write Empty MI Files;

The Read File command is extremely useful for placing commands that remain unchanged or are common for a group of simulations in another file (eg. the MI Projection command will be the same for all runs within the same study area). This reduces the size/clutter of .tcf files and allows easy global changes to a group of simulations to be made.

Other commonly used or useful commands are: BC Event Name; BC Event Text; Cell Wet/Dry Depth; Cell Side Wet/Dry Depth; Instability Water Level; Read MI FC; Read MI IWL; Read MI PO; Screen/Log Display Interval; Set IWL; Start Map Output; Start Time Series Output; Viscosity Coefficient; Viscosity Formulation; Write PO Online.

```
# This is an example of a simple .tcf file
! Comments are shown after a "!" or "#" character.
! Blank lines are ignored.  Commands are not case sensitive.

! Set the geographic projection
MI Projection == ..\model\mi\Projection.mif

BC Control File == ..\model\boundaries.tbc  ! boundary control file
Geometry Control File == ..\model\topography.tgc   ! topography control file

Start Time (h) == 0.
End Time (h) == 12.
Timestep (s) == 5

Read Materials File == ..\model\n_values.tmf ! .tmf is for Materials File
```

14 GIS Layers

GIS data layers are transferred into and out of xp2D using the MapInfo data exchange MIF/MID format. This format is documented and in text (ASCII) form, making it easy to transfer GIS data. It is also available for import and export from most mainstream CAD/GIS platforms.

All GIS layers imported or exported by xp2D must be in the **same geographic projection.** To ensure this occurs use the MI Projection and Write Empty MI Files commands.

xp2D interprets MIF/MID GIS data and the data objects (points, lines, etc) as described in the following sections.

To appreciate how xp2D interprets MIF/MID data it is important to understand the following.

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

15 "MI" Commands

Commands containing "MI" (eg. Read MI Zpts) read and/or write **both** .mif and .mid files. The geographical location of objects in the GIS layer **is important** as this controls which part of the model they affect.

When specifying the .mif/.mid file, the extension may be omitted, or either of the .mif or .mid extensions may be used. For example, all of the lines below would be interpreted in the same way:

```
Read MI Code == ..\model\mi\2d_code_buildings.mif
Read MI Code == ..\model\mi\2d_code_buildings.mid
Read MI Code == ..\model\mi\2d_code_buildings
```

When digitising objects, it is preferable that they do not snap to the 2D cell sides or corners as this may produce indeterminate effects.

16 "MID" Commands

Commands containing "MID" (eg. Read MID Zpts) only read the .mid file. The .mif file is not used. These commands rely on the first two columns of attribute data to define the cell reference (ie. n,m or row,column). Data in subsequent columns depends on the data type. It is not necessary for the user to create these layers manually, as xp2D produces them, or than can be easily created using the GIS check files that xp2D writes.

Moving an object in a layer that is read by a "MID" command should never occur and has misleading effects.

In earlier xp2D versions, only the MID option was available, however, the MID option is now normally only used for Zpts (see Read MID Zpts).

The filename must specify the .mid extension, for example:

```
Read MID Zpts == ..\model\mi\2d_zpt_DTM.mid
```

Object Type	**xp2D Interpretation**
Used Objects	
Point	Refers to the cell that the point falls within. Points snapped to the sides or corners of a 2D cell may give uncertain outcomes as to which cell the point refers to.
Line (straight line)	Affects a continuous line of 2D cells. The algorithm for selecting the cells has varied with different builds. See Line Cell Selection and Boundary Cell Selection.
Pline (line with one or more segments)	As for Line above.

Region (polygon)	• Either effects any 2D cell or cell mid-side/corner (eg. Zpt) that falls within the region. If the command is modifying a whole 2D cell, it uses the cell's centre to determine whether the cell falls inside or outside of the region. If the cell's centre, mid-side or corner lies exactly on the region perimeter, uncertain outcomes may occur. Holes within a region are accepted except for polygon objects in shape layers used for TIN boundaries. • Or, just the centroid is used. Examples are the original flow constriction layers (2d_fc) and time-series output locations (2d_po).
Multiple (Combined) Objects	In later versions of xp2D, multiple point, polyline and region objects are generally accepted (ERROR or WARNING messages are given if not the case).
Unused (Ignored) Objects	
Arc	Ignored (do not use).
Collections	Not supported. Collections are groups of objects of differing type.
Ellipse	Ignored (do not use).
none	These objects are ignored and most commonly occur when a line of attribute data is added that is not associated with an object. In MapInfo, this occurs when a line of data is added to a Browser Window.
Roundrect (Rounded Rectangle)	Ignored (do not use).
Rect (Rectangle)	Ignored (do not use).
Text	Ignored.

17 2D Domains (.tgc File)

2D domains are created by building them through a series of commands contained in .tgc files. The .tgc file contains, or accesses from other files, information on the size and orientation of the grid, grid cell codes, bed/ground elevations, bed material type or flow resistance value, and optional data such as ripple height, wave climate, wind field, etc.

A 2D domain is automatically discretised as a grid of square cells. Each cell is given characteristics relating to the topography such as ground/bathymetry elevation, bed resistance value and initial water level, etc.

Only one .tgc file per 2D domain is specified in the .tcf file using Geometry Control File**.**

The .tgc (Geometry Control) File

Rather than contain all the 2D grid information in one file, the .tgc file is a series of commands that **builds** the model. The commands are applied in sequential order, therefore, it is possible to override previous information with new data to modify the model in selected areas. This is very useful where a base data set exists, over which areas need to be modified to represent other scenarios such as a proposed development. This eliminates or minimises data duplication.

The commands can occur in any order (as long as it is a logical one!).

If an unrecognisable command occurs, xp2D stops and displays the unrecognisable text.

Notes & Tips:

7 Any command can be repeated any number of times.

8 Commands are executed in the order they occur. If the data for a 2D cell or Zpt is supplied more than once, the last data read is that used, ie. the latter data for a cell overrides any previous data for that cell.

9 The .mid file is a comma delimited text file. It can be created not only by exporting a MapInfo table but also by using Excel, a text editor or a purpose written translator.

10 The .mid file accessed by a Read MI or Read MID command does not have to contain data for the entire model. If you wish to modify just a few cell values or Zpts, the file only needs to contain these cells/Zpts.

11 Use Write Check Files commands to cross-check and carry out quality control checks on the final 2D grid and Zpts.

An example of a .tgc file is shown below.

```
# Setup the location of the 2d domain
Read MI Location == ..\model\gis\2d_loc_my_model.mif  ! Locate 2D domain
Cell Size == 10.  ! Set cell size to 10m

# Setup the base topography
Set Code == 1  ! Set everywhere as water
Read MI Code BC == ..\model\gis\2d_bc_my_model.mif   ! Read codes from 2d_bc

# Setup the base topography
Set Zpts == 100.  ! Set elevations everywhere to 100 m
Read MID Zpts == ..\model\gis\2d_zpt_dtm.mid  ! Read the DTM Zpts
Read MI Z Lines == ..\model\gis\2d_zlr_levees.mif  ! Apply 3D lines along levees
Read MI Zpts == ..\model\gis\2d_zpt_fill.mif  ! Proposed filling of floodplain

# Setup the materials
Set Mat == 1  ! Set default material value to 1
Read MI Mat == ..\model\gis\2d_mat_land use.mif   ! Read materials distribution
```

2D Grid Orientation and Dimensions

Each 2D domain is a rectangle at any orientation. The orientation and dimensions are defined using .tgc file commands. For the orientation it is recommended that the X-axis falls between 90° and –90° of East as it is preferable to view the 2D grid within this range and some post-processing software only operate within this range.

Several options are available for setting the grid location and orientation as a result of a number of new commands being introduced over the years. In all cases, Cell Size must be specified. The options are:

- Using a four-sided polygon in a GIS layer to define the 2D grid orientation and dimensions (see Read MI Location).
- Using a line (two vertices only) in a GIS layer to define the orientation of the X-axis (see Read MI Location), and Grid Size (N,M) or Grid Size (X,Y) to set the 2D grid X and Y dimensions.
- Using Origin, Orientation or Orientation Angle, and Grid Size (N,M) or Grid Size (X,Y). No GIS layers are required for this option.

It is not essential at any point to specify dimensions that are an exact multiple of Cell Size.

2D Cell Codes

Each cell in a 2D domain is assigned a code to indicate its role. It must have a value of one of the types in below table, the default code value is one (1) or “water”.

Commands used to modify the cell codes are Set Code, Read MI Code (or Read MI Code BC), Read MID Code in the .tgc file, and Read MI BC in the .tbc file automatically sets the Boundary Cell code of 2 along external boundaries.

Type	Code	Description
Null Cells	-1	Inactive cells used to deactivate cells within the active domain. Null cells are often preferred to land cells as they are not excluded when xp2D outputs in SMS format. For two simulations to be compared in SMS, they must have exactly the same mesh. If an area in a model is removed (eg. filling part of a floodplain), use null cells or raise the ground elevations in preference to using land cells so that the two simulations can be compared. **Note:** In earlier versions of xp2D null cells were used to indicate the outside side of an external boundary – this is no longer the case. Cells on the outside of a boundary can be either a land or a null cell.
Land or Redundant Cells	0	Land cells are cells that are totally removed from the computation. The name "land" comes from coastal hydraulic studies where the land was the permanently dry area. Maximising the area of land cells reduces computation time and output file sizes.
Water or Active Cells	1	Water cells are active cells that can wet and dry.
Boundary Cells	2	Boundary cells indicate water cells that have an external boundary (including some types of 2D/1D dynamic links). At an external boundary there must be a water cell on one side and a null or land cell on the other. **Note:** It is not necessary to manually specify each boundary cell. Boundary lines are digitised in the GIS and xp2D automatically assigns the boundary code to the cells (see Section **Error! Reference source not found.** and Read MI BC).

