

Triangle-Quadrilateral Grid Generation (TQGG) User Manual

Roy Walters Clayton Hiles
Ocean-River Hydrodynamics Cascadia Coast Research Ltd.

Kristoffer Lorentsen
Cascadia Coast Research Ltd.

May 27, 2015

Table of Contents

Contents

1	TQGG Program Description	3
1.1	Introduction	3
1.2	Use of TQGG	3
2	Grid generation procedures	3
2.1	Sampling data	4
2.2	Generating a grid	5
2.3	Local refinement	6
2.4	Grid quality	6
3	Menu description	6
3.1	Contents of top menu	6
3.2	Contents of menu: File	6
3.3	Contents of menu: View	9
3.4	Contents of menu: Info	11
3.5	Contents of menu: GridGen	17
3.6	Contents of menu: NodeEdit	20
3.7	Contents of menu: GridEdit	23
3.8	Contents of menu: Polygons	26
3.9	Contents of menu: NodeInPoly	28
3.10	Contents of menu: GridInPoly	29
3.11	Contents of menu: Help	33
4	Input and output formats	33
4.1	Neighbour (NGH) format	35
4.2	Node (NOD) format	37
4.3	Element (ELE) format	39
4.4	Grid (GRD) format	40
4.5	Node (XYZ) format	42
4.6	Section (XSC) format	43

1 TQGG Program Description

1.1 Introduction

This program uses interactive graphics to read geometric and grid data, to create a background grid from the geometric data, to create a model grid, and to allow examination and modification of an existing grid. At the end of an editing session, this program can output a triangle list in addition to the modified grid. The program permits the user to display various properties of the grid, such as coordinates of individual vertices. It also provides means of displaying various properties of vertices and triangles that normally cannot be judged by eye. For instance, colour markers can be placed at all vertices where water depth exceeds a specified value.

Changes can also be made to the grid. For instance, vertices and connections between vertices can be added or deleted; vertices can be moved or merged with one another; and triangle shape can be adjusted. The user directs changes in purely graphical terms, by suitable positioning of a cursor on the displayed grid; the program keeps account of all corresponding changes in vertex coordinates and interconnections between vertices. Any proposed changes are displayed immediately, for confirmation or cancellation.

1.2 Use of TQGG

This program is invoked from the command line by typing the name of the executable file (typically 'TQGG'). A frame for the grid editing area is drawn in black on the screen and a menu along the top of the panel allows interaction with the program. At this point, the user chooses a menu item that leads to node or grid input, whichever is desired. The initial display shows the entire grid being edited. If the grid is larger than 1000 nodes, only the outline is shown initially. The entire grid can be displayed by selecting **{View}Redraw** from the View menu.

Editor options are presented to the user in menus that appear across the top of the screen, and additional prompts are displayed when necessary. Selections from the menus are made by means of the mouse, but some editing operations require keyboard input also.

The various editor options are discussed below, menu-by-menu. One important editor facility which should be noted is the save option; it is recommended that the current version of the grid should be saved at regular intervals during long editing sessions, in case of power failures or computer gremlins.

2 Grid generation procedures

Generating an unstructured grid relies on a series of subtasks that depend primarily on the study objectives and the data or initial grid available for input. If the study is a one-off, then usually the study area is treated in some detail and the grid becomes coarser out to where the boundary conditions are applied. If

the grid is a basic resource for several current and future studies, then it may be better to create a larger scale background grid with good quality everywhere, and refine the grid in specific areas as necessary.

The initial data for the grid can come from a wide range of sources. Generally, it is easier to start with an existing grid and make modifications than to start from scratch with sampling. Two types of data are required: data that approximate the external and internal boundaries, and data that provide the topography. Here we use a right-hand coordinate system with x directed eastward, y directed northward, and z directed upward, although the (x,y) plane can be rotated if appropriate. The dimensional units are either degrees for spherical polar coordinates or meters otherwise. Arbitrary units can also be used if care is taken to not invoke inappropriate transformations such as UTM or Polar.

The program has sufficient flexibility that tasks can usually be performed using several methods. In the following subsections, we present methods to accomplish some of the major grid generation tasks. The user may find other methods that are more suitable for their problem and that is all in keeping with the design of the program. Other more specific operations can be found while perusing the menu items.

2.1 Sampling data

Boundary data is obtained from an existing grid by reading the grid using **[File]/OpenGrid** and then transforming the grid file to a node file using **[GridEdit]/Grid2Nodes**. Save the node file and then read it back using **[File]/OpenNode**. At this point, the data is represented as a series of boundaries starting at the outer boundary, followed by island boundaries followed by line constraints, and finally all the interior nodes. Delete the interior points by defining a polygon with **[Polygon]/Whole** and then **[NodeInPoly]/DeleteInt**. The result is a data set with external boundary, islands (if any), and interior line constraints (if any). Save this file for future use and editing with **[File]/SaveAs**. Generally, it is safer to save a file that has been modified then read it again because this operation cleans up the indexing associated with deleted nodes. At this point, the boundary resolution can be changed in individual sections using **[NodeEdit]/ReSample** or within a polygon using **[NodeInPoly]/ReSample**. The boundaries can also be edited using items in the menu **[NodeEdit]** or **[NodeInPoly]**. Island boundaries and line constraints from other files can be added through the menu item **[File]/AddNode**.

Boundary data can also be obtained from an existing set of coordinates for point data through the menu item **[File]/Sample**. For instance, the input data can be a text file with lines containing (x,y) or (x,y,z) coordinates generated from vector shorelines or other sources. The program will prompt for a minimum distance between data points and prefilter the data to reduce the size of high resolution data sets. The minimum distance should be some fraction of the desired final resolution ($\approx 1/4$ or so) in order to maintain accuracy when sampling. Save each input file as a node file using **[File]/SaveAs**. Each output file contains one boundary arranged in order from one end to the other and

the boundaries from each file can be added together using **[File]/AddNode** and adding each boundary in sequence. These boundaries can then be joined, reversed, and manipulated using items in **[NodeEdit]**. As an alternative, the files may include multiple segments with lines containing (x,y,z,s), where s is a segment identifier. All points with the same segment identification should be arranged in order from one end to the other and the points will be automatically arranged into a separate boundary. Multiple segment data can be manipulated in the same manner as the single segment data.

The end product after editing is a node file containing boundaries at the specified resolution. This file is used directly by the grid generation procedures. AN IMPORTANT NOTE: the node spacing along boundaries should vary smoothly otherwise the grid generation procedure that starts on the boundary will generate highly irregular elements corresponding to the degree of irregularity of the boundary data. The resample options provide an easy method to smooth the distance between boundary points. Choosing menu item **[Info]/BoundaryCheck** will run a procedure that will indicate whether there are errors in any of the boundaries. If the data pass this test, the next step is grid generation.

2.2 Generating a grid

An important prerequisite to generating a grid is to verify that distances measured along the x and y axis are the same; that is, the aspect ratio (x distance/y distance)=1. For grid units of meters (igridtype=1 or 2), this is satisfied. For user defined units (igridtype=3) this may not be the case so that the x or y dimensions must be scaled so that aspect=1. Otherwise, the elements that are generated will be distorted and stretched in one axis direction, leading to reduced accuracy in a numerical model. This issue also applies to spherical polar coordinates (igridtype=0) since horizontal distances in degrees decrease as the poles are approached. For the latter, this problem is resolved by transforming to a local coordinate system using **[View]/PolarTransform** so that the aspect=1. However, after grid generation and before saving the file, the grid must be transformed back to the global polar coordinates by invoking **[View]/PolarTransform** again.

The main grid generation algorithm is a frontal marching method that starts at the boundaries and attempts to create equilateral triangles as it marches into the interior. The size of the triangles can be controlled in 2 ways. In the first case if no reference grid is chosen, the size will vary linearly between boundaries. This gives a relatively uniform element size that grades from the inner boundaries to coarser open boundaries. In the second case if a reference grid is chosen, the depth in the reference file is used to determine element size; ie, deep areas have larger elements than shallow areas. Limits are placed on the rate the element size can grow or decrease in order to maintain a smooth variation in sizes.

The overall procedure then is to read in the node boundary data using **[File]/OpenNode**. Then use **[GridGen]/OneFront** in order to examine the results from the generation of the first front. If these results are acceptable, then read the node file again and select **[GridGen]/AllFronts**. The program

will then generate interior elements throughout the area within the external and island boundaries.

