Conservative Treatment of 3D Multi-Block Unstructred Mesh Interfaces for Finite Volume Computations of Fluid Flows With Moving Boundaries

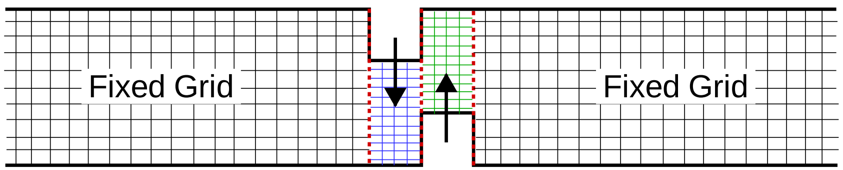
D. Vigneron and J.-A. Essers

Aerospace and Mechanical Engineering Department

Multi-block meshes

Use of Multi-block meshes for finite volume methods

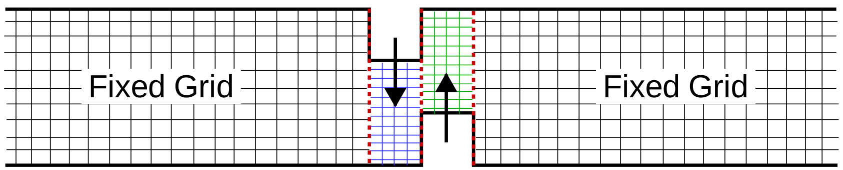
- Generate meshes more easily
- Use of independent dynamic meshes



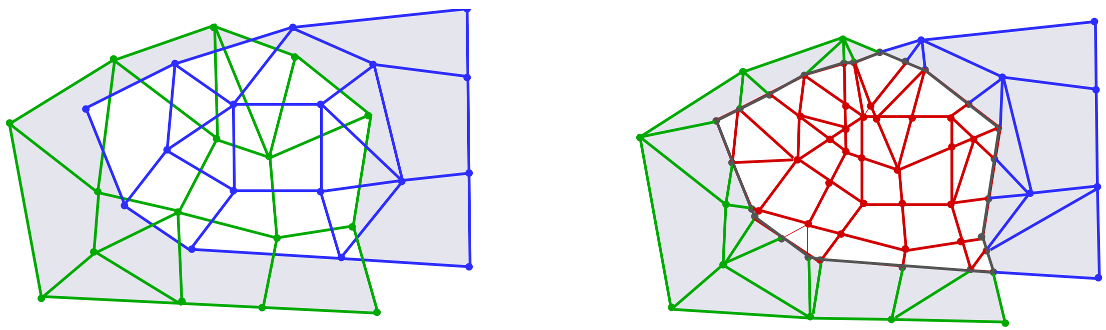
Multi-block meshes

Use of Multi-block meshes for finite volume methods

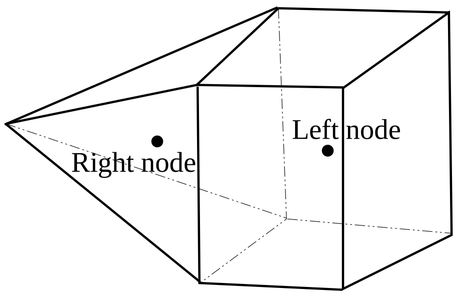
- Generate meshes more easily
- Use of independent dynamic meshes

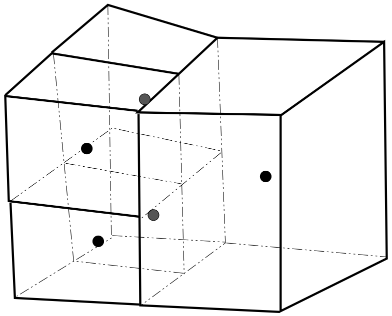


- Treatment of 3D unstructured meshes interface



Cell Centered Finite Volume Method





- Integration of advective and viscous fluxes on faces

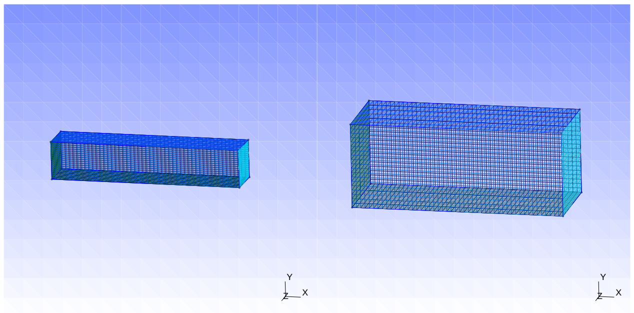
$$\mathbf{Rhs} = \sum_i \int\int_{\triangle_i/\square_i} \left[\tilde{\mathbf{f}}_n^a\left(\tilde{\mathbf{w}}_L, \tilde{\mathbf{w}}_R, \tilde{v}_n^g\right) + \mathbf{f}_n^d\left(\tilde{\mathbf{w}}, \tilde{\nabla\mathbf{w}}\right)\right] dS_i$$

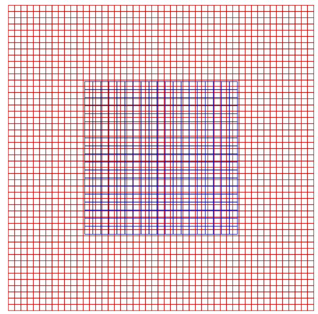
$$\tilde{w}_{L/R} = w_{L/R} + \left(\mathbf{x} - \mathbf{x}_{L/R}\right)^T \nabla(w)\rfloor_{L/R} + \ldots$$

$\Rightarrow$ FVM allows the use of non conformal meshes

Algorithm description

Two mesh blocks example - Boundary must be detected



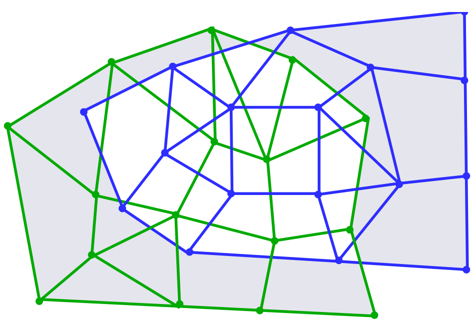


- The interface is composed of two boundary meshes (mesh A and mesh B)
- Mesh A and mesh B faces have only one left neighbour node
- A new mesh C must be created with faces having one left and one right neighbour nodes

Algorithm description

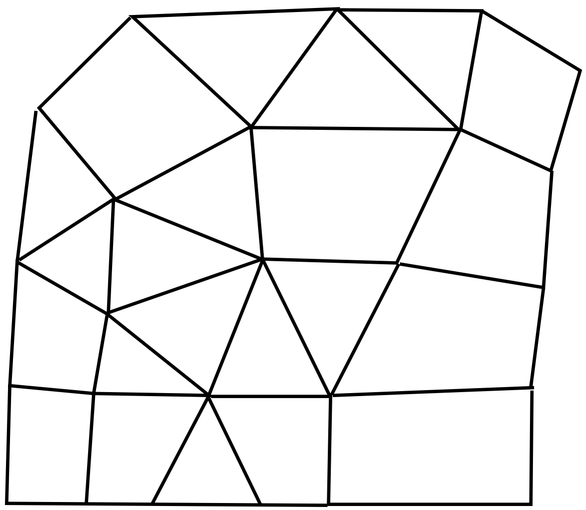
Three steps

- Locate mesh B vertices in mesh A faces or on boundary.
- Compute mesh B edges and mesh A edges intersection, cut all edges and create new mesh C.
- Build macro-faces having one left and one right neighbour node. Cut macro-faces.



Algorithm description

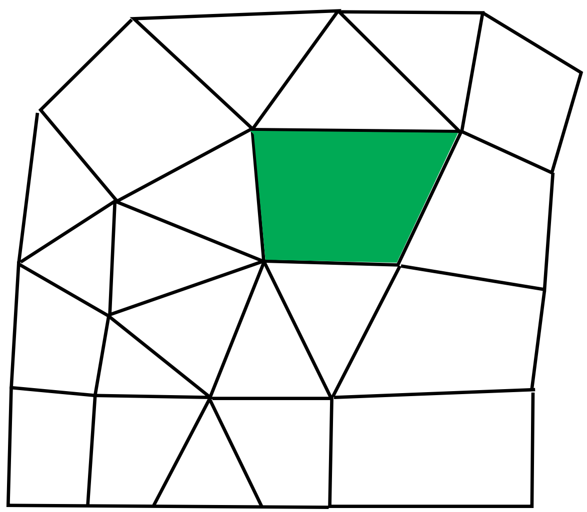
Data for boundary meshes



Algorithm description

Data for boundary meshes

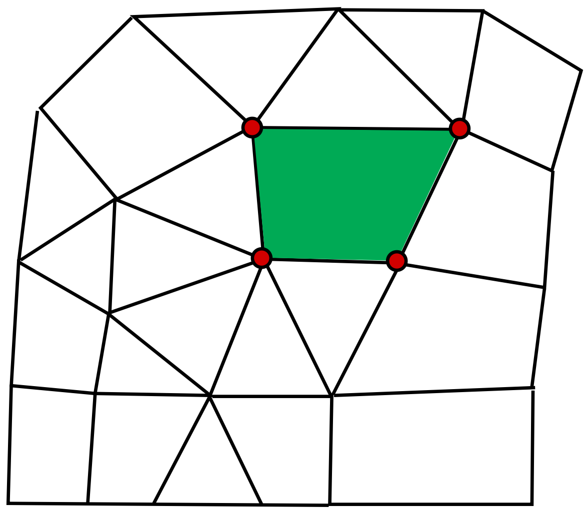
- Faces :



Algorithm description

Data for boundary meshes

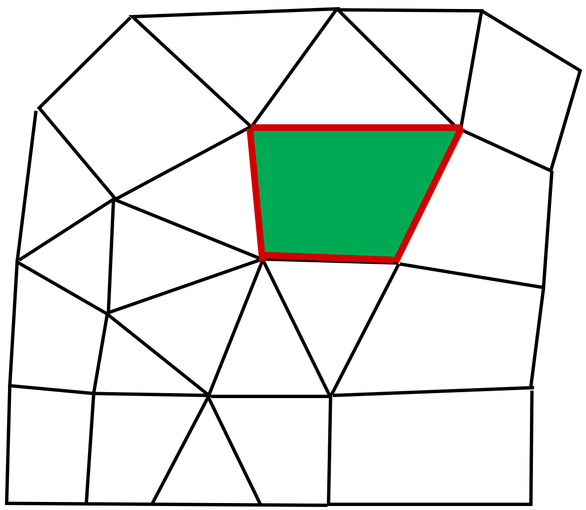
- Faces : list of vertices



Algorithm description

Data for boundary meshes

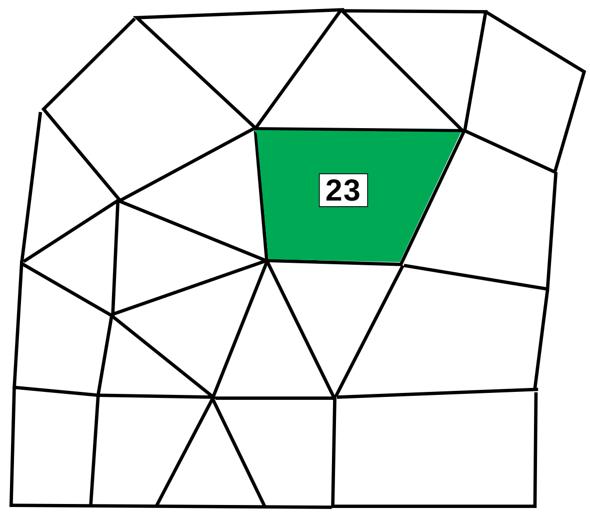
- Faces : list of vertices and list of edges



Algorithm description

Data for boundary meshes

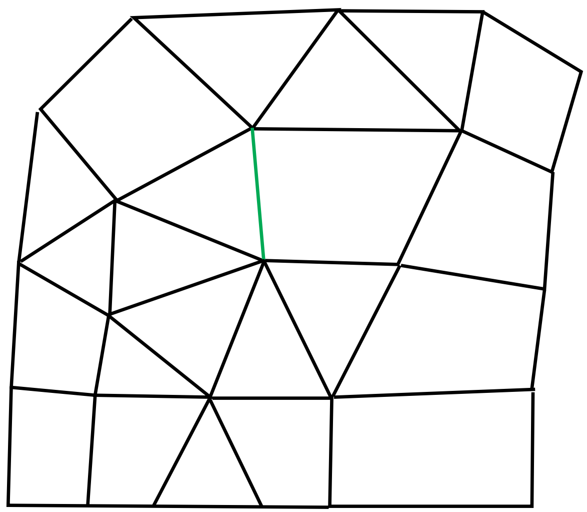
- Faces : list of vertices and list of edges
- Faces : left node number



Algorithm description

Data for boundary meshes

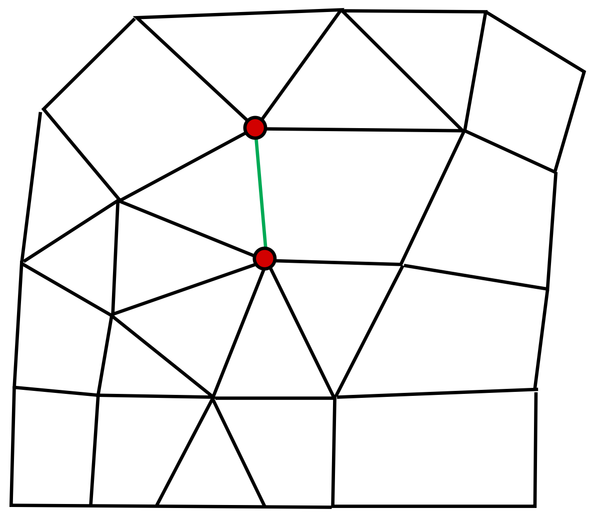
- Faces : list of vertices and list of edges
- Faces : left node number
- Edges :



Algorithm description

Data for boundary meshes

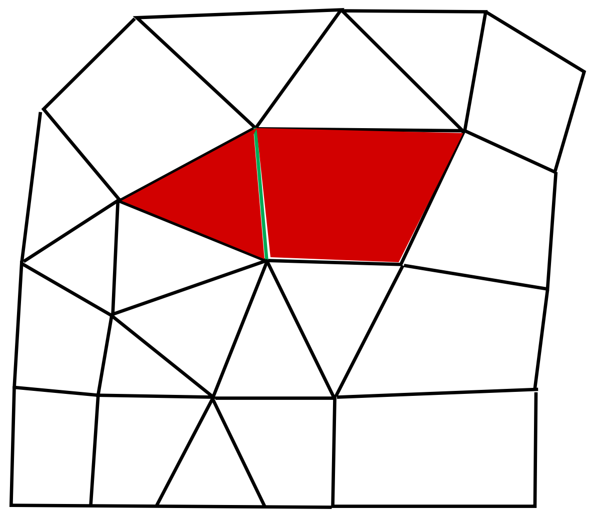
- Faces : list of vertices and list of edges
- Faces : left node number
- Edges : list of vertices



Algorithm description

Data for boundary meshes

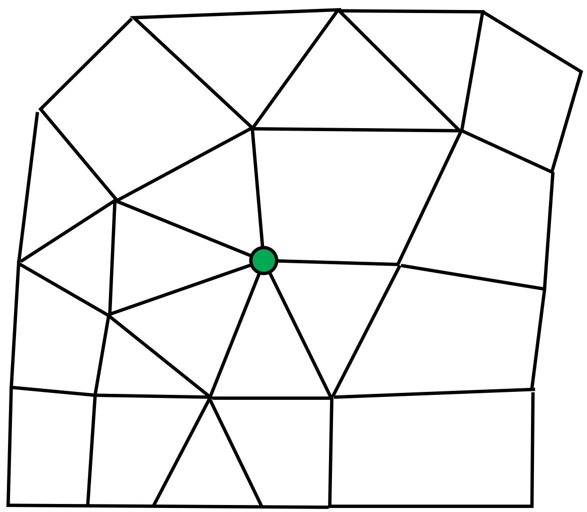
- Faces : list of vertices and list of edges
- Faces : left node number
- Edges : list of vertices and list of faces