ZC, ZU, ZV and ZH Points (Zpts)

The model topography is defined by elevations at the cell centres, mid-sides and corners. Each cell has the following elevations assigned to it as shown in below figure

- "C" Zpt (ZC) – middle of cell
- "U" Zpt (ZU) – middle right of cell
- "V" Zpt (ZV) – middle top of cell
- "H" Zpt (ZH) – top right hand corner of cell

One of most important aspects of xp2D modelling is to understand the roles of the elevation (Zpt) points.

The ZC point:

- defines the volume of active water (cell volume is based on a flat square cell that wets and dries at a height of ZC plus the Cell Wet/Dry Depth);
- controls when a cell becomes wet and dry (note that cell sides can also wet and dry); and

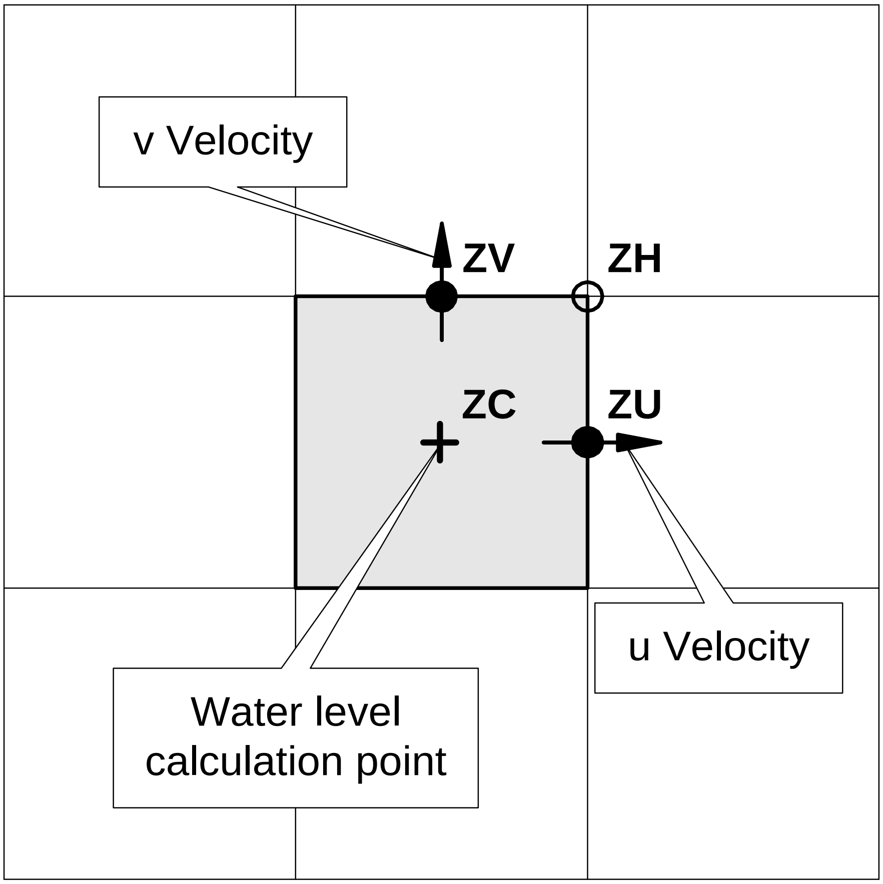
- is used to determine the bed slope when testing for the upstream controlled flow regime.

The ZU and ZV points:

- control how water is conveyed from one cell to another;
- are where the momentum equation terms are centred and where upstream controlled flow regimes are applied;
- are deactivated if the cell has dried (based on the ZC point) and cannot flow; and
- can wet and dry independently of the cell wetting or drying (see Cell Wet/Dry Depth and Cell Side Wet/Dry Depth) which allows the modelling of "thin" obstructions such as fences and thin embankments relative to the cell size (eg. a concrete levee).

ZH points:

- play no role hydraulically;
- are, by default, the only elevations to be written to the .2dm mesh file (by default, all output is interpolated/extrapolated to the cell corners.



Location of Zpts and Computation Points

Building the Topography Layer by Layer

A powerful feature of xp2D is its capacity to build the 2D elevations from any number of GIS layers and/or TINs. The general approach is as follows:

1 A default elevation is specified first in the .tgc file using Set Zpt. A flood-free elevation is usually specified.

2 The elevations sampled from a DEM using Read MID Zpts is next. These elevations will override those previously specified wherever they exist.

3 If there are areas where there are missing or erroneous data, these areas are easily interpolated across using Read MI Z Shape regions or polygons. Simply digitise a polygon around the missing or erroneous data and xp2D interpolates elevations across the region based on the existing Zpts around the perimeter of the polygon. (Note: missing or erroneous data often occurs with aerial surveys where the sampling in areas of thick vegetation, water or the removal of elevated objects (eg. buildings) gives a poor representation of the topography.)

4 Sometimes there is a need to remove cut and fill works from the topography. For example, if modelling an historical event to calibrate the model, some of the levees, road embankments and developments may not have existed at the time of the event, and need to be removed. This is easily done using Read MI Z Shape by simply digitising polygons around the various works.

5 The base elevations set up in the previous step(s) can be modified to represent hydraulic controls, proposed works, failure of a flood defence wall, etc. Some examples are:

(a) The crests of road/rail embankments, levees, fences and other solid obstructions are easily inserted using Read MI Z Line, Read MI Z Shape or Read MI Z HX Line. Read MI Z Shape is particularly powerful as 3D lines can be given a thickness making it very easy to quickly raise, or lower, elevations along a road alignment or a diversion channel where the width of the embankment/channel is wider than the 2D cell size.

(b) The proposed cut and fill for a development or other works can be incorporated using Read MI Zpts, Read MI Z Shape, Read TIN Zpts or Create TIN Zpts. These powerful commands can set elevations based on regions. Within regions TINs can be generated from points and lines, and the perimeter of the TIN can be automatically merged with the existing Zpts. A TIN of the cut and fill produced by SMS or 12D can be read directly if so desired using Read MI Zpts.

6 If there is a need to simulate the failure of a flood defence wall or road/rail embankment, or the collapse of fences, Read MI Variable Z Shape can be used to control the collapse of the embankment or fences over time. The collapse can be triggered to occur at a specified time or when a water level, or a water level difference between two locations, elsewhere in the model is reached.

A 2D domain's Zpts are built up using one or more of the commands shown in the below table:

Command	Description
Set Zpt	Sets all Zpts over the whole 2D domain to the same value. Useful for providing an initial elevation prior to other commands as some Zpts in inactive (land) parts of the model may not receive a value. The default value for all Zpts is 99999.0; every Zpt must be assigned a value, essentially making this command mandatory.
Read MID Zpts	Normally used to set the Zpts generated from a DTM. Use the Write MI Zpts command for xp2D to create a GIS layer of Zpts with no elevations (xp2D only writes out Zpts at active (non-land) cells). This layer is then imported to a GIS, each Zpt assigned an elevation from a DTM and then exported for use by the Read MID Zpts command. Note the use of **MID** in the command, which indicates that only the .mid file is read (ie. do not specify a .mif file when using this command).
Read MI Zpts	Note, this command is different to the Read MID Zpts command above. It is typically used for simple modifications of sections of the topography. Examples are filling an area (defined by a region or polygon object) to the same elevation, or dredging (lowering) a section of river using Read MI Zpts ADD.
Read MI Z Line	Reads 3D breaklines (defined as a polyline with elevation points) to modify the nearest Zpts to the height of the line. A powerful command for ensuring the crest height of ridges (levees, embankments, etc) is correctly modelled. A number of options exist for this command. Also see Allow Dangling Z Lines and Pause When Polyline Does Not Find Zpt.
Read MI Z HX Line	Similar to Read MI Z Line, but uses HX lines and ZP points in a 2d_bc layer to adjust the 2D cell elevations along HX lines.
Create TIN Zpts	Useful for having xp2D create a TIN using polygons for TIN boundaries and points and lines within the polygons to create the TIN. If no points are snapped to the perimeter vertices of a polygon, the elevations around the polygon's perimeter are merged with the current Zpt values. The resulting TIN can optionally be exported to SMS and 12D TIN formats.
Read TIN Zpts	Reading of a TIN in SMS or 12D format to set the Zpt values within the TIN.

Command	Description
Read MI Z Shape	Powerful command to modify Zpt values using points, lines and polygons. Lines can vary in width from just the cell sides (thin), whole cells (thick) or be assigned a width (thickness) in metres. TINs are created within the polygons and incorporate elevations from points and lines that fall within the polygon. The perimeter of the polygon can be merged with the current Zpt values in part, or in its entirety. Read MI Z Shape and Create TIN Zpts are excellent for removing bad data areas and for filling in null areas where the aerial survey has provided poor or no data. Another example is to remove buildings from a DEM.
Read MI Variable Z Shape	Allows the user to define the eroded 3D shape of a section of the 2D domain, specify the period for collapse, and how the collapse is triggered (ie. at a specified time or when a water level is reached or when a water level difference is exceeded). Raising of the Zpts over time is also permitted (eg. to model the influence of a landslide filling a river).
Maximum Points Maximum Vertices	Use these commands to increase the maximum number of elevation points or maximum number of vertices in a single polyline or polygon.
Default Land Z	Now rarely used in lieu of Set Zpt.
Interpolate ZC Interpolate ZHC Interpolate ZUV Interpolate ZUVC Interpolate ZUVH	Allows the interpolation of Zpts from other types of Zpts. Now rarely used as nearly all models assign values directly to all the Zpts. The original xp2D code only required input of ZH points, and Interpolate ZUVC provided a tool for interpolating the other Zpts. Models with "bumpy" terrain, such as that from airborne laser surveys, might benefit from using Interpolate ZHC or Interpolate ZUV. Models through urban areas where the DTM includes the buildings may benefit from using Interpolate ZC ALL LOWER, which reduces the amount of cells that become blocked out due to high ZC elevations from buildings.
ZC == MIN(ZU,ZV)	Rarely used.