[Under construction]

2.3 Local refinement

[Under construction]

2.4 Grid quality

The overall objective is to generate an unstructured grid that is free of errors, grades smoothly from one area to the next, and contains roughly equilateral triangles and/or square quadrilaterals. Errors can be eliminated completely but the grading and element shape can only be optimized due to geometric constraints.

[Under construction]

3 Menu description

The top menu has entries that lead to similar groups of operations. For instance, **File** contains the operations to read and write data into files and **View** contains the operations to change the display (zoom, pan, etc.). Selecting a top menu item brings up a submenu with more specific operations. An attempt has been made to make the menus and submenus both logical and intuitive.

3.1 Contents of top menu

When the program begins, the following top-level menu appears across the top of the screen:

File View Info GridGen NodeEdit GridEdit Polygons NodeInPoly GridInPoly Help

The entries in this menu indicate further sub-menus, which can be selected by placing the cursor on the appropriate word in the menu and then clicking the mouse, i.e. pressing any button on the mouse.

In the following discussion, each menu item is prefixed or followed by a reminder in curly brackets of which menu the option appears in, e.g. option EditNode in the TOP menu will be referred to as "{TOP}**EditNode**", "**EditNode** in {TOP}" or as **{EditNode}**.

In all editing operations, which involve moving nodes, such as Move or Reshape, depths at new locations of nodes affected by the changes are evaluated automatically by linear interpolation among existing nearby depths. In other cases, the user is offered the choice of setting depth by linear interpolation or by entering a value via the keyboard.

3.2 Contents of menu: File

When this option is picked, the following menu options are displayed:

Table 1: File menu items.

File :
OpenGrid
AddGrid
OpenNode
AddNode
Sample
XSection
IntrimSave
SaveAs
Print
Quit

These menu items are described next.

3.2.1 Menu item OpenGrid

This option allows for reading a new grid file in NGH or GRD format (see Section 4). Any existing data is replaced.

3.2.2 Menu item AddGrid

This option allows for reading a new grid file in NEIGH format (see Section 4). This grid is merged with any existing data so that this action is used to join grids together. Along the edge of the grids that are merged, the boundary node codes are changed to 90, which make this a line that is fixed in space. After merging files, use the **{Info}NodeCheck** to examine the codes at the ends of the lines where the grids are joined. In many cases, the codes at these points need to be changed to 1, 5, or 6, depending on the type of boundary (land, open, or junction of land and open).

When joining grids in this way, it is helpful to begin the editing session with the two sub-grids displayed in different colours. This can be done by suitable choice of the SECONDARY COLOUR INDEX, as explained in the description of option **{Configure}ConfigGrid** later in this Chapter.

The two sub-grids may have to be stitched together as required. Some nodes may have to be deleted if the two sub-grids overlap; conversely, extra nodes may have to be added if there is a substantial gap. Connections between

nodes of the two sub-grids may be established using **{GridEdit}AddLine** or the node-merging capability available through **{GridEdit}Move**. To use this later, which is very convenient where there are pairs of adjacent nodes from the two sub-grids, it is first necessary to invoke **{GridEdit }GridMerge**.

3.2.3 Menu item OpenNode

This option allows for reading a new node file in NODE format (see Section 4). Any existing data is replaced. After picking this option, the user is prompted for the file name of a NODE format file.

3.2.4 Menu item AddNode

This option permits reading a NODE format file and sub-sampling the boundaries and interior nodes by index or by distance. If there are no nodes in existence prior to choosing this option, a NODE format file can be sub-sampled. Otherwise, the nodes are added to the existing nodes. If the prompt option is ON, the user is prompted before any set of nodes is added. This is useful for selectively adding boundaries or interior nodes. [THIS SECTION NEEDS TO BE UPDATED. I AM NOT SURE THAT THE PROGRAM BEHAVIOUR IS STILL AS DESCRIBED]

3.2.5 Menu item Sample

This menu item is used to sample *.xyz files into TQGG. The workspace will be cleared prior to sampling the files. The user will be prompted to input a gridtype and minimum subsample spacing. The subsample spacing is defined in meters and will subsample the boundaries by only including new nodes that are farther away from the previous one than the minimum spacing.

3.2.6 Menu item XSection

The following option is used for creating grids from cross section data. A file containing cross section data is read and a grid is created from this data. Options in the right-hand-panel define the number of nodes to create across the section, and the number of nodes to create between each cross section. All these nodes are interpolated using a cubic spline algorithm. See Section 4 for a description of files in XSEC format. [THIS SECTION NEEDS TO BE UPDATED. I AM NOT SURE THAT THE PROGRAM BEHAVIOUR IS STILL AS DESCRIBED]

3.2.7 Menu item InterimSave

Invoking this option leads to output of the current version of the grid in ‘.NGH’ format or node file in a ‘.NOD’ format. The resulting file is less compact than the NEIGH file and NODE file obtained with the usual EXIT procedure (see below). But it is a useful facility and should be used at regular intervals during a long editing session to avoid losing one’s work in the event of power failure or other

interruption. Alternate interim saves are written to files named interim1.*** and interim2.***; the name of the last interim save file output can be checked via the **{Info}Files** option.

3.2.8 Menu item SaveAs

This is the option normally used after the completion of a grid or modification of a grid and node file. It brings up a request for the name of the file in which the final version is to be saved. If the file does not exist then a prompt appears asking the user whether or not to create the file. If no file output is selected or 'CANCEL' is selected, then the program exits with open error message.

3.2.9 Menu item Print

This option is identical in all the interactive programs. Picking this option brings up the Windows print manager. Note that for all printing it is advisable to change the background to white using Background in **{View}**. [This menu is not implemented yet in Motif version of TQGG.]

3.2.10 Menu item Quit

This allows exit from the Editor without output of any files. The user is prompted to answer whether an exit is really desired. This helps prevent accidental termination of the program.

3.3 Contents of menu: View

This option provides control over windowing and uses the same module as is used in the other interactive graphics programs.

When the View menu is chosen from **{TOP}**, the options appear as shown in Table 2.

The functions in this menu provide control over windowing (zoom) and refreshing the display. Up to [?] levels of windowing are allowed.

3.3.1 Menu item Redraw

This function forces a refresh of the current display. It may be used for instance when grid lines have become partially erased during removal of markers. It is also used in some instances to get a clean display after turning off options such as display of the original digitized boundaries.

3.3.2 Menu item Outline

This option sets a switch that allows only the boundary to be drawn. This speeds the redraw considerably and is useful in the manipulation of large grids.

Table 2: View menu items.

View:
Redraw
Outline
FullSize
Zoom
ZoomOut
Pan
LastView
Scale
Shift
Rotate
PolarTransform
UTMTransform

3.3.3 Menu item FullSize

After a set of windowing, this option will bring the display back to the full size displaying the whole grid.

3.3.4 Menu item Zoom

Selection of this option with the mouse permits windowing in (zoom in) on any square sub-area of the current window. The area are specified by assigning to opposite corners of the screen. To specify the corners, click and hold the mouse button, move the mouse to a new location and release. The corners are defined by the location for click and location for release. The new window is automatically squared off and refreshed.

3.3.5 Menu item ZoomOut

When ZoomOut is selected with the mouse, the program will zoom out to the lowest zoom level and refresh the window.

3.3.6 Menu item Pan

This option permits the user to move the current window in discrete steps in any direction over the grid, that is, to display, at the same level of magnification, a portion of the grid adjacent to that currently being displayed. This can be used, for instance, to tour a boundary at a high level of magnification. After choosing

Pan, press and hold the mouse button at any location. To specify pan direction and distance, move the mouse in desired direction and distance, and release the mouse button. The window selection will move and refresh accordingly.

3.3.7 Menu item LastView

This section returns the viewing window to its previous state (before a zoom, pan, FullSize, etc. operation).

3.3.8 Menu item Scale

This menu selection allows the x, y, or z dimensions to be linearly scaled. The user will be prompted to input scale factor for each dimension.

3.3.9 Menu item Shift

This allows shifting of the x and y coordinates of all the grid nodes. The user will be prompted to input the amount to add to each dimension.

3.3.10 Menu item Rotate

This selection allows the grid to be rotated by a given angle. The user will be prompted to input the angle of rotation. The grid will be rotated clockwise. After rotating, it might be necessary to adjust zoom or pan the window.

