Hyb2D

(A Hybrid Grid Generator in 2D)

(Version 1.0)

Algorithm Overview and Description

Cristina Manzi, Rosa Maria Spitaleri and Luca Formaggia

November 25, 1998

Contents

Introduction

In this report we present the hybrid grid generation algorithm implemented in the generator Hyb2D . We will describe the numerical methods and the algorithmic components used by the version 1.0 of this generator which combines the elliptic generation of boundary-fitted grids (BFGs) with an advancing front technique for unstructured mesh generation.

A companion report [6] contains the instructions on how to run the actual code.

The report is organized as follows. First a brief review on mesh generation techniques for structured, unstructured and hybrid grids is given in section 1. Section 2 presents the grid generation method used to create the structured grid. It is an elliptic generation technique which provides BFGs. Section 3 is devoted to the illustration of the unstructured mesh generation. In section 4 there is the description of the overall hybrid mesh generation strategy using the two presented algorithms for the structured and unstructured part. A short conclusion completes the report.

1 Overview of techniques for hybrid mesh generation

1.1 Hybrid grid: why?

During the past decade there has been an ever increasing demand to perform flow simulations that incorporate the complete details of geometry as well as flow physics.
Computational simulation of fluid flow around complex configurations has progressed significantly, thanks to the developement of numerical algorithms that can simulate the actual flow phenomena with great fidelty.
However the success of these algorithms hinges on the grid that models the geometry.
Several approaches have evolved for grid generation. Two of the most common are

- structured grid generation and
- unstructured grid generation.

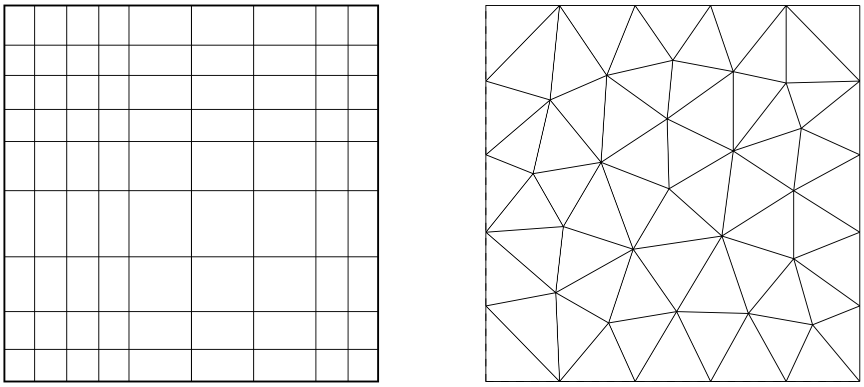

Figure 1: An example of a structured cartesian grid (on the left) and an unstructured triangular grid (on the right)

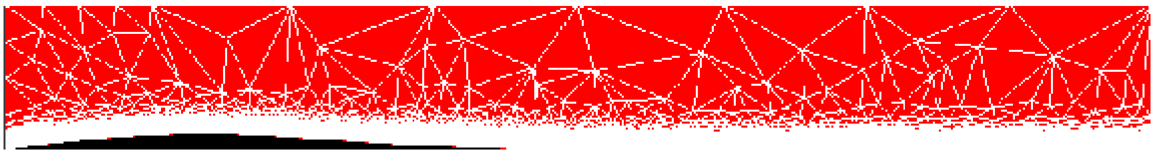

Figure 2: An example of a hybrid grid

In a structured grid every internal node is connected to a fixed number of mesh neighbors, forming a regular lattice (see fig. 1). In a quadrilateral (exahedral) cell grid each interior node belongs to four quadrilateral (eight exahedral) cells of the grid. In [12] you may find all the details about structured grid generation.
In an unstructured mesh a mesh node can be connected to an arbitrary number of adjacent points (see fig. 1). The connection of any node must be explicitly defined.
Unstructured grids may furnish a high geometrical flexibility (i.e. complex configurations may be handle quite naturally) and give an efficient control of the mesh resolution allowing highly localized refinements. Moreover adaptive techniques can be implemented in order to produce the "optimal" mesh with respect to the required parameters.

On the other hand the algorithms which operates on structured meshes are less expensive per mesh node than their unstructured grid based counterparts and also in the cases of non-adapted meshes around particulary simple geometries, specialized structured mesh generator can be devised, which are normally faster and require less user intervention.