2D ZPT Commands

Zpt (2d_zpt) Layers

2d_zpt layers contain points at the ZC, ZU, ZV and ZH locations, with each point being assigned an elevation. The layer is usually initially generated by xp2D using Write MI Zpts, which creates the .mif and .mid files, and elevations are assigned to the fourth (Elevation) attribute using a DEM in 3D surface modelling software (for example, by using the Point Inspection Analysis tool in Vertical Mapper). The 2d_zpt attributes are listed the below table:

Unlike nearly all other layers, xp2D only reads the .mid file of the 2d_zpt layer. The .mid file contains the attribute data in comma delimited format. Read MID Zpts (note the MID) reads the attribute data and locates the elevation within the 2D domain using the n, m and Type attributes. n and m (row and column) defines the 2D cell, and Type defines whether it is the ZC, ZU, ZV or ZH elevation point. The fourth attribute is the elevation value (note, that is does not have to be called Elevation, in fact it is good practice to label the fourth attribute after the DEM for data traceability).

Read MID Zpts can be used as many times as needed. Usually the first time is to read in the DEM elevations. If there are elevations that need to be manually modified, the points can be selected in the GIS, saved as a separate layer (note, only save the points that need to be modified), and the Elevation attribute values edited (do not change any of the first three attribute values). In the .tgc file, a second instance of Read MID Zpts can be used to read in the new elevation values to override the original ones.

No.	GIS Attribute	Description	Type
1	n	2D grid row	Integer
2	m	2D grid column	Integer
3	Type	One of “C”, “U”, “V” or “H” to indicate whether the elevation is at the ZC, ZU, ZV or ZH location	Char(1)
4	Elevation	Elevation of the point.	Float

3D Breakline (2d_zln) Layers

2d_zln layers are used for adjusting the closest Zpt elevations to lines representing the crests or highpoint of levees, flood defences, roads, railways and other embankments that obstruct flow. The lines can be 3D in shape and height using elevation points. For further information see Read MI Z Line.

One of the advantages of continuing to use 2d_zln layers is that they only require one attribute (Z or Elevation) and are therefore simpler to use.

Users should also be aware of the new Read MI Z HX Line for assigning Zpt elevations along HX lines.

3D TIN (2d_ztin) Layers

TINs (triangulations) of elevation points and 3D lines within a polygon can be carried out using Create TIN Zpts. This is particularly useful for modifying the Zpt elevations where there have been, or are proposed, changes to the base DEM Zpt values.

This feature can also be used as an alternative to using Read MID Zpts for the base DEM elevations, although it is noted that if dealing with large data sets, it is likely to be much more efficient to use 3D surface modelling software to triangulate the data, and Read MID Zpts to read the data into xp2D.

The protocols applied are:

1 A TIN is created for each polygon in the 2d_ztin layer.

2 Any points found within a polygon are used when generating the TIN.

3 Any lines are converted to points, and those points falling within the polygon are used for the TIN creation. Lines are converted to points as follows:

(a) All vertices (nodes) of the line are converted to points.

(b) The dMax attribute is used to insert additional vertices between the line's vertices. For example, if dMax is set to 10, then additional intermediate vertices are inserted at least every 10 metres (or less) between the existing vertices where the distance between the existing vertices exceeds 10m. If the dMax attribute does not exist or is zero, half the 2D domain's Cell Size is used as the dMax value.

(c) If there are any points snapped to the line's vertices, the elevations of these points are used to set the elevations at all the vertices generated along the line. In this way, a 3D breakline effect can be produced within the TIN. If there are no points snapped to the line, the line's elevation is used giving the effect of a horizontal line.

4 The perimeter of the polygon/TIN can either be merged with the current Zpt values or have its own values as follows:

(a) If there are no points snapped to the perimeter of the polygon, the elevations of the polygon's perimeter vertices, and of any automatically inserted vertices, are based on the current Zpt values (ie. the Zpt values assigned by any prior commands).

(b) If there are one or more points snapped to the polygon's perimeter vertices, the perimeter is not merged with the Zpt values, and the elevations of the snapped points are used to assign elevations to the perimeter vertices and any automatically inserted vertices.

(c) The frequency of any automatically inserted points around the perimeter is controlled by the dMax attribute. If the dMax attribute does not exist or is zero, half the 2D domain's Cell Size is used.

No.	GIS Attribute	Description	Type
1	Z	Point: Elevation of the point. Line: Elevation of the line. Ignored if there are any points snapped to the line's vertices. Polygon: Not used.	Float

No.	GIS Attribute	Description	Type
2	dMax (optional)	Point: Not used. Line: Maximum distance between automatically added intermediate vertices. If set to zero or this attribute does not exist, half the 2D domain's Cell Size is used. If less than zero no intermediate vertices are inserted. Polygon: Same as for Line above.	Float

Z Shape (2d_zsh) Layers

Read MI Z Shape offers a wide range of options for manipulating and modifying the Zpt values. These include 3D breaklines and TINs.

When a mixture of different shapes and shape options occur within the same layer the following protocols are used to control how Zpt values are modified.

5 The order in which objects are processed is:

(a) **Polygons:** All polygons are triangulated according to the process described for Create TIN Zpts in Section 0. The only difference to note is that if a line is to be used for the TIN generation, the Shape_Option attribute for the line must include the keyword "TIN".

(b) **Wide Lines:** Wide lines are lines that have a Shape_Width_or_dMax attribute value greater than or equal to 1.5 times the 2D Cell Size. A buffer polygon is created along the line, and all Zpts falling within the buffer polygon are assigned elevations based on a perpendicular intersection with the line. Note that wide lines are processed in the order that they occur in the GIS layer, so if the buffer polygons of two wide lines overlap each other, the latter one prevails. In this situation it would be wise to separate the two lines into two different layers. Buffer polygons can be viewed in the 2d_sh_obj_check.mif layer.

(c) **Thin and Thick Lines:** Thin lines have a Shape_Width_or_dMax value of zero and Thick lines a value less than 1.5 times the 2D Cell Size. For a more detailed description of Thin and Thick lines see Read MI Z Line. Thin and Thick lines are applied depending on the Shape_Option attribute setting as follows:

(i) All ADD lines are applied first.

(ii) Then lines without any option (these will modify all Zpts affected by the line).

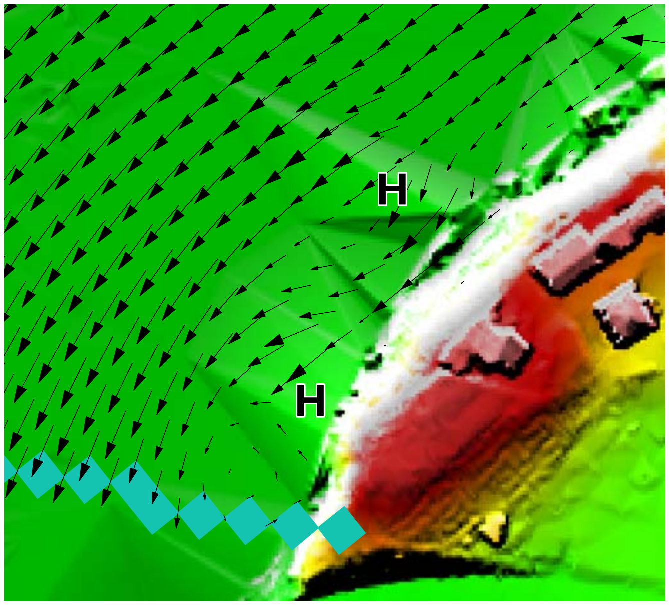
(iii) Followed by GULLY, LOWER or MIN lines.

(iv) And finally any RIDGE, RAISE or MAX lines.

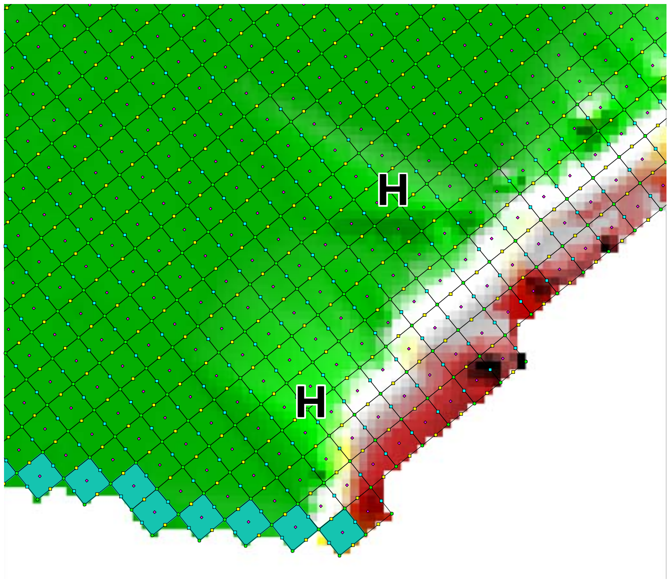
6 The priority can of course be further controlled by using different layers and controlling when those layers are processed in the .tgc file.

7 Note that Read MI Z Shape uses a new GULLY line approach (for thin and thick lines) compared with the Read MI Z Line approach. The new approach assigns Zpt values that are interpolated using the perpendicular or nearest point along the breakline.