3.3.11 Menu item PolarTransform

This selection converts the present x and y coordinate using a spherical polar transform. Selecting this menu item once performs the forward conversion (from degree to spherical polar coordinates). Selecting this item a second time performs the inverse operation.

3.3.12 Menu item UTMTransform

This selection converts the present x and y coordinate using a UTM Transform. If the current grid type is Lat/Lon it will convert to UTM. If the current grid type is UTM, it will convert to Lat/Lon. The user will be prompted to input a UTM zone for the conversion.

[TODO: Transform menu item descriptions should be support with a description of each type of transform]

3.4 Contents of menu: Info

When the Info menu is chosen from **{TOP}**, the options appear as shown in Table 3.

These permit placing of coloured markers on the screen to mark locations or to display properties of vertices and triangles. Solid colouring of triangles is available as an alternative to colour markers.

Table 3: Info menu items.

Info:
NodeInfo ElementInfo
NodeCheck ElementCheck BoundaryCheck EraseChecks
PMarkers EraseLast EraseAll
SetRange TooClose
Files Limits

3.4.1 Menu item NodeInfo

Selection of this menu item prompt the user to select a node. In the Motif version of TQGG this is done using the mouse to click on the node. The following information is displayed in the terminal window. Note that work is ongoing to translate output to a dialogue box.

- Index number
- Node Code
- Coordinates (X, Y, Z)

A marker is placed at each vertex examined. If a number of points are examined, it may be desirable to erase existing markers in order to be able to spot new markers. **{View}Redraw** erases all markers.

3.4.2 Menu item ElementInfo

Selection of this option prompts the user to select an element. In the Motif version of TQGG this is done using the mouse to click on the element. The following information is displayed in the terminal window. Note that work is ongoing to translate output to a dialogue box.

The following information on the selected element appears:

- Index number
- Coordinates
- Element code (type)
- List of nodes in the element

A marker is placed in each element examined. If a number of them are examined, it may be desirable to erase existing markers in order to be able to spot new markers. **{View}Redraw** erases all markers.

3.4.3 Menu item NodeCheck

This enables labelling of grid vertices with coloured markers according to certain built-in criteria or an external list, so providing an efficient visual means of simultaneously checking specific properties at all vertices visible in the window. When this option is selected, appropriate information appears in a dialogue box with available criteria as shown below. [This dialogue box is still under construction and not all options work].

The final option EXT instructs the Editor to read in a prepared external file (in EXTVER format), each line of which contains the number of a vertex where a marker is to be placed, followed by an integer specifying the colour of the marker. [This option not yet implemented in Motif version of TQGG].

The name of the external criterion file is not requested until the EXT option is invoked; consequently, it is possible for the user to colour vertices according to different criteria in succession by preparing as many external files as required. The name of the external file currently assigned is displayed in the information panel. A different external file can be assigned by clicking the name of the existing file in the information panel.

Finally, when all details of the criterion or criteria to be displayed have been decided, the user should press the "Run Check" button in order to view the appropriate coloured markers.

3.4.4 Menu item ElementCheck

To permit convenient monitoring of certain triangle properties, default or user defined colour tables are used to place coloured markers and solid colours in triangles. When the colouring option is invoked, necessary list of all triangles and their calculated triangular properties are updated. This enables the user to check effects on triangle properties of any editing grid immediately.

When the ElementCheck option is selected, a dialogue box appears with options for selecting the colouring mode and the type of test as shown below.

Table 4: Vertex criterion

Option	Description	Default
C0	Computational code equals 0	OFF
C1	Computational code equals 1	OFF
C2	Computational code equals 2	OFF
C3	Computational code equals 3	OFF
C4	Computational code equals 4	OFF
C5	Computational code equals 5	OFF
C6	Computational code equals 6	OFF
C7	Computational code equals 7	OFF
C8	Computational code equals 8	OFF
C9	Computational code equals 9	OFF
NC0	Computational code not equal to 0	OFF
C=?	Computational code to be set by user	OFF
DLT	Depth less than d1	OFF
DGT	Depth greater than d2	OFF
DBTW	Depth greater than or equal to d3 and less than or equal to d4	OFF
NBGT	No of neighbours greater than n1	OFF
NBLT	No of neighbours less than n2	OFF
NBE	No of neighbours equals n3	OFF
EXT	Mark according to external list	OFF

Colour Triangles By Criteria
Select Colouring Mode:
Full Color Color Marker
Select Criteria
EQL DEP A2D CCW G90 14 COD

The colouring mode can be selected between "Full Colour" and "Colour Marker" with the appropriate radio buttons. "Colour Marker" indicates that coloured markers will be placed in the triangles, while "Full Colour" indicates solid colouring of triangles. Markers are preferable if editing operations are to be carried out on the grid.

Picking the radio buttons beside EQL, DEP, A2D, CCW, G90, or COD determines which of the following internally evaluated triangle properties is to be displayed:

- EQL - measure of equilateral shape, defined as the ratio of the sum of the squares of the sides of the triangle-to-triangle area, normalized in such a way that an equilateral triangle has this ratio equal to unity. This ratio or shape factor increases as triangle shape departs from equilateral. For instance, a right-angled isosceles triangle has a shape factor of 1.154
- DEP - mean depth (average of depths at vertices)
- A2D - area/mean depth
- CCW - clockwise test (+1 if vertices ordered counter clockwise in triangle list, -1 if clockwise). The default colour scale for counter clockwise is black and red for clockwise.
- G90 - flags triangles with angles greater than 90 degrees. The default colour scale is black if less than 90-degree angles, red otherwise.
- COD - element code. The default colour scale follows the colour indices.

In addition, an external file (in EXTCRI format) containing a list of triangles and corresponding values of any quantity defined by the user can be read in, using option EXT.

After the needed check and options are selected, the check can be run by pressing RUN CHECK. The markers or coloured triangles will be displayed. To erase the checks, use **{Info}EraseChecks**

3.4.5 Menu item BoundaryCheck

Selection of this option will perform boundary checks on the boundary. The check is intended to be run prior to doing a triangulation to make sure that the grid is defined correctly. The following checks will be performed:

Boundary Orientation All boundaries orientation are checked. Clockwise boundaries are then displayed in *red*, while counter-clockwise boundaries are display in *green*

Outer Boundary Location The outer boundary is found and if this is not the first boundary in the data-arrays, the user will be prompted to move it.

Outer Boundary Orientation The outer boundaries orientation should be CCW. If it is CW, the user is prompted to reverse it.

Island Boundary Orientation All other boundaries than the outer boundary should be defined CW. If any are CCW, the user will be prompted to reverse them.

Number of nodes on boundaries If there are boundaries that have 3 or less nodes, the user will be prompted to delete these.

Node code reset All nodecodes are reset to: 1 - outer boundary, 2 - island boundary

Triangulation Check TQGG will perform a test-triangulation of the current boundaries. If any errors occur, the user will be notified and a permanent marked will be placed on the location where the error occurred. Errors include: intersecting boundaries, coincident nodes and more.

3.4.6 Menu item EraseChecks

This option turns off place, vertex and triangle marking.

3.4.7 Menu item PMarkers

Selection of this option allows the user to place a coloured marker anywhere in the current window by means of the mouse. These markers remain displayed until **EraseAll** is invoked. **EraseLast** erases the last marker created. Markers survive windowing and consequently, one of the purposes they can be used for is to identify an area of interest on the grid.

3.4.8 Menu item EraseLast

Selection of this option will delete that last marker that was placed on the grid.

3.4.9 Menu item EraseAll

Selection of this option will delete all markers on the grid.

3.4.10 Menu item SetRange

Selection of this option will allow the user to input a range for the **{Info}TooClose** menu item. The range is in meters.

3.4.11 Menu item TooClose

The TooClose option allows automatic detection of nodes that are too close to one another (coincident). All the nodes in the workspace will be sorted along the x-axis, any node that has another node within its range in the positive x-direction will be flagged to be deleted and marked in the workspace. The

range is set using **{Info}SetRange** menu item. If no range has been set when invoking this option, the user will be prompted to input one. The distance between nodes is calculated using the *Haversine* formula for Lat/Lon grids and *Pythagoras' theorem* for meters based coordinates.

After nodes are flagged and marked, the user is prompted to delete them. If confirmed all the flagged nodes will be deleted.

3.4.12 Menu item Files

This option brings up a list of files currently assigned to the Editor. Filenames cannot be changed via the display panel under this option.

3.4.13 Menu item Limits

This option displays the maximum number of nodes allowed, the number of nodes used at present, maximum neighbours allowed and maximum boundaries allowed.

