Preprocessor Janet

Grid Generation

---

for UnTRIM

smile consult GmbH
www.smileconsult.de

Version 1.0 (19.5.2005)

Hannover, May 2005 Christoph Lippert

smile consult GmbH
Vahrenwalder Strasse 7
30165 Hannover

Fon 0511/9357–620
Fax 0511/9357–629

lippert@smileconsult.de
www.smileconsult.de

Introduction

1

The following document describes the methods and strategies implemented in the preprocessor Janet for generating, analyzing and optimizing unstructured orthogonal grids.

The second chapter deals with the basics of unstructured orthogonal grids with focus on the UnTRIM model. Basic properties such as the definition and calculation of the centerpoints are presented. Furthermore, quality measurements for unstructured orthogonal grids are defined. The chapter finally focuses on basic operations on unstructured orthogonal grids for UnTRIM.

The next chapter illustrates the grid generation techniques implemented in the preprocessor. The different grid generation strategies are presented by various examples.

Grid optimization with focus on optimizing grids for orthogonality is shown in the forth chapter. This chapter also centers on shape and patch optimization.

Basics

2

2.1 Unstructured Orthogonal Grids

The horizontal computational domain (x,y) must be covered with a set of non–overlapping convex polygons. Each side of a polygon is either a boundary line or a side of an adjacent polygon. Moreover, it is assumed that within each polygon there exists such a point (hereafter called a *center*) that the segment joining the centers of two adjacent polygons and the side shared by the two polygon have a non empty intersection and are orthogonal to each other (Validation Document UnTRIM 2004, BAW)

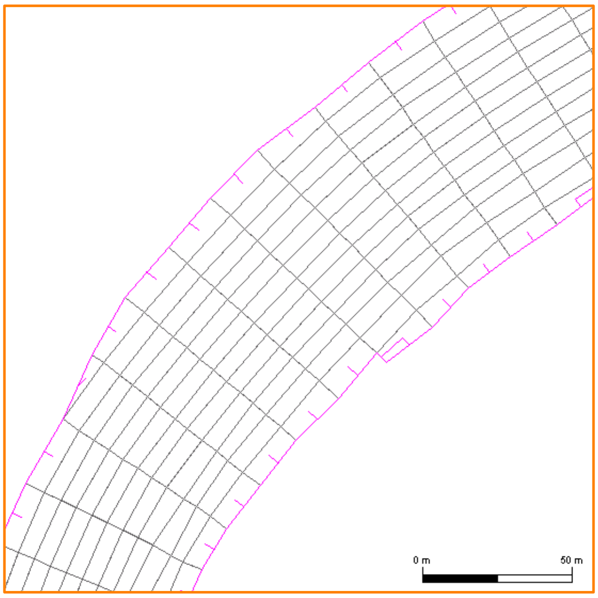


Figure 2-1. Example of an unstructured orthogonal grid

The specific geometric properties of two adjacent polygons are shown in the following figure.

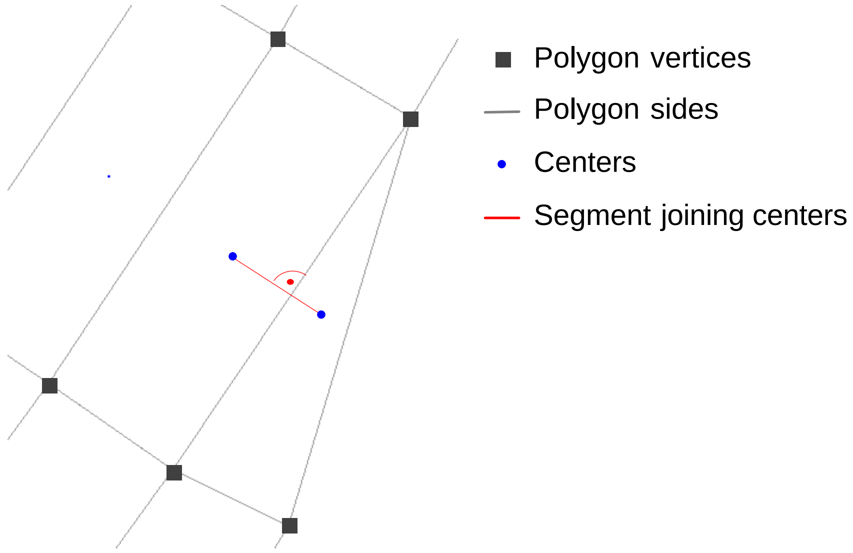


Figure 2-2. Some key terms of unstructured orthogonal grids

2.1.1 Center Point Calculation

An unstructured grid consisting of three and foursided polygons is strictly orthogonal if for each polygon its vertices are located on the shared circumcircle. The center of each polygon is therefore calculated as the center of the circumcircle.

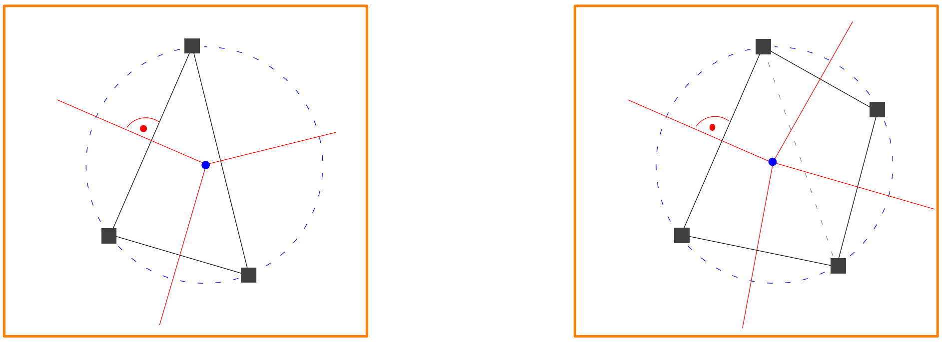


Figure 2-3. Strict orthogonal polygons

Further, the resulting center must be located within the polygon as shown in Figure 2-3. For Polygons that do not fulfill these criterias the center point is recalculated using the following two rules:

- Center point is outside the polygon:
  The center is moved to the intersection point of the “violated” polygon side and the orthogonal straight line.

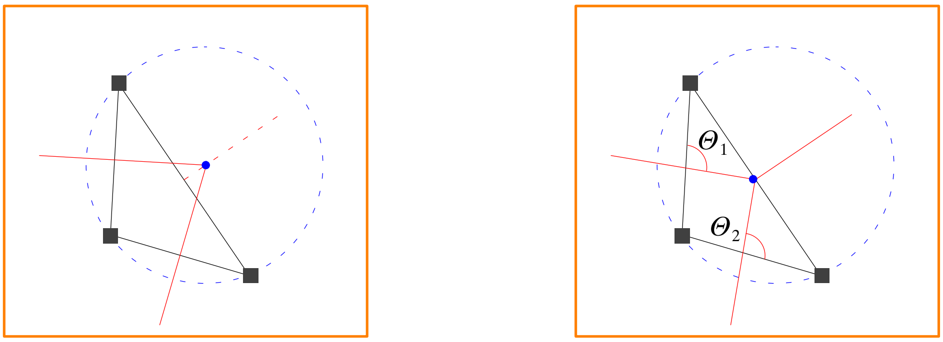


Figure 2-4. Approximated center for a non-orthogonal triangle

- Polygon vertices are not located on a shared circumcircle:
  This case is limited to foursided polygons. The center is recalculated as the geometric mean of the two circumcircles given by three vertices of the quadrilateral polygon.

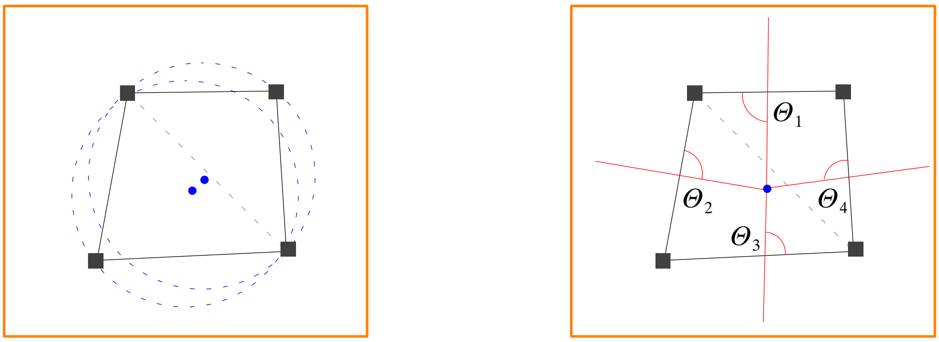


Figure 2-5. Recalculated center for a foursided polygon

The recalculation leads to a deviation from orthogonality indicated by the angles Θ_1 ... Θ_n as shown in the figures. The deviation from orthogonality can be calculated by $|90^\circ - \Theta_1|$ ... $|90^\circ - \Theta_n|$.

2.1.2 Grid Quality Measurements

Deviation from Orthogonality

The deviation from orthogonality can be measured by the maximum deviation (Max DO) and the mean deviation (Mean DO) for a grid. Furthermore the distribution of the deviation across all polygon sides is an indicator for grid quality.

$$Max\ DO = \max\left(\left|90^\circ - \Theta_{i,j}\right| \right)$$

$$Mean\ DO = \frac{1}{ns} \sum \left(\left|90^\circ - \Theta_{i,j}\right| \right)$$

where ns is the number of polygon sides of the entire grid.

Shape Parameter

The shape parameter indicates a polygon’s shape compared to an optimal shaped triangular or quadrilateral polygon. It is calculated as the sum of the deviation from the optimal angle for a single polygon.

$$SH_{Triangle} = \left(\left(\sum |\alpha_i - \Pi/3|\right)/3\right)/1.3963$$

$$SH_{Quad} = \left(\left(\sum |\alpha_i - \Pi/2|\right)/4\right)/1.5708$$

The range of the shape parameter is between 0.0 for an optimal shaped polygon and 1.0 for a degenerated polygon. For the entire grid a maximum (Max SH) and a mean shape parameter (Mean SH) value are defined as follows:

$$Max\ \ SH = \max(SH_i)$$

$$Mean\ \ SH = \frac{1}{np}\sum(SH_i)$$

where np is the number of polygons in the grid.