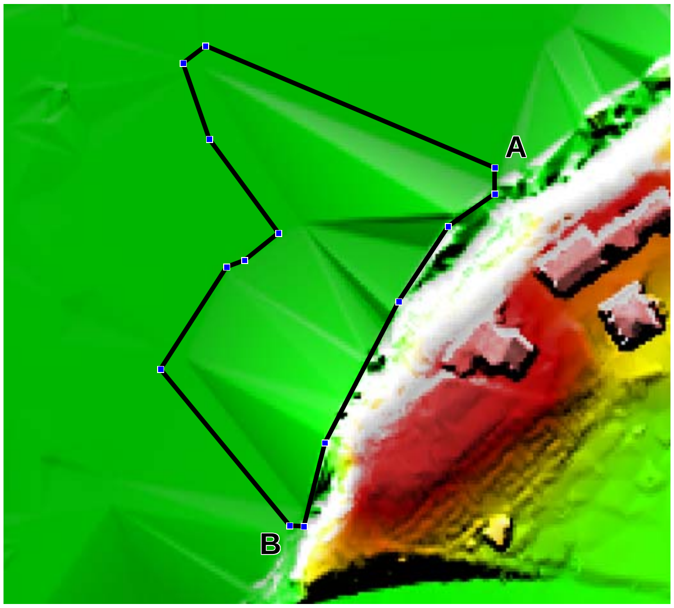
An example of using Read MI Z Shape to simply modify Zpt elevations is given below:



The image shows a couple of false ridges indicated by the H letters that were caused by a poor triangulation of the TIN software used to create the DEM. These ridges caused unrealistic flow patterns as shown by the velocity vectors. The blue cells at the bottom are the downstream boundary cells.

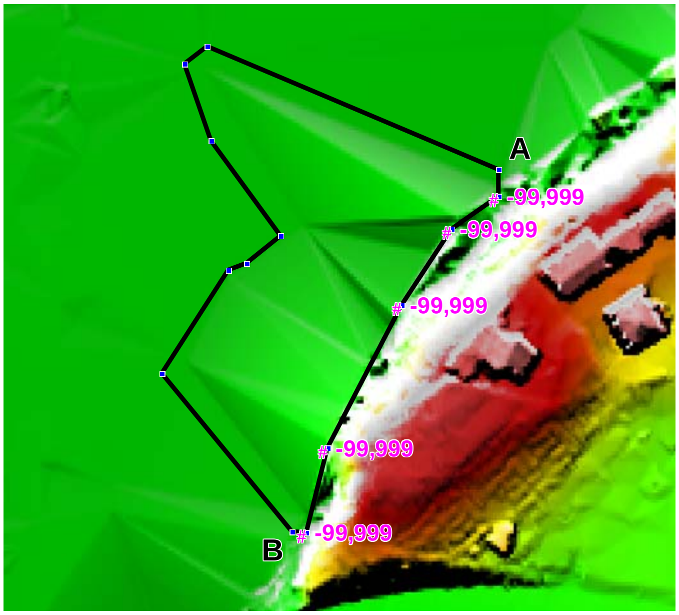


This image is of a DEM produced from the 2d_zpt_check.mif layer (which is shown as the points). This is how xp2D sees the DEM data with the false ridges clearly shown.

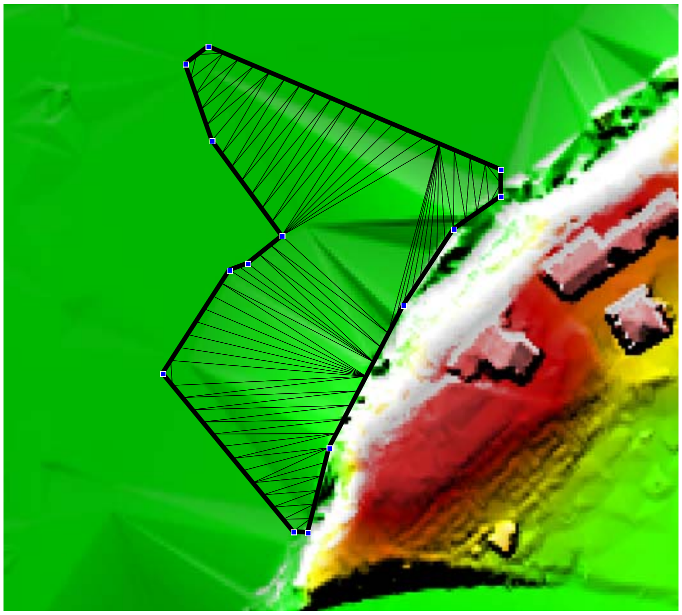


To remove the ridges, import an empty 2d_zsh layer, and digitise a polygon around the ridges as shown. By default (ie. using the default attribute values), the elevations assigned around the perimeter of the polygon are interpolated from the current Zpt values.

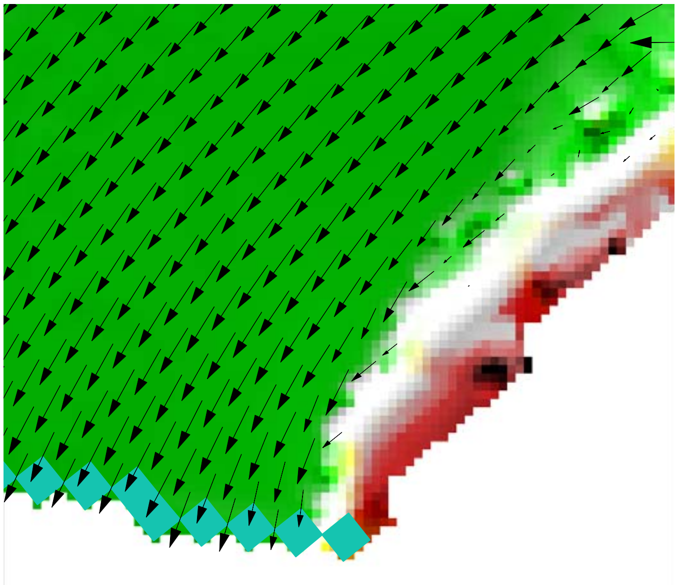
One problem with this approach is that the elevations along the right hand side (ie. along the edge of the floodplain between Locations A and B) are interpolated from the high Zpts along this boundary.



To solve this problem, digitise points either into the 2s_zsh layer or into a 2d_zsh…_pts layer that snap to the vertices of the polygon where the high elevations occur. Assign a Z attribute of -99999 to each point as shown in the image. The -99999 indicates to not interpolate an elevation from the existing Zpts. Instead, the elevations at Locations A and B are used to interpolate elevations at vertices where -99999 has been assigned.



Use Read MI Z Shape to process the polygon and points and generate the TIN as shown in the image. The TIN can be viewed by importing the 2d_sh_obj_check.mif layer.



This image shows a DEM created from the 2d_zpt_check.mif layer with the above 2d_zsh layer applied. As can be seen, the ridges have been removed and the flow patterns are now realistic.

No.	GIS Attribute	Description	Type
1	Z	**Point:** Elevation (or change in elevation for ADD option) of the point. Points are only used to assign elevations along lines and polygon perimeters, or for the creation of TINs within polygons. They do not modify Zpts directly. An elevation of -99999 has a special meaning when the point is snapped to a vertice of a polygon. The -99999 indicates to ignore the elevation at that vertice and of any automatically inserted vertices between that vertice and the two neighbouring vertices. Instead the elevations are based on the elevations of the neighbouring vertices. If a neighbouring vertice also has a -99999 point snapped to it, the next vertice is used, and so on. This feature is very useful, as illustrated in the example above. **Line:** Elevation of the line if no points are snapped to the line, otherwise ignored. **Polygon:** Used to set the elevation of the polygon perimeter only if NO MERGE option is specified and there are no points snapped to the perimeter. Otherwise ignored.	Float
2	dZ	**Point:** Change in elevation (Z) value at the point. To leave Z unchanged, set to zero. Useful for easily adjusting the height of the shape object without having to change the original Z value. **Line:** Change in elevation (Z) value for all vertices/points along the line irrespective of whether there are points snapped to the line or not. To leave elevations unchanged, set to zero. For example, to raise an embankment by 0.2m, set dZ to 0.2. **Polygon:** Adjusts the height of all points inside or snapped to the perimeter by dZ. Only adjusts the elevations of the polygon perimeter if NO MERGE option is specified, otherwise ignored.	Float

No.	GIS Attribute	Description	Type
3	Shape_Width_or_dMax	**Point:** Not used. **Line (no TIN):** Provided "TIN" is not specified for the Shape_Option attribute, this attribute specifies the width (thickness) of the breakline in metres. If equal to zero, only the elevations along the cell sides are adjusted (ie. a thin breakline). If less than or equal to 1.5 times the 2D Cell Size, a line of whole cells is adjusted (ie. a thick line). Otherwise all Zpts within a distance of half Shape_Width_or_dMax are adjusted. **Line (TIN):** If "TIN" is specified for the Shape_Option attribute, this attribute controls the maximum distance between automatically added intermediate vertices. If set to zero half the 2D domain's Cell Size is used. If less than zero no intermediate vertices are inserted. **Polygon:** The maximum distance between intermediate points inserted around the polygon's perimeter to interpolate elevations from the current Zpts. If equal to zero, half the 2D Cell Size is used. If less than zero no intermediate vertices are inserted. Not used if the NO MERGE option is specified.	Float

No.	GIS Attribute	Description	Type
4	Shape_Options	**Point:** Not used. **Line or Polygon:** ADD: Adds the shape elevations to the current Zpts. MAX, RIDGE or RAISE: Only changes a Zpt elevation if the Z Shape elevation at the Zpt is higher. MIN, GULLY or LOWER: Only changes a Zpt elevation if the Z Shape elevation at the Zpt is lower. **Line Only:** TIN: Indicates the line is only to be used for generation of TINS from polygons (therefore, only sections of TIN lines that fall within the polygon(s) are used). **Polygon Only:** If none of the options below are specified, the elevations at perimeter vertices that do not have an elevation point snapped to them are merged with the current Zpt values. MERGE ALL: Ignores elevations from any points snapped to the perimeter and merges all perimeter vertices with the current Zpt values. NO MERGE: Does not merge the perimeter elevations with the current Zpt values.	Char

Variable Z Shape (2d_vzsh) Layers

The Zpts can be varied in elevation over time to simulate, for example, breaching of embankments, or even the filling of a river due to a landslide. The 2d_vzsh layer is used to define the final topographic shape at the end of the breach and/or fill period. The first four attributes are the same as for the 2d_zsh layer, plus there are some additional attributes to define how/when the breach commences and for how long. The breach/fill can be triggered by: a specified time; when the water level reaches a specified height at a specified (trigger) location; or the water level difference between two triggers exceeds a specified amount.