3.5 Contents of menu: GridGen

On picking the GridGen option from {TOP} with the mouse, the options appear as shown in Table 5:

Table 5: GridGen menu items.

GridGen:
OneFront
Clusters Options
AllFronts Options
OverlayHex OverlaySquares OverlayMixed
Triangulate

These options are used for the creation of nodes and grids within the working polygon (the currently-activated polygon) and the last option generates triangles from nodes.

3.5.1 Menu item OneFront

[NEEDS UPDATE]

3.5.2 Menu item Clusters

The Clusters option provides a method of creating a set of nodes whose spacing is a function of water depth. As explained in Henry (1988), when the model domain is subdivided into cells whose areas are proportional to water depth and the centres of area of these cells are taken as the basis for a triangular network, the areas of the triangles in the network are also approximately proportional to water depth and the spacing of the nodes is such that the Courant criterion is satisfied approximately throughout the grid, that is, node spacing is proportional to the local phase speed of shallow water waves. It should be noted that triangle area could be made proportional to any scalar quantity defined over the domain, by providing a reference grid for that quantity at stage GenerateMesh above, in place of a depth grid.

The model domain is subdivided into cells by forming appropriately-sized compact clusters of squares belonging to the fine resolution Cartesian grid laid out over the domain using option GenerateMesh. Cluster (cell) area is related to water depth by an expression $A0 + A1*DEPTH + A2*DEPTH**2$, where the coefficients A0, A1, A2 can be set by the user. The linear case described in the preceding paragraph corresponds to $A0 = A2 = 0$.

Choice of option Clusters brings up an information panel on the right, which allows the user to reset various parameters and display conditions. The information panel contents are:

Minimum cluster size (default =1) is considered in Cartesian grid squares. The maximum number of new nodes shown is the maximum node array size permitted taken away from the number of existing nodes in the whole domain. If a starting location for cluster formation is required other than at the point of maximum water depth, the starting location has to be specified in terms of number of rows and columns in the fine Cartesian grid, counting from the lower left-hand corner. The outlines of the clusters of squares from the Cartesian grid can be displayed as picking the option Draw Clusters forms them. Cluster formation begins on picking option ACCEPT.

3.5.3 Menu item Options

When selecting this menu item, the user will be prompted with the following questions:

Display Mesh? Yes or No

Display Ref Grid? Yes or No

Display Clusters? Yes or No

[NEEDS UPDATE]

3.5.4 Menu item AllFronts

Generates a unstructured grid using all boundaries. [NEEDS UPDATE]

3.5.5 Menu item Options (AllFronts)

Menu item under construction

3.5.6 Menu item OverlayHex

[NEEDS UPDATE]

3.5.7 Menu item OverlaySquares

This option places a new interior node at the centre of each square of the Cartesian grid whose centre lies within the active polygon and more than half a mesh interval from the nearest land boundary, that is, at the centre of each square assigned a positive depth as a result of operation GenerateMesh above.

3.5.8 Menu item OverlayMixed

[NEEDS UPDATE]

3.5.9 Menu item Triangulate

The following option is used for triangulation of a set of nodes; that is, it converts the data from NODE format to NEIGH format with a neighbour list and triangle list. The triangulation algorithm used in GRIDIT was devised originally by Cline and Renka and modified by Bova and Carey to handle boundaries. It yields what is known as a Delaunay triangulation, one in which the triangles formed are as near equilateral as possible for the given positions of the nodes.

After triangulation, the grid should be checked both visually and by using various operations under Tests in the {TOP} menu. The reason for checking the output is that although the triangulation algorithm is fairly robust, it can produce various errors, some immediately obvious and some not. For instance, when the coordinate datum is poorly positioned, remote from the domain being modelled, consequent error in single precision subtractions can lead to very obvious misconnections between nodes far removed from one another. To minimize round off errors, coordinate reference should be placed within or immediately adjacent to the grid.

On the other hand, less obvious but serious error can occur if the first few nodes on a boundary are collinear or approximately collinear. Then some or all of the nodes involved, and even some nodes further along the boundary, may have surplus connections to nodes which are not their immediate neighbours in the input node set. Automatic detection and correction of these errors is part of the triangulation algorithm. Checking for this type of error is advisable with the help of the EditGrid option. It can be prevented by modifying the arrangement of the corresponding block of nodes in the input file. Since each boundary corresponds to a closed curve, collinear nodes at the beginning of a boundary block can be moved to the end of the block (using a text editor), where they normally cause no problem.

The triangulation algorithm can be confused by certain complicated coastal geometry. When an island lies partly in a coastal bay and both features have relatively few boundary nodes, spurious connections may be set up; the island coastline may be erroneously incorporated into the outer boundary. Another common error occurs along some nearly straight boundary segments. There are boundary connections that lie outside the boundary and define long, thin triangles that are difficult to see. Normally these connections are removed automatically. They can also generally be detected by using option `{Tests}Nodecheck`, as will be explained later.

These errors are not peculiar to the Renka algorithm. All methods of triangulation used so far by the authors suffer from some problems similar to those mentioned, so it is advisable to have good error-checking capability, no matter what triangulation method is used.

3.6 Contents of menu: NodeEdit

Picking `EditNode` with the mouse brings up a menu with options as shown in Table 6. It gives access to operations affecting individual nodes rather than groups of nodes.

Table 6: NodeEdit menu items.

NodeEdit:
DeleteNode
MoveNode
AddBndNode
Reverse
Join
Split
ReSample
Reselect
AddBndLine
DeleteIsland
AddIntNode
AddIntLine
PolyDeleteBnd
PolyDeleteInt
PolyDeleteAll

NOTE: The subsections below are no longer accurate.

3.6.1 Menu item DeleteNode

This menu options allow for deleting single nodes. Point the mouse at the node that is to be deleted, and click.

N.B. This DeleteNode option should not be used for complete deletion of an island, which requires more radical changes to the NODE file. Instead, use DeleteIsland, described later.

3.6.2 Menu item MoveNode

This menu options allow for moving single nodes. Point the mouse at the node that is to be moved, and click. The node will be highlighted. Click on the location where the node should be moved.

3.6.3 Menu item AddBndNode

This menu options allow for adding a boundary node to an existing boundary. Click on the location where the node is to be added. TQGG will connect the node to the existing boundary and the user will be prompted to confirm the connection.

3.6.4 Menu item Reverse

This option is used to reverse the direction of a boundary. After choosing this option, click on the boundary that is to be reversed.

3.6.5 Menu item Join

This option is used to join two boundarie endpoints. After choosing this option, click on the first boundary, then the second boundary endpoint that is to be joined. The user will be prompted to confirm the action.

3.6.6 Menu item Split

This option is used to split a boundary. After choosing this option, click on a boundary where you want to delete it. After doing so, TQGG will suggest to split the boundary in the direction of the next node that is farthest away from the selected node. If this is rejected, TQGG will suggest to split in the opposite direction. If confirmed, the boundary will be split. Note that endpoints can not be split from the boundary.

3.6.7 Menu item ReSample

This option performs an uneven ReSample along a boundary line. When selected, the user should select two nodes on the same boundary. TQGG will then perform a resample so that the resolution is evenly increasing/decreasing from the resolution at the first node, to the resolution at the second node.

3.6.8 Menu item Reselect

This option allows the user to replace any designated string of boundary nodes with a fresh selection of nodes from a boundary data file in DIGIT or NODE format. Selection may be made on the basis of distance between points or by choosing every Nth digitized point. The following information panel is displayed:

BOUNDARY NODE
RESELECTION

File Type: NODE
File Name:
NONE

Show Bndry from file: NO
Show Bndry Connect: YES
Sampling Rate:
Nth Point: 10
Distance: 1.000
PICK ENDPOINTS
RESELECT
TOP: DISPLAY
TOP: INFO
QUIT

NODE and File Boundary Blocks in Same Order:
NO

Pick [NONE] under "File Name:" as the first operation in this panel. Enter the file name of the file to reselect from to the following prompt. Pick [NODE] next to "File Type:" if the file is in DIGIT format. The asterisk indicates which selection criterion is in effect. To reselect a stretch of boundary nodes, first pick the first and last nodes on the stretch to be replaced. The program will then search the digitized boundary data file and display the points, which it identifies to be the first and last points of the corresponding stretch of digitized boundary data. On confirming this identification, the user can make a fresh selection of nodes by picking ReSelect. Once reselection of the nodes is complete the new nodes will be displayed and the user may save or cancel the reselection at the following prompt. If the reselection is saved a prompt will appear asking if depth for the reselected nodes should be interpolated or assigned from the input file used. Pick QUIT to leave this procedure. The final option in the panel permits some saving of time in the case of very large digitized boundary files. If it is known that the blocks of data (digitized data or nodes) representing islands are in exactly the same order in both the DIGIT and NODE files, then some search time can be saved by toggling NO to YES. Do not use this facility if islands have been deleted with the corresponding updating of the digitized boundary file.