Center Distance DX and Centerpoint-Edge-Ratio DX/L

Very small distances between center points may lead to numerical instabilities and may result in a large computational effort to solve the system of equations. The analysis of the minimum center distance

$$Min\ \ DX = \min(DX_i)$$

helps to optimize grid quality with respect to these aspects. In cases of adjacent triangles, a very small center distance can be avoided by replacing adjacent triangles by quadrilaterals. Therefore, the ratio of the center distance to the length of the shared polygon side indicates that the vertices form a quadrilateral polygon. The ratio is defined as

$$R = \frac{d_{Centers}}{L_{side}} = \frac{DX}{L}$$

and is used to algorithmically merge triangles to quadrilaterals.

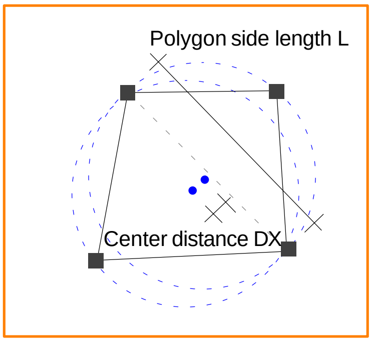


Figure 2-6. Center distance to polygon side length ratio

Ratio of a Polygons’s Minimum and Maximum Center Distances

The analysis of a polygon’s minimum and maximum center distance can be applied by the ratio

$$R_{centers} = \frac{DX_{\min}}{DX_{\max}}$$

The detection of polygons with very small ratios may also help to avoid numerical instabilities. A gradual change in centerpoint distances DX indicated by higher ratios should be achieved especially for triangular polygons. In case of flow aligned quadrilaterals small ratios can be allowed.

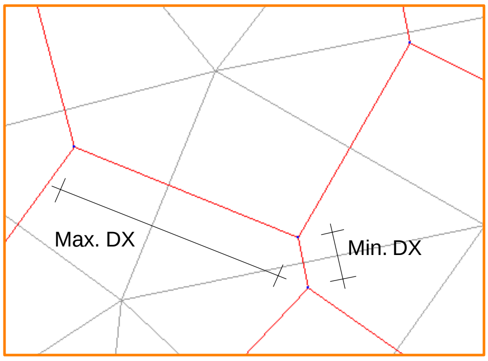
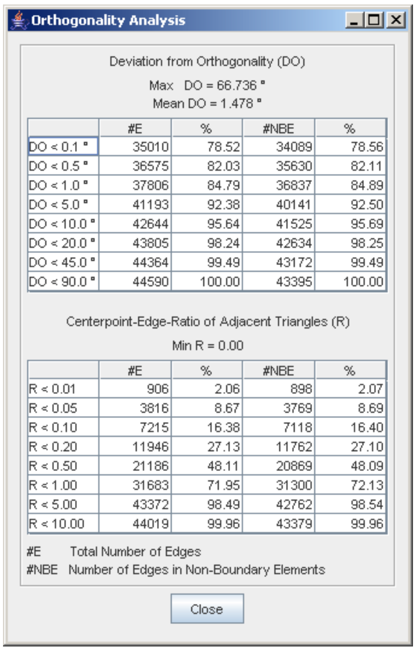


Figure 2-7. Minimum and Maximum DX of a triangle

2.1.3 Strict Orthogonal Grids

The Preprocessor Janet is designed to support grid generation for unstructured orthogonal with special respect to the orthogonality criterium. A special analysis function helps to evaluate grids concerning strict orthogonality. The above mentioned deviation from orthogonality and the centerpoint-to edge-ratio are calculated for each polygon side of a specific grid. Besides the maximum and mean deviation and the minimum ratio, a predefined set of interval bounds for the deviation and ratio are tested and the number of polygon sides fulfilling these criteria are listed (Figure 2-8). A grid is regarded as strictly orthogonal if 100% of all centerpoint connections have a deviation from orthogonality across the shared polygon side below 0.1°.


Orthogonality Analysis

Deviation from Orthogonality (DO)

Max DO = 66.736 °

Mean DO = 1.478 °

	#E	%	#NBE	%
DO < 0.1 °	35010	78.52	34089	78.56
DO < 0.5 °	36575	82.03	35630	82.11
DO < 1.0 °	37806	84.79	36837	84.89
DO < 5.0 °	41193	92.38	40141	92.50
DO < 10.0 °	42644	95.64	41525	95.69
DO < 20.0 °	43805	98.24	42634	98.25
DO < 45.0 °	44364	99.49	43172	99.49
DO < 90.0 °	44590	100.00	43395	100.00

Centerpoint-Edge-Ratio of Adjacent Triangles (R)

Min R = 0.00

	#E	%	#NBE	%
R < 0.01	906	2.06	898	2.07
R < 0.05	3816	8.67	3769	8.69
R < 0.10	7215	16.38	7118	16.40
R < 0.20	11946	27.13	11762	27.10
R < 0.50	21186	48.11	20869	48.09
R < 1.00	31683	71.95	31300	72.13
R < 5.00	43372	98.49	42762	98.54
R < 10.00	44019	99.96	43379	99.96

#E Total Number of Edges
#NBE Number of Edges in Non-Boundary Elements

Close

Figure 2-8. Example of an orthogonality analysis report

Further, the function distinguishes between the total number of polygons and non-boundary polygons. This distinction supports a more detailed analysis of a grid. The creation of very accurate non-boundary polygons and boundary polygons with a slightly higher deviation is a possible application. This allows a better boundary fitting for more complex boundary geometries.

2.2 Depth Approximation in the UnTRIM Model

Depth values are assigned to each polygon side. Each side is assumed to have a constant depth value for the entire side. The depth of the polygon is defined as the maximum depth of all polygon sides.

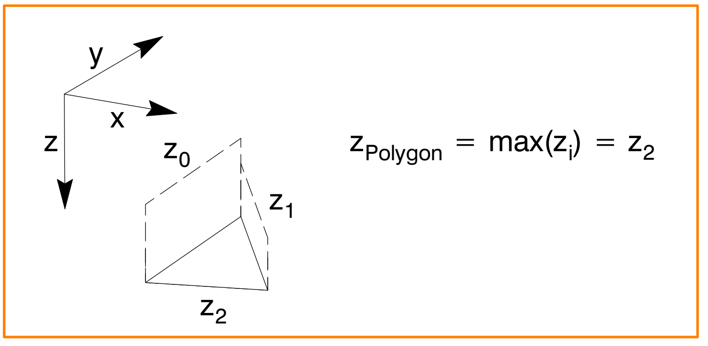


Figure 2-9. Depth Approximation for polygon sides in the UnTRIM model

2.2.1 Interpolation on Polygons

A point (x,y) inside a polygon is interpolated by detecting the nearest polygon side and assigning the side's depth value.

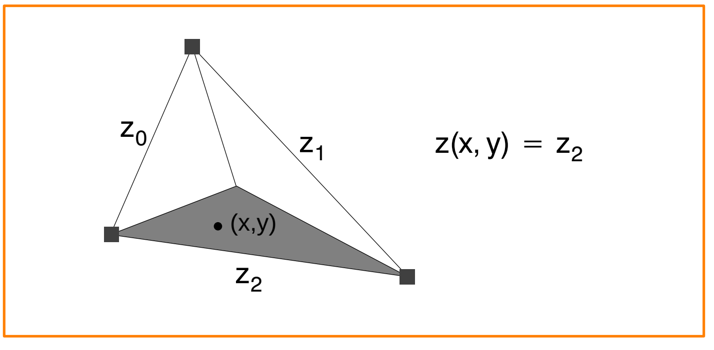


Figure 2-10. Interpolation scheme

2.2.2 Volume Calculation for a Polygon

The volume of a single polygon below a horizontal layer Z_H is calculated by

$$V = (Z_{Polygon} - Z_H) \cdot A$$

where $Z_{Polygon}$ is the maximum side depth and A the polygons's area.

2.3 Basic Operations for UnTRIM Grids

2.3.1 Mapping Depths from a Digital Terrain Model to an UnTRIM Grid

This basic operation assigns depth values derived from a Digital Terrain Model to the polygon sides. A representative location on each side is used for the interpolation. The location is defined as the intersection point of the segment joining the centers and the shared polygon side.

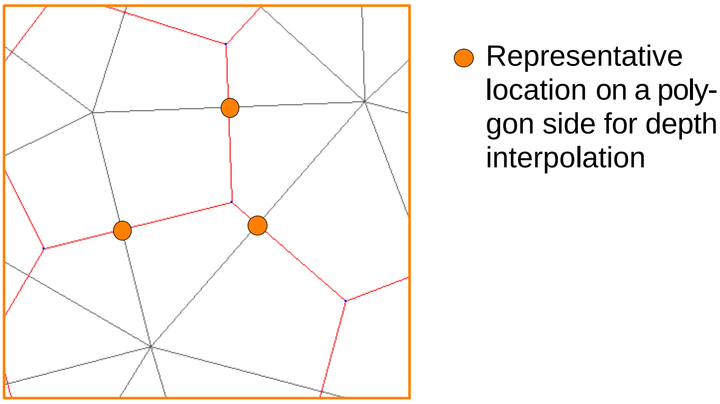


Figure 2-11. Interpolation points for polygon sides

A Digital Terrain Model is defined as bathymetric data combined with an interpolation method. Various interpolation and approximation methods are offered, such as:

- Bivariate Interpolation
- Nearest Neighbour
- Natural Neighbour (Sibson)
- Inverse Distance Interpolation (Shepard)
- Mean, minimum, maximum and median depth of all data points within a specific distance to the interpolation point
- Mean, minimum, maximum and median depth of all data points within an area as shown in Figure 2-12

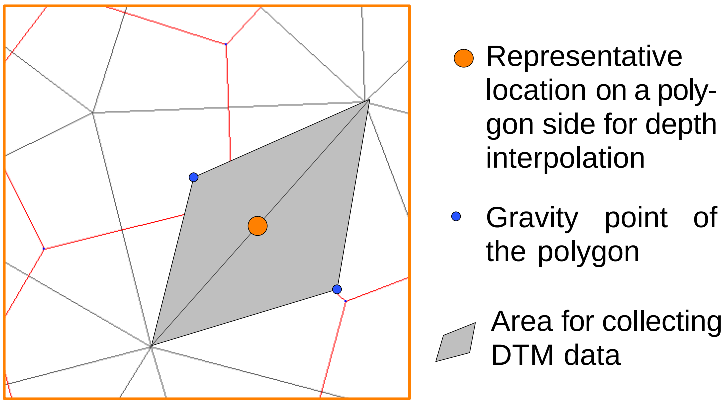


Figure 2-12. Interpolation points for polygon sides

In addition to the depth interpolation on a Digital Terrain Model, the resulting depths can be recalculated with an algorithm developed by Prof. Casulli in cooperation with the BAW. The recalculation leads to a smoother bathymetry in order to reduce artificial damping in case of three-dimensional computations (terrace shaping algorithm).
The effect of the terrace shaping algorithm is illustrated by the following example of an idealized river profile.