Algorithm description

Data for boundary meshes

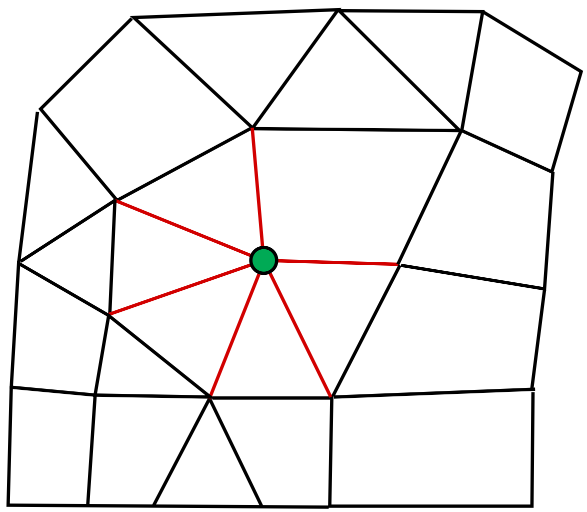
- Faces : list of vertices and list of edges
- Faces : left node number
- Edges : list of vertices and list of faces
- Vertices :



Algorithm description

Data for boundary meshes

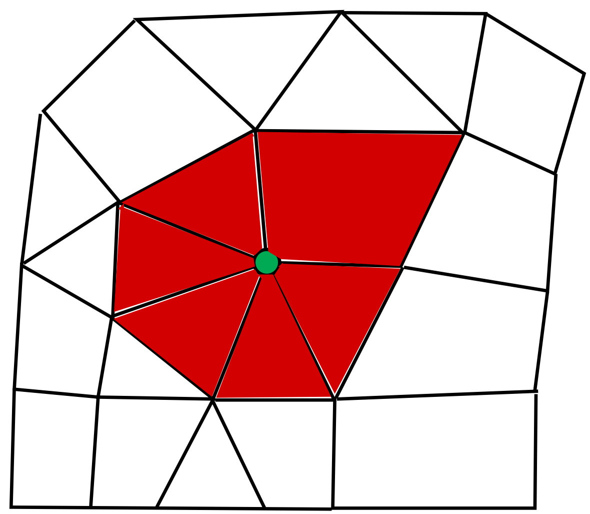
- Faces : list of vertices and list of edges
- Faces : left node number
- Edges : list of vertices and list of faces
- Vertices : list of edges



Algorithm description

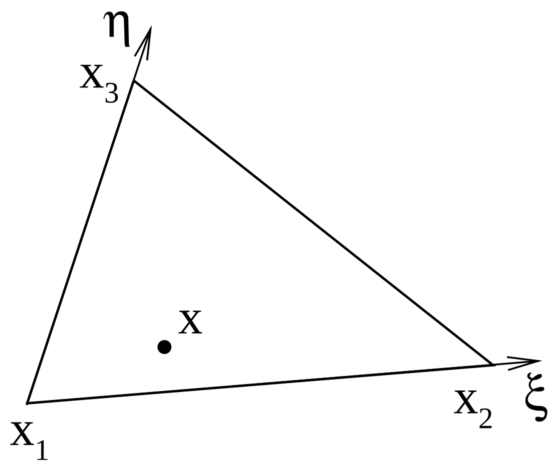
Data for boundary meshes

- Faces : list of vertices and list of edges
- Faces : left node number
- Edges : list of vertices and list of faces
- Vertices : list of edges and list of faces



Some Easy Problems

Is one vertex is in a given triangle ?



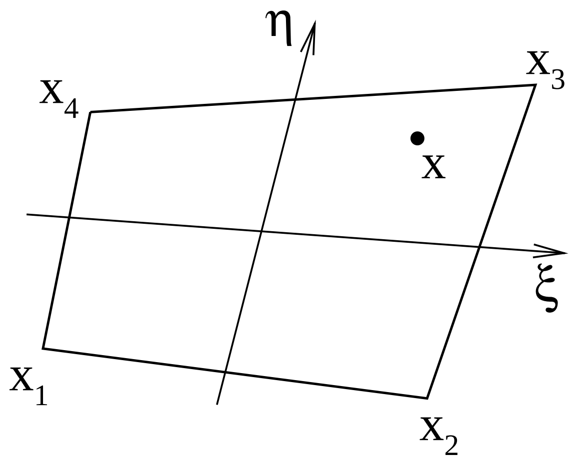
Knowing $\mathbf{x}$, find (ξ, η) parameters by solving

$$\mathbf{x}\,(\xi,\eta) = \mathbf{x}_1 + \xi\,(\mathbf{x_2} - \mathbf{x_1}) + \eta\,(\mathbf{x}_3 - \mathbf{x}_1)$$

if $\xi > 0$ and $\eta > 0$ and $1 - \xi - \eta < 0$, vertex is in triangle.

Some Easy Problems

Is one vertex is in a given quadrangle ?



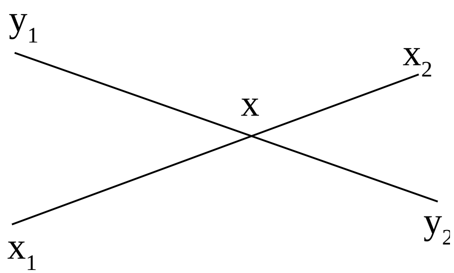
Knowing $\mathbf{x}$, find (ξ, η) parameters by solving

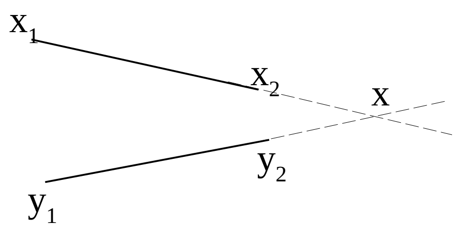
$$\begin{aligned}\mathbf{x}(\xi,\eta) = \quad & \tfrac{1}{4}(1-\eta)\,(1-\xi)\,\mathbf{x}_1 + \tfrac{1}{4}(1-\eta)\,(1+\xi)\,\mathbf{x}_2 \\ & +\tfrac{1}{4}(1+\eta)\,(1+\xi)\,\mathbf{x}_3 + \tfrac{1}{4}(1+\eta)\,(1-\xi)\,\mathbf{x}_4\end{aligned}$$

if $-1 < \xi < 1$ and $-1 < \eta < 1$, vertex is in quadrangle.

Some Easy Problems

Do two edges intersect each other ?





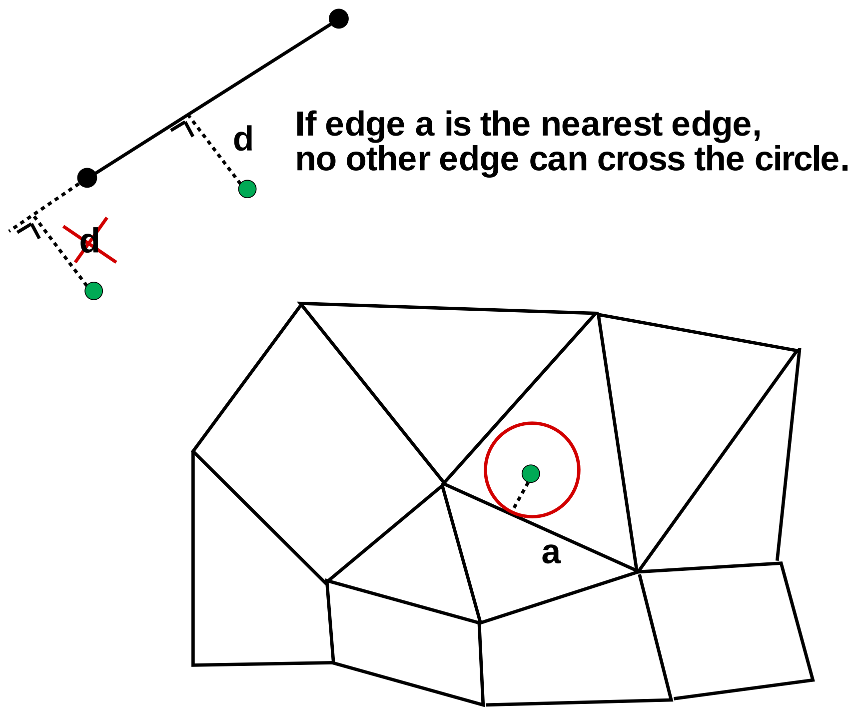
Find (ξ, η) parameters such as

$$\frac{1}{2}(1-\xi)\ \mathbf{x}_1 + \frac{1}{2}(1+\xi)\ \mathbf{x}_2 = \frac{1}{2}(1-\eta)\ \mathbf{y}_1 + \frac{1}{2}(1+\eta)\ \mathbf{y}_2$$

if $-1 < \xi < 1$ and $-1 < \eta < 1$, the intersection exists.

Interface treatment algorithm

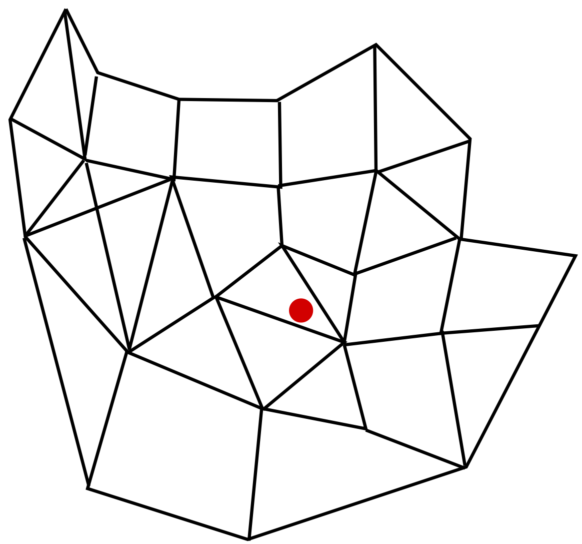
Locate mesh B vertices in mesh A faces
Compute distances between the vertex and all edges.



$\Rightarrow$ If edge a is the nearest edge then the vertex stands in one of the neighbouring faces of edge a.

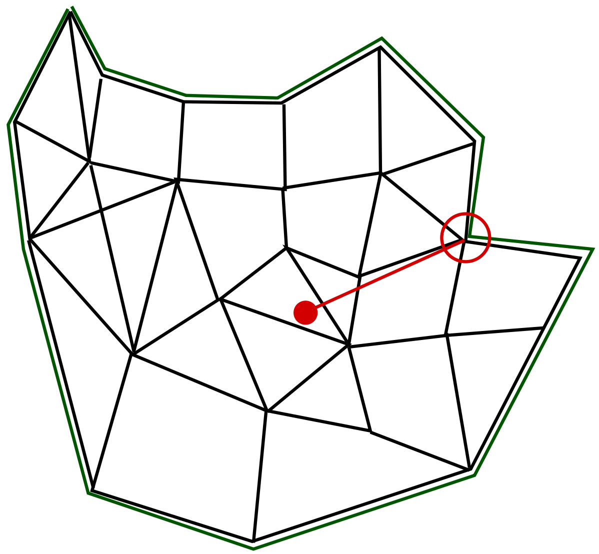
Interface treatment algorithm

Locate mesh B vertices in mesh A faces



Interface treatment algorithm

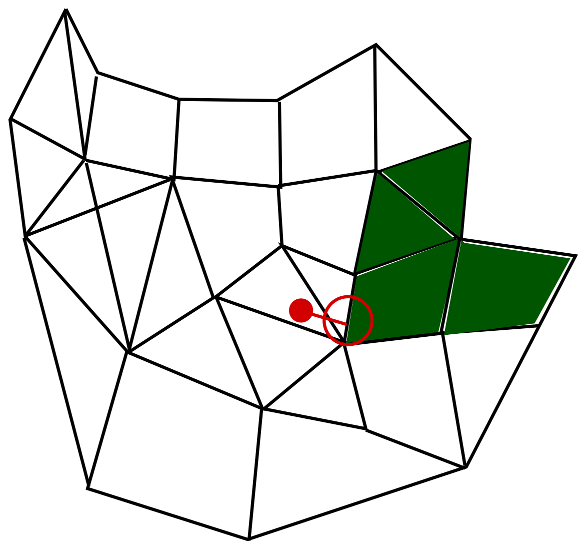
Locate mesh B vertices in mesh A faces



Find nearest boundary edge or vertex

Interface treatment algorithm

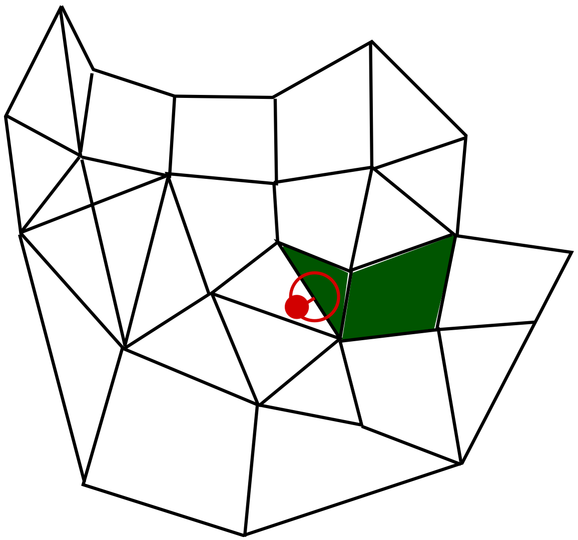
Locate mesh B vertices in mesh A faces



Find nearest surrounding edge or vertex
Repeat operation

Interface treatment algorithm

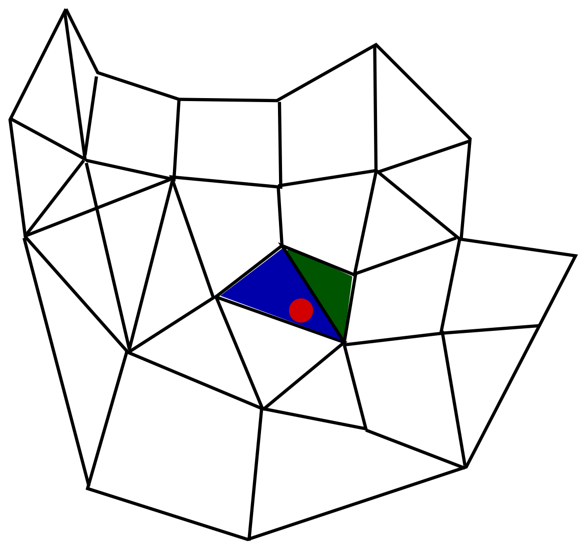
Locate mesh B vertices in mesh A faces



Find nearest surrounding edge or vertex
Repeat operation

Interface treatment algorithm

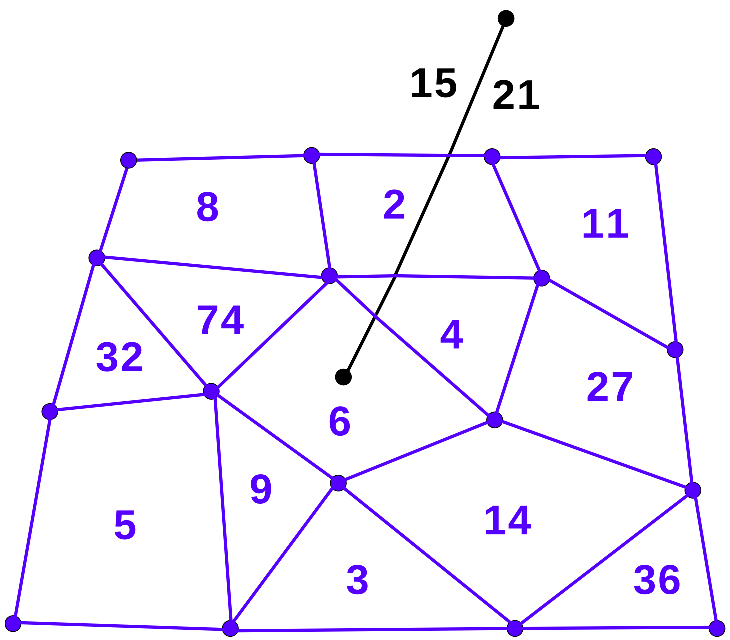
Locate mesh B vertices in mesh A faces



Check if the vertex is in one of the surrounding faces

Interface treatment algorithm

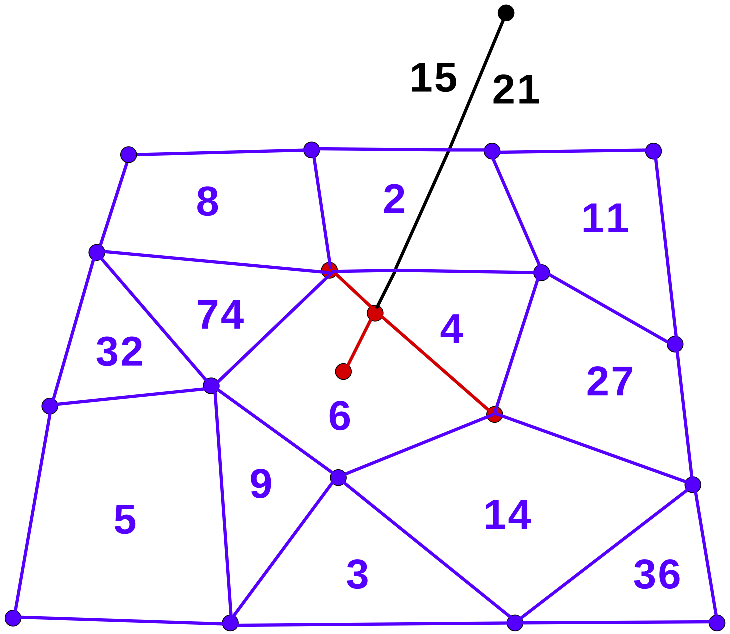
Cut egdes and build new mesh C



Find intersections, cut edges and create new vertices for new mesh C.