3.6.9 Menu item DeleteIsland

This option deletes all boundary nodes associated with an island, that is, complete removal of an island from the node file. The user selects the island to be deleted by picking any boundary node of the island.

3.6.10 Menu item AddIntNode

[NEEDS UPDATE]

3.6.11 Menu item AddIntLine

[NEEDS UPDATE]

3.7 Contents of menu: GridEdit

This menu option leads to menu items that provide for manipulation of triangular grids, including editing, merging, and splitting. When the GridEdit menu is chosen, the options in Table 7 appear:

Table 7: GridEdit menu items.

GridEdit:
AddLine
DeleteLine
AddNode
DeleteNode
Move
Merge
CleaveNode
Insert
Exchange
DeKite
ReShape
GridToNodes

These permit a wide variety of changes to be made to the displayed grid, as described below.

NOTE: menu items below are out of date.

3.7.1 Menu item AddLine

Option AddLine permits addition of a connection between two vertices. Pick the two nodes by point and click.

Note: When selecting a node by means of the cursor, the message ‘ERROR - Invalid point’ may be displayed. When this happens, a second attempt to select the required node will normally prove successful, since the cursor range is increased automatically. The cursor range can also be changed directly as described in {Info}CursorRange or turned off using {Info}AutoRange.

3.7.2 Menu item DeleteLine

Option AddLine permits the removal of the connection between two vertices. Pick the line that is to be removed by clicking its center point.

3.7.3 Menu item AddNode

Choice of option AddNode permits the user to add a new vertex to the grid. To add a new node - point and click.

3.7.4 Menu item DeleteNode

This option allows deletion of a vertex and its connections to other vertices. To delete a node - point and click.

3.7.5 Menu item Move

The Move operation consists simply of moving a designated vertex to a new location, which should lie strictly within the polygon formed by the neighbours of the vertex, otherwise some line segments will cross. The vertex and its new location are selected with the mouse and cursor. The display shows the revised positions of the line segments linking the moved vertex to its neighbours, so that the move can be confirmed or cancelled. In the latter case, the vertex and its connections are redisplayed in their original configuration.

3.7.6 Menu item Merge

The Merge operation combines two nodes into one. First, use the mouse to choose the node that should be merged into another, then choose the node that is should be merged into.

Values of depth and computational code are changed automatically when vertices are merged. If the vertex being moved is labelled A, and B designates the stationary vertex with which A is merged, then the depth and code for the merged vertex at B are set as follows:

- If B and A are both interior points of the grid, B retains its original code (0) and depth.

- If A is an interior point and B is a boundary point, B retains its original code and depth.
- If A is a boundary point and B is an interior point, B assumes the code and depth of A.
- If B and A are both boundary points, B retains its original code and depth. If the points originally lay on boundaries of different types, the user should check whether the code and depth at B should equal the original code and depth at B or A, and reset them with option {EditGroup}Grids.NodeCode if necessary. Depths and codes at individual nodes can be changed if necessary by means of options in {EditGroup}.

N.B. A node can only be merged with one of its neighbours.

3.7.7 Menu item CleaveNode

Cleave allows the user to replace an interior vertex with two new vertices, each of which is connected to roughly half of the neighbours of the vertex being replaced. To carry out a cleaving operation, it is necessary only to use the cursor to select the point to be replaced. The purpose of this option is to allow convenient insertion of extra vertices. Cleaving is not allowed on boundary vertices or with interior vertices having only three or four neighbours. The depth at each new vertex position is computed automatically by linear interpolation. The Reshape option is often used following the cleave operation in order to improve triangle shape.

3.7.8 Menu item Insert

The Insert option is used in two situations, where adding a line connection to the grid implies creation of a new vertex.

First, a line segment can be added from an existing interior node to a point X on the boundary. This requires creation of a new grid node on the boundary, since every line of the grid must end at a node. This operation can be carried out by placing the cursor at or near the mid-point of a boundary segment. If the boundary points on either side of the new boundary vertex have the same computational code, the editor assigns that code to the new vertex also, otherwise the user is asked to enter an appropriate code. The option of setting the depth at the new vertex manually or automatically is then offered. Automatic depth setting means that the new vertex is assigned a depth equal to the average of the depths at the two neighbouring boundary points. If the new vertex lies outside the former boundary, a yellow marker is placed at the new vertex as a reminder that the reference DEPTH GRID should be updated correspondingly.

The second application of Insert is to add certain extra connections within a quadrilateral consisting of two triangles sharing a common side. This is done by placing the cursor at or near the mid-point of the connection between 2 interior nodes. An extra vertex is added along with new connections. If the

user approves the new configuration displayed, a choice of manual or automatic calculation of depth is offered. If automatic evaluation is chosen, the depth at the new vertex is found by linear interpolation if the four surrounding vertices are all interior points of the grid. In the event that one or more of the surrounding vertices is on a boundary, the depth at the new vertex is set equal to the average of the depths at those surrounding points which have non-zero depths.

In both of the uses of Insert described above, the new vertex is placed at the position of the cursor, i.e. close to the mid-point of the existing connection. Its position can be adjusted subsequently using the Move option in {EditGrid} if required.

3.7.9 Menu item Exchange

The function of this editing operation is to swap a diagonal connection in a quadrilateral formed by 2 triangles. In order to perform this process, place the cursor near the mid-point of diagonal line.

3.7.10 Menu item Dekite

For removing kites (elements with 4 nodes). [NEEDS UPDATE]

3.7.11 Menu item Reshape

Reshape provides a method for forming more equilateral triangles in the grid by making appropriate adjustments in the positions of interior vertices. Use of this option is recommended after any editing operations that involve adding or deleting any vertices or connections between vertices. Reshape makes three passes through the grid, treats the interior vertices in order of their indices, and leaves a vertex in its original position if the computed adjustment is less than about 1% of the linear dimensions of the polygon formed by its neighbours. The depth at each new vertex position is computed automatically by linear interpolation.

Unlike all the other editing operations, Reshape cannot be reversed; it is recommended that the current grid be saved using {File}InterimSave if there is any likelihood of requiring the grid as it is prior to reshaping.

3.7.12 Menu item GridToNodes

Use this option to save a Node file from the existing grid.

3.8 Contents of menu: Polygons

This option permits creation, saving, retrieving, activation and deletion of polygonal areas of the model domain in which the user wishes to carry out editing functions accessed subsequently through the Top menu. The menu is shown in Table 8 :

These options are described next.

Table 8: Polygons menu items.

Polygons:
Create
Whole
Cycle
Delete
Read
Write

3.8.1 Menu item Create

The Create option permits design of a new polygon within which to edit nodes. The user picks successive vertices of the required polygon with the cursor, finishing by picking the first vertex a second time to complete the polygon. Once confirmed, the newly designed polygon becomes the active (yellow) polygon (see Activate below).

3.8.2 Menu item Whole

This option creates a polygon that includes the entire grid.

3.8.3 Menu item Cycle

Repeated picking of option Cycle permits the user to display in turn all, none or individual members of the list of stored polygons. Non-active polygons are outlined in red, whereas the active polygon is outlined in yellow.

3.8.4 Menu item Delete

Delete is used in conjunction with Cycle to delete the currently active polygon from the stored list of polygons.

3.8.5 Menu item Read

This option allows the user to read in a named file containing polygons designed and saved during some earlier node-editing session by means of Write. For instance, when running the demonstration case supplied, use Read to read a file named POLYEAST.DAT, which defines a particular polygon to be used if exact comparison with subsequent test outputs to be possible.

3.8.6 Menu item Write

This saves all currently defined polygons to a file named by the user. It may be used in conjunction with Read below.

3.9 Contents of menu: NodeInPoly

This option gives access to various editing operations that can be carried out on groups of nodes once one or more working polygons have been set up. The menu is as listed in Table 9

Table 9: NodeInPoly menu items.