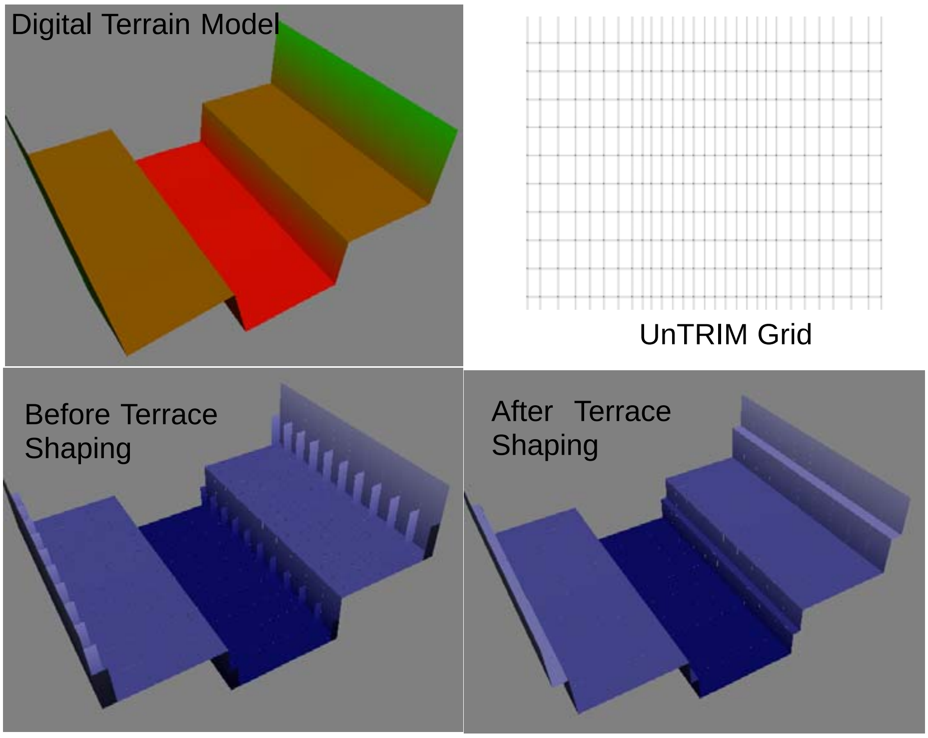


Figure 2-13. Effect of the terrace shaping algorithm

2.3.2 Generating a Grid with depth differences for two UnTRIM Grids

For each polygon side of the two grids the depth difference is calculated at the gravity point of the side. The calculation of the difference value is done by finding the polygon containing the difference node and interpolating the

depth on each grid, as described in chapter 2.2.1.
The resulting grid with depth differences is computed by meshing all the vertices where differences in depth have been calculated.

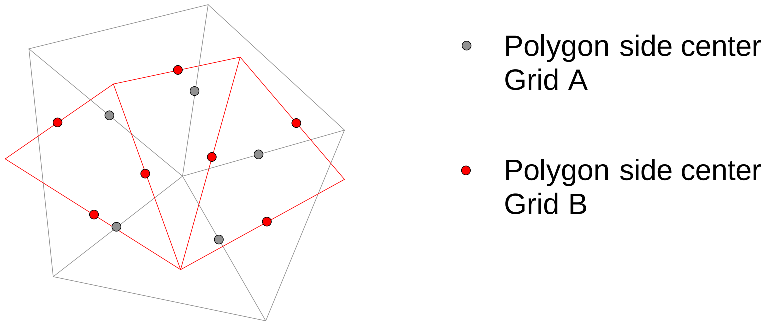


Figure 2-14. Scheme for generating depth differences

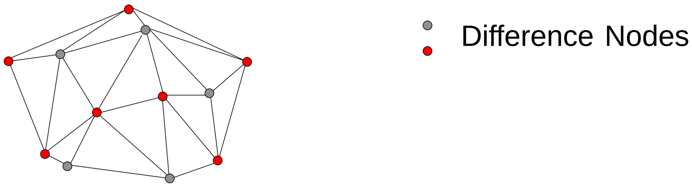


Figure 2-15. Resulting grid with depth differences

2.3.3 Generating the Volume of an UnTRIM Grid to a horizontal Z-Layer

The volume calculation for an UnTRIM Grid to a horizontal Z-Layer Z_H is computed by

$$V = \sum\left(Z_{Polygon_i} - Z_H\right) \cdot A_i$$

where $Z_{Polygon_i}$ is the maximum side depth of polygon i and A_i the specific polygon’s area.

Grid Generation for UnTRIM

3

3.1 Grid Generation Concept

The Preprocessor Janet is designed to support grid generation for different numerical models. The basic concept is illustrated in the following figure.

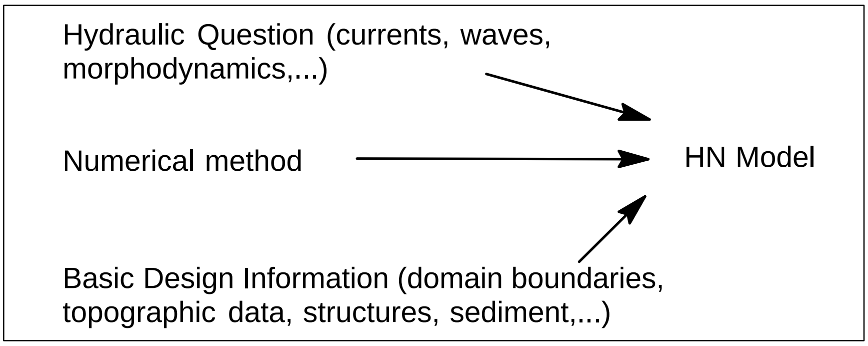


Figure 3-1. Basic concept for HN Model setup

This concept implies that different strategies and techniques for grid generation are necessary to meet the requirements of the numerical models, the question to be answered and moreover to meet the complexity of the domain for which the model is setup.
Transferred to a software concept, main software components can be identified by

- Digital Models to represent the basic design information
- Different methods for grid generation that rely on the basic design information
- Coupling sub grids of different grid structure to setup complex unstructured models
- Analyzing and optimizing the model
- Export to a model's specific file format

The preprocessor Janet supports this approach by numerous Digital Models such as Digital Terrain Models, Digital Structure Models, Density Functions, etc. These models serve as basic input for the grid generation modules, which support the following grid generation techniques:

- Refinement based grid generation
- Quad grid generator
- Structured grids: finite difference grids, curvilinear grids (in progress)

The different grid generation strategies enable the definition of sub domains with varying grid structures. These sub grids are coupled to an entire unstructured grid with the preprocessor's sub grid module which allows splitting, merging and copying sub grids.
Various analysis functions allow a detailed assessment of the model. In connection with the grid analysis, different optimization methods help to improve the grid for specific properties.

3.2 Refinement Based Grid Generation

3.2.1 Refinement Methods

The refinement based grid generation follows the approach of succesively inserting new vertices into an existing grid. The preprocessor supports various refinement strategies which support different tasks.

Barycentric Refinement
This approach is restricted to triangles. A new vertex is inserted at the gravity point of an existing triangle. The triangulation is therefore updated by replacing the triangle by three new triangles. After insertion, the new triangles are checked for the delaunay criterium.

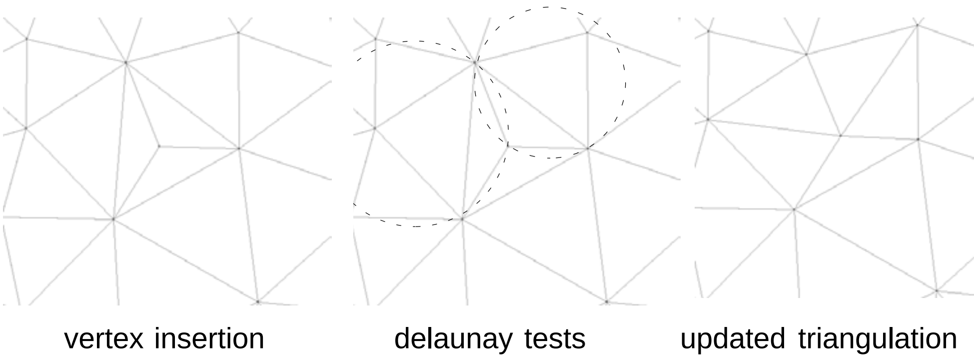


Figure 3-2. Barycentric refinement

This refinement strategy is well suited for local refinements of triangular grids. It allows a smooth gradual change in polygon sizes.

Natural Refinement
This strategy is used for quadrilaterals. The natural refinement splits a quadrilateral into four quadrilaterals. The basic properties of the replaced quad (e.g. orthogonality) are preserved.

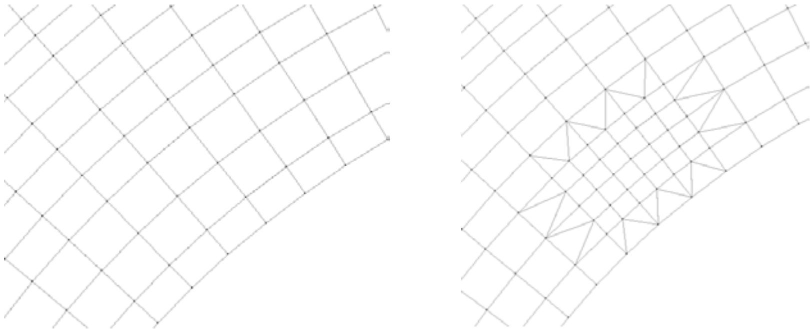


Figure 3-3. Natural refinement of quadrilaterals

This refinement strategy can be applied to flow aligned quad grids for local refinement.

Circumcenter Refinement
A triangle is refined by inserting a new vertex at the location of the triangle's circumcenter (the center of its circumcircle). The new vertex need not be located inside the triangle. This refinement approach is used for a "minimum angle refinement" approach as described by Jonathan Richard Shewchuk. This approach enables grid generation with a user defined angle that restricts the minimum angle of all triangles in a grid.