Providing hybrid grids can be more appropriate than just structured or unstructured point distributions.
We divide the computational domain into two adjacent regions.
The first will be discretized by a structured mesher into quadrilateral elements [9], the second by the unstructured triangular mesh generator MeSh2D [2, 3].
The hybrid grid is the final mesh obtained combining the two grids which join at the boundary interface.
Regarding structured discretization, we will not map the whole physical domain into one rectangular block but just a restricted zone around solid boundaries. Such generation is often the only way to achieve a satisfactory numerical solution in the computational simulation of application problems involving boundary layers.
Indeed the generation of boundary-fitted curvilinear co-ordinate systems by solution of elliptic partial differential equations is a powerful and widely used approach to the structured discretization of the domains.
We assume Poisson's equations in a physical zone, appropriately selected, and the transformed system in the transformed computational rectangle [8, 9, 10, 11].
We have defined central differencing computation to solve the transformed generation system.
We can also employ line-spacing control by assuming non zero control functions in the governing equations.
Elliptic generation has been appropriately combined with an unstructured generator to easily handle geometrically complex configurations.
The unstructured grid technique makes use of the advancing front method which can start from either the physical or artificial boundary. Information related to the geometry of the interface between the two grids and the grid size distribution in the structured region are used to have a smooth global mesh.
During the unstructured grid generation, mesh spacing and stretching are governed through a metric tensor, which provides a high flexibility.
Anisotropic control can be also exerted. The spacing control on the unstructured part has been implemented by automatically generating sources around the structured-unstructured interface for the metric function.
This allows imposing spacing in the vicinity of the interface which is consistent with the grid density on the structured part, which results in a smooth transition (see fig. 2).

1.2 Overview of hybrid mesh generation methods

Genaration of body-conforming grids is a difficult task.
In case of structured meshes, a large number of separate blocks needs to be defined and the elements are generated within each block. In addition a special algorithm is required in order to match the grids of neighboring blocks.
Unstructured grids provide flexibility in grid generation since they can cover complicated topologies easier compared to the structured meshes. Unstructured grid generators have fo-

cused on producing valid grids for complex domains. However the generation of these kind of meshes is quite difficult near the boundary layers.
In these sense hybrid grids combine features from both techniques.
A hybrid type of grid which combines elements of different orientation appears to be much more flexible to conform flow features (such as boundary layers, wakes, shock waves and vortices) [4].
Koomullil et al. [5] have presented a hyperbolic type marching scheme to generate structured grids near the solid boundaries. A local elliptic solver is utilized for smoothing the grid lines and to avoid grid line crossing. They have described a method for trimming the overlaid structured grid. Delaunay triangulation is employed to generate the unstructured grid in the regions away from the body. Then, the structured and unstructured grid regions are integrated together to form a single grid for the solver.
Chappell et al. [1] have employed a marching method for structured generation. Generation propagates from a boundary to an outer boundary, the exact shape or location of which cannot be predetermined. The process goes on one layer at a time. Unstructured grids are generated via the Delaunay algorithm in the parametric space. Element quality measures and metric data calculated in the physical space ensure the production of a grid of isotropic triangles with smoothing varying point density. Grid density is controlled by the definition of boundary point distributions. Interfacing the structured and unstructured grid (in particular in the three dimensional case) is a non trivial problem. In [1] the authors present two approaches to this problem.
In [7], Noack et al. have described a method which combines structured grids with unstructured triangular meshes and Cartesian quadtree grids. This method utilizes as input a set of structured quadrilateral cell grids that may overlap each other and may not completely cover all the computational domain. An advancing front grid generation algorithm is used to trim the structured grids and remove the overlapping parts.

2 Structured mesh generation

In this section we present the structured mesh generation method implemented in Hyb2D. It consists in the differential generation by a system of elliptic equations providing BFGs [8, 9, 10, 11].

2.1 Elliptic grid generation

Structured grid generation techniques provide cartesian and curvilinear body comforming grids. There are two main families of methods:

- algebraic methods;
- differential methods.

The first methods are based on interpolation from grid boundaries while the differential methods consist in the numerical solution of elliptic, parabolic or hyperbolic partial differential equations.
Our approach is of the second kind: we assume two-dimensional Poisson's equations in the physical domain, which allow line-spacing control, and the transformed system in the computational domain.