NodeInPoly:
ReSample
DeleteBnd
DeleteInt
DeleteAll

3.9.1 Menu item ReSample

Use this this option to resample all boundary nodes within the current active polygon. The user will be prompted to input a resample distance in meters. TQGG will then resample all boundaries in the active polygon so that they are at that specified resolution.

3.9.2 Menu item DeleteBnd

Use this menu options to delete all boundary nodes within the currently active polygon.

3.9.3 Menu item DeleteInt

Use this menu options to delete all internal nodes within the currently active polygon.

3.9.4 Menu item DeleteAll

Use this menu options to delete all nodes within the currently active polygon.

3.10 Contents of menu: GridInPoly

This option gives access to various editing operations that can be carried out on groups of nodes once one or more working polygons have been set up. The menu is as listed in Table 10

Table 10: GridInPoly menu items.

GridInPoly:
NodeCode
ElementCode
DeKite
ReShape
DeleteGrid
SplitGrid
RefineGrid
CutOffGrid
SetDepth
ReDepth

3.10.1 Menu item NodeCode

This option permits changing all computational codes for nodes within a polygon. For instance, all the nodes with code = 1 along a section of land boundary can be changed to code = 5 for an open boundary. Then the endpoints are set manually to code = 6 to describe a node at the junction of a land and open boundary.

3.10.2 Menu item ElementCode

This option allows the element code to be changed for groups of elements in the current polygon, or by reading a polygon file. When a grid is set up, separate polygons should be created using DefineGroup to define separate element types. All these polygons should be saved in a file and then read to set element codes for any modification of this grid. The first polygon (usually the whole polygon) defines element code 1 and the second code 2 etc.

3.10.3 Menu item Dekite

Reshapes all kites (elements with 4 nodes) in the active polygon. The operation is similar to that in {EditGrid}Dekite.

3.10.4 Menu item Reshape

This option allows a reshape of the elements in the current polygon. The operation is similar to that in {EditGrid}ReShape.

3.10.5 Menu item DeleteGrid

This option deletes the grid section selected by the current polygon.

If a polygon is not active a message is displayed asking the user to define a polygon first.

3.10.6 Menu item SplitGrid

The split option permits division of an existing grid into two separate grids. It is used most frequently to remove surplus parts of a grid outside the open boundaries, after the latter have been positioned. It can also be used to split a large grid into smaller parts, either temporarily, to facilitate editing, or permanently, to provide grids for smaller models. (The opposite process of joining together small grids to make a larger one can be carried out by means of the {File}AddGrid option). In order to use this option, an active "splitting" polygon must be created first using DefineGroup. In this case, grid parts inside and outside the polygon are separated into two self-consistently numbered grids. Before splitting can be carried out, the user must select suitable vertices near the polygon and move them to the nearest side of the polygon, otherwise the grids will have a gap between them. As well as facilitating movement of vertices, the program also reminds the user to enter appropriate computing codes for each vertex moved to the new boundaries formed on splitting.

Before choosing the SplitGrid option, the user must design a polygon that demarcates the intended division. That part of the initial grid, which lies inside the splitting polygon, will eventually be output as an independent grid, and the remaining part outside the polygon will be output as another grid. Each of the output grids will be in NEIGH format and will have its nodes numbered consecutively from 1 upwards. Note: A current bug in the Splitter makes it inadvisable to place a polygon vertex directly at a boundary node of the grid: a triangle may consequently be dropped from the sub-grid lying inside the polygon when the split takes place. Avoid this by finding the coordinates of the boundary node using the {Info}Node option before choosing SplitGrid and extrapolate the required splitting line to find a suitable vertex position outside the grid.

The following first prompt appears after the option is chosen.

First vertex of side to work on by C - cursor, X - xy, Q - quit?

Normally, the polygon vertices will be placed by means of the cursor, option C, but the X option is useful occasionally, e.g. when repeating a previous run, for instance when practising with the demonstration data. Coordinates are always given in problem length units. On exit from the Splitter, a file of polygon vertex coordinates with the filename POLY.DAT is output automatically to facilitate such repeat runs. The next prompt asks the user to indicate the desired location for the first vertex of the polygon, after which the following prompt appears:

V - pick next vertex L - pick last vertex Q - quit

Vertices should be entered either in clockwise order or counter clockwise order, since they are connected and displayed in order of entry. Choice of location by cursor or coordinates is offered for each vertex. Currently, up to ten vertices are allowed. After the last vertex has been positioned, the user can either confirm or cancel the displayed polygon. Assuming that a satisfactory polygon has been drawn, the next step is to decide which side of the polygon to work on, i.e. to choose one of the sides of the polygon which actually intersects the grid, and then move nearby nodes to this line, so that both sub-grids produced will have boundary nodes lying on the splitting polygon. The relevant prompts are:

Pick 1st vertex of next side to work on by C - cursor, X - xy

Pick 2nd vertex of next side to work on by C - cursor, X - xy

At this point, in response to the prompt “A - automatic M - manual movement to polygon boundary Q -quit”, there is a choice of methods for moving nodes to the splitting line. Manual movement of nodes to the splitting polygon requires choosing both the node and its new position by cursor. Under option A - automatic, so far as internal nodes are concerned, the user merely has to indicate with the cursor which node to move and the node is then moved to the current working side of the splitting polygon along a perpendicular through its original position. However, boundary nodes are still moved manually, as the automatic move option might result in a node being moved away from the original boundary, particularly when the boundary and the splitting line are not approximately perpendicular to one another. Whether automatic or manual moving of nodes is chosen, each move is displayed for confirmation or cancellation.

Note that in choosing nodes to move to the splitting line, it is important that consecutive nodes moved should be connected to one another, i.e. be neighbours. An example of an improper sequence of nodes is A-B-C-D in a figure not yet available. In this case, the connection EF still crosses the splitting line after nodes A, B, C and D have been moved. On the other hand A-B-F-D or A-E-F-D would be acceptable sequences. Where one of the sub-grids output by the Splitter is going to be discarded, for example when some unwanted extension of a grid is being removed, it is normally better to move unwanted nodes to the splitting line, rather than choosing nodes that will be retained in any case. It is then easier to maintain well-shaped triangles in the remaining sub-grid. Prior to moving a node to the splitting line, the user has a choice of the following:

M - Move Node W - Windowing D - Done this side

Option W is not accessible from this menu.

When moving nodes to the splitting polygon, it is advisable to zoom in until nodes to be moved are at least 0.5 cm apart on the screen. This may involve displaying only part of the current polygon side; opportunity to zoom out again is offered at appropriate times. When ready to move a node to the current working side of the splitting polygon, choose option M (move node) from [3]. This will bring up further prompts appropriate to automatic or manual node moving. After the node has been moved, the user will be asked to confirm or cancel the move. One situation in which a move may have to be cancelled is if the wrong node is selected due to insufficient sensitivity of the cursor.

The program will correct this automatically, if the move is cancelled and the selection is repeated. There is a bug in the Splitter at present, which results in occasional failure to restore the previous display when a move is cancelled; this can be ignored, since the program in fact handles the grid file correctly. Continue moving nodes to the splitting line until connections between nodes coincide with the line along its whole length. Note that a node should be "moved" even if it originally lies on the line; otherwise the program is not informed that such a node is now a boundary node. If in doubt whether a node has been moved or not, check the colour of lines connecting it to its neighbours: all of these will be in the "modify colour" if the node has been moved. There is no harm in moving a node twice, if in doubt. The completion of a side is signalled by using option D in [3], but before doing so, it is necessary to window out, using option W, to the extent that the whole length of the current working side is visible in the display. Then when D is subsequently entered, it will always be possible to see which node prompt [5], below, refers to. When moving nodes to the current working side of the splitting polygon is complete and this has been signalled to the Splitter program by entering D in answer to prompt [3], a marker is placed at each moved node in turn and the following prompt appears:

Enter a non-zero computational code for this boundary point:

The code entered by the user will be assigned to the node in question in both sub-grid files output by the splitter. Water depth at the new location of a moved node is evaluated by linear interpolation, if it is originally an internal node. For a moved boundary node, the user has the option of leaving the depth equal to its value before the node was moved or entering a new value of depth. When codes have been entered for all nodes moved to the current working side, the following prompt appears:

F - finished M - if more sides to do

If more than one side of the splitting polygon intersects the grid, and not all such sides have had nodes moved to them, option M should be chosen. The first subsequent step is to window up, if necessary, to make sure that the whole of the next side to be worked on is visible. The user will then be led through the same procedure for the next working side as for the previous one. Eventually, when nodes have been moved to all sides of the splitting polygon which intersect the grid, choose option F, which initiates the actual splitting process.