Interface treatment algorithm

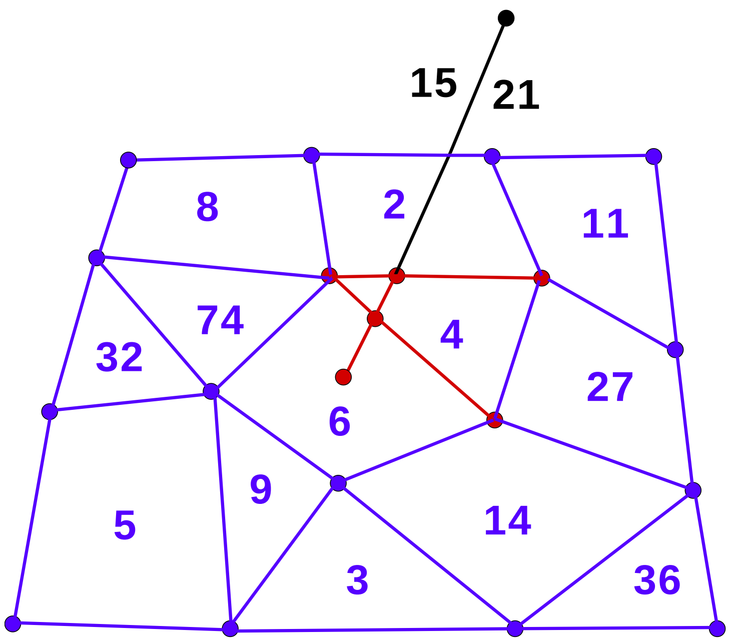
Cut egdes and build new mesh C



Find intersections, cut edges and create new vertices for new mesh C.

Interface treatment algorithm

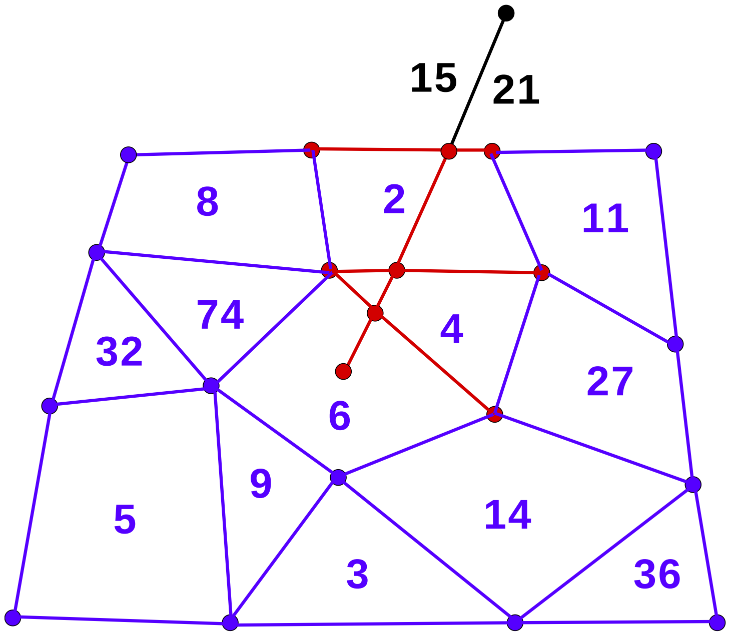
Cut egdes and build new mesh C



Find intersections, cut edges and create new vertices for new mesh C.

Interface treatment algorithm

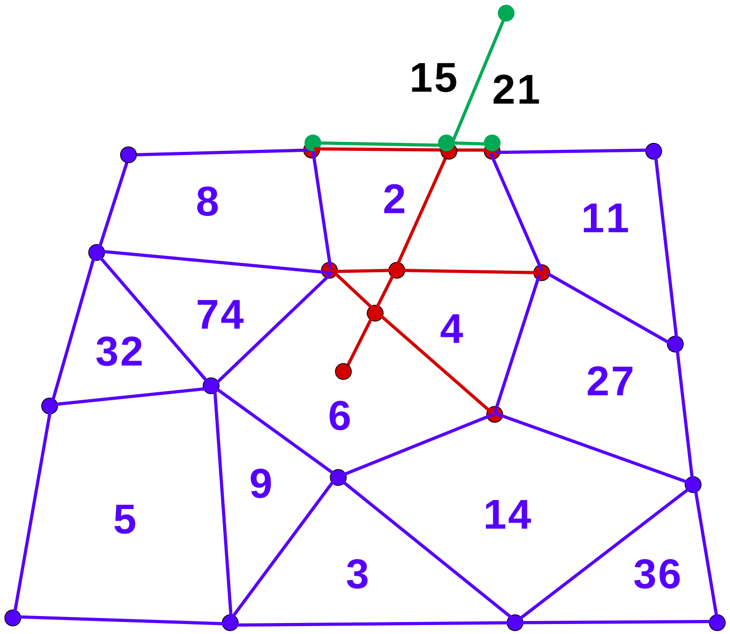
Cut egdes and build new mesh C



Find intersections, cut edges and create new vertices for new mesh C.

Interface treatment algorithm

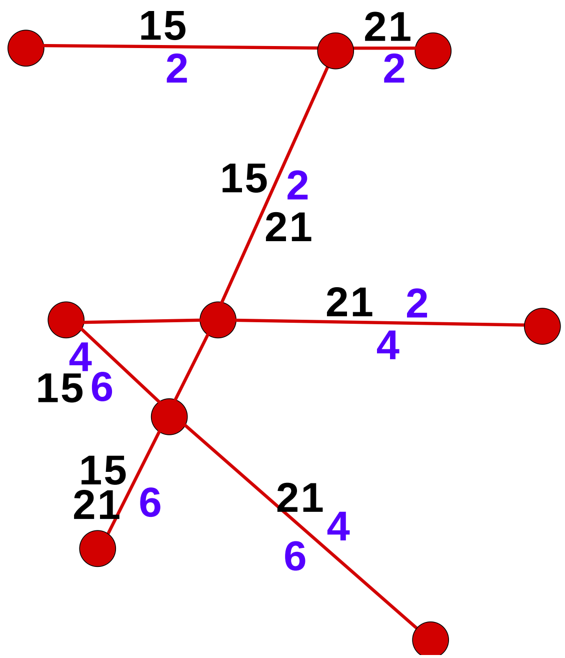
Cut egdes and build new mesh C

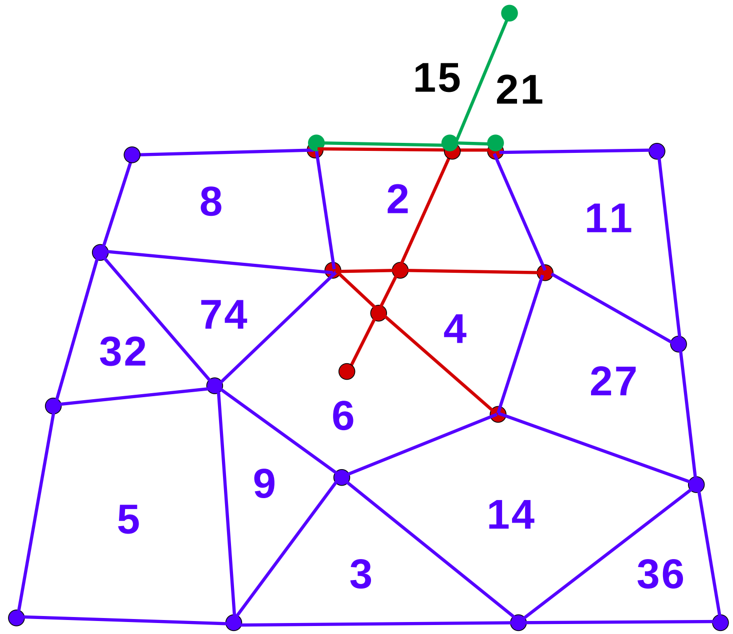


Find intersections, cut edges and create new vertices for new mesh C.

Interface treatment algorithm

Cut egdes and build new mesh C

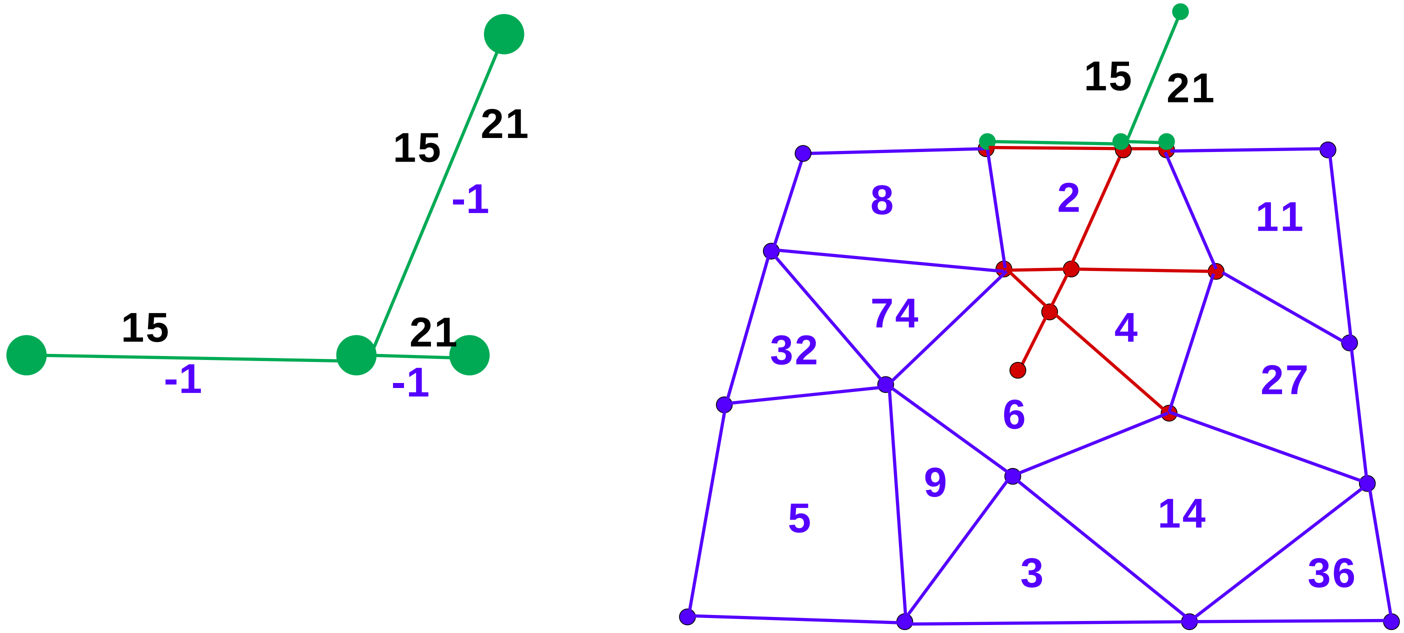




Store for each new edge the identification number of neighbour nodes

Interface treatment algorithm

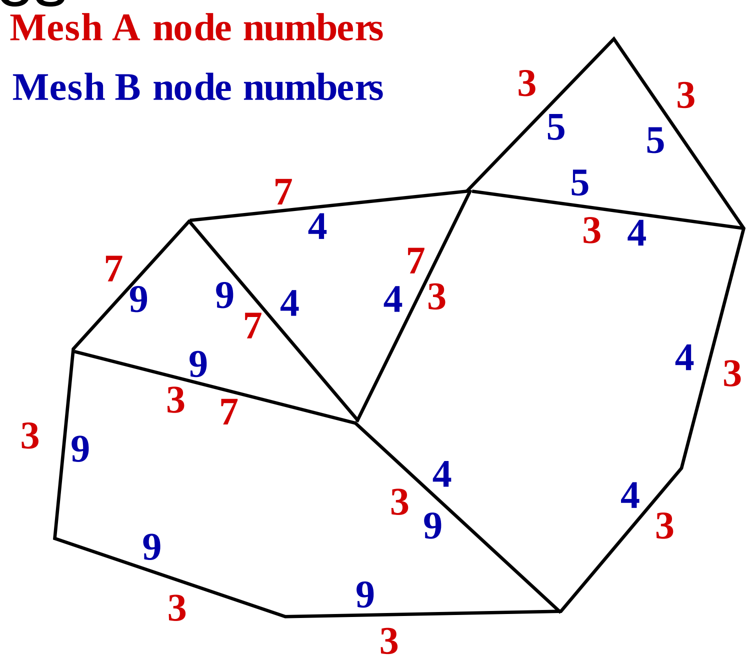
Cut egdes and build new mesh C



Store for each new edge the identification number of neighbour nodes

Interface treatment algorithm

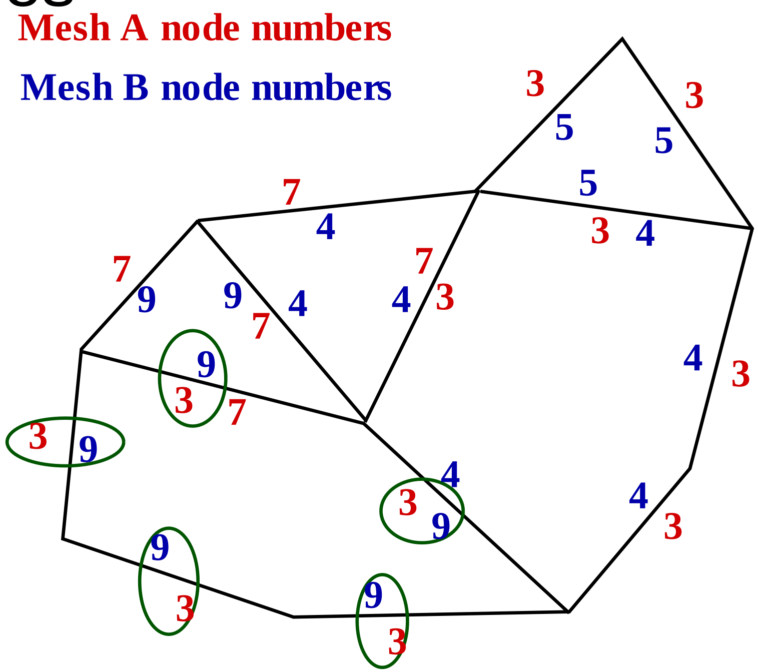
Build macro faces having one left and one right neighbour nodes



Create macro face with edges having the same nodes in mesh A and B.