Below table presents the attributes of a 2d_vzsh layer, and an example is discussed and illustrated in the next section

No.	GIS Attribute	Description	Type
1	Z	Same for previous table	Float

No.	GIS Attribute	Description	Type
2	dZ	Same for previous table The exception is for a thin line, for which the final elevations along the line can be set to dZ metres above the current Zpt values. For example, if a thin line represents a fence that is being collapsed, and the finished height of the collapsed fence is to be 0.2m above existing ground levels, specify a dZ value of 0.2. Note this only applies if the NO MERGE Shape_Option has not been specified.	Float
3	Shape_Width_or_dMax		Float
4	Shape_Options	**Point:** TRIGGER: Indicates the point is not an elevation point, but a trigger location. The trigger must be given a name using the Trigger_1 attribute. **Line or Polygon:** Same for previous table. except the ADD option is not supported. **Thin Line:** NO MERGE: For thin lines (Shape_Width_or_dMax = 0), the final elevations along the line are as specified. If NO MERGE is not specified for a thin line, the final elevations are set to be the same as the current Zpt values plus the dZ value.	Char(20)
5	Trigger_1	**Point:** If Shape_Options is set to TRIGGER, enter the name of the trigger location. The name can contain any characters and can include spaces. Otherwise not used. **Line or Polygon:** To commence the failure at a specified time leave blank. To commence failure based on reaching a water level elsewhere in the model, enter the name of the trigger location. **Thin Line:** For thin lines there are two special options as follows. Specify DEPTH to have the failure commence once the depth of water adjacent to the cell side exceeds the amount specified for Trigger_Value. Specify DEPTH DIFF to have the failure commence once the difference in water level across the cell side exceeds the amount specified for Trigger_Value.	Char(20)

No.	GIS Attribute	Description	Type
6	Trigger_2	**Point:** Not used. **Line or Polygon:** The name of a second trigger location (only needed if the breach is to be initiated on a water level difference between two trigger locations).	Char(20)
7	Trigger_Value	**Point:** Not used. **Line or Polygon:** If Trigger_1 is blank, the simulation time in hours that the breach is to commence. If Trigger_1 is specified and Trigger_2 is left blank, the water level in metres at Trigger_1 that needs to be reached to trigger the failure. If both Trigger_1 and Trigger_2 are specified, the water level difference in metres between Trigger_1 and Trigger_2 that needs to be exceeded to trigger the failure. The water level difference is taken as the absolute of the difference between Trigger_1 and Trigger_2, so there is no need to specify a negative value. **Thin Line:** If "DEPTH" is specified for Trigger_1, the depth in metres adjacent to the cell side that needs to be exceeded to trigger the failure at the cell side. If "DEPTH DIFF" is specified for Trigger_1, the water level difference in metres across the cell side that needs to be exceeded to trigger the failure.	Float
8	Period	**Point:** Not used. **Line or Polygon:** Time in hours over which the variation in Zpt elevations occurs.	Float

Using a Separate Points Layer

GIS layers used for Read MI Z Line, Read MI Z HX Line, Create TIN Zpts, Read MI Z Shape, Read MI Variable Z Shape, Read MI FC Shape and Read MI Layered FC Shape can be split into two layers, one containing the elevation points, and the other the lines and polygons. This is useful in terms of managing the data, and especially when interrogating and/or viewing the data in GIS. It is also more compatible with the ArcGIS software, which does not allow layers to contain a mixture of different object types (eg. points and lines can not exist in the same layer).

There is only one attribute for the points layer, ie. Elevation (or Z) as described in below table. This also keeps the data sets more simple and easier to manage.

Using the line below as an example.

Read MI Z Shape == mi\2d_zsh_haul_road.mif

1 The specified layer, 2d_zsh_haul_road.mif, is opened. This layer may or may not contain elevation points. If any elevation points exist they are used.

2 A separate points layer can optionally be used to specify additional points or all of the points. The layer can be specified in one of two ways:

(a) Entering the pathname of the points layer after the main layer. A "|" must be used to separate the two layers. The points layer must be the second layer specified. For example the example above would look like:

Read MI Z Shape == mi\2d_zsh_haul_road.mif | mi\2d_zsh_haul_road_points.mif

(b) Alternatively, name the points layer the same as the main layer, but with a "_pts" extension. If a layer exists with the "_pts" extension, xp2D automatically assumes this layer is associated with the main layer, and includes all points within this layer when applying the above commands. For this example, the layer would be named 2d_zsh_haul_road_pts.mif.

(c) The first approach (a) above prevails over the second (b) if both apply.

(d) If neither (a) or (b) apply, xp2D assumes there is no separate points layer.

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

Building the Bed Resistance (Materials)

The bed resistance values for 2D domains are created by using GIS layers containing polygons of different bed resistance. Bed resistance formulation is set to one of Manning's n, Manning's M (1/n) or Chezy using Bed Resistance Values in the .tcf file. The default is Manning's n. Chezy can be specified as a direct value or by bed ripple heights. Manning's n is the only supported option. Manning's n values can also be varied with depth.

The recommended approach is to use materials to define how the bed roughness over the model varies. Each material represents a different roughness category. GIS layers of land-use or vegetation often make excellent material layers. Examples of different materials are River In-Bank, River Banks, Pasture, Roads, Buildings, Forest, Mangroves, etc. Each material is assigned a constant Manning's n value, or varying Manning's n with depth.

The default is to interrogate the material polygons directly at the 2D cell mid-sides. This produces a higher resolution representation of the bed roughness, and gives improved flow patterns and results especially in urban areas where large and sudden variations in bed roughness occur due to the presence of roads (very slippery) and buildings (major obstruction, ie. very rough).

xp2D also offers two alternatives to direct interrogation of material/roughness values at cell-sides. The cell centre values can still be used with the cell mid-side values being averaged as either the average of the n values or the average of the M values. To use these options see Bed Resistance Cell Sides.

The most common approach is to digitise one or more materials layers (2d_mat) and assign Manning's n values to the materials using Read Materials File and a .tmf file. This approach allows the easy adjustment of Manning's n values during model calibration.

In creating the base 2d_mat layer, it is good practice to not digitise the most common or the most difficult to digitise material, and use the following commands in the .tgc file.

Use Set Mat to set the most common material to all cells in a 2D domain.

Use Read MI Mat to allocate the remaining material values.

The Read MI Mat command may be used as many times thereafter to further modify the materials in parts of a 2D domain.

Note that, the default material value is now zero (previously one). As a material value of zero is not allowed, every cell and cell-side must now be assigned a material value using Set Mat and Read MI Mat in the .tgc file (it is good practice to always set a default materials value using the Set Mat as the first material command in the .tgc file). For backward compatibility, Change Zero Material Values to One can be used to set any zero material values to one.

Material values must be within the range 1 to 32,767 (in previous builds a value up to 99,999 could be used).

If using the Chezy formula, a number of commands have been setup to provide backward compatibility. These are Depth/Ripple Height Factor Limit and Recalculate Chezy Interval.

18 Multiple 2D Domains

Any number of 2D domains of different cell size and orientation can be combined to form one model. The 2D domains can be linked by 1D domains, with 2D/2D linking under development and planned for a future release. For example, a 1D domain of a river system may have several 2D domains embedded to represent several townships where a more detailed analysis is required. The combination of 1D and 2D domains forms one overall model. Direct 2D to 2D linking can be achieved by using the 2d_bc 2D link Type.

To specify more than one 2D domain use Start 2D Domain and End 2D Domain in the .tcf file to start and end blocks of commands applicable for each 2D domain.

The mandatory .tcf commands that occur within a 2D domain block are:
BC Control File
Geometry Control File
Timestep

Optional commands that can be used are:
CellWet/DryDepth

Cell Side Wet/Dry Depth

Set IWL

Read MI FC

Read MI GLO

Read MI IWL

Read MID IWL

Read MI LP

Read MI PO

Instability Water Level

Note that specifying one of the above commands outside a Start/End 2D Domain block does not apply that command to all 2D domains. For example, specifying Cell Wet/Dry Depth outside a block will not set the Cell Wet/Dry Depth value to all 2D domains.

An example of using 2D domain blocks is given below.

```
# This is an example of a simple .tcf file with multiple 2D domains

MI Projection == ..\model\mi\Projection.mif


Start 2D Domain == East Domain

BC Control File == ..\model\east domain.tbc

Geometry Control File == ..\model\east domain.tgc

Timestep (s) == 10

Set IWL == 1

End 2D Domain


Start 2D Domain == West Domain

BC Control File == ..\model\west domain.tbc

Geometry Control File == ..\model\west domain.tgc

Timestep (s) == 5
```

```
Set IWL == 1

End 2D Domain



Start Time (h) == 0.

End Time (h) == 12.



Read Materials File == ..\model\n_values.tmf ! .tmf is for xp2D Materials File
```

19 Time-Series Output Locations

Plot Output (PO, LP) from 2D Domains

Time-series data output from 2D domains is available for a range of hydraulic parameters as listed in below table. Output takes the form of time-series hydrographs (referred to as PO – Plot Output) or longitudinal profiles over time (LP). All data types are available for PO and only H_ (water level) is available for LP.