2.1.1 2D generation equations

Let Ω be the physical domain to discretize. We assume the two-dimensional Poisson's equations in the curvilinear co-ordinates in an appropriate zone $\hat{\Omega} \subset \Omega$, and the transformed systems

$$\begin{cases} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2(Px_\xi + Qx_\eta) = 0 \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(Py_\xi + Qy_\eta) = 0, \end{cases} \tag{1}$$

in the cartesian co-ordinates, which are defined in the computational domain Ω^* transformed of $\hat{\Omega}$.
The model coefficients have this form:

$$\begin{array}{l} \alpha = x_\eta^2 + y_\eta^2, \;\; \beta = x_\xi x_\eta + y_\xi y_\eta, \\ \gamma = x_\xi^2 + y_\xi^2, \;\; J = x_\xi y_\eta - x_\eta y_\xi. \end{array} \tag{2}$$

The control function are defined as follow:

$$P(\xi,\eta) = -\sum_{i=1}^{n} a_{\xi_i} sign(\xi - \xi_i) e^{-c_{\xi_i}|\xi-\xi_i|} - \sum_{j=1}^{m} b_{\xi_j} sign(\xi - \xi_j) e^{-d_{\xi_j}\sqrt{(\xi-\xi_j)^2+(\eta-\eta_j)^2}} \tag{3}$$

The control function $P(\xi,\eta)$ allows both n and m attraction sources, ξ_i-line and (ξ_j,η_j)-points respectively, to be activated with the following specific controls:

- the amplitude a_{ξ_i} controls the attraction of the ξ-lines towards the ξ_i-line;
- the amplitude b_{ξ_j} controls the attraction of the ξ-line towards the grid point (ξ_j,η_j);
- the decay factor c_{ξ_i} controls the extension of the ξ_i-line attraction;

- the decay factor d_{ξ_j} controls the extension of the (ξ_j, η_j)-point attraction;
- the functiona *sign* allows attraction to be exercised on both sides of each source.

Attraction becomes repulsion by negative amplitudes.
The control function $Q(\xi, \eta)$ has the same form of the function $P(\xi, \eta)$, with ξ appropriately sobstituted by η so that it allows analogous control on the η-line spacing (see fig 3 and fig. 4).

2.2 Anisotropy control

For efficient applications we would like to focus on and analyze anisotropic behaviours arising in grid generation problems. We put in relation the isotropic model:

$$ar_{\xi\xi} + br_{\eta\eta} = f(\xi, \eta), \quad a \approx b > 0$$

and the elliptic grid generation system (1).
The generation model which has solution dependent coefficients, could slip from isotropic to anisotropic behaviours by losing balanced coefficient values.

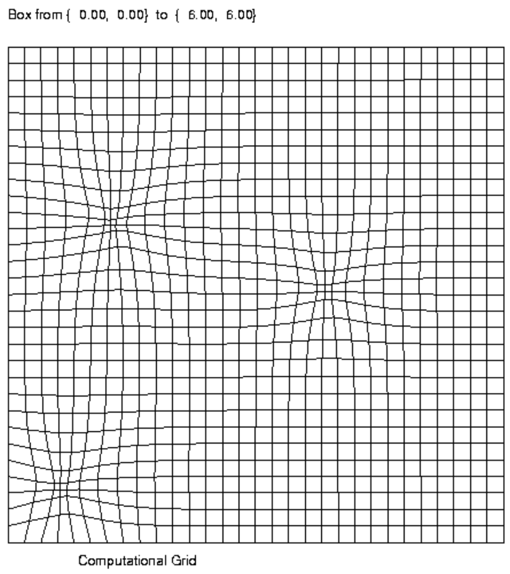

Figure 3: A structured grid with three point sources

From the geometrical point of view strong line concentration in the physical domain corresponds to anisotropy in the principle parts of the equations. Indeed if we assume that grids are near to be orthogonal, the mixed derivative coefficient can be considered to be relatively small and thus negligible with respect to the coefficients of the second order derivatives [11]. In case of isotropic models, which are discretized by standard symmetric differences, on a