Interface treatment algorithm

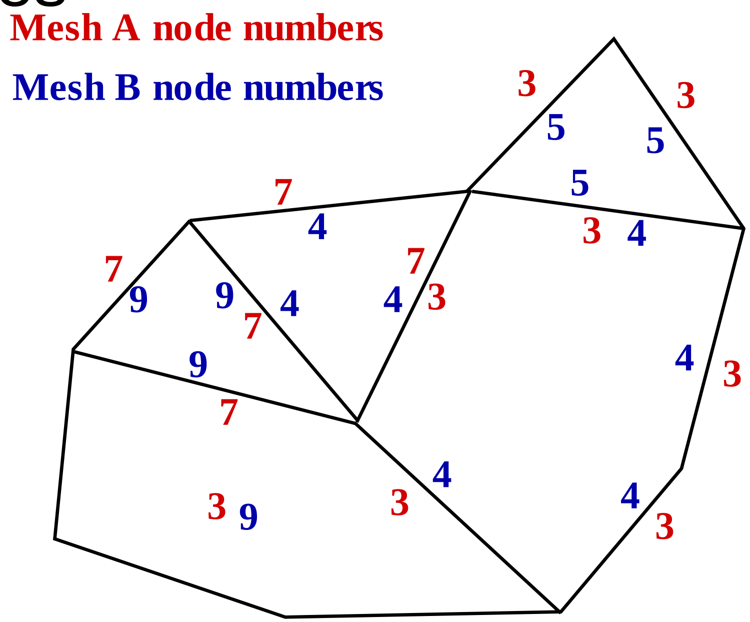
Build macro faces having one left and one right neighbour nodes



Create macro face with edges having the same nodes in mesh A and B.

Interface treatment algorithm

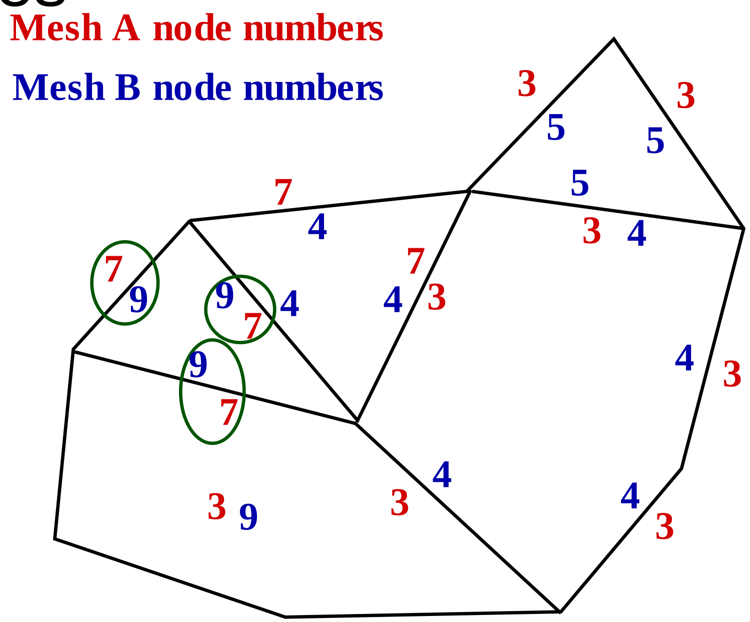
Build macro faces having one left and one right neighbour nodes



Create macro face with edges having the same nodes in mesh A and B.

Interface treatment algorithm

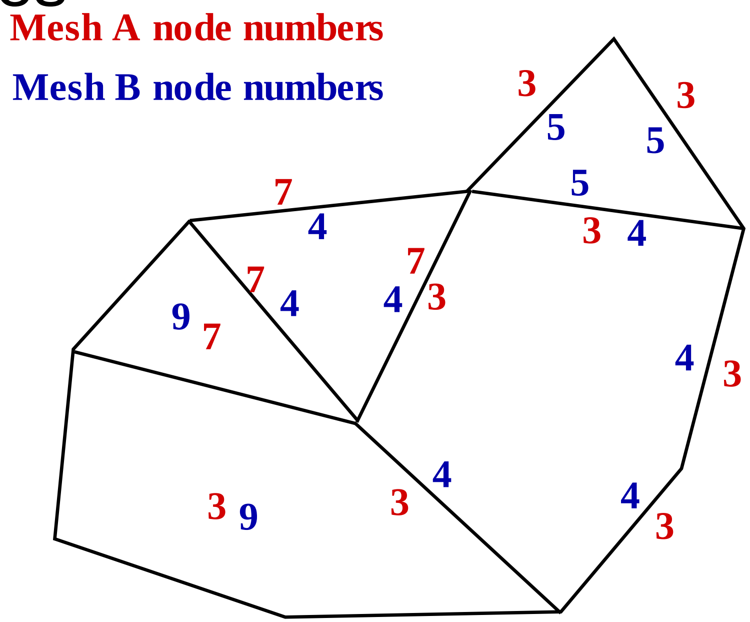
Build macro faces having one left and one right neighbour nodes



Create macro face with edges having the same nodes in mesh A and B.

Interface treatment algorithm

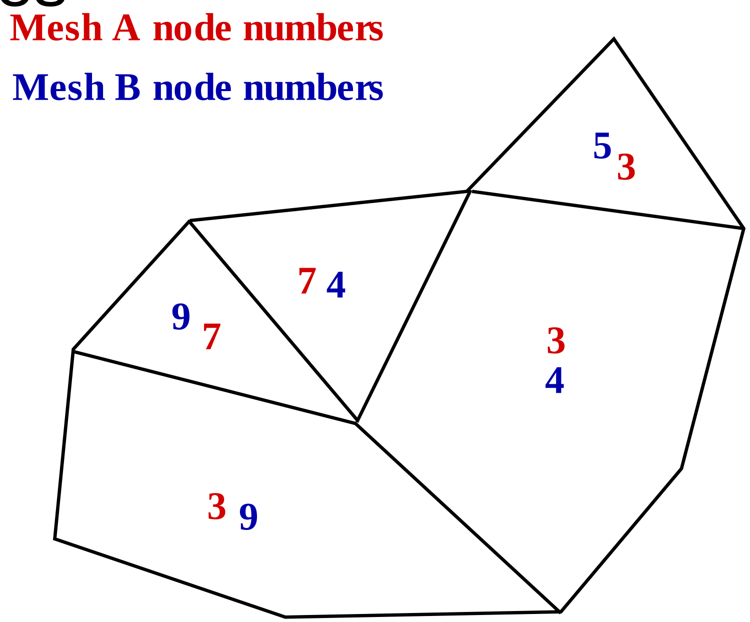
Build macro faces having one left and one right neighbour nodes



Create macro face with edges having the same nodes in mesh A and B.

Interface treatment algorithm

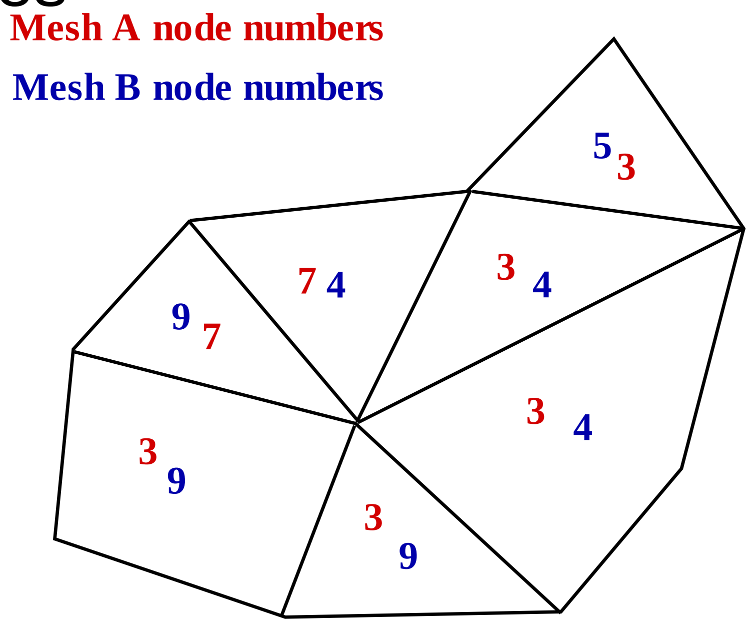
Build macro faces having one left and one right neighbour nodes



Create macro face with edges having the same nodes in mesh A and B.

Interface treatment algorithm

Build macro faces having one left and one right neighbour nodes



Create macro face with edges having the same nodes in mesh A and B.
Cut macro faces into triangles or quadrangles

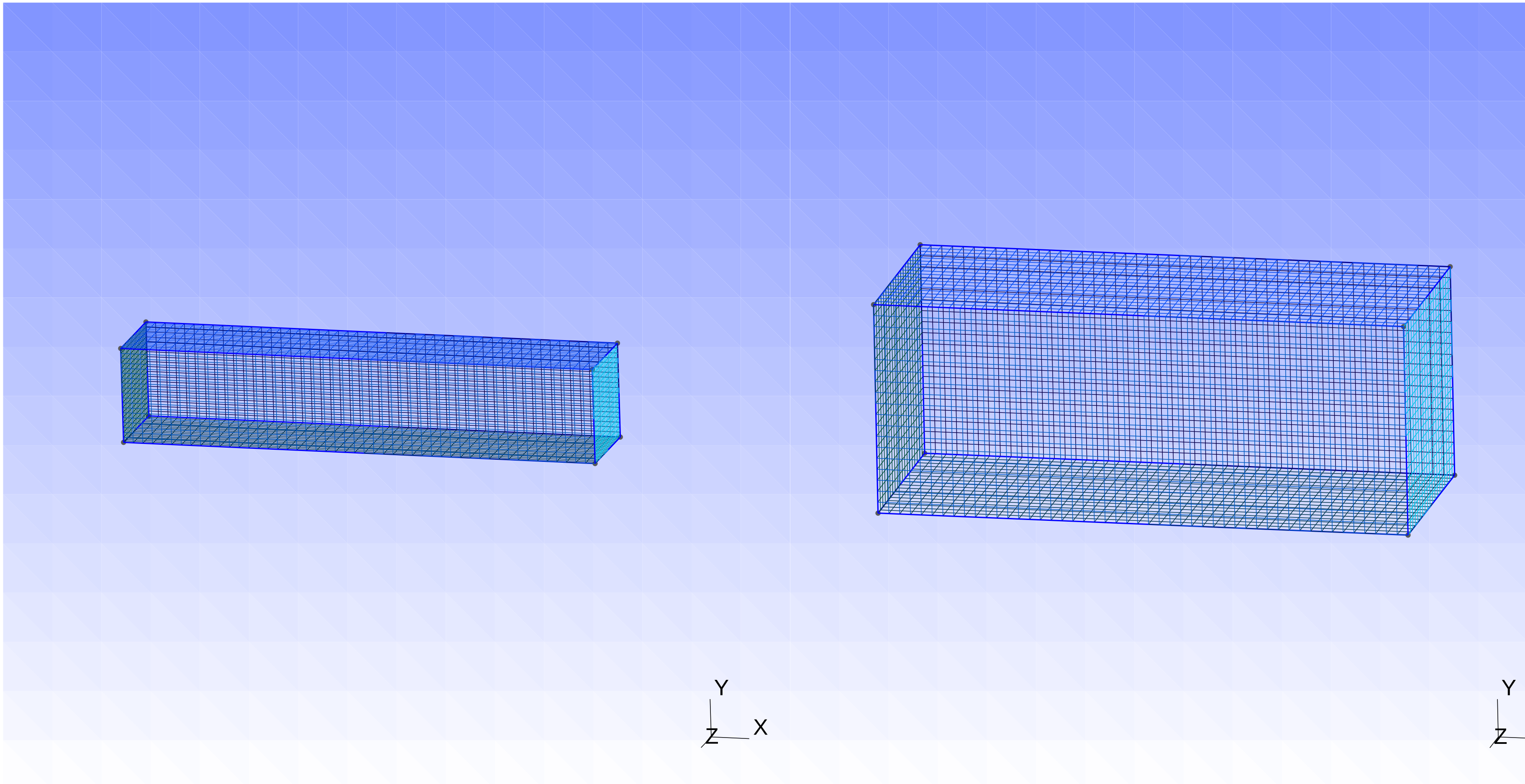
Algorithm description

Parallel computation

- Impose all nodes next to an interface to stand on the same processor
- Distribute interfaces on different processors
- Partition all the other nodes with Metis