The locations of PO and LP output must be defined prior to a simulation. This is carried out by creating one or more GIS layers containing points, lines and polylines that define the locations of PO and LP output.

The start time for PO and LP output and the output interval is set separately to map based output using Start Time Series Output and Time Series Output Interval. If no start time or interval is set, output occurs from the beginning of the simulation at every timestep.

The output is in the form a .csv files and also to the _TS.mif file. 2D domain time-series (PO) output is synchronised with 1D domain output by default. This allows both 1D and 2D time-series to be placed in the _TS.mif file. Set Output Times Same as 2D to OFF in the .ecf file if 1D and 2D time-series data is not to be synchronised. In this case, no 2D PO is written to the _TS.mif file.

Flag	Description	Point	Line (or Polyline)
H_	Water Level (Head)	Water level of the h-point of the nearest cell. If the cell is dry, the ground level (ZC) is output.	The average water level of all wet cells along the line. If all cells are dry, the lowest cell's ground level (ZC) is output. If a polyline is used, the average water level along each line segment is output.

Flag	Description	Point	Line (or Polyline)
Q_	Flow or discharge.	N/A (zero flow results).	The flow crossing the line. For a polyline, the sum of the flows crossing each polyline segment. The flow across a line or polyline segment is determined by summing the flow across cell sides whose perpendiculars' intersect the line The sign of the flow across a line or polyline segment is positive if the flow is in the same direction when looking in a direction perpendicular to the line with the start of the line on your left and the end on your right. If digitising a flow line across a 1D channel that is carved through the 2D domain, ensure that the line is digitised so that it crosses the 1D channel where there is a change in colour of the linked 2D HX cells as shown in the 1d_to_2d_check.mif or _TSMB1d2d.mif layers.
QA	Flow Area (m^2).	N/A (zero area results).	The flow area is calculated using the same cell sides as for Q_. An adjustment for obliques lines is made.
QI	Integral Flow (m^3)	N/A (zero integral flow results).	Integrates the flow (as determined for Q_ above) over time (ie. the area under a Q_ time-series curve). If Write PO Online is set to ON, the integral flow is not calculated until the simulation is complete.
QX	Flow in X-direction.	N/A (zero flow results).	The X component of Q_ (ie. the sum of the flows at the u-points).
QY	Flow in Y-direction.	N/A (zero flow results).	The Y component of Q_ (ie. the sum of the flows at the v-points).
V_	Velocity (m/s)	The magnitude of the resolved vector based on the two u-points and two v-points of the cell in which the point falls. Exactly which cell is selected may be uncertain if the point falls exactly on a cell's side.	N/A. Do not use lines or polylines for velocity output. At present uses the cell in which the line or each polyline segment starts. Future release plan to calculate Q_ and QA, and output the velocity as Q_/QA (ie. the depth and width averaged velocity along the line).

Flag	Description	Point	Line (or Polyline)
VA	Velocity Angle (degrees relative to east where east is zero, north is 90, etc.).	The angle of V_.	N/A. See comments above for V_.
Vu	u-point velocity (m/s)	The magnitude of the u-point velocity (ie. across the right hand side of the cell).	N/A. See comments above for V_.
Vv	v-point velocity (m/s)	The magnitude of the v-point velocity (ie. across the top side of the cell).	N/A. See comments above for V_.
VX	Velocity in X-direction (m/s).	The magnitude of the average of the u-point velocities (ie. across the left and right hand sides of the cell).	N/A. See comments above for V_.
VY	Velocity in Y-direction (m/s).	The magnitude of the average of the v-point velocities (ie. across the bottom and top sides of the cell).	N/A. See comments above for V_.
W_	Wind (undocumented and yet to be tested feature on PC version).		

Time-Series (PO) Data Types

20 Initial Water Levels (IWL)

2D Domains

Initial water levels (IWL) are set globally as a constant using the Set IWL (.tcf file) or Set IWL (.tgc file) command. IWLs can also vary spatially using one or more GIS layers. This is particularly useful for setting initial water levels in lakes, dams, etc.

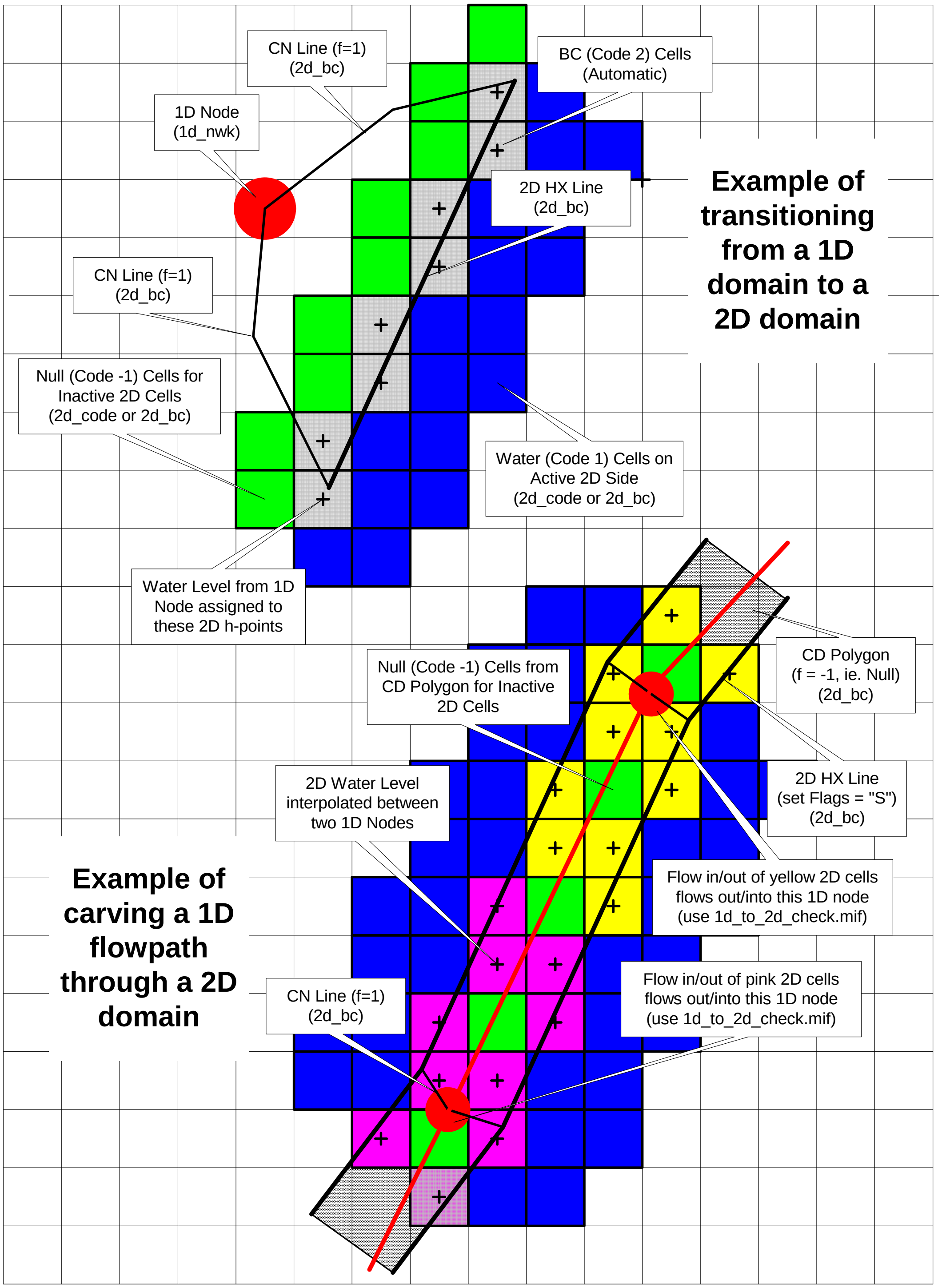
Any number of IWL layers may be used, noting that if a cell's IWL occurs more than once, the last occurrence prevails, ie. xp2D overwrites any previous IWL already set.

The Read MI IWL or Read MID IWL command maybe used in the .tcf and .tgc files, noting that the .tgc file is processed before the .tcf file (ie. any IWL commands in the .tcf file will override those in the .tgc file). Using the IWL command in the .tcf file allows the initial water levels to be set independently of the geometry file. This is useful where several simulations of different events use the same .tgc file but have different initial conditions, thereby removing the need to have separate geometry files for each event. IWL commands are 2D domain dependent.

21 Linking 1D and 2D Domains

To link 1D and 2D domains use the 2D interface and 2D connections lines:

- 2D Interface boundaries are preferred for transitioning between 1D domains and 2D domains, or when carving a 1D network through a 2D domain. They can also be successfully used for connecting to 1D structures embedded into a 2D domain, especially where preservation of momentum from 2D to 1D to 2D is important.


CN Line (f=1)
(2d_bc)
BC (Code 2) Cells
(Automatic)
1D Node
(1d_nwk)
2D HX Line
(2d_bc)
Example of transitioning from a 1D domain to a 2D domain
CN Line (f=1)
(2d_bc)
Null (Code -1) Cells for Inactive 2D Cells
(2d_code or 2d_bc)
Water (Code 1) Cells on Active 2D Side
(2d_code or 2d_bc)
Water Level from 1D Node assigned to these 2D h-points
Null (Code -1) Cells from CD Polygon for Inactive 2D Cells
CD Polygon
(f = -1, ie. Null)
(2d_bc)
2D Water Level interpolated between two 1D Nodes
2D HX Line
(set Flags = "S")
(2d_bc)
Example of carving a 1D flowpath through a 2D domain
Flow in/out of yellow 2D cells flows out/into this 1D node
(use 1d_to_2d_check.mif)
CN Line (f=1)
(2d_bc)
Flow in/out of pink 2D cells flows out/into this 1D node
(use 1d_to_2d_check.mif)

CN Line = 1D/2D Connection Line

HX Line = 1D/2D Interface Line

22 Presenting 1D Domains in 2D Output (1d_wll)

Output from 1D domain(s), can be combined with the 2D map output viewed. This allows easier viewing of results and the ability to animate the 1D results in combination with the 2D results. 1d_wll GIS layer(s) are used to define and control the 1D map output. The layer(s) contain lines (called WLLs or Water Level Lines) that cross 1D channels and/or nodes. A WLL is essentially a line of horizontal water level, and should be digitised on this basis (ie. perpendicular to the flow direction).