Raster Refinement
This refinement method inserts vertices of a structured grid in an unstructured triangulation. Different options for this refinement strategy are available which mainly differ in the alignment of the vertices. The alignment either leads to equilateral triangles (Figure 3-4) or the vertices are aligned to a raster. The refinement method allows nesting of rasters of different widths.

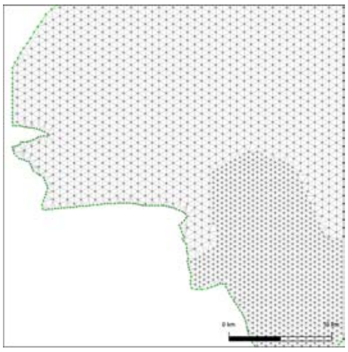
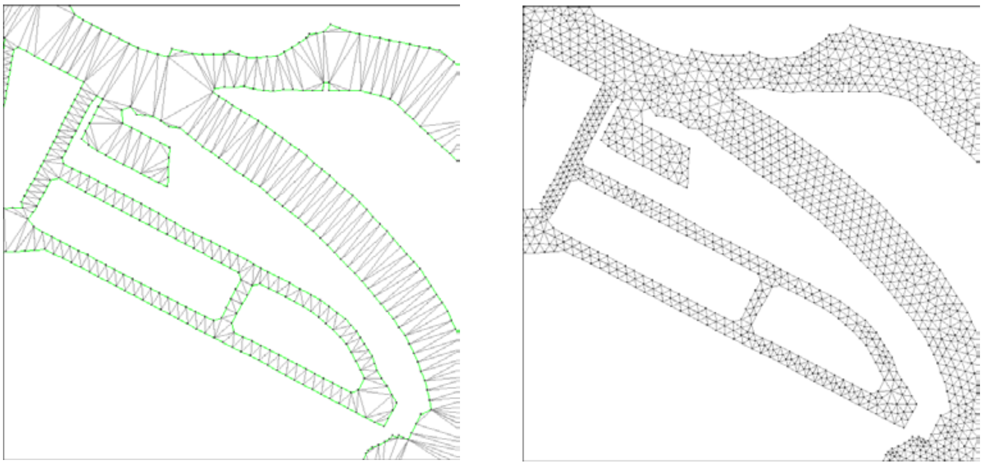


Figure 3-4. Unstructured triangular grid with nested raster refinements

Advancing Front Refinement
Beginning with an initial set of boundary segments, the advancing front refinement constructs almost equilateral triangles towards the interior of the domain. The newly generated triangles form the new boundary segments for the "advancing front". The refinement ends if the grid is covered with almost well shaped triangles.



initial grid (triangulated boundary polygon) refined grid

Figure 3-5. Triangular grid generated with an advancing front refinement

3.2.2 Criterion Based Refinement

The criterion based refinement combines a refinement method with an additional condition. The refinement of a triangle is only applied if the condition is fulfilled for the specific triangle. The preprocessor Janet offers a set of predefined criterias, among these are:

Depth Difference Criterion

This criterion is based on the barycentric refinement method and uses the gravity point for calculating a depth difference between a Digital Terrain Model and the depth interpolated by bivariate interpolation on the triangle itself. If the calculated value is above a user defined maximum absolute or relative difference, the refinement is proceeded.

The application of this refinement criterion leads to an increased grid resolution in areas of steep bathymetric gradients and is well suited to optimize the depth approximation of a grid in comparison to the Digital Terrain Model.

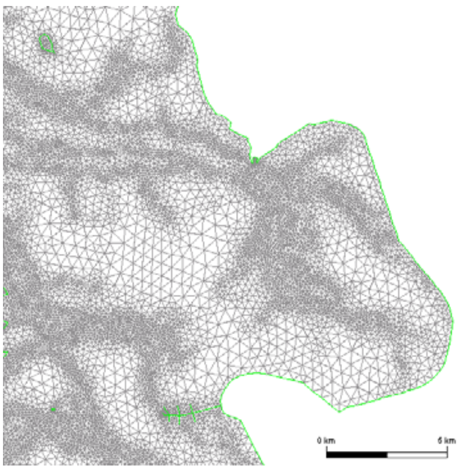


Figure 3-6. Triangular grid refined with depth difference criterion

Minimum Angle Criterion
This criterion is based on the circumcenter refinement method. It refines a triangle if the minimum angle of a triangle falls below a user defined value. Refinement with this criterion is suggested to generate an inital elementation for a grid that is derived by triangulation of the boundary polygons.

Edge Length Criterion
This criterion is based on the barycentric refinement method. The criterion condition tests all edge lengths of a triangle. The refinement is proceeded if the maximum length is above a user defined value. The refinement strategy can be used to limit the maximum vertex distances of a grid.

3.2.3 Application of the Refinement Based Grid Generation

The refinement based grid generation with the preprocessor Janet is proceeded in several basic steps. Although not all of the steps presented are obligatory for a specific task, the following basic steps can be distinguished:

1. Provide a Suitable Digital Terrain Model

2. Setup the Domain Boundary
The boundary polygon of the grid is created using the prepocessor’s polygon editor. The boundary polygon can either be composed of imported GIS data or manually defined.

raw polygon data — created boundary polygon

Figure 3-7. Creation of a boundary polygon

3. Select and Setup Polylines as Geometric Constraints
The preproceesor Janet allows the definition of geometric constraints which define fixed vertex and polygon side positions. These constraints are integrated into a grid by polylines. The next step of the grid generation process is the selection and modification of these polylines.

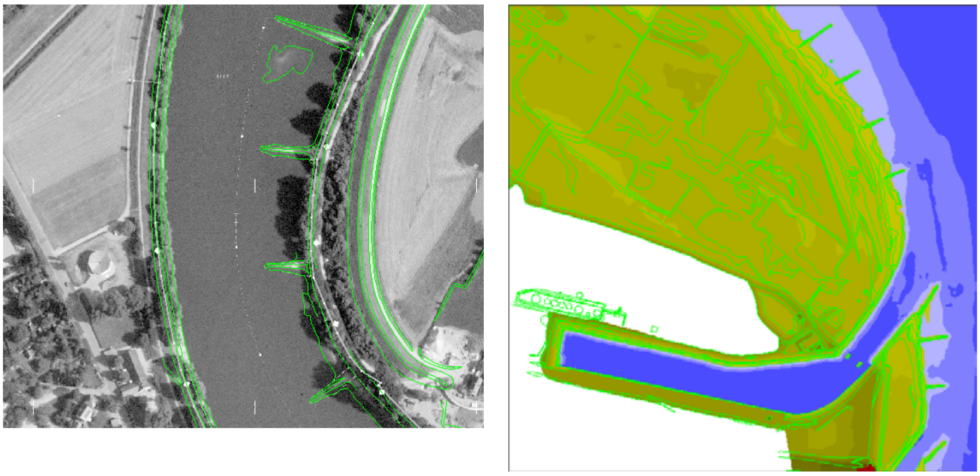


Figure 3-8. Selection of polylines with the help of additional information (georeferenced images, a Digital Terrain Model)

4. Adjust Resolution of Selected Polylines

A suitable resolution for the domain boundary and the selected polylines that serve as geometric constraints is created with the functions of the polygon editor. The functions contain simplification of polylines, algorithmically refinement and coarsening and furthermore interactive modification.

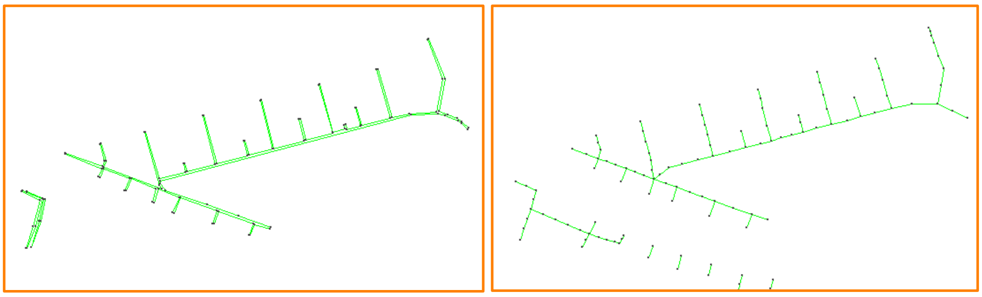


Figure 3-9. Modified polylines

5. Define Sub-Domains

Sub-domains such as fairways with special local requirements can be defined, created and integrated in the over-all model.

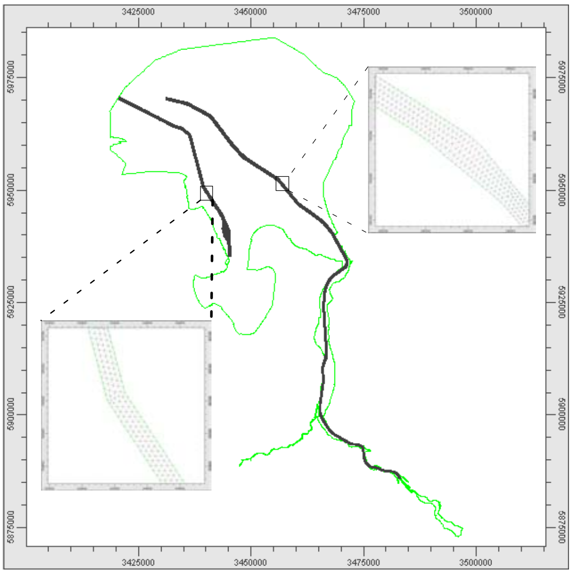


Figure 3-10. Sub-domains of the Jade-Weser-Estuary

5. Create the Inital Triangulation

An initial triangulation is generated with the preprocessor's triangulation methods. The polylines that were created as geometric constraints are integreted via constrained polygon sides ("Constrained Delaunay Triangulation").