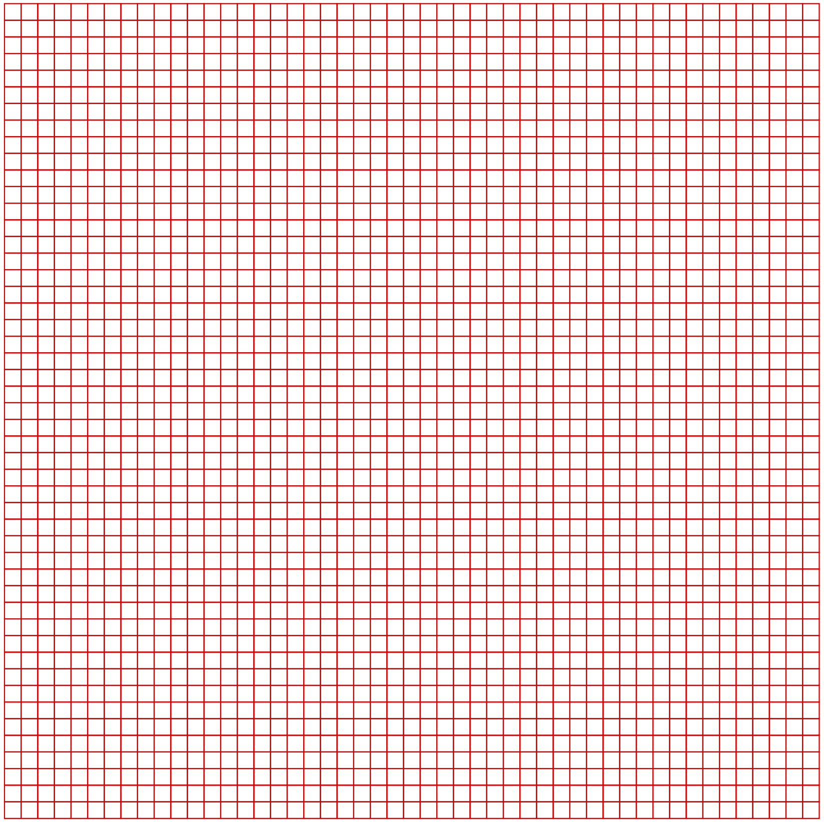
Duct flow

Two mesh blocks - Boundary must be detected



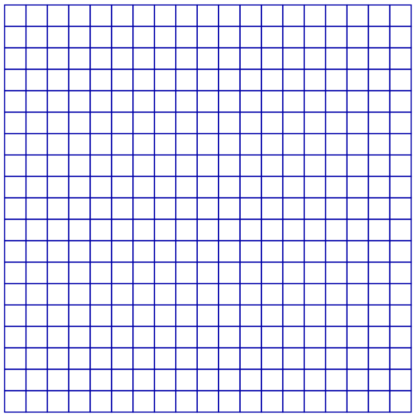
Duct flow

One left mesh



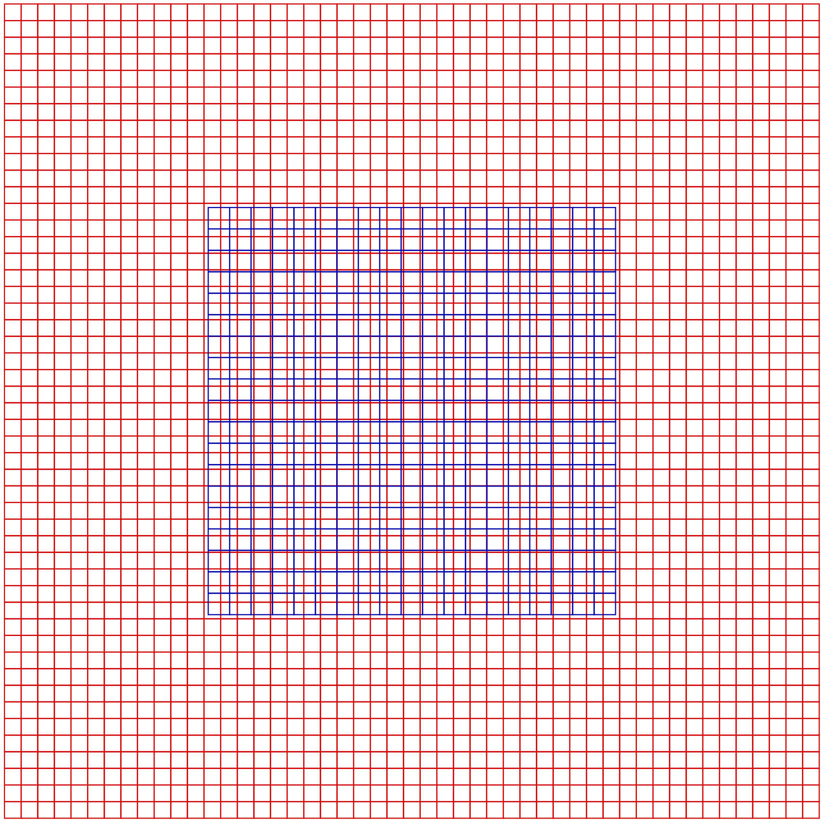
Duct flow

One right mesh



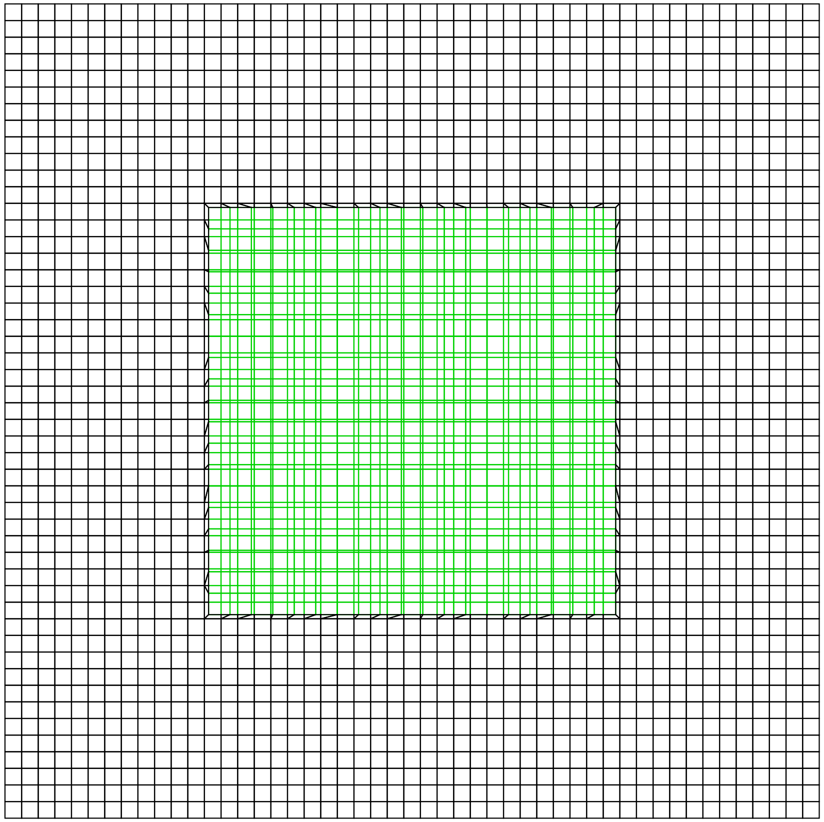
Duct flow

Interface to be computed and boundary to be detected



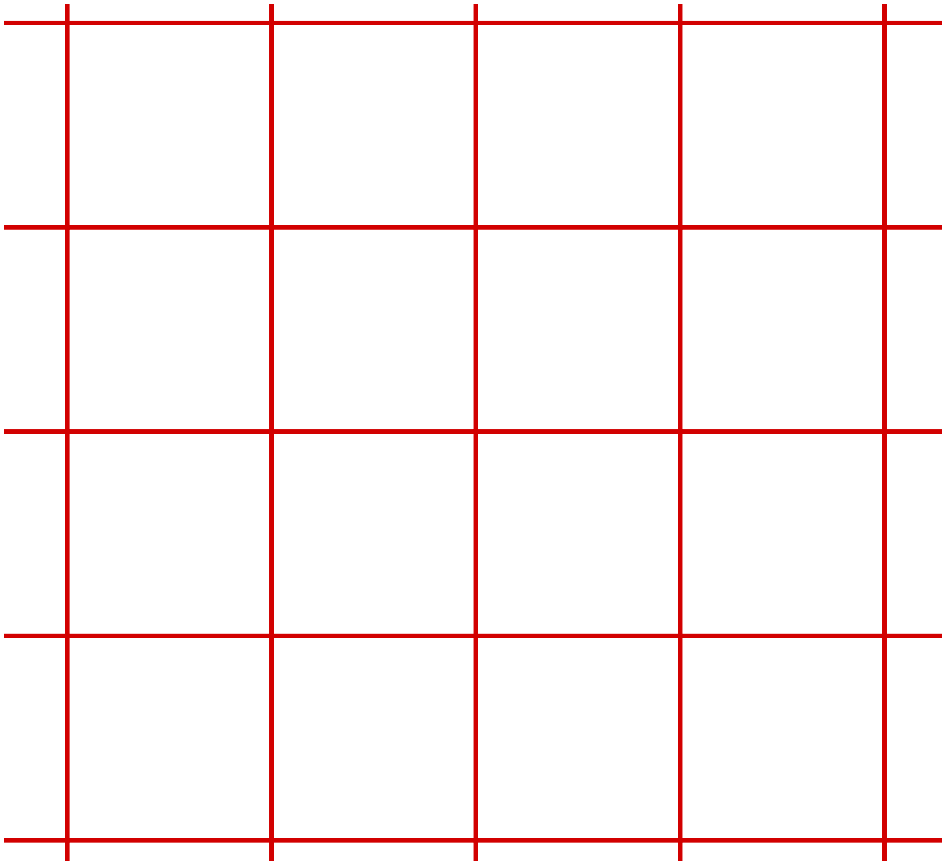
Duct flow

Result



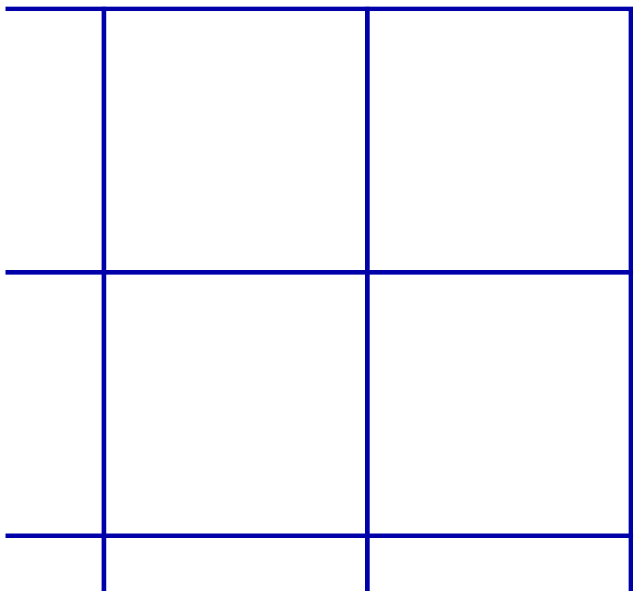
Duct flow

One left mesh



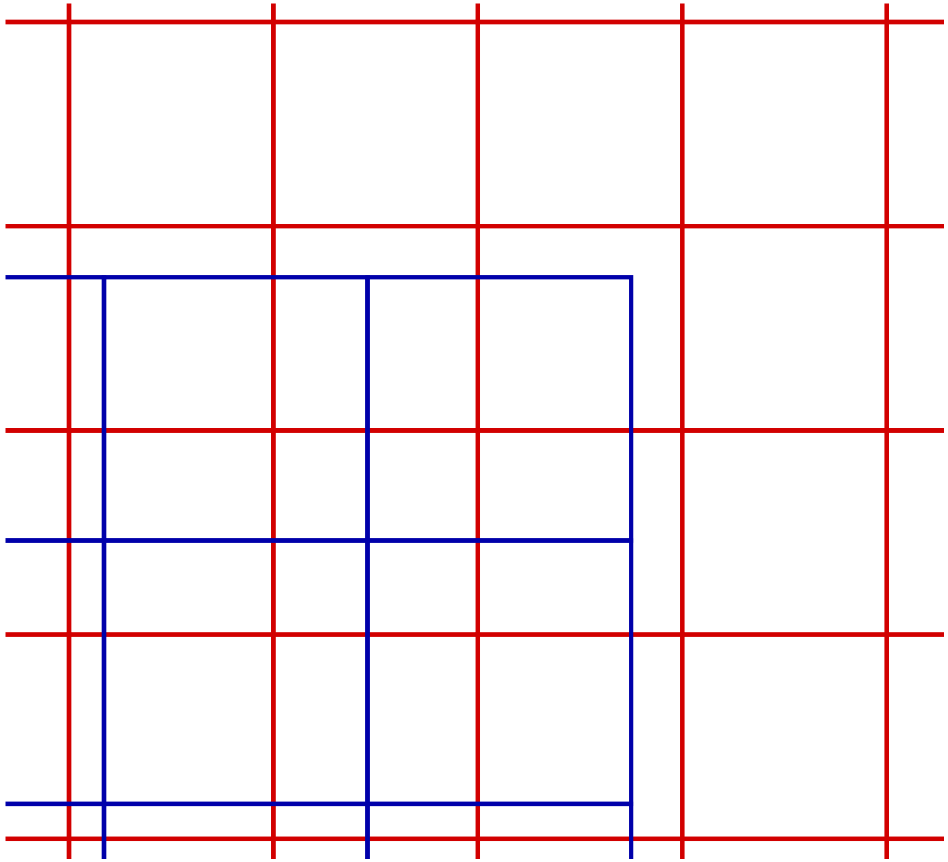
Duct flow

One right mesh



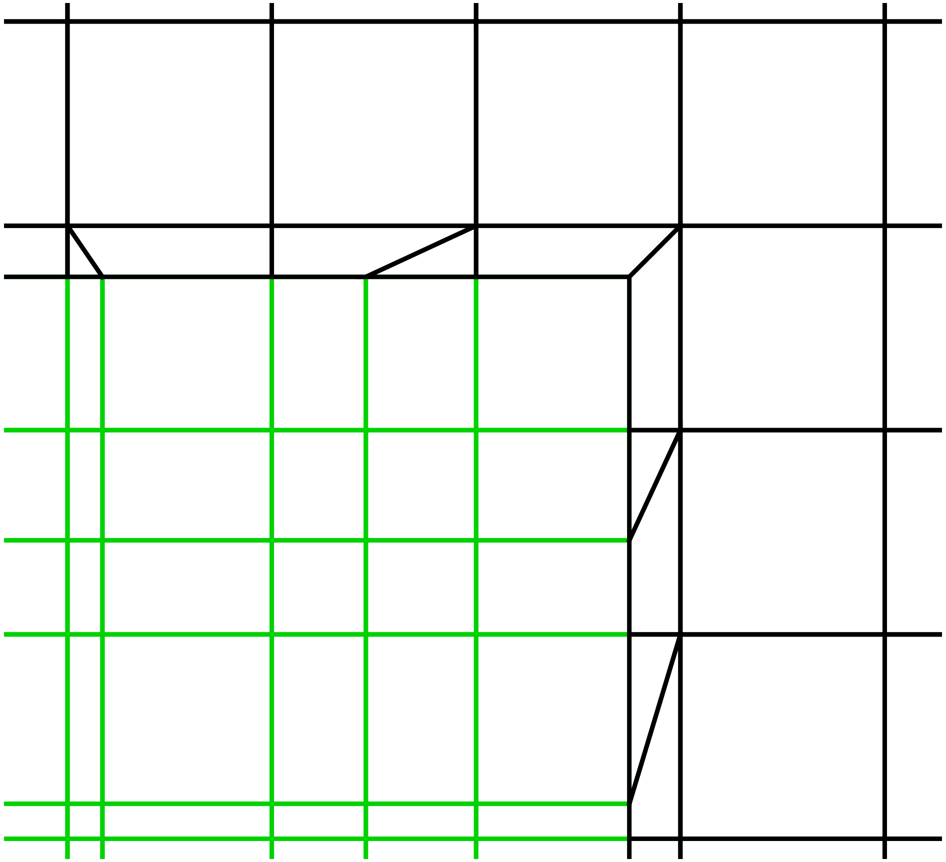