The direction of digitising the WLLs is important. They must be from left to right looking in the positive direction of the channel.

WLL method allows elevations and optionally material (Manning's n) values to be assigned to points along a WLL. A more accurate representation of velocity and flood hazard from the 1D domain can be mapped. The velocity at a point on the WLL is estimated by carrying out a parallel channel analysis along the WLL using the flow in the channel the WLL is connected to. The analysis estimates the water surface slope at the WLL based on the conveyance of the profile along the WLL. Implicit in this analysis is that the cross-section selected for the channel produces an average water surface slope representative of that along the length of the channel. The water level at the WLL still remains the linearly interpolated water level between the upstream and downstream nodes.

This can have 2 and 3 vertex WLLs , as well as WLLs with no limit on the number of vertices. For 4 or more vertices, one of the vertices (except for the two end vertices) must snap to a vertex on the channel line.

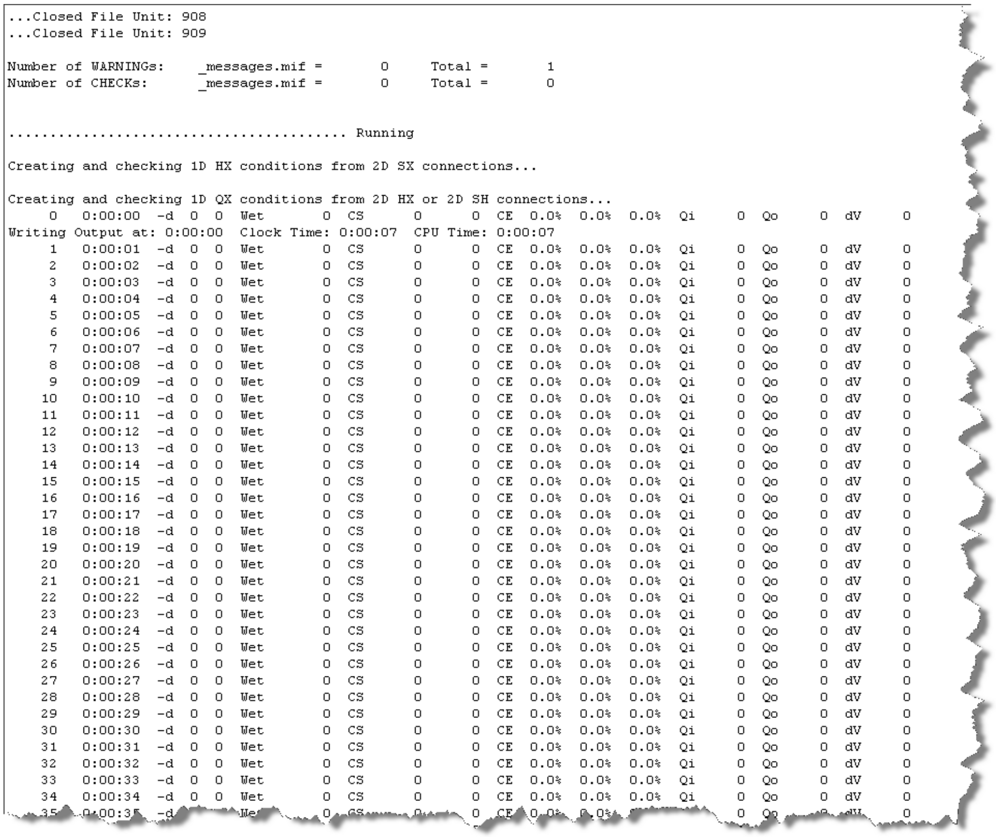
There is one attribute required for this method. The attribute, dMin, is the minimum distance in metres along which to generate elevation points for that line. If dMin is zero, only elevations at the mid and end points on the WLL are generated.

Read MI XP WLL in the **.tcf** file to read the 1d_wll layer(s). The XP_SWMM links will also need to be built into a GIS layer and read using or Read MI XP Network from the **.tcf** file.

GIS Attribute	Description	Type
Read MI WLL Command		
dMin	The minimum distance interval along the WLL to generate elevation and material sampling points (WLL Points). These points form the corners of the triangulation. If set to zero, no additional points are generated (ie. only the existing vertices along the WLL are used).	Float

23 Data Output

xp2D displays a lot of information to the *.tlf file. Once the simulation has started, the simulation status at each timestep is displayed to change the frequency of display). Where different timesteps are used for different domains, the display interval is based on the largest timestep.



```
...Closed File Unit: 908
...Closed File Unit: 909

Number of WARNINGs:     _messages.mif =       0     Total =       1
Number of CHECKs:       _messages.mif =       0     Total =       0


..................................... Running

Creating and checking 1D HX conditions from 2D SX connections...

Creating and checking 1D QX conditions from 2D HX or 2D SH connections...
     0   0:00:00  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
Writing Output at: 0:00:00  Clock Time: 0:00:07  CPU Time: 0:00:07
     1   0:00:01  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
     2   0:00:02  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
     3   0:00:03  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
     4   0:00:04  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
     5   0:00:05  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
     6   0:00:06  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
     7   0:00:07  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
     8   0:00:08  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
     9   0:00:09  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    10   0:00:10  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    11   0:00:11  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    12   0:00:12  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    13   0:00:13  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    14   0:00:14  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    15   0:00:15  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    16   0:00:16  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    17   0:00:17  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    18   0:00:18  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    19   0:00:19  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    20   0:00:20  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    21   0:00:21  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    22   0:00:22  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    23   0:00:23  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    24   0:00:24  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    25   0:00:25  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    26   0:00:26  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    27   0:00:27  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    28   0:00:28  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    29   0:00:29  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    30   0:00:30  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    31   0:00:31  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    32   0:00:32  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    33   0:00:33  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    34   0:00:34  -d  0  0  Wet       0  CS      0       0  CE  0.0%  0.0%  0.0%  Qi     0  Qo     0  dV     0
    35   0:00:35  -d        Wet       0  CS      0       0  CE  0.0%  0.0%              0  Qo     0  dV     0
```

Along each line the following information is shown:

- Number of timesteps completed based on the largest timestep of all 1D and 2D domains.
- Simulation time in hhhh:mm:ss.
- "-d" followed by two numbers:
  - o The maximum number of 1D nodes per timestep that experienced negative depths below -0.1m since the previous display line.
  - o The maximum number of 2D cell sides per timestep that experienced negative depths below -0.1m since the previous display line.

  The locations of these negative depths are output as warnings in the _messages.mif file. Negative depths indicate the model is having difficulty in convergence at that location, which may lead to an instability.
- "Wet" followed by number of wet or active 2D cells.
- If automatic weir switching is active (see Free Overfall) the next information is "CS" (Cell Sides) followed by two numbers as follows:
  - o The number of cell sides where upstream controlled friction flow occurred (see Supercritical).

- The number of cell sides where upstream controlled broad-crested weir flow occurred (see Free Overfall).

- If the free-overfall algorithm is set to ON WITHOUT WEIRS (see Free Overfall), the next information is "FO" followed by the number of cell sides where the free-overfall algorithm is being applied. Note: this option is now rarely used in lieu of the automatic weir and supercritical flow options.

- If Display Water Level was specified, the next piece of information is a "GL" (Gauge Level) followed by the water level at the location indicated. This is useful to monitor the rise and fall of the water level at a key location.

- If Mass Balance Output is set to ON, "CE" (Cumulative Error) followed by three percentages is displayed to show the cumulative mass error as follows:

  - The whole of model % cumulative mass error for all 1D and 2D domains.

  - The % cumulative mass error for all 1D domains.

  - The % cumulative mass error for all 2D domains.

- If Mass Balance Output is set to ON the following are displayed after the "CE" percentages:

  - "Qi" followed by the total flow into the model (all domains) in m^3/s. If the inflow exceeds 999,999m^3/s or falls below -99,999m^3/s, the flow is expressed in units of 1,000m^3/s and a single quote symbol is displayed after the number. A double quote symbol indicates the flow is expressed in units of 1,000,000m^3/s.

  - "Qo" followed by the total flow out of the model (all domains) in m^3/s. If the outflow exceeds 999,999m^3/s or falls below -99,999m^3/s, the flow is expressed in units of 1,000m^3/s and a single quote symbol is displayed after the number. A double quote symbol indicates the flow is expressed in units of 1,000,000m^3/s.

  - "dV" followed by the change in volume in m^3 of the model (all domains) since the last display time. If the change in volume exceeds 999,999m^3 or falls below -99,999m^3, the amount is expressed in units of 1,000m^3 (mL) and a single quote symbol is displayed after the number. A double quote symbol indicates the change in volume is expressed in units of 1,000,000m^3.

- The negative depth numbers, cumulative error percentages, inflow, outflow and change in volume figures are very useful to gauge the health of the model. Frequent negative depths, poor cumulative error (>1%, noting that some models will show a high mass error at the start, which can be acceptable provided it diminishes quickly) and "bouncy" inflow, outflow and change in volume values are all indicators of an unhealthy model.