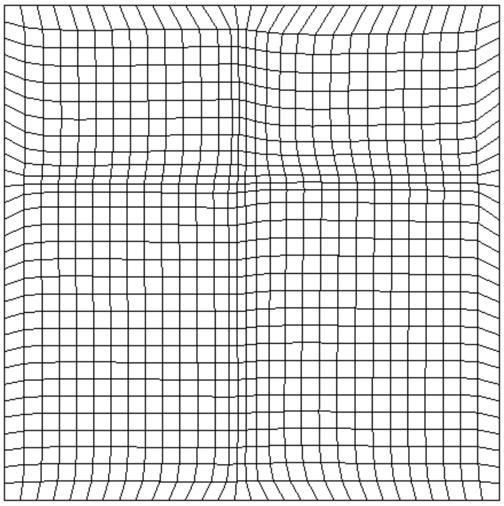

Figure 4: A structured grid with line sources

uniform grid, we may efficiently apply Gauss-Siedel relaxation.
Approximation of the model (1), (2), (3) by central differencing leads to a diagonally dominant system if the coefficients of the first order derivatives are "small enough" on a grid with given mesh size.
When the discrete equations are anisotropic, convergence becomes as slower as stronger the anisotropy is. In fact relaxation have not enough smoothing capability in the direction of the smallest coefficient and works just in the remaining directions.
Let L_h be the two-dimensional centred difference operator on the grid G_h and α_h and γ_h the discrete coefficients of the principal terms, that is associated to α and γ in (2). We can use the following criteria to investigate anisotropy in the structured zone:

- *Isotropic Operator L_h:*
  $\alpha_h \sim \gamma_h$ on $G_h \Leftrightarrow \alpha_h/\gamma_h \in [0.01, 100.]$ on G_h;
- *Anisotropic Operator L_h:*
  $\exists G_h^* \subseteq G_h : \alpha_h >> \gamma_h$ on $G_h \Leftrightarrow \alpha_h/\gamma_h > 100.$ on G_h.

These criteria can be used to immediately capture unbalanced coefficient values of the discrete generation model and, from the geometrical point of view, too strong grid line concentration. Anisotropy appearance causes convergence troubles. For the iterative solution of the system, (3) we apply Gauss-Seidel relaxation with lexicographic, red-black, four-color and zebra-line orderings [10, 11].

3 Unstructured mesh generation

The generation of the unstructured grid makes use of the front advancing method. We have used MeSh2D [2, 3], which is a triangular mesh generator, able to treat quite complex domains. It also permits a fine control of the mesh spacing and it allows for anisotopic grids. Also it implementes mesh adaption procedures.
To specify the parameter of the mesh (i.e. the shape and misures of the elements) MeSh2D makes use of a metric tensor which is defined by the user on an initial grid (*background grid*).

3.1 Definition of the metric

Using unstructured grids we normally desire to have a more refined mesh in the areas where we expect large solution gradients. Indeed we need the control of the mesh in the vicinity of the boundaries in order to maintain a good accuracy in the geometry definition.
To define the spacing and the strecthing of all the elements of the unstructured grid we use a metric tensor $\mathbf{M}$ and to define the element aspect ratio in restricted zone we use the definition of point and line sources [2].

3.1.1 The metric tensor

If Ω is the computational domain we can define on $\overline{\Omega} \subset \Omega$ a symmetric definite positive second-order tensor $\mathbf{M}$

$$\mathbf{M} = \mathbf{M}(\mathbf{x}) \qquad \mathbf{x} \in \overline{\Omega}.$$

A mesh side AB with Euclidean length l is called *optimal* if its legth according to the metric $\mathbf{M}$ is unitary, that is if it satisfies the following relation:

$$||AB||_{\mathbf{M}} = \int_0^l \sqrt{\mathbf{t}^T \mathbf{M} \mathbf{t}} d\xi = 1, \tag{4}$$

where $\mathbf{t}$ is the versor of AB.
Normally in equation (4) it is used a constant value of the metric $\mathbf{M}$, for example an everage value $\overline{\mathbf{M}}$.
We can say that an element is *optimal* if its sides are optimal.
So, given a metric function $\mathbf{M}$ the mesh generator will produce a mesh whose elements are as close as possible to the optimality condition. In this sense the metric $\mathbf{M}$ can be used to define the spacing control of the mesh.
For each point of the computational domain we can define two characteristics lengths δ_1 and δ_2 with associated orthogonal directions $\alpha_{\mathbf{1}}$ and $\alpha_{\mathbf{2}}$, defined as unitary vectors.
An optimal element, with respect to the metric defined as

$$\mathbf{M} = \sum_{i=1}^{2} \delta_i^{-2} \alpha_{\mathbf{i}} \otimes \alpha_{\mathbf{i}}, \tag{5}$$

will have dimensions in accordance with the given parameters.
If we use the matrix $\mathbf{T}$ defined as

$$\mathbf{T} = \sqrt{\mathbf{M}} \tag{6}$$

we can see that this is a local mapping between the physical space and a "normalized" space where optimal sides have unitary length (see fig. 5).