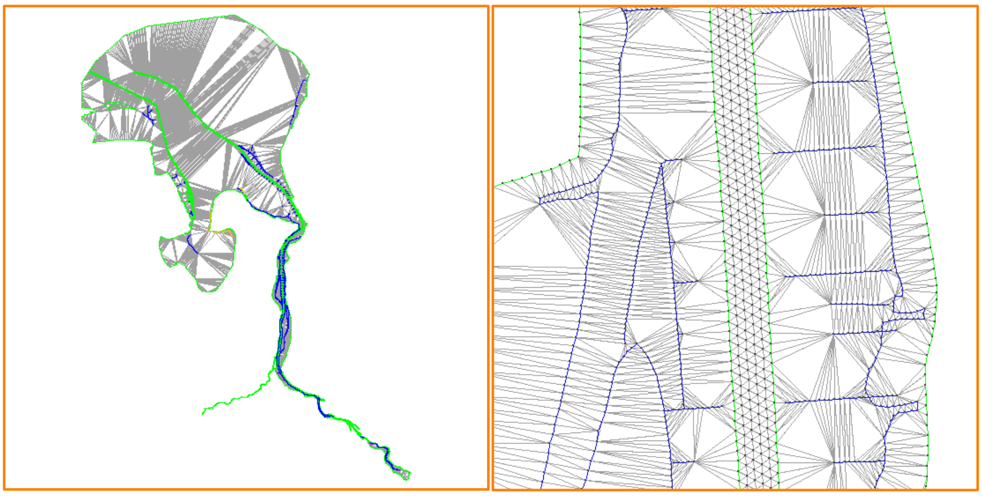


Figure 3-11. Initial triangulation of the Jade-Weser-Estuary

6. Criterion Based Refinement

A further refinement is generated using the refinement methods. The application of different refinement criterions ensures that a suitable grid resolution is computed.

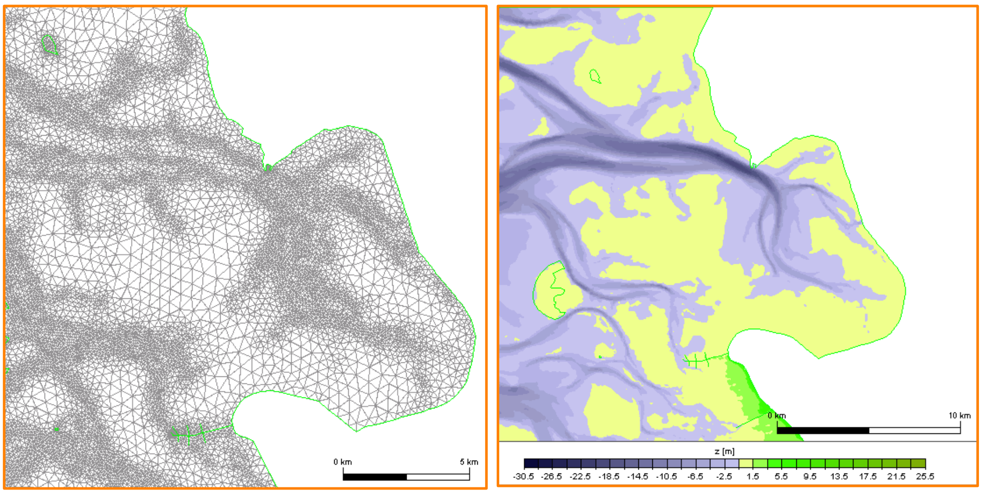


Figure 3-12. Refined grid with contour plot

3.3 Quad Grid Generator

The quad grid generator is designed to allow a more constructive approach to grid generation. Basic fields of application are the generation of boundary approximating flow aligned grids for narrow channels, the creation of flow aligned sub-domains and the generation of sub-grids for structures (e.g. dykes).

Basic input data for this grid generation method is given by a polyline that is used for aligning the quadrilaterals and several characteristic “profiles” where the user defines the widths of each row of quadrilaterals. This approach is supported by a “Quad Grid Model” that allows the user to define, edit, load and save this model. The quad generation algorithm uses this model and optionally a boundary polygon and a Digital Terrain Model for the grid generation process.

3.3.1 Grid Generation for a Narrow Channel

The first example shows basic steps to generate a grid for a narrow channel. Special requirements for the modelling are often given by discretizing the channel with a constant number of polygons per cross section and allowing asymmetric profiles for a better alignment of the polygons to the isobaths of the Digital Terrain Model. Furthermore, sharp curvatures might have to be considered during the grid generation process.

The design steps for the quadrilateral grid are as follows:

1. Provide a Suitable Digital Terrain Model

2. Provide a Domain Boundary

The boundary polygon for the grid is imported from GIS data or manually defined.

3. Setup the Alignment Polyline

The polyline that is used to align the quadrilaterals is created with the preprocessor’s polygon editor. Various automatic functions help to modify the polyline to the user's needs. The resolution of the polyline in flow direction defines the side lengths of the quadrilaterals in this direction.

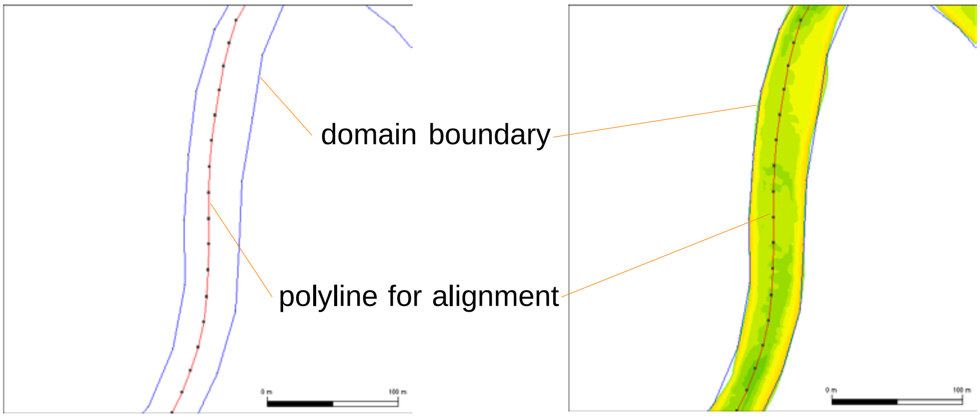


Figure 3-13. Alignment polyline, domain boundary and contour plot of the DTM

4. Running the Quad Grid Generator

The quad grid generator is started with the options "boundary approximation" and "5 polygons per cross section". The generated quad grid is shown in the next figure.

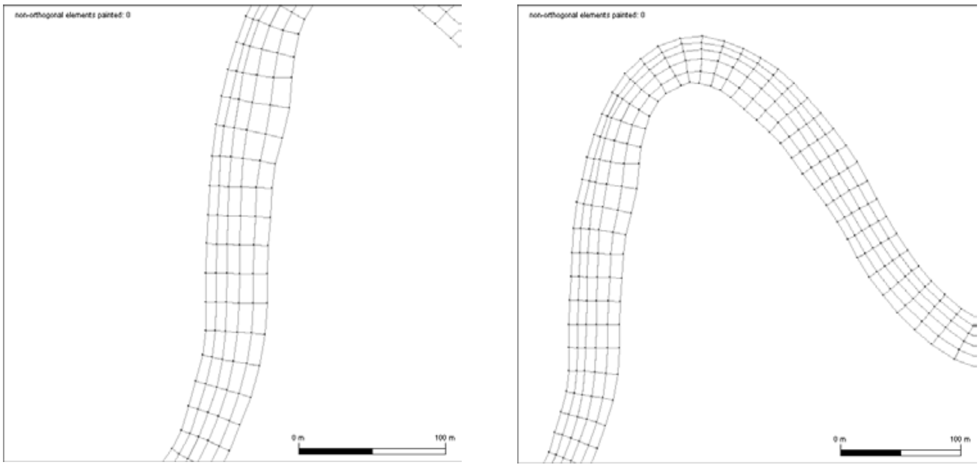


Figure 3-14. Details of the generated quad grid

The presented procedure might be restricted by further requirements. One of these is given by mapping groines to the qudrilateral grid. A suitable modification of the model's alignment polyline supports this task.

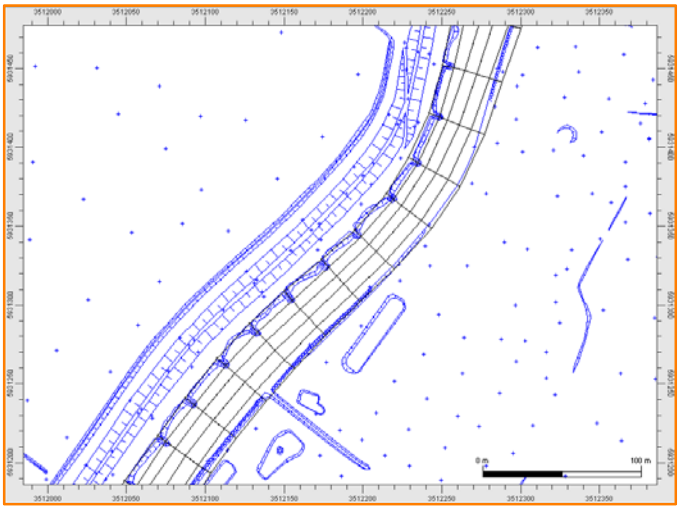


Figure 3-15. Alignment of polygon sides of the quad grid to groins

3.3.2 Quad grid for a sub-domain

The next example shows a slightly different approach of generating a flow aligned grid. The task is to create a sub-grid for the fairway of the Elbe estuary. This sub-grid is used for insertion into an unstructured triangular grid of the entire estuary model.

1. Provide a Suitable Digital Terrain Model

2. Setup the Alignment Polyline

3. Define Characteristic Profiles
With the help of the plotted isobaths of the Digital Terrain Model several characteristic “profiles” are created. A profile consists of a list of row widths that are used by the quad grid generation algorithm to gradually change the widths of the generated quadrilaterals.

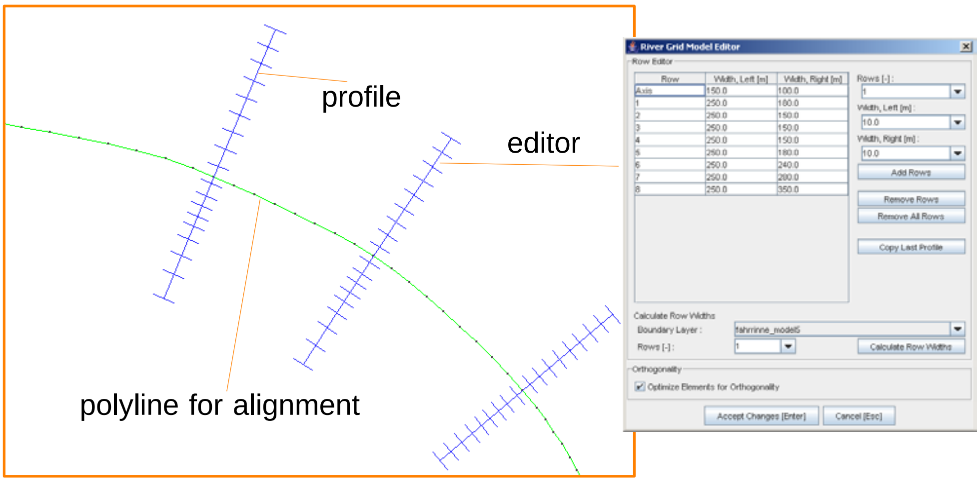


Figure 3-16. Editing the Quad Grid Model

4. Running the Quad Grid Generator

The quad grid generator is started with the created Quad Grid Model as basic input. Figure 3-17 shows the resulting grid.