Duct flow

Interface to be computed and boundary to be detected



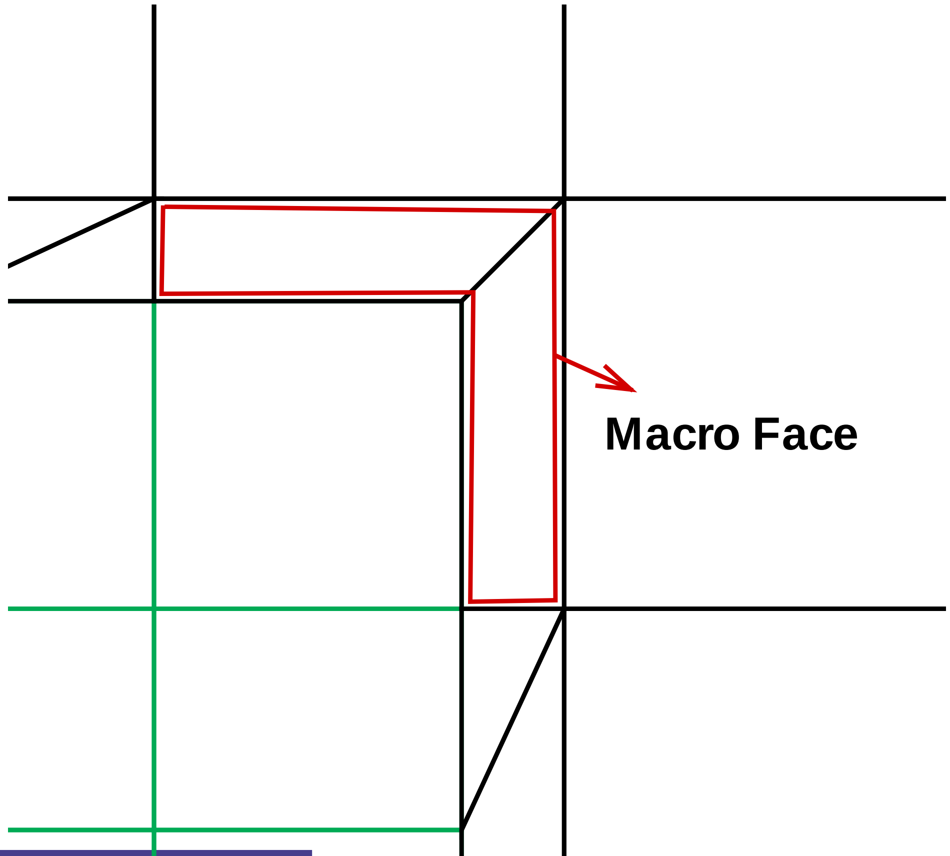
Duct flow

Result

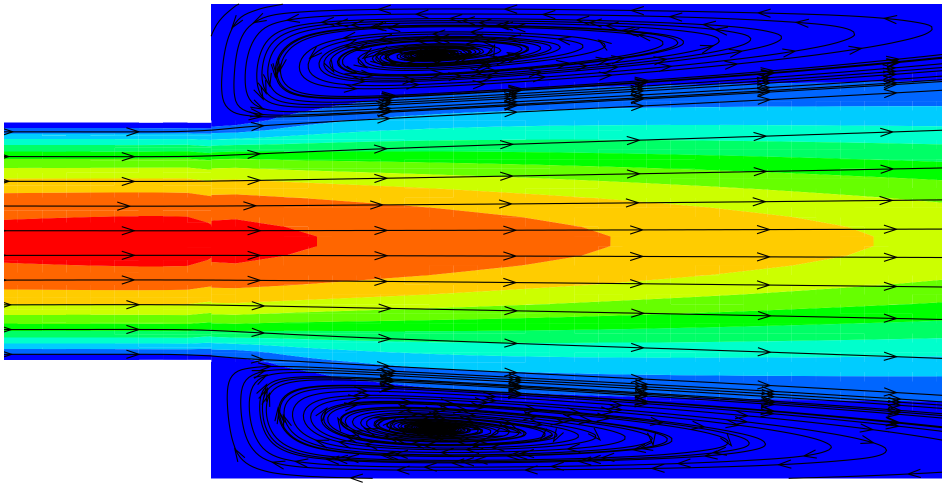


Duct flow

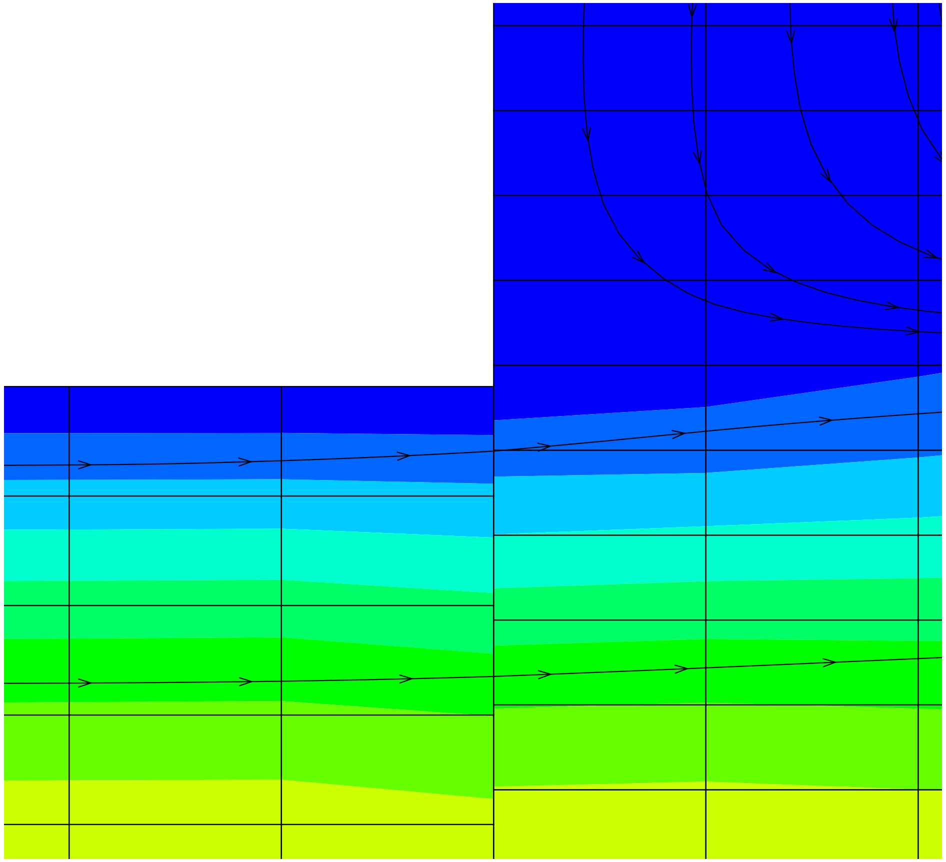
Macro face treatment



Duct flow

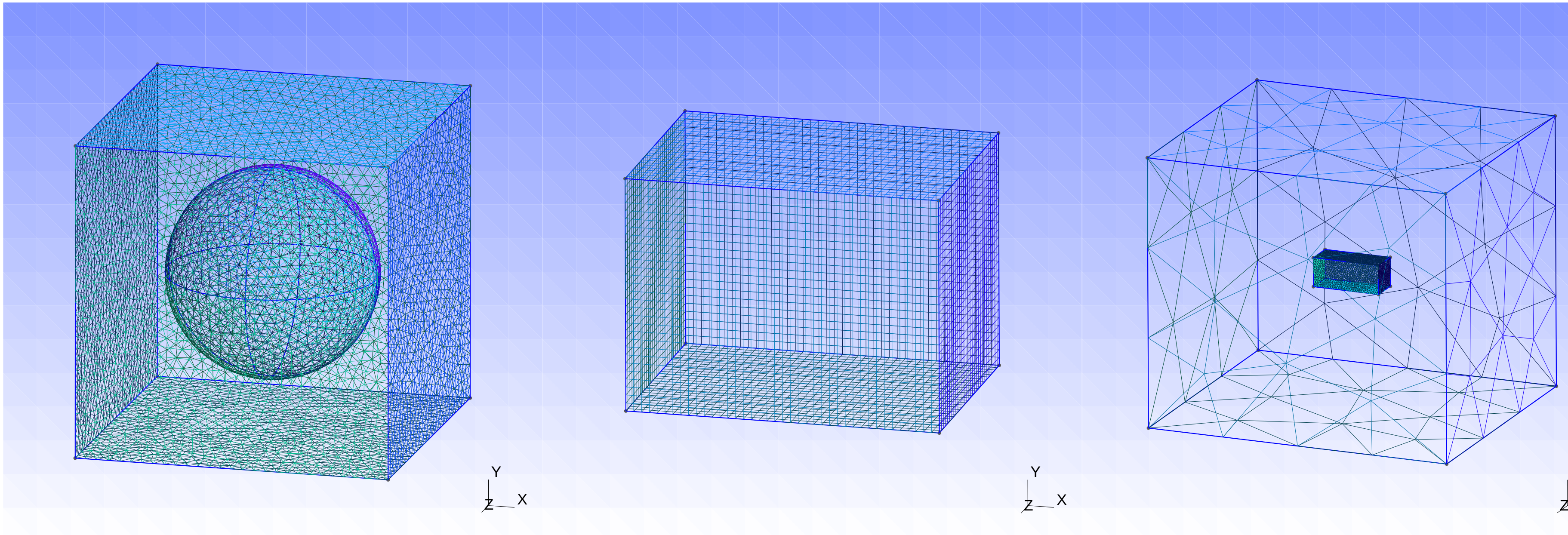


Duct flow



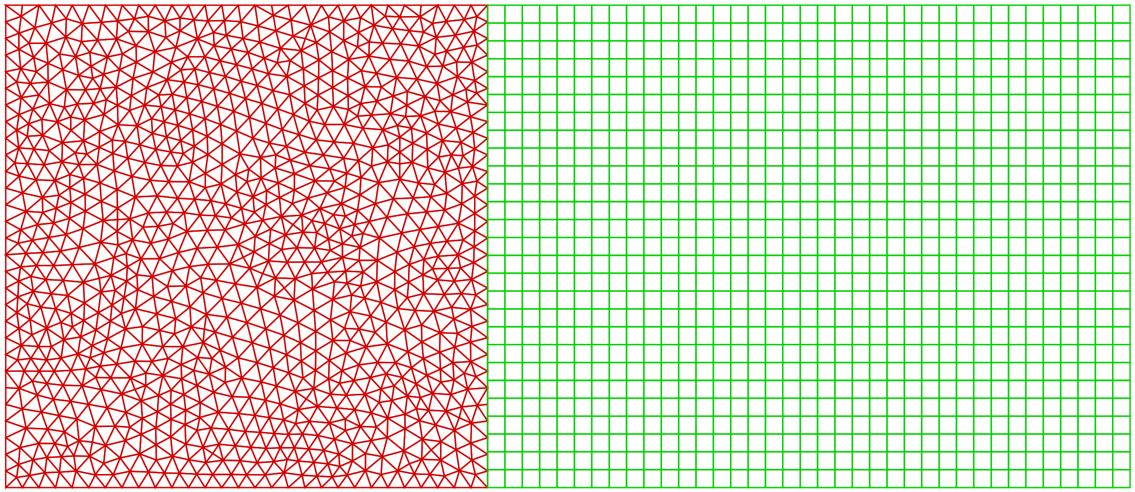
Flow past a sphere

Three mesh blocks



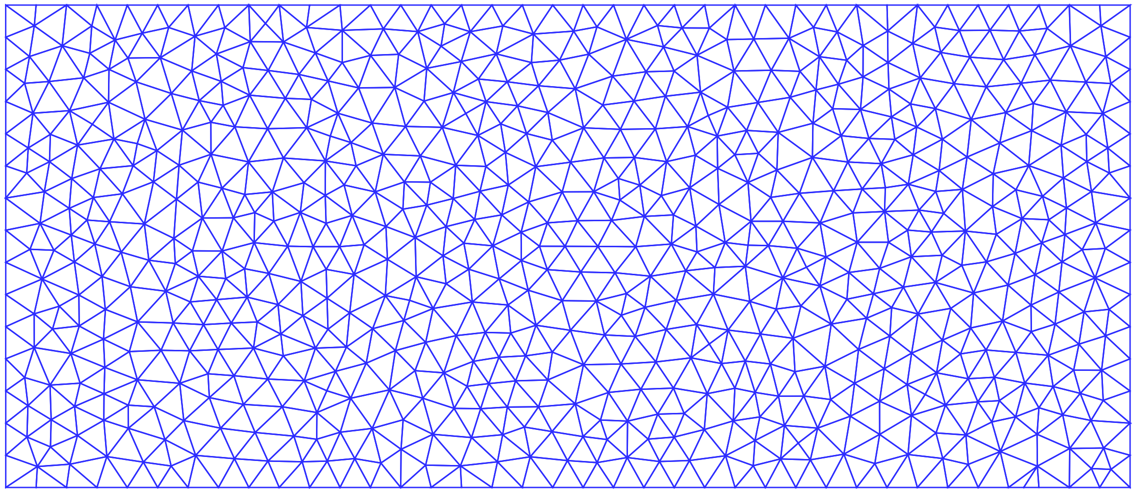
Flow past a sphere

Two left meshes



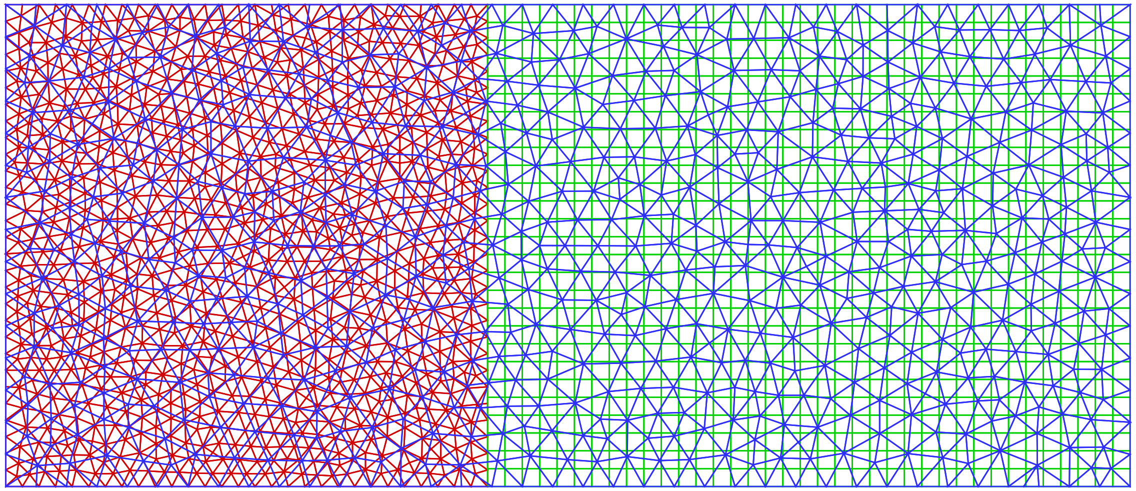
Flow past a sphere

One right mesh



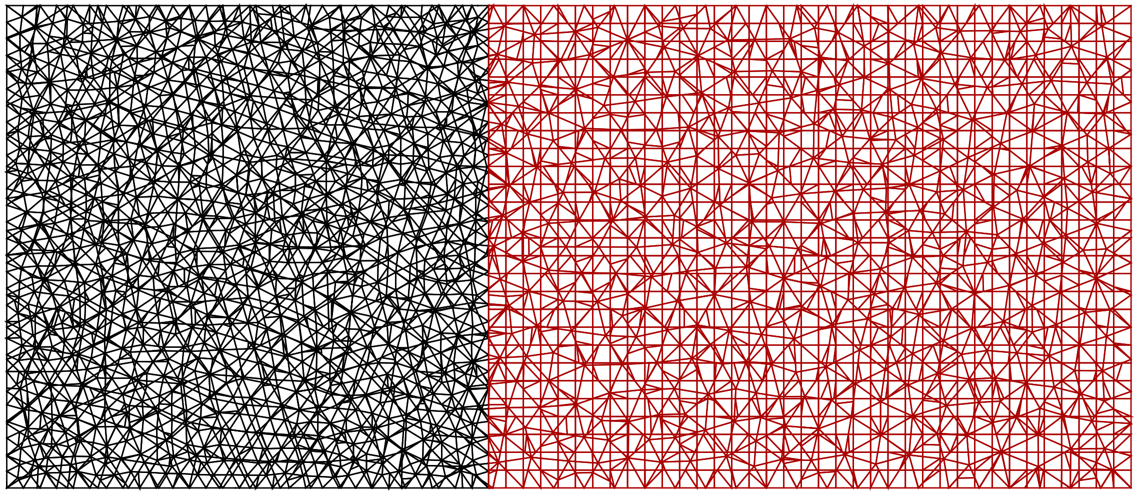