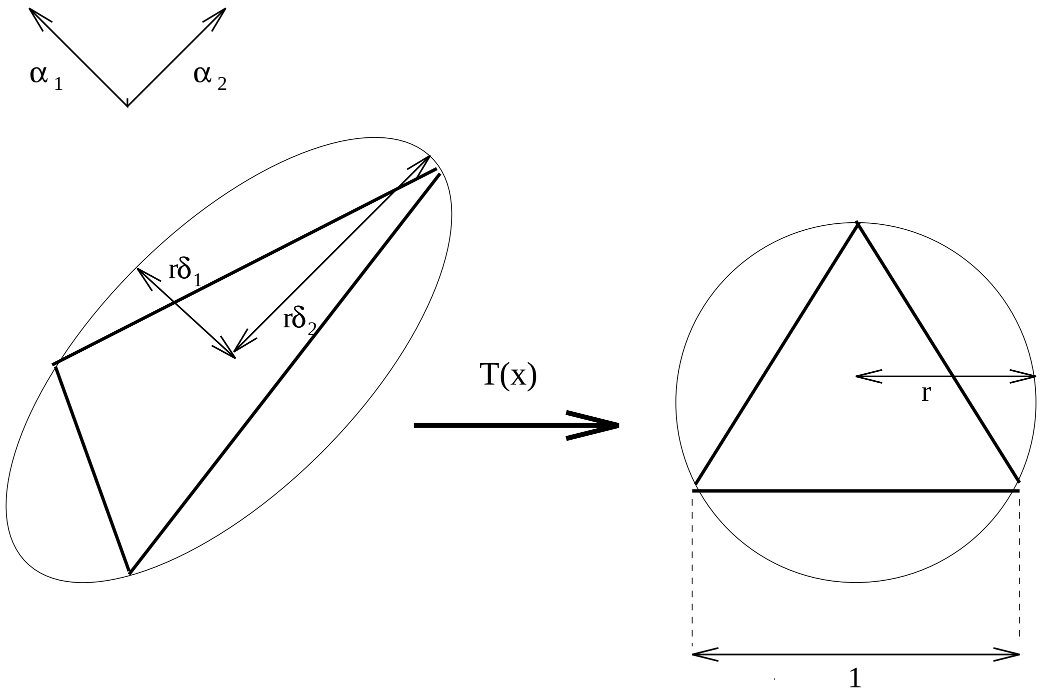

Figure 5: The transformation **T** converts an optimal triangle into an equilateral triangle with unitary lenght side

Clearly this transformation is only local since, in general, the Riemann-Christoffel tensor associated to **M** does not vanish at every point of the computational domain Ω, hence a global mapping can not be found.
For practical use it is better to define two quantities, δ and s, which are defined as follow:

$$\delta = \delta_1, \quad s = \delta_2/\delta_1, \qquad \text{where} \qquad \delta_1 = \min(\delta_1, \delta_2).$$

If we can indicate $\alpha_\delta = \boldsymbol{\alpha_2}$ and $\alpha_{\mathbf{s}} = \boldsymbol{\alpha_2}$, the grid parameters are:

- the misure δ of the element;
- the stretching factor s;
- the stretching direction, defined by the versor $\alpha_{\mathbf{s}}$ and by the versor α_δ, orthogonal to $\alpha_{\mathbf{s}}$.

The control of the mesh caracteristics is obtained defining a spatial distribution of these parameters. To define the metric we implemented two strategies which work together:

- the C^0-interpolation on a background grid and
- the point and line sources technique.

3.1.2 Background grid technique

To define the metric tensor on the whole computational domain we adopted the *backgroung grid technique*. It consists of the definition of an initial grid which complitely covers all the domain $\overline{\Omega}$ and on which it is possible to define a discrete number of values of the tensor $\mathbf{M}$.

Using these informations on the nodes of this background grid, MeSh2D will compute the value of the metric at each point of the domain, using a C^0 interpolation.
The use of the background grid makes the implementation of mesh adaption techniques based on re-meshing straightforward.
If we compute a solution on a preliminary grid and by analyzing the obtained solution we have ways of estimating a new metric function at the grid points, we can use directly the preliminary grid as background grid for the new adapted mesh [2].