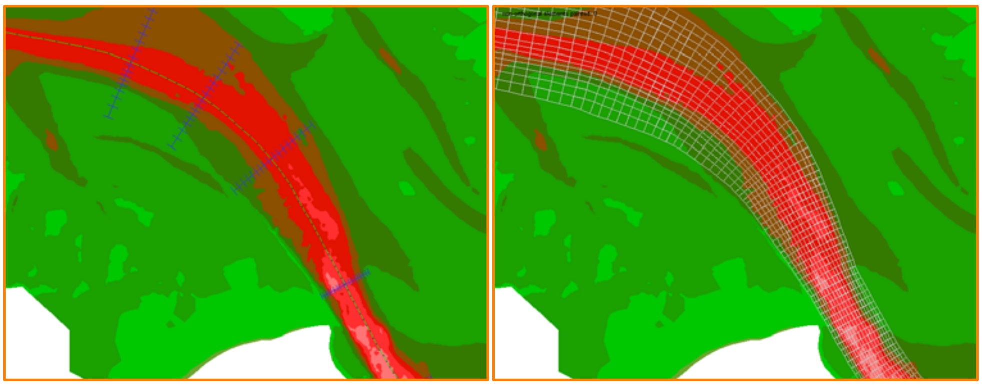


Figure 3-17. Quad Grid Model and generated quadrilateral sub-grid

The presented procedure is well suited for different types of application. Two further examples are demonstrated in Figure 3-18.

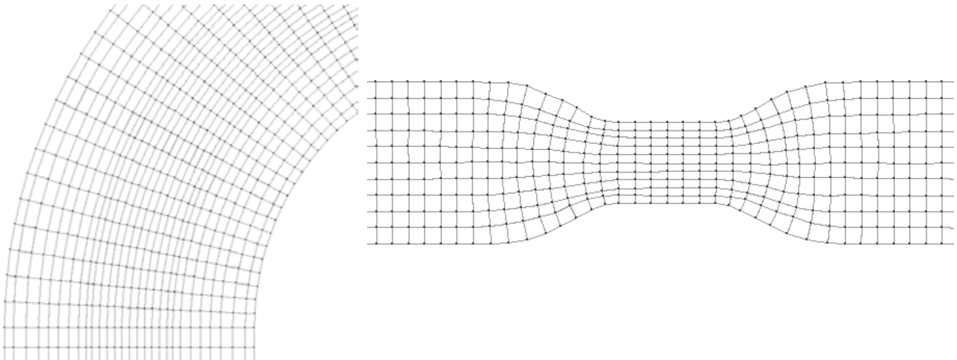


Figure 3-18. Grids constructed with the quad grid generator

3.4 Coupling Unstructured Grids

Unstructured grids derived from different grid generation techniques can be combined to an entire grid with the preprocessor's sub-grid module. A grid can be inserted into an unstructured model with a special subroutine. All polygons that cover the overlapping area are automatically updated. The next figure shows the insertion of quadrilateral sub-grids into a triangulation.

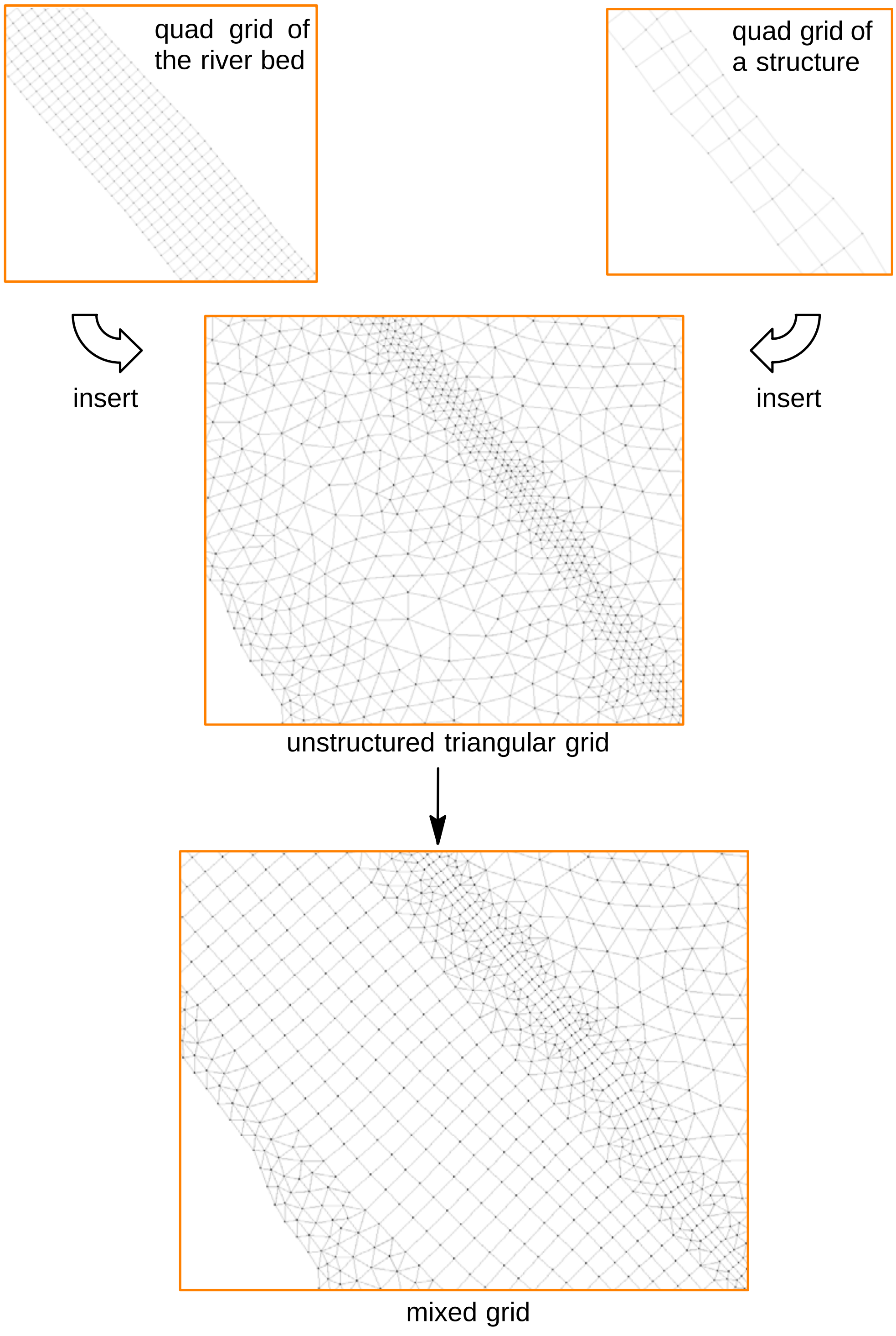


Figure 3-19. Grid insertion with the sub-grid approach

Grid Optimization

4

4.1 Orthogonality Optimization

4.1.1 Basic Geometric Operators

A basic geometric operation is the construction of strictly orthogonal three– and foursided polygons. The operator takes into account that on the one hand all polygon vertices must be located on a circumcircle and on the other hand the centerpoint must be located within the polygon bounds. For a given polygon side the possible location of the third polygon vertex for constructing a triangle respectivly the fourth vertex for generating a quadrilateral are shown in Figure 4-1.

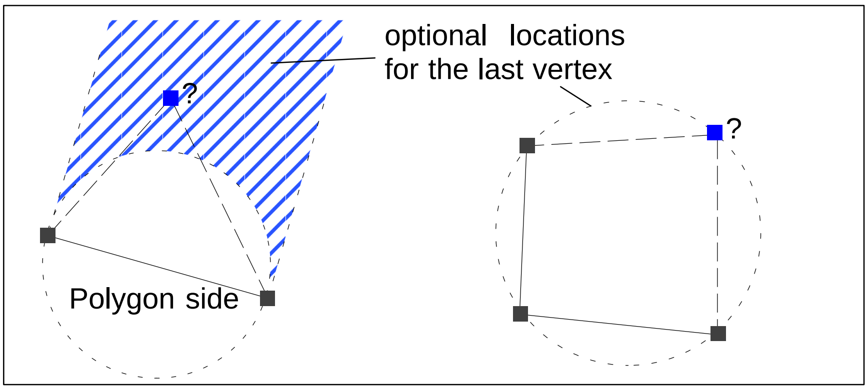


Figure 4-1. Construction of triangular and quadrilateral polygons

The construction schemes presented in the last example only dealt with a single polygon. The task becomes more complex for adjacent polygons.

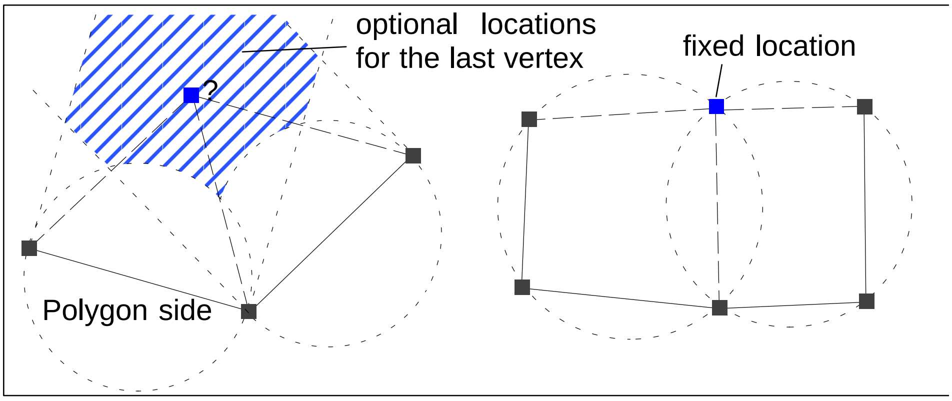


Figure 4-2. Construction of adjacent triangular and quadrilateral polygons

The operator applied to quadrilaterals leads to a well defined location for one of the vertices on the shared polygon side. This location can be calculated as the intersection point of the circumcircles given by the three remaining vertices of each polygon. For adjacent triangles possible locations are given by the area indicated in Figure 4-2. The location within this area that leads to an optimal shape parameter of the two triangles is therefore chosen.