- Whenever the map output is written (see Start Map Output and Map Output Interval), a line "Writing Output at:" is displayed followed by the simulation time, the clock time and the CPU time. If the CPU time is significantly lower than the clock time, then either the simulation was paused for awhile the CPU is overloaded or the CPU is not being fully utilised.

Check Files

xp2D produces check files for quality control of a model's input data. It is strongly recommended that models are quality controlled through reviews of the check files. Effective use of the check files can save days during a model's development and application.

ERROR, WARNING and CHECK Messages (in .tlf and _messages.mif Files)

Error, warning, check and other useful messages that are output to the Console window and log file are also output to a .mif file provided the message can be geographically located within the model domains.

24 Quality Control

Unhealthy Models

Proficient and effective 1D and 2D hydrodynamic modelling is a skill that takes years to develop to the point where the modeller can produce unproblematic, healthy models that consistently simulate floods and tides without drama! During the development of these skills, most modellers produce "unhealthy" models at some point, ie. models that are problematic in that they regularly go unstable, produce strange flow patterns, and so on. While in most cases the reasons for problems are within the quality of the input data, other reasons include poor model schematisation, and, of course, human error. With mentoring from experienced modellers, and/or following an iterative testing process, unhealthy models can be turned into healthy ones, and hydrodynamic modelling skill levels greatly enhanced. This section attempts to try and convey some of the ways of identifying problematic areas within an unhealthy model, and solutions to resolving the problem. Any constructive suggestions on how to expand this section further are welcome!

Unhealthy models usually exhibit one or more of the following characteristics:

- The model only remains stable if using a smaller than recommended timestep.
- Poor mass error (> ±1 to 3%) as indicated by the "CE" percentages displayed to the Console Window (see Section **Error! Reference source not found.**), and output to the various mass balance files.
- "Unnatural" fluctuations of flow in/out and change in volume values (ie. the Qi, Qo and dV values displayed to the Console Window) discussed in Section **Error! Reference source not found.**.
- Locations in the model that repetitively have negative depth WARNINGs. These repeatedly appear as a message such as:

  "WARNING 2991 - Negative U depth at [030;088], Time = 0:01:30, Depth = -0.4…" The occurrence of the message several times at a location is usually not an issue (this means that the model experienced a short and slight numerical disturbance), however, if it repeatedly occurs for a period of time, it is good practice to resolve the problem as this numerical disturbance is likely to be causing mass errors, possibly forcing the use of a too small a timestep, and may initiate an instability in a future simulation.

If one or more of the above apply, the model needs to be reviewed and reasons identified. This can be a daunting and difficult task for inexperienced modellers; however, the guidelines in the sections below are hopefully of some assistance!

Tip: Take an iterative approach to solving problems as one problem often causes other problems. When searching through the _messages.mif file, resolve the problems in order of occurrence (ie. in the order the messages appear in a MapInfo Browser Window).

The most common cause for an unhealthy model is poor underlying topography. In the case of 2D domains, the quality of the DTM is often the problem, therefore, investing in time to create truly representative, well constructed, DTMs is highly recommended from both the modelling perspective and the quality of the inundation mapping.

For 1D domains, topographic inaccuracies in cross-section data and at structures is often a problem, although as 1D modelling is more of an art than 2D modelling (there is much more a modeller can intentionally, or unintentionally, "fiddle" in a 1D model), the selection of cross-section locations and schematisation of the 1D domain is often an issue.

The underlying message is invest in good quality input data and experience! As the saying goes:

"garbage in, garbage out"

Timestep

There is a tendency for hydraulic modellers to "solve" an instability by simply reducing the timestep. Whilst this may "work", it is usually not solving the fundamental cause of the model's poor hydraulic performance or instability.

For the majority of flood models, the 2D Timestep in seconds should be somewhere between ¼ to ½ of the 2D Cell Size in metres. For example, a 10m 2D grid should use a timestep of between 2 and 5 seconds. 2D domains with predominantly sub-critical flow usually can have timesteps larger than those for steeper models with significant areas of supercritical flow.

For coastal models, models with large cell sizes (>50m) or models with significant areas of deep water (>5m), the above rule-of-thumb may not apply with the timestep often being smaller. This is due to the Courant condition.

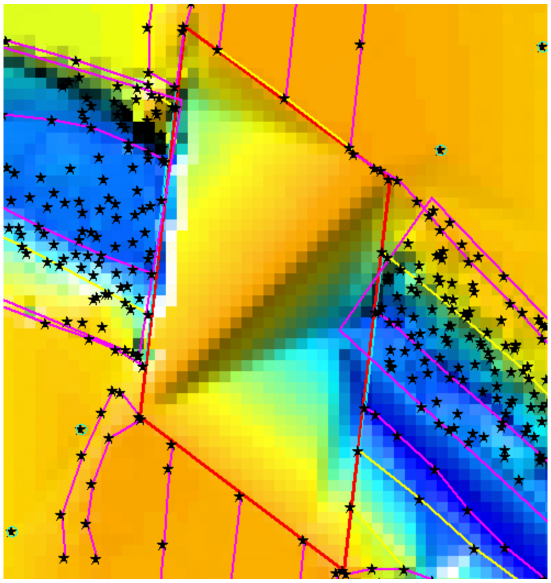
Using too small a timestep can tend to mask fundamental problems in the input data, and hide mistakes in the construction of the model.

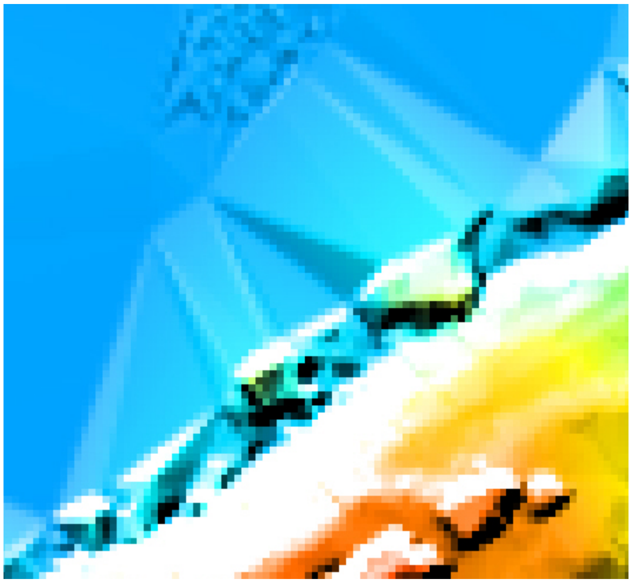
Using a too large a timestep will cause mass errors. If the model runs stable without any negative depth warnings, yet the cumulative mass error is poor throughout the simulation, this is often an indication that the 1D and/or 2D timesteps are too large.

Tips for an Unhealthy 2D Domain

To identify problematic areas within a 2D domain, the most common approach is to run the model at a reasonable timestep and investigate where the WARNING 2991 messages occur. If mass error is an issue in the 2D domain, the _MB1.dat output (see Map Output Data Types) may be of use.

Of particular note is that models based on high quality DTMs are as-a-rule rarely problematic, however, if the DTM is rough or “bumpy” (as a lot of air-borne laser DTMs are), or has poor triangulation causing sharp ridges as illustrated in the images below, models based on these DTMs are much more likely to be problematic in/near these areas.





The following steps are useful to help resolve the problem:

3 Set the 2D Timestep to somewhere between ¼ to ½ of the cell size (in metres).

4 Run the model until it becomes unstable or has generated the WARNING 2991 messages.

5 Import the _messages.mif layer to the diagnostic layer

6 Try to trace back from the “UNSTABLE 2999” messages to the initial “WARNING 2991” messages that were most likely the trigger for the instability.

7 In the Browser Window select a few of these WARNINGs.

8 Using the various _check.mif layers, carry out some fundamental checks:

(a) 2d_zpt_check: Check the Zpt values are as you would expect. If the Zpt values are particularly “bumpy”, try smoothing the ones at/near the WARNING 2991 messages. Where the Zpts change in elevation rapidly (eg. the outside bend of a river), deep ZU and ZV elevations can be problematic. If modifying the ZU and ZV values, tend to assign an elevation closer to the higher of the two ZC values either side of the ZU/ZV point. To modify Zpts don’t edit the original 2d_zpt layer, instead:

(i) Select the Zpts you wish to modify from the 2d_zpt_check.mif layer;

(ii) File, Save As…, and choose Selection and provide a name such as 2d_zpt_for_stability.tab;

(iii) Open 2d_zpt_for_stability.tab (this will be the first file on the recent files list);

(iv) Make the layer editable, label the points with their elevations and using the Info tool, edit the elevation attribute (must be the fourth attribute) to smooth/change the topography.

(v) Save and export 2d_zpt_for_stability.tab as a .mif file.

(vi) Add a line such as Read MID Zpts == mi\2d_zpt_for_stability.mid to the .tgc file after all the other Zpt commands, noticing that this is a Read MID, not a Read MI command.

(b) 2d_uvpt_check: check the Manning's n values are as expected, if not identify why.

Other points to note for 2D domains are:

9 If the model becomes unstable quickly and the instability location is near a 2D water level boundary check that the initial water level setting is compatible with the starting water level of the boundary.

10 Direct rainfall modelling on high elevations may experience unacceptable mass errors (>100m) due to a floating point imprecision problem. These models need to use the double precision version of engine.

XP Solutions, Inc.

North American Office:
5415 SW Westgate Drive
Suite 150
Portland, OR 97221
Phone: +1-888-554-5022
Fax: +1-888-554-5122

Australian Office :
8-10 Purdue Street
Belconnen ACT 2617
Postal Address:
PO Box 3064
Belconnen ACT 2616
Phone: +61 (02) 6253 1844
Fax: +61 (02) 6253 1847

U.K. Office:
Jacobs Well
West Street
Newbury, Berkshire
RG14 1BD
Phone: +44 0 1635 582555
Fax: +44 0 1635 582131