3.1.3 Point and line sources

However, for a finer control of mesh size in particolar zones of the domain, it may be necessary to have a more complicated background grid. This is why MeSh2D [2] integrates background grid informations with the mesh size control based on *point* and *line* sources.
Using a point source it is possible to define a transformation matrix with associated decay function, while using a line source it is possible to define the transformation at the end points of a segment, identifying two spacing directions (one aligned and the other orthogonal to the segment).
It is important to say that the sources do not substitute the background grid which is required to provide the "background" value of the metric.

3.2 Front advancing technique

MeSh2D implements the front advancing technique
to create the elements of the mesh (see fig 6).
This technique consists first in the creation of the corners points, i.e. the vertices of the domain. Then the generation discretizes the domain boundary curves by placing nodes along them. The generation front is defined as the union of the oriented segments that enclose the region that has still to be triangulated.
At the beginning of the process the front will contain just the boundary sides. The generation proceeds then by selecting one of the sides in the front as the base for a new triangle. The third node to build the triangle can be either a new node or an existing active one. The generation goes on by selecting a new front side and it stops when the generation front is empty.
The generation process may be summarised as follow:

- vertices are taken as mesh nodes;
- nodes are generated inside of each edge, according to the given metric;
- a triangular mesh is generated on each region and the regions are assembled.

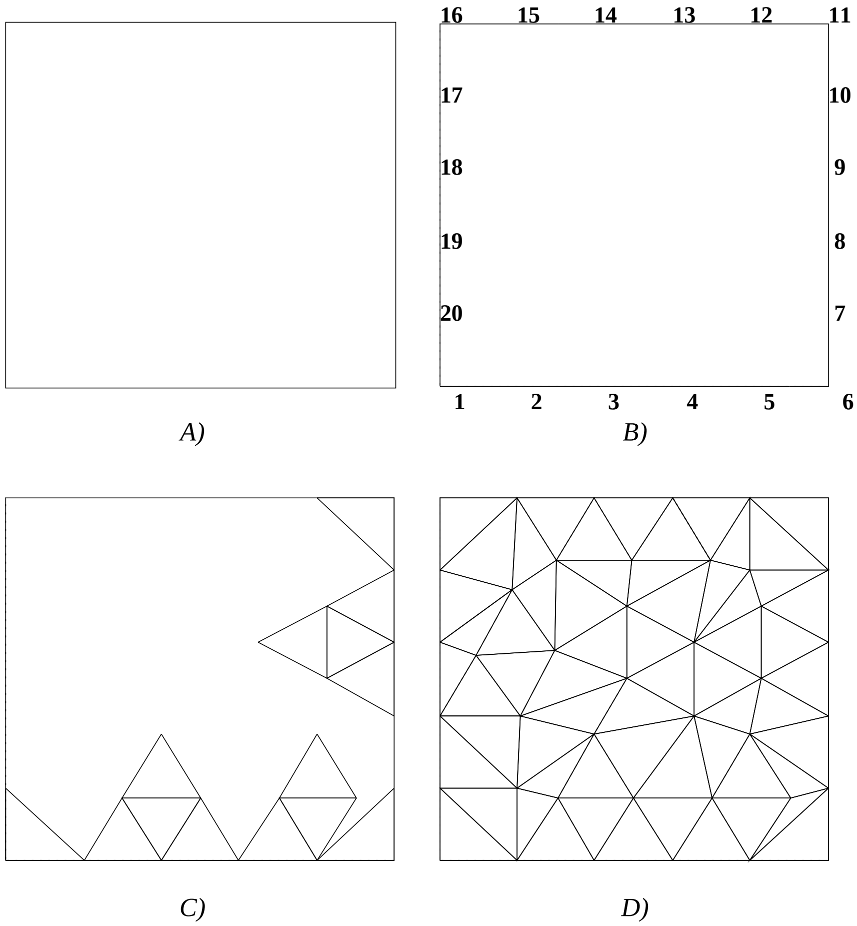

Figure 6: A) Domain definition. B) Boundary discretization. C) A generation process step. D) The final grid.

3.3 Mesh enhancement

The grid generated by the front advancing technique may create some badly shaped elements. To enhance mesh quality MeSh2D implements the following different procedures:

1. node movement;
2. mesh side swapping and
3. elimination of "3-connected nodes" (see fig. 7).