4.1.2 Orthogonality Operator

The strategies described above for constructing orthogonal polygons are used for a special smoothing operator that is applied to a patch of polygons. The smoothing operator calculates an optimal location for two adjacent polygons each and finally moves the patch node to the geometric center of all calculated locations.

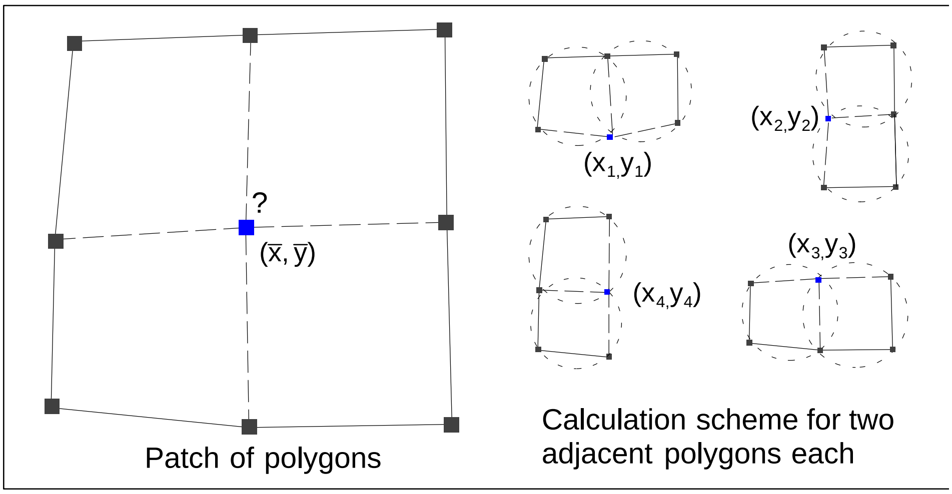
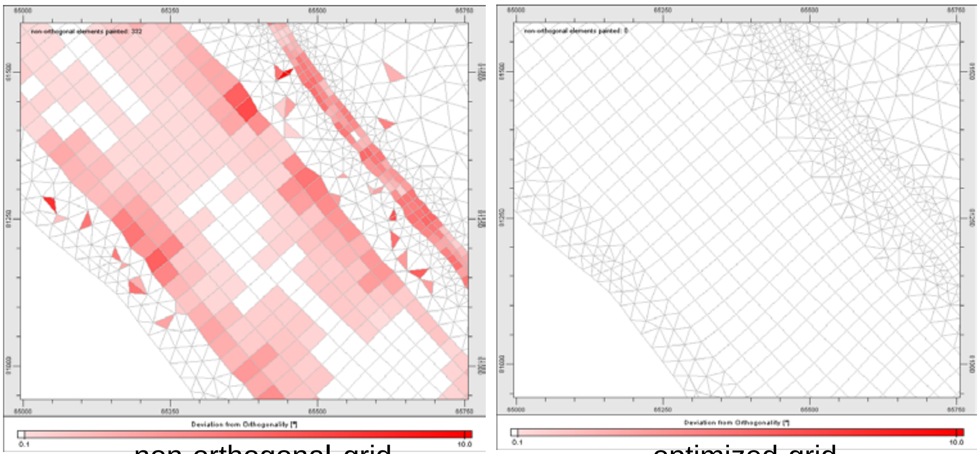


Figure 4-3. Calculation of the patch node location

To avoid degeneration of polygons the resulting triangles and quadrilaterals are tested for shape parameters and orthogonality values in comparison to the initial situation. The smoothing operation is undone if no improvement is gained.
The calculation scheme illustrated for a patch of quadrilaterals is applied in the same way to mixed patches of triangular and quadrilateral polygons and to patches containing triangles only.

Optimizing an entire grid is finally achieved by calculating optimal locations for all patch vertices. This loop over all vertices is repeated in several steps.



non-orthogonal grid   optimized grid

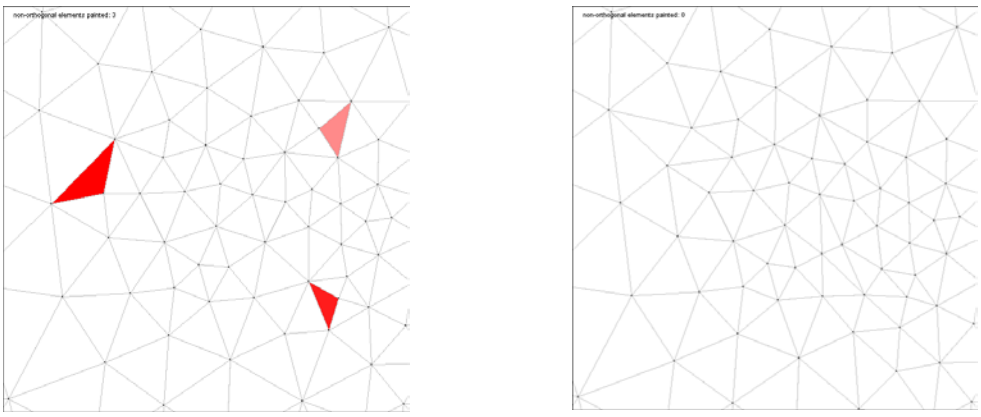
Figure 4-4. Example for orthogonality optimization

4.1.3 Strict Orthogonality

Optimizing a grid for orthogonality using the orthogonality operator does not guarantee strict orthogonality across all polygon sides. The resulting maximum deviation from orthogonality is mainly influenced by

- the topology of the grid (e.g. patch configurations)
- geometric constraints (boundaries, user defined break lines, inner boundaries, etc.)

If strict orthogonality is required, the grid has to be optimized in regard to its topological properties. This is done by a patch size optimization, which in general requires a grid refinement and a grid coarsening. Thus, this preparatory step affects polygon and vertex sizes.



not suitable patch configurations   optimized patch configurations

Figure 4-5. Improving patch configurations to gain strict orthogonality

Non-orthogonal polygons caused by geometric constraints need further discussion in respect of orthogonality optimization. The preprocessor's concept allows the definition of geometric constraints such as break lines, structure geometries, etc. These constraints are used to protect polygon vertices from any modification which often excludes these polygons from strict orthogonality. The following figure illustrates this aspect.

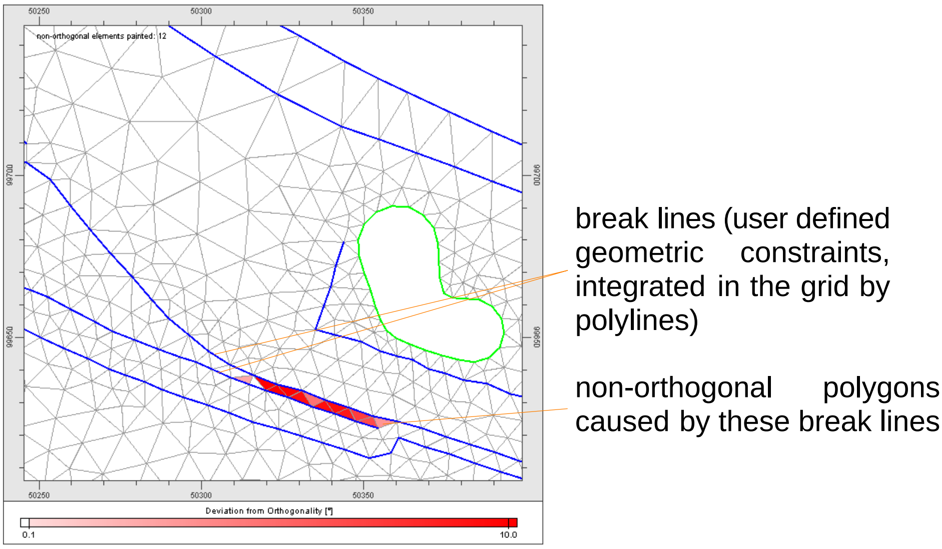


Figure 4-6. Non-orthogonal polygons caused by geometric constraints

Different strategies are suggested to face this problem:

- using well prepared polylines as geometric constraints
- allowing a deviation from orthogonality
- allowing a slight spatial tolerance

Using suitable polylines generally results in a large effort to prepare these polylines and is often not recommended for large grids. Since a deviation from orthogonality is opposed to the basic mathematical assumptions of unstructured orthogonal grids, allowing a slight spatial tolerance is the strategy suggested by the preprocessor Janet to deal with this problem.

The example shown in Figure 4-6 contains non-orthogonal polygons with a maximum deviation from orthogonality of 11.97°. The applied grid optimization using a slight spatial tolerance (Figure 4-7) results in a maximum spatial deviation of 0.45m (15.2% of polygon side length).

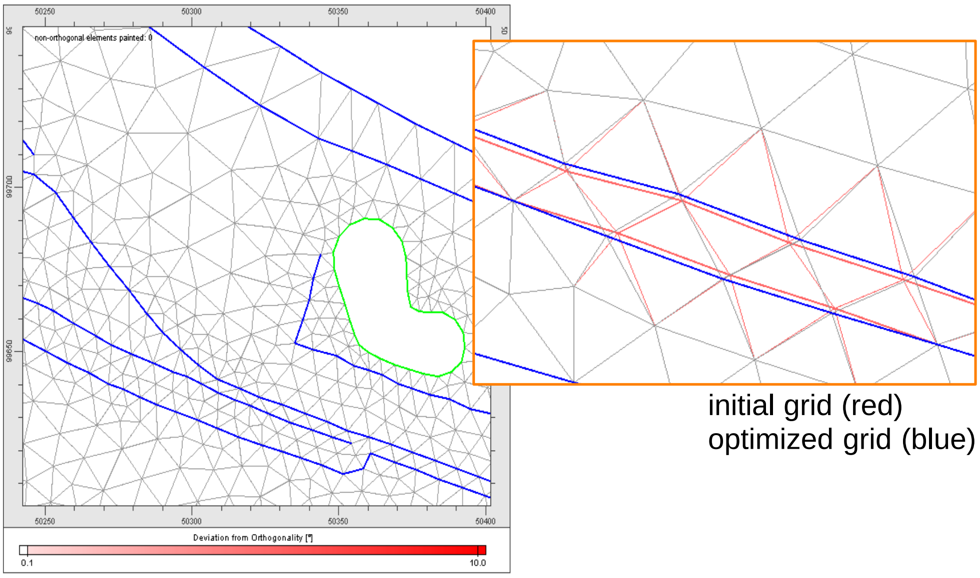


Figure 4-7. Optimized grid with detailed view on breaklines

A further example shows a narrow channel discretized by quadrilaterals. If the grid is restricted to fit the boundary, strict orthogonality cannot be achieved. The grid contains a maximum deviation from orthogonality of 6.3° and a mean deviation of 2.1°. By allowing the boundary vertices to be slightly moved, strict orthogonality is reached. The maximum spatial deviation can be measured to 0.36m (6.3% of polygon side length) for this example.