Flow past a sphere

Interface to be computed



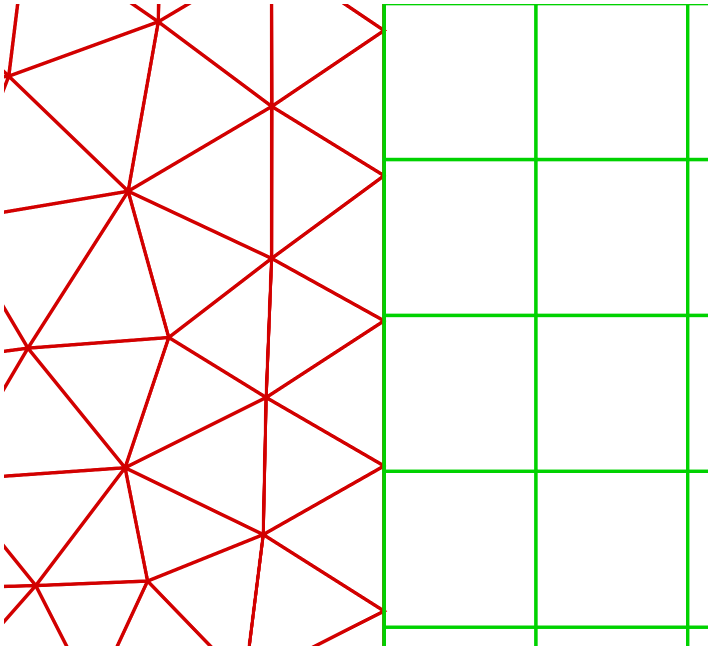
Flow past a sphere

Result



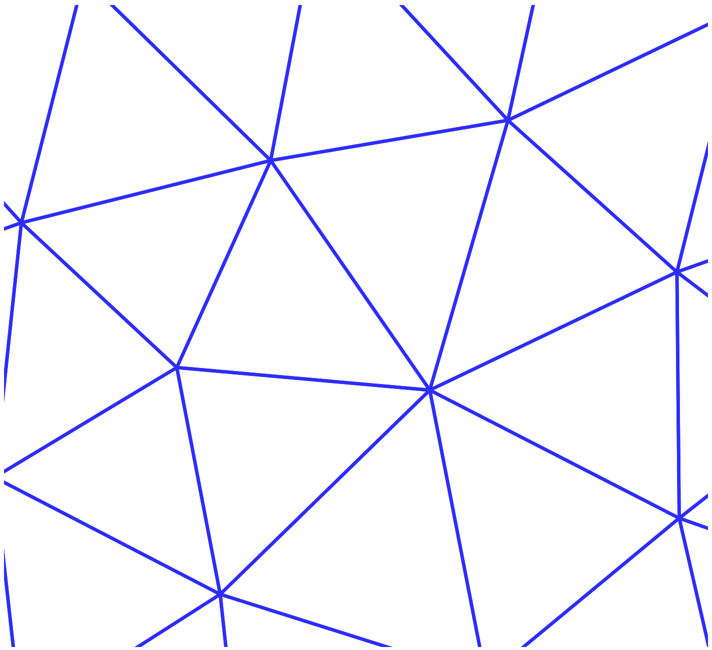
Flow past a sphere

Two left meshes



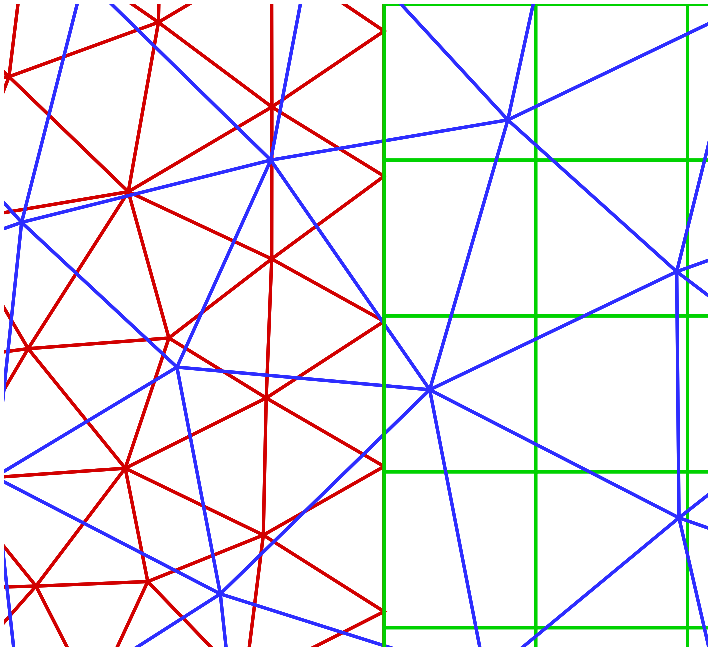
Flow past a sphere

One right mesh



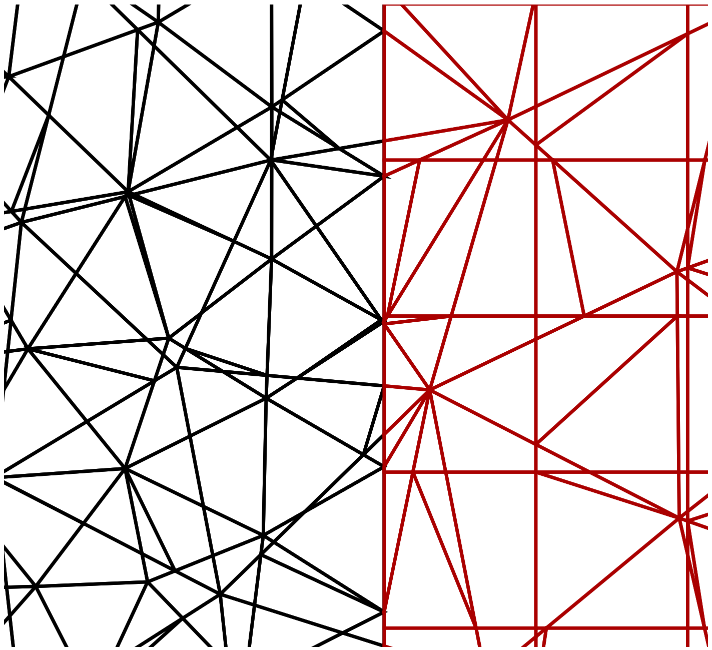
Flow past a sphere

Interface to be computed

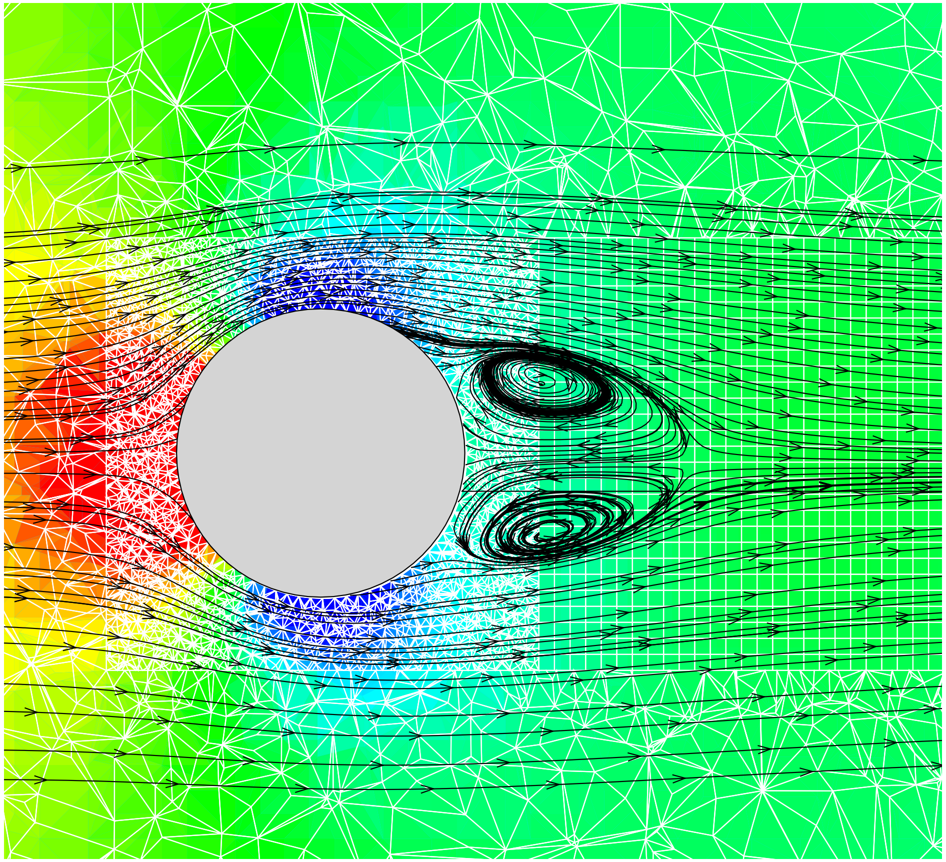


Flow past a sphere

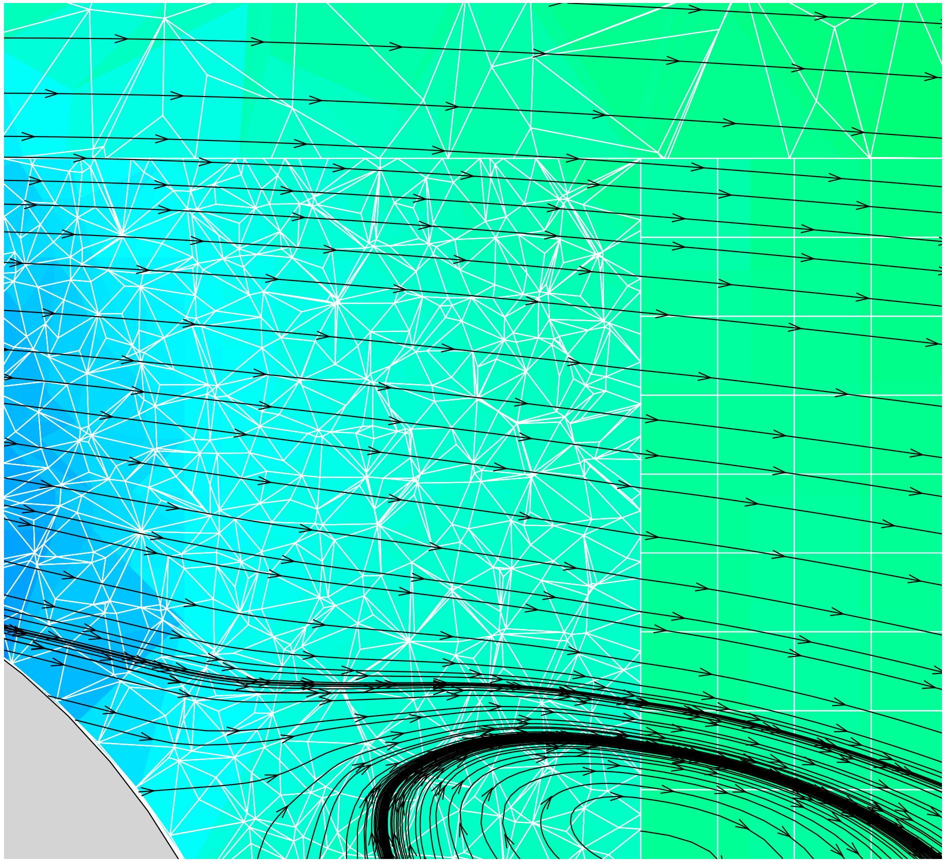
Result



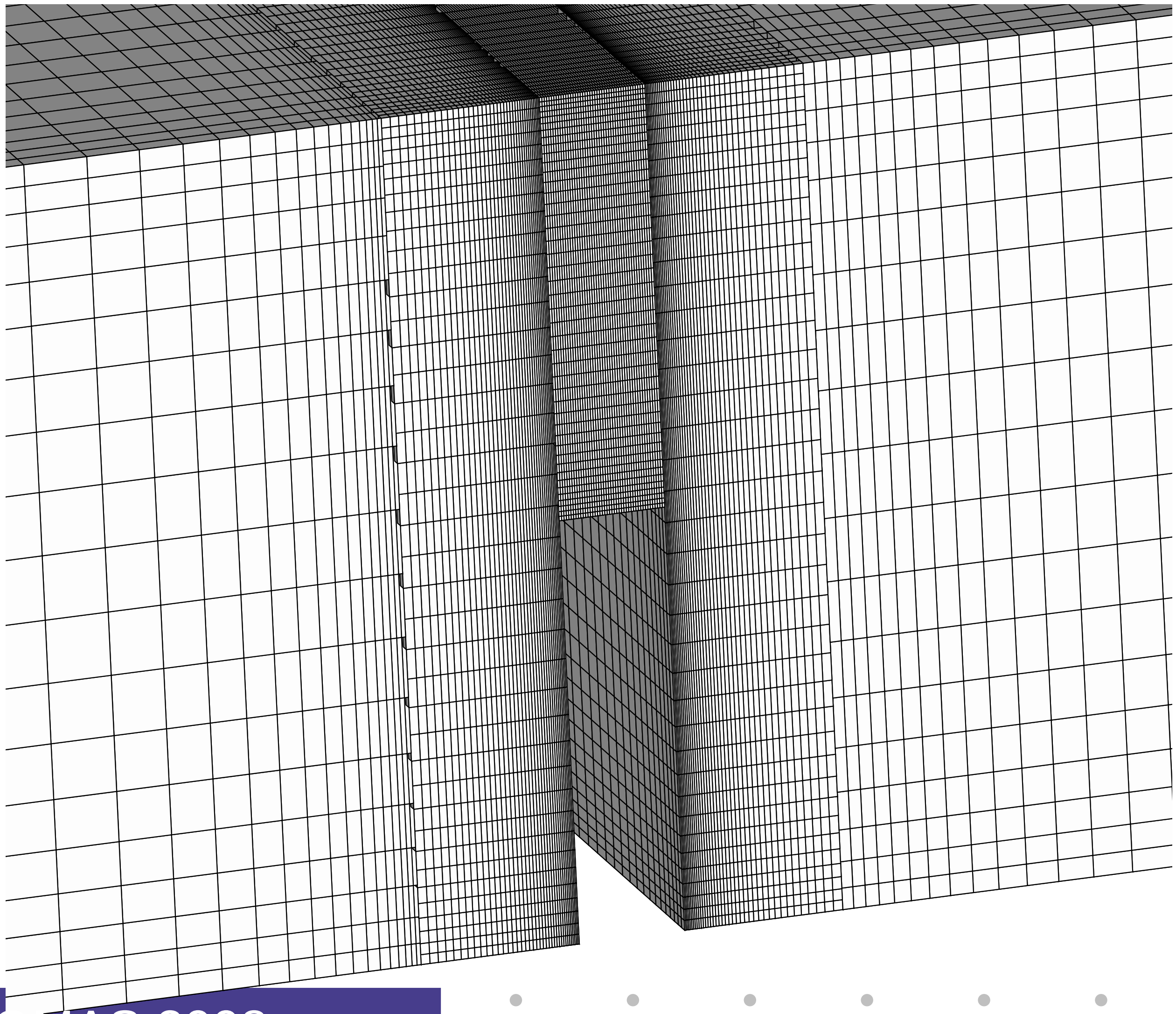
Flow past a sphere



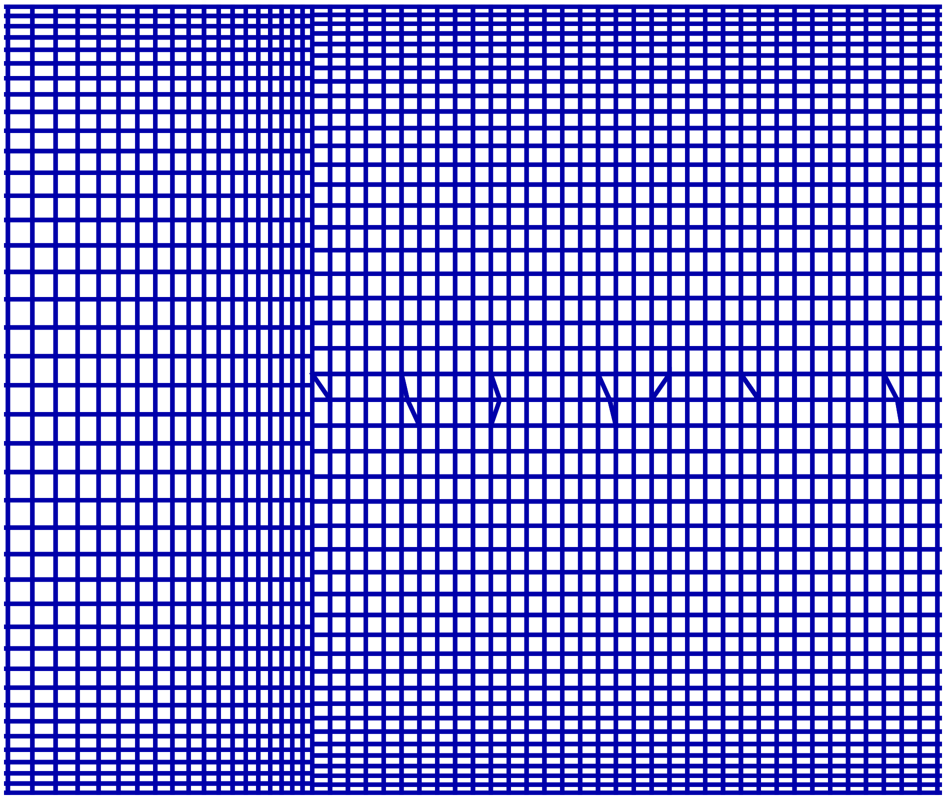