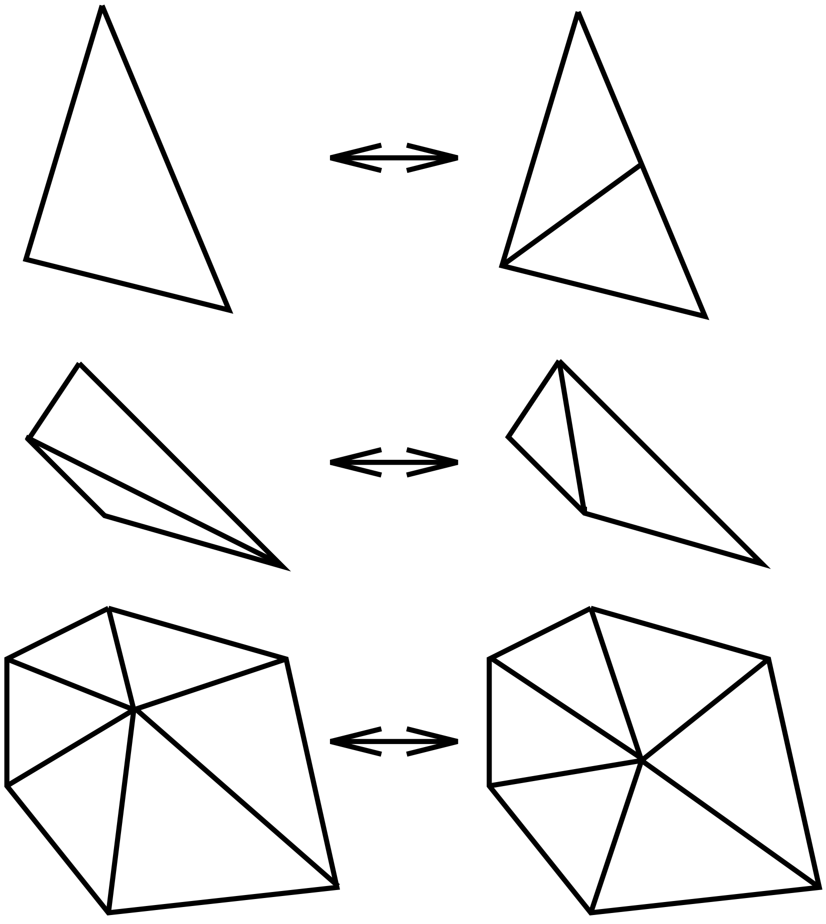

Figure 7: The three different mesh enhancement procedures. Starting fron the top: mesh refinement and derefinement; edge swapping; node movement.

3.3.1 Node movement

Moving mesh nodes is a common technique to enhance the quality of a mesh. MeSh2D makes use of this procedure keeping unchanged the grid topology.
Most of the procedures of this kind are based on the minimization of a special function related in some way to the mesh quality measure.
MeSh2D implements the "laplacian smoothing" technique [2].

3.3.2 Mesh side swapping

Diagonal swapping is a technique which allows to modify the grid topology while mantaining unchanged the number and position of the mesh nodes. This is a technique often used in unstructured mesh generation in order to improve the mesh quality.
Three different strategies have been implemented in MeSh2D:

- point connectivity strategy;
- element quality indicator strategy and
- a mixture of the two previous techniques [2].

A mesh side is "swappable" if the configuration obtained after the swapping is admissible. This means that the boundary sides are not allowed to be swapped and the elements in the swapped configuration have positive areas. An internal side is swappable if and only if the quadrilateral formed by two adjacent triangles is strictly convex.

3.3.3 Elimination of "3-connected nodes"

MeSh2D implements the de-refinement obtained by eliminating an internal node with node connectivity equal to 3.
This operation is not strictly necessary, but a node with three neighbors is often a "spurious" point in the triangulation and has been generated when the front has collapsed into itself at the last step of the triangulation process. This node may be normally eliminated without problems.

4 Hybrid mesh generation

In this section we will see how the structured and unstructured generators "join together" to create a hybrid grid.
First we produce the structured grid and then, the unstructured grid. To have a smooth hybrid grid it is necessary to define the spacing control at the interface. The information about the interface elements can be transfered to the metric tensor to produce triangles in accordance with the spacing defined for the structured part near the interface boundary.

4.1 Starting from a domain

First of all, given the computational domain it is possible to individuate the interface boundary between the two grids.
To define the structured part, as it has been seen in section 2, it is necessary to give all the boundary nodes and the sources.
The order in the definition of the boundary nodes for the structured domain has an important role in the generation of the mesh. The user has to define the nodes starting from a vertex point of the domain and going on in the clock-wise sense around the boundary sides.
The sources can be attractive or repulsive lines or points. We also need the amplitude and decay constant parameters for each source.
For the unstructured domain we need the boundary conditions for all the curves of the domain.
During the generation the points on the interface boundary will be set as fixed nodes in order to create a conformal grid.
The unstructured generation will start from the interface boundary keeping into account all the information about the shape and stretching of the elements. This informations are given by the metric tensor.
Some enhancements are possible for the unstructured grid to avoid badly shaped elements that the front advanging technique could create.