The user is then led through a series of steps concerning display and output of the two sub-grids produced. When these have been carried out, splitting is complete.

3.10.7 Menu item RefineGrid

When this option is chosen, all triangles within the active polygon are refined.
[NEEDS UPDATE]

3.10.8 Menu item CutOffGrid

Cuts a section of the grid off, specified by the active polygon. [NEEDS UPDATE]

3.10.9 Menu item SetDepth

[NEEDS UPDATE]

3.10.10 Menu item ReDepth

Load a reference NEIGH file to apply new depth parameters to the grids nodes. This function is useful when you have altered the existing grid.

3.11 Contents of menu: Help

When the item Help is chosen, the options in Table 11 appear.

Table 11: Help menu items.

Help:
TQGG help
About

3.11.1 Menu item TQGG Help

[Needs update]

3.11.2 Menu item About

[Needs update]

4 Input and output formats

All input routines accept free format input files, i.e. data fields in each record must be separated by at least 1 space character. In general, the first line of the main files is a description of the file contents and the first 4 characters on the line define the file type (see below). Older file formats do not have the file description on the first line and the data may be arranged differently. When saving files, they are written using the latest default format which contains the file description. In this way, legacy formats are updated automatically.

The program will attempt to select the correct file format by first assessing the file delimiter (*.ngh, *.grd, *.nod, or *.xyz,) to determine the general type of file, then by reading the file type at the beginning of the file to determine

the format, and finally by parsing the file to find the correct variables. An error message is returned if this selection is unsuccessful.

There are 2 broad categories of files: Grid files that contain nodal and connection information (*.ngh and *.grd) , and Node files that contain only nodal information (*.nod, *.xyz, and *.xsc). Grid and node files are used at different stages in grid generation and can be converted back and forth with the menu item GridGen/Triangulate and GridEdit/Grid2Nodes.

4.1 Neighbour (NGH) format

4.1.1 Description

Neighbour format files can have 3 formats: default (current) format, old format, and really old (original) format. These are defined below.

The default format file can contain comment lines beginning on line 2 and are identified by a `#` character in the first position on the line. The first line after comments contains scaling and grid type information. When the file is read, the scaling factors `scaleX` and `|scaleZ|` are applied to convert to the correct units (degrees or meters). Then `scaleX` is reset to 1. and `scaleZ` is reset to `sign(scaleZ)*1`. The sign of `scaleZ` then indicates whether the z axis is positive upwards (+1.) or positive downwards (-1.). The offsets from the global coordinate system are not applied but retained in the data set for reference in later calculations.

The old file format originally contained (`xmax,ymax,xmin,ymin`) instead of (`x_origin, y_origin, scaleX, scaleZ`) on the third line. The max/min values are unnecessary and were only used for plotting purposes. They have been replaced with scaling information such as in the default format. **IMPORTANT NOTE:** Older files with max/min values **MUST** be updated to contain scaling information or the units will be nonsense.

Since the old and really old formats do not contain a grid type, the user is prompted for this when the file is read. The really old format does not contain scaling information so this must be applied by the user when editing the grid.

Data layout - default format:

filetype (string, free format)
x_origin, y_origin, scaleX, scaleZ, igrdtype, [UTMzone] (free format)
np (integer, free format)
nnb (integer, free format)
np lines of
id,x(id),y(id),code(id),z(id),(nbr(id,j),j=1,nnb)

Data layout - old format:

np (integer, free format)
nnb (integer, free format)
x_origin, y_origin, scaleX, scaleZ (free format)
np lines of
id,x(id),y(id),code(id),z(id),(nbr(id,j),j=1,nnb)

Data layout - original format:

np (integer, free format)
nnb (integer, free format)

np lines of
id,x(id),y(id),code(id),z(id),(nbr(id,j),j=1,nnb)

Definitions:

filetype (4 characters) - an indicator of the file type. For ngh files it is #NGH.
x_origin, y_origin (real) - the grid origin (0,0) in global coordinates in the x and y dimensions.
scaleX (real) - scaling for the grid in the x and y dimensions so that the resultant units are degrees (igridtype=0) or m.
scaleZ (real) - scaling of the grid in the z dimension so that the resultant unit is m directed upwards.
igridtype (integer) - the type of coordinates used by the grid. 0=(lat,long), 1=UTM, 2=m, 3=general.
UTMzone (3 characters) - optional UTM zone if igridtype=1.
np (integer) - number of points (nodes) in the grid.
nnb (integer) - maximum number of neighbours at a node.
id (integer) - index (number) of node , (id = 1, np).
j (integer) - neighbour counter, (j=1,nnb).
x(id),y(id) (real) - x,y coordinates of idth node.
code(id) (integer) - identifies the type of node (boundary,interior,etc).
z(id) (real) - the value of bottom elevation at the idth node.
nbr(j,id) (integer) - node adjacency array that contains the index of the jth neighbour of the idth node.

4.1.2 Example of data file in NGH format

```
#NGH
# Comments go here
0.0000E+00    0.0000E+00    1.0000E+00    -1.0000E+00    3
505
6
1    1.525    12.469    1    0.000    0    2    485    0    0    0
2    1.480    12.280    1    0.000    1    3    400    485    492    0
3    1.310    12.070    1    0.000    2    4    311    477    492    0
4    1.160    11.860    1    0.000    3    5    477    0    0    0
5    1.030    11.640    1    0.000    4    6    381    391    477    0
6    0.910    11.400    1    0.000    5    7    391    482    494    0
7    0.740    11.200    1    0.000    6    8    494    504    0    0
..    .....    .....    .    .....    ..    ..    ...    ...    .    .
..    .....    .....    .    .....    ..    ..    ...    ...    .    .
502    5.249    6.259    0    1.100    122    123    471    486    493    0
503    5.635    6.518    0    1.060    120    121    475    484    496    0
504    0.779    11.055    0    1.030    7    8    401    468    494    0
505    8.486    6.300    0    1.010    68    69    369    457    0    0
```

4.2 Node (NOD) format

4.2.1 Description

Node format files can have 2 formats: default (current) format and old format. These are defined below.

Data layout - default format:

filetype - (string, free format)
x_origin, y_origin, scaleX, scaleZ, igridtype, [UTMzone] (free format)
np - (integer, free format)
nb,nib - (2 integers, free format)
nbp(1) - (integer, free format)
x,y,z - (nbp(1) lines, 3 reals, free format)
:
nbp(nb) - (integer, free format)
x,y,z - (nbp(nb) lines, 3 reals, free format)
:
npi - (integer, free format)
x,y,z - (npi lines, 3 reals, free format)

Data layout - old format:

np - (integer, free format)
nb,nib - (2 integers, free format)
nbp(1) - (integer, free format)
x,y,z - (nbp(1) lines, 3 reals, free format)
:
nbp(nb) - (integer, free format)
x,y,z - (nbp(nb) lines, 3 reals, free format)
:
npi - (integer, free format)
x,y,z - (npi lines, 3 reals, free format)

Definitions:

filetype (4 characters) - an indicator of the file type. For node files it is #NOD
x_origin, y_origin (real) - the grid origin (0,0) in global coordinates in the x and y dimensions.

scaleX (real) - scaling for the grid in the x and y dimensions so that the resultant units are degrees (igridtype=0) or m.

scaleZ (real) - scaling of the grid in the z dimension so that the resultant unit is m directed upwards.

igridtype (integer) - the type of coordinates used by the grid. 0=(lat,long), 1=UTM, 2=m, 3=general.

UTMzone (3 characters) - optional UTM zone if igrdtype=1.
np (integer) - total number of nodes
nb (integer) - total number of boundaries.
nib (integer) - number of internal boundaries (line constraints).
nbp(i) (integer) - number of nodes on ith boundary.
x,y,z (real)- x,y co-ordinates and bed elevation at node.
npi (integer) - number of internal nodes.

NOTES: - outer boundary nodes must be all in one block and must be the first boundary.
- outer boundary must be in counter clockwise order.
- all inner boundaries (islands) must be in clockwise order.
- all internal boundaries must be last in the boundary list.