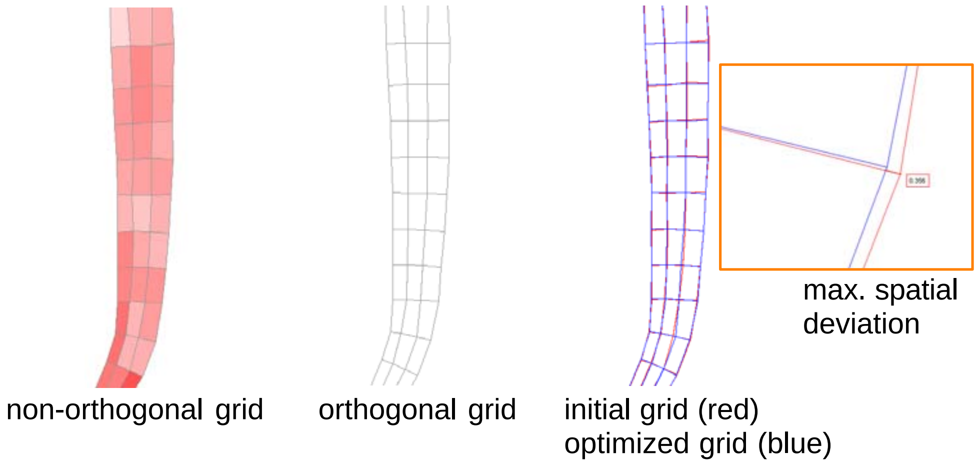


Figure 4-8. Narrow channel discretized by quadrilaterals

In general the preprocessor supports all strategies discussed to deal with non-orthogonal polygons caused by geometric constraints. The suggested spatial tolerance is not obligatory. The hydraulic problem for which a grid is generated should be taken into account for choosing a suitable strategy.

4.2 Shape Optimization

Shape optimization is offered by a Laplacian smoothing operation. For a patch of polygons, a new location of the patch vertex is calculated by moving the vertex to the geometric center of all polygon's gravity points. The operation can be applied to both mixed patches and patches consisting of only triangles or only quadrilaterals.

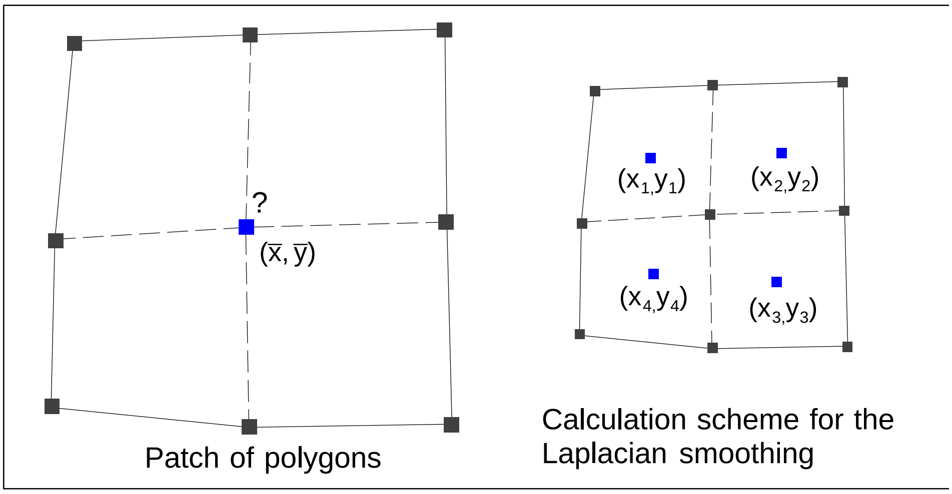
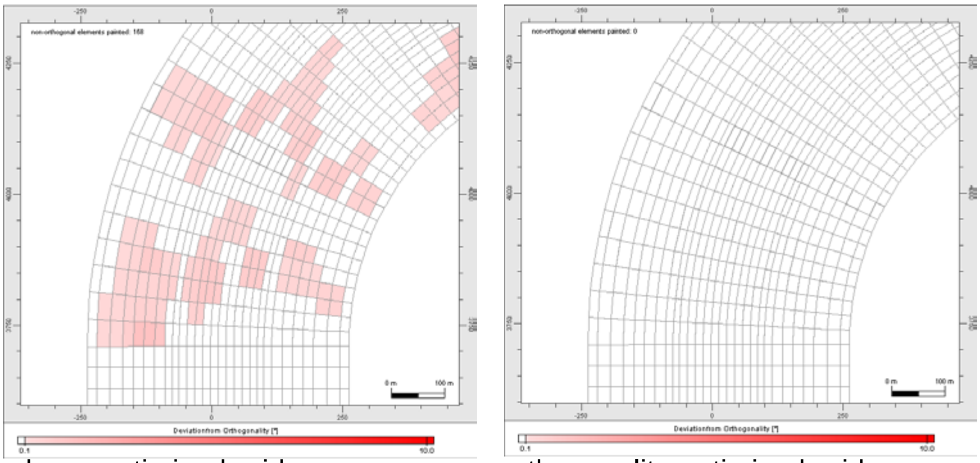


Figure 4-9. Calculation of the patch node location

Optimizing an entire grid is finally achieved by calculating optimal locations for all patch vertices. This loop over all vertices is repeated in several steps.

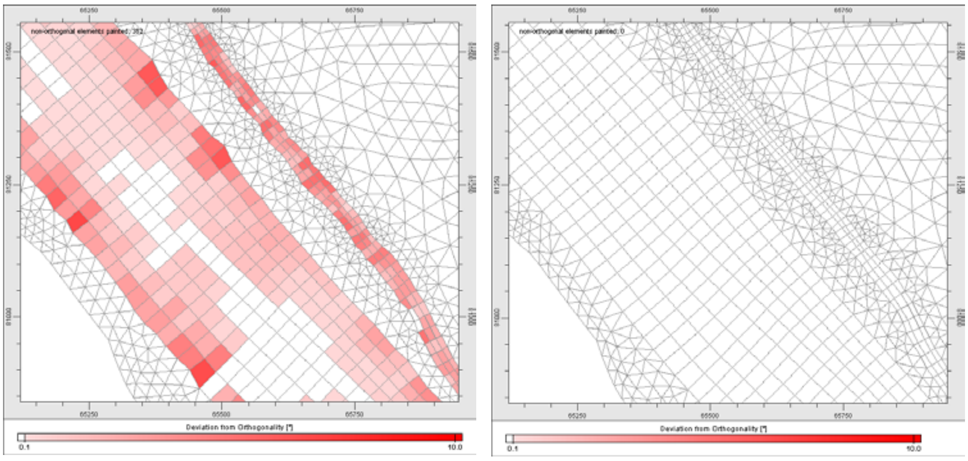
In connection with unstructured orthogonal grids the effects of shape optimization and orthogonality optimatization in respect of orthogonality is a point of interest. The following examples illustrate the shape optimization compared to a grid already optimized for orthogonality.



shape optimized grid	orthogonality optimized grid
Max SH = 0.029, Mean SH = 0.008	Max SH = 0.031, Mean SH = 0.008
Max DO = 1.245, Mean DO = 0.044	Max DO = 0.000, Mean DO = 0.000

Figure 4-10. Comparison of shape and orthogonality optimization for a quad grid

As expected the maximum and mean values for the shape parameter and orthogonality are reduced by the respective optimization operations. The differences are, however, very small for a grid consisting of quadrilaterals only. Differences become more significant if mixed grids are considered. The next example shows the same comparison for a mixed grid.



shape optimized grid	orthogonality optimized grid
Max SH = 0.385, Mean SH = 0.131	Max SH = 0.539, Mean SH = 0.157
Max DO = 12.18, Mean DO = 0.103	Max DO = 0.000, Mean DO = 0.000

Figure 4-11. Comparison of shape and orthogonality optimization for a mixed grid

The results presented in Figure 4-11 illustrate that more significant differences can be found if mixed grids are considered. In particular the quadrilaterals show a higher deviation from orthogonality after the shape optimization has been applied to the grid. The higher values are mainly caused by the mixed patches at the border of the river channel. Therefore shape optimization is not a suitable method for mixed grids to ensure strict orthogonality.

4.3 Topological Patch Optimization

The topological properties of a grid are improved by a patch optimization which improves a grid's patch configurations. Especially patches with four polygons and patches with eight or more polygons are modified by this operation.
Fields of application are given by the needs of optimizing a grid for orthogonality as described in chapter 4.1, enabling gradual changes in polygon size from higher discretized to more coarse areas and finally for gaining a more optimized configuration of flow directions.

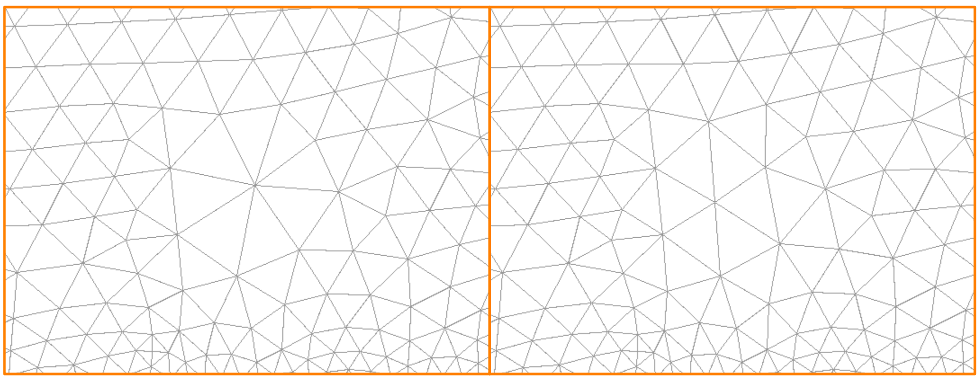


Figure 4-12. 8-Patch before (left side) and after optimization (right side)