4.2 Smoothness of the grid

The spacing control on the unstructured part has been implemented by automatically generating sources around the structured-unstructured interface for the metric function.
This allows imposing spacing in the vicinity of the interface which is consistent with the grid density on the unstructured part, which results in a smooth transition (see fig. 8).

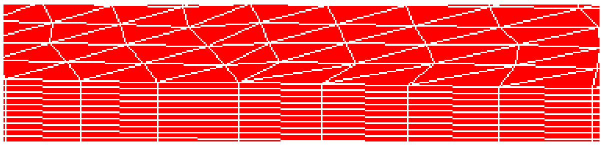

Figure 8: Zoom near the interface region.

5 Final considerations

We have presented the hybrid grid generation method and the related algorithm implemented in Hyb2D and experimented on test problems. The hybrid mesh generator method in two dimensions, combines an elliptic generator of boundary-fitted grids with an advancing front method for unstructured mesh generation. The code Hyb2D integrates efficiently the structure and unstructured computational strategies.

The generation of boundary-fitted grids is reserved to the region surrounding solid boundaries and the unstructured generation provides triangular grids on the remaining zone.

The generation of boundary-fitted curvilinear co-ordinates systems by the solution of elliptic partial differential equations is a powerful and widely used approach to the structured discretization of the domain.
We assume Poisson's equations in one physical zone and the transformed system in the transformed computational rectangle. We have defined central differencing computation to solve the transformed generation system.
The generation of the unstructured mesh, which involves the creation of both the mesh points and their connectivities, is performed simultaneously using the advancing front technique.
An appropriate technique for matching the two grids together in order to have a smooth transition between the structured and unstructured grids, has been devised.

From the already obtained numerical results we can conclude that:-an effective hybrid generation method and code are available with enlarged capabilities for applications respect to just structured or unstructured generation tools,- they can be a promising starting point for future development and improved performances.

List of Figures

References

[1] J.A. Chappell, J.A. Shaw, and M. Leatham. The generation of hybrid grids incorporating prismatic regions for viscous flow calculations. *Proceedings IMACS-WC97*, 1997.

[2] L. Formaggia and F. Rapetti. Mesh2d (unstructured mesh generator in 2d), version 1.0. Algorithm overview and description. Technical report CRS4-TECH-REP-96/1, CRS4, February 1996.

[3] L. Formaggia and F. Rapetti. Mesh2d (unstructured mesh generator in 2d), version 1.0. An user's manual. Technical report CRS4-TECH-REP-96/2, CRS4, February 1996.

[4] Y. Kallinderis. Discretization of complex 3-d flow domains with adaptive hybrid grids. *Proceedings IMACS-WC97*, 1997.

[5] R.P. Koomullil, B.K. Soni, and H. Chih-Ti. Flow simulations on generalized grids. *Proceedings of the* 5^{th} *International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, 1996.

[6] C. Manzi, R. M. Spitaleri, and L. Formaggia. Hyb2D (A Hybrid Mesh Generator in 2D), version 1.0. An user's manual. 1998.

[7] R.W. Noack, J.P. Steinbrenner, and D.G. Bishop. A three-dimensional hybrid grid generation technique with application to bodies in relative motion. *Proceedings IMACS-WC97*, 1997.

[8] R.M. Spitaleri. Full-FAS multigrid grid generation algorithms. *Proceedings IMACS-WC97*, 1997.

[9] R.M. Spitaleri. Computational and visual advances in multigrid grid generation. *App. Num. Math. 30 (to appear)*, 1998.

[10] R.M. Spitaleri and V. Micacchi. A multiblock multigrid grid generation for complex simulations. *Math. & Comp. Sim.*, 1997.

[11] R.M. Spitaleri and V. Regolo. Multiblock multigrid grid generation methods overcoming multigrid anisotropy. *Appl. Math. & Comp.*, 84(2), 1997.

[12] J.F. Thompson, Z.U.A. Warsi, and C.W Mastin. Boundafy-fitted coordinate systems for numerical solutions of partial differential equations - a review. *Journal of computational Physics*, 1982.