4.2.2 Example of data file in NODE format

```
#NOD
# Comments go here
0.000000 0.000000 1.000000 1.000000 0
479
2 0
146
1.57      13.47      0.00
1.46      13.24      0.00
1.44      13.00      0.00
:         :         :
2.70      13.24      0.00
2.30      13.30      0.00
1.90      13.40      0.00
41
4.09      8.08       0.00
3.92      8.27       0.00
3.70      8.44       0.00
:         :         :
4.32      7.74       0.00
4.19      7.97       0.00
292
4.29      0.61       5.00
4.17      1.01       5.00
4.06      1.41       5.00
:         :         :
2.61      11.90      30.00
2.61      12.30      30.00
2.66      12.70      30.00
```

4.3 Element (ELE) format

4.3.1 Description

The element files can have 2 formats: one with 3 integers per line that is suitable for triangle elements only, and one with 5 integers per line that is suitable for quadrilateral or triangular elements. If using the latter, the 4th vertex number is 0 for a triangle. The program will parse the first line to determine the correct format to read.

Data layout - default format:

vertex 1, vertex 2, vertex 3, vertex 4, tcode - (5 integers, free format)

Data layout - triangle format:

vertex 1, vertex 2, vertex 3 - (3 integers, free format)

Definitions:

vertex 1 to vertex4 - node index for element vertices in CCW order.

4.3.2 Example of data file with default element format

1	2	48	0	1
1	48	41	0	1
2	3	26	0	1
2	26	29	0	1
.
.
32	37	36	0	2
34	45	35	0	2
34	35	46	0	1

4.3.3 Example of data file in triangle only format

1	2	48
1	48	41
2	3	26
.	.	.
.	.	.
32	37	36
34	45	35
34	35	46

4.4 Grid (GRD) format

Grid (GRD) format files contain both node location and element vertex information. Node adjacency lists are generated when the file is read. This is different than neighbour files which contain node adjacency information and the element list is generated when they are read.

4.4.1 Description

The format for grid (GRD) files has 2 forms. The default format contains a file type identifier and origin and scaling information. The simple format just reads the number of nodes and elements, then reads a list of node locations followed by a list of element vertices.

Data layout - default format:

filetype (string, free format)
x_origin, y_origin, scaleX, scaleZ, igrdtype, [UTMzone] (free format)
np,ne - (2 integers, free format)
x,y,z - (np lines, 3 reals, free format)
OR data layout with node codes:
x,y,z,code - (np lines, 3 reals, 1 integer, free format)
vertex 1, vertex 2, vertex 3, vertex 4, tcode - (ne lines, 5 integers,free format)
OR simple layout for triangles and no Tcodes:
vertex 1, vertex 2, vertex 3 - (ne lines, 3 integers, free format)

Data layout - simple format:

np,ne - (2 integer, free format)
x,y,z - (np lines, 3 reals, free format)
OR data layout with node codes:
x,y,z,code - (np lines, 3 reals, 1 integer, free format)
vertex 1, vertex 2, vertex 3, vertex 4, tcode - (ne lines, 5 integers,free format)
OR simple layout for triangles and no Tcodes:
vertex 1, vertex 2, vertex 3 - (ne lines, 3 integers, free format)

Definitions:

filetype (4 characters) - an indicator of the file type. For grid files it is #GRD
x_origin, y_origin (real) - the grid origin (0,0) in global coordinates in the x and y dimensions.

scaleX (real) - scaling for the grid in the x and y dimensions so that the resultant units are degrees (igrdtype=0) or m.

scaleZ (real) - scaling of the grid in the z dimension so that the resultant unit is m directed upwards.

igrdtype (integer) - the type of coordinates used by the grid. 0=(lat,long),

1=UTM, 2=m, 3=general.

UTMzone (3 characters) - optional UTM zone if igrdtype=1.

np - number of points (nodes) in the grid.

ne - number of elements in the grid.

x,y,z - coordinates for x,y and bed elevation.

code - node code for boundary conditions and other purposes.

vertex 1 to vertex4 - node index for element vertices in CCW order.

TCode - element code for assigning friction type.

4.4.2 Example of data file in default GRD format

```
#GRD
# Comments go here
0.000000 0.000000 1.000000 1.000000 0
    477    795
1.5700001E+00 1.3470000E+01 0.0000000E+00 1
1.4299999E+00 1.3180000E+01 0.0000000E+00 1
    .
    .
    .
5.3130002E+00 1.1940000E+00 1.5000000E+01 0
5.5460000E+00 7.5000000E-01 1.5000000E+01 0
    1      2      204      0      1
    1      204    126      0      1
    2      3      204      0      1
    3      4      203      0      1
    .
    .
    .
    .
468      471      472      0      1
469      470      471      0      1
```

4.4.3 Example of data file in simple GRD format

```
    477    795
1.5700001E+00 1.3470000E+01 0.0000000E+00 1
1.4299999E+00 1.3180000E+01 0.0000000E+00 1
    .
    .
    .
5.3130002E+00 1.1940000E+00 1.5000000E+01 0
5.5460000E+00 7.5000000E-01 1.5000000E+01 0
    1      2      204      0      1
    1      204    126      0      1
    2      3      204      0      1
    .
    .
    .
    .
468      471      472      0      1
469      470      471      0      1
```

4.5 Node (XYZ) format

This format provides a general method to import nodal data from other datasets such as chart data and shoreline data. It is used in the sampling routines invoked by the menu option File/Sample.

4.5.1 Description

The input routines will parse the file to determine the format and variables present, then read to the end of the file to determine the number of data points. Three types of formats are supported. In the first, only the x,y coordinates of the point are read which is primarily used for shoreline data. In the second, x,y coordinates and bed elevation of a point are read which applies to boundary data with depth information. In the third, the data is read as segments in pairs of data points. When the segment identifier is the same, the segments will be joined together. Otherwise, the segments can be manipulated with editing tools.

Data layout:

x,y - (2 reals, free format)
OR with full coordinates
x,y,z - (3 reals, free format)
OR multi-segement with segment number
x,y,z,s - (3 reals, 1 integer, free format)

Definitions:

x,y,z - x,y coordinates and bed elevation.

s - segment number such that all segments where s is the same are joined together.

4.5.2 Example of data file in XYZ format

1.5700001E+00	1.3470000E+01	0.0000000E+00
1.4299999E+00	1.3180000E+01	0.0000000E+00
1.4800000E+00	1.2880000E+01	0.0000000E+00
1.5300000E+00	1.2580000E+01	0.0000000E+00
1.4800000E+00	1.2280000E+01	0.0000000E+00
1.2700000E+00	1.2020000E+01	0.0000000E+00
1.0900000E+00	1.1750000E+01	0.0000000E+00
9.4999999E-01	1.1460000E+01	0.0000000E+00
.	.	.
.	.	.

4.6 Section (XSC) format

4.6.1 Description

[THIS SECTION NOT YET UPDATED]

Data layout:

nxp
Next nxp pairs of lines:
ns,xl,yl,xr,yr,xref
((distr(j), depth(j)) j=1,ns)

Definitions:

nxp - number of cross-sections to be input.
ns - number of points in the input cross-sections.
xl,yl - (x,y) coordinates of left bank, looking downstream.
xr,yr - (x,y) coordinates of right bank, looking downstream.
zref - elevation reference for cross-section.
distr - distance from right bank.
depth - depth below zref at point distr.

4.6.2 Example of data file in XSEC format

```
21
9 -6.06    3.50    6.06   -3.50    9.15 0. 0. 2.00 1.2 2.50 1.5 3.00 1.8 3.33
2. 4.06 2. 4.80 2. 5.53 2. 6.27 2.
9  3.20    17.48    13.69    8.22    9.145 0. 0. .77 .45 1.54 .9 2.33 1.35 3.11
1.8 3.88 2.25 4.67 2.25 5.44 2.15 6.22 2.0
9 18.51    28.48    21.72    19.00    9.14 0. 0. .77 .45 1.54 .9 2.33 1.35 3.11
1.8 3.88 2.25 4.67 2.5 5.44 2.5 6.22 2.05    .    .    .    .
.
9 206.79   -18.15   209.47   -27.64    9.06 0. 0. .77 .45 1.54 .9 2.33 1.35 3.11
1.39 3.88 1.43 4.67 1.47 5.44 1.51 6.2 1.5
9 214.74   -8.04   225.23   -17.31    9.055 0. 0. .77 .45 1.54 .9 2.33 1.35 3.11
1.45 3.88 1.55 4.67 1.65 5.44 1.75 6.2 1.8
```