Flow past a sphere



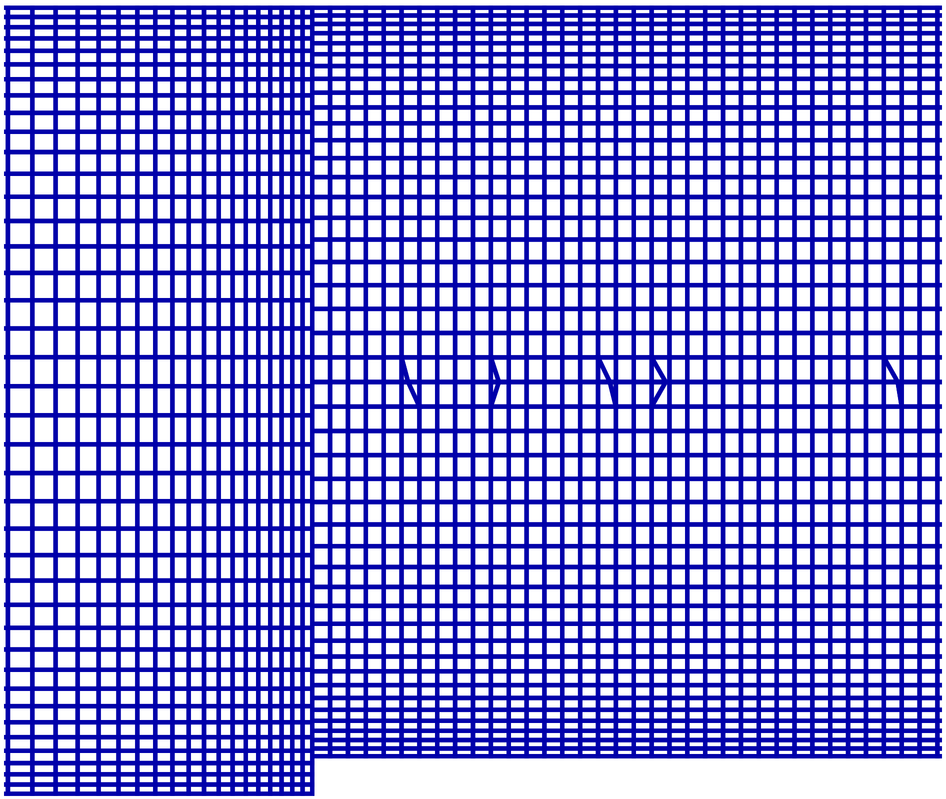
Unsteady flow with moving boundary



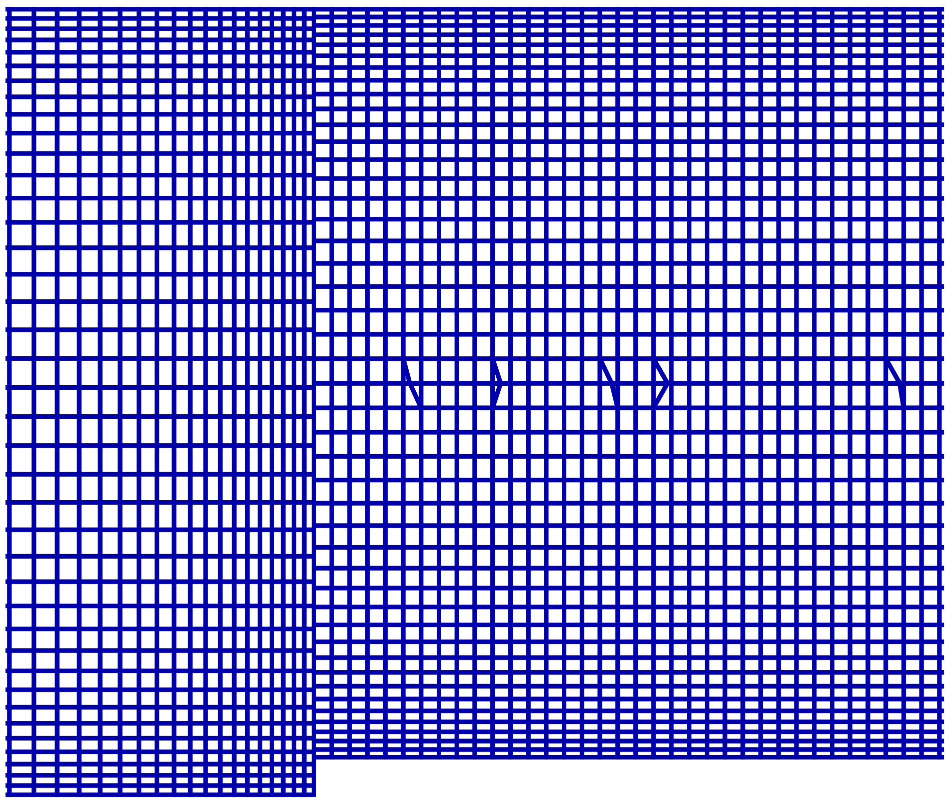
Unsteady flow with moving boundary



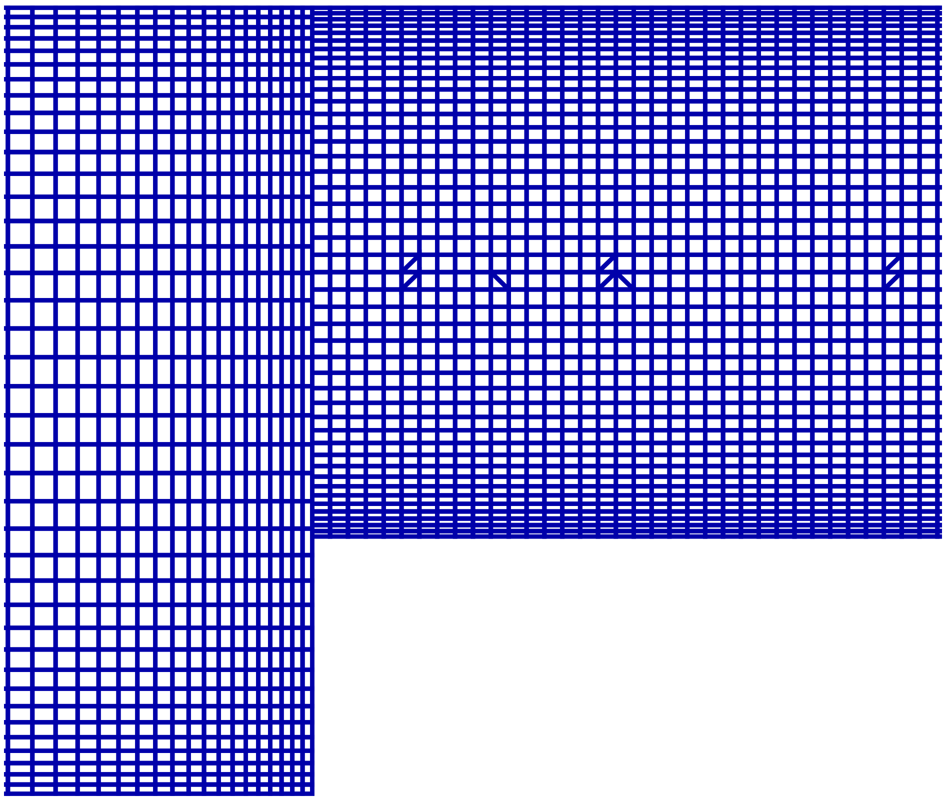
Unsteady flow with moving boundary



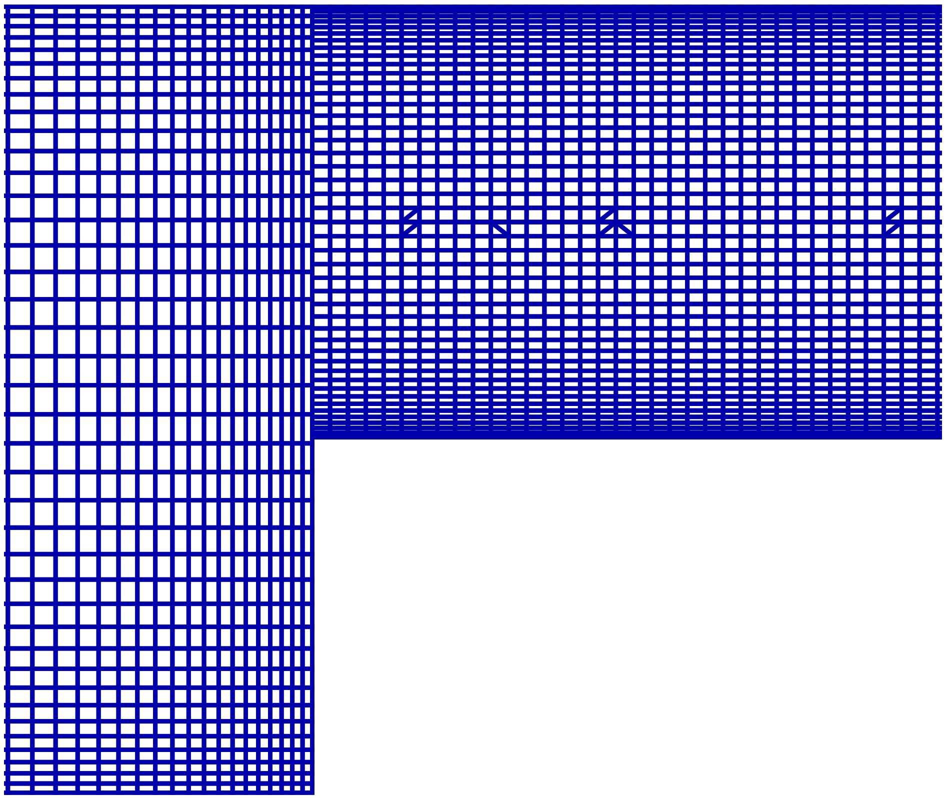
Unsteady flow with moving boundary



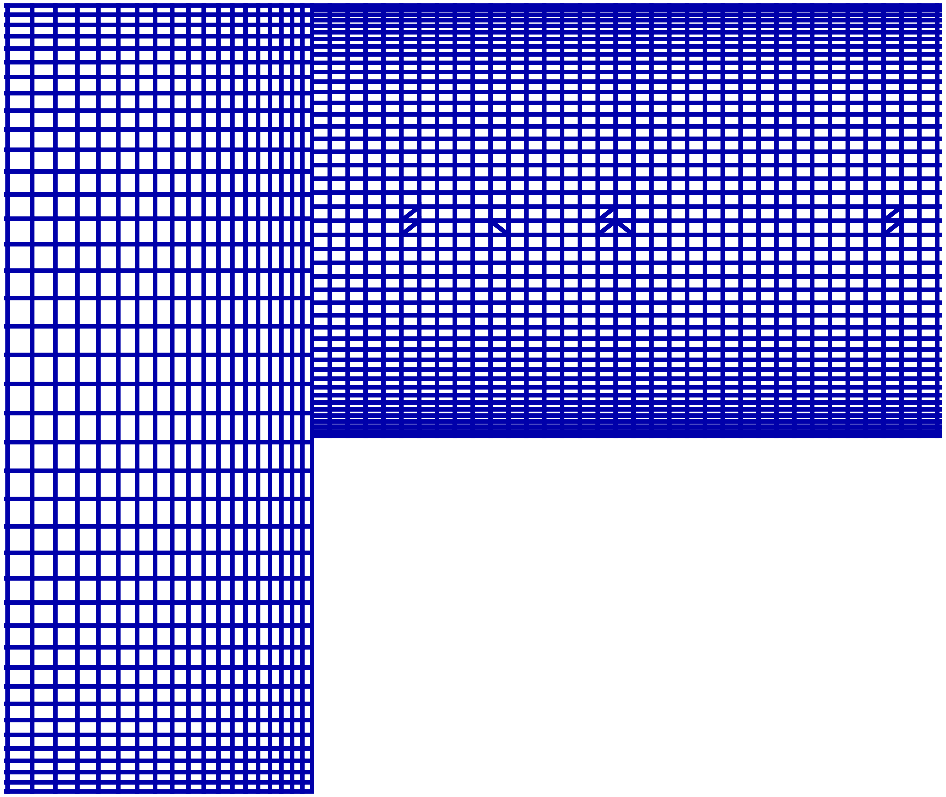
Unsteady flow with moving boundary



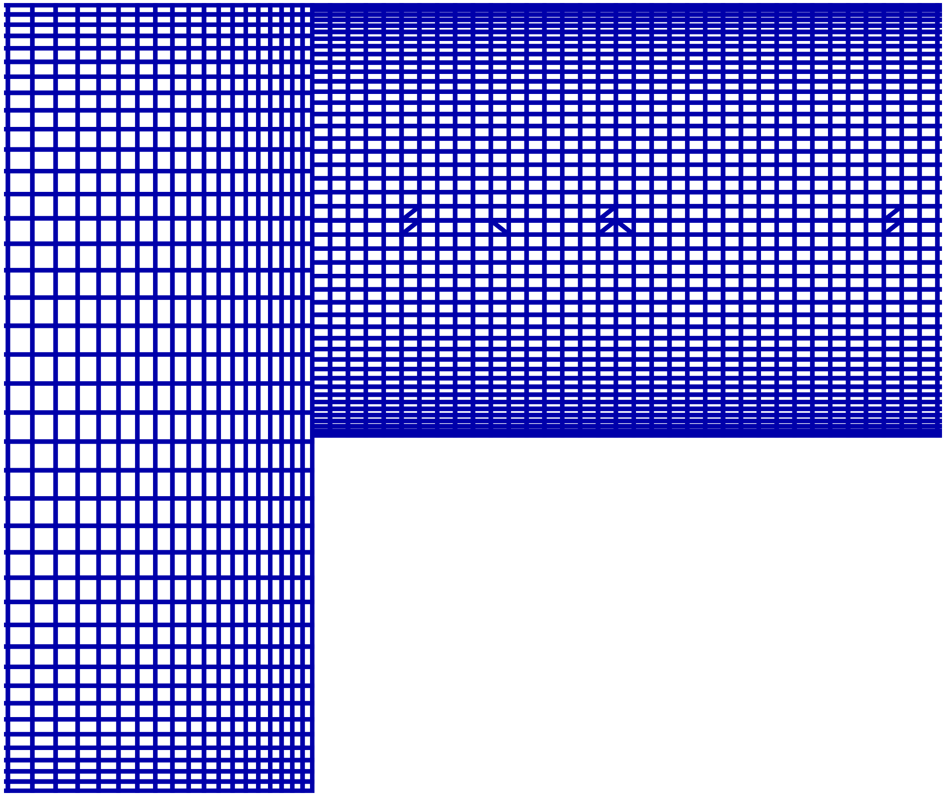
Unsteady flow with moving boundary



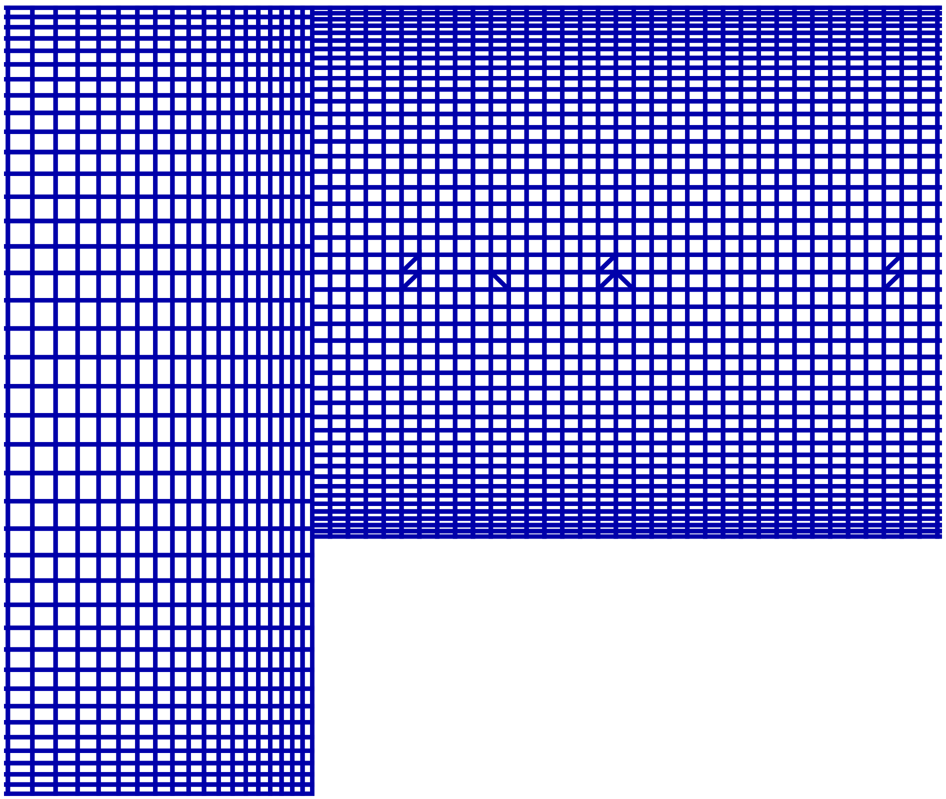
Unsteady flow with moving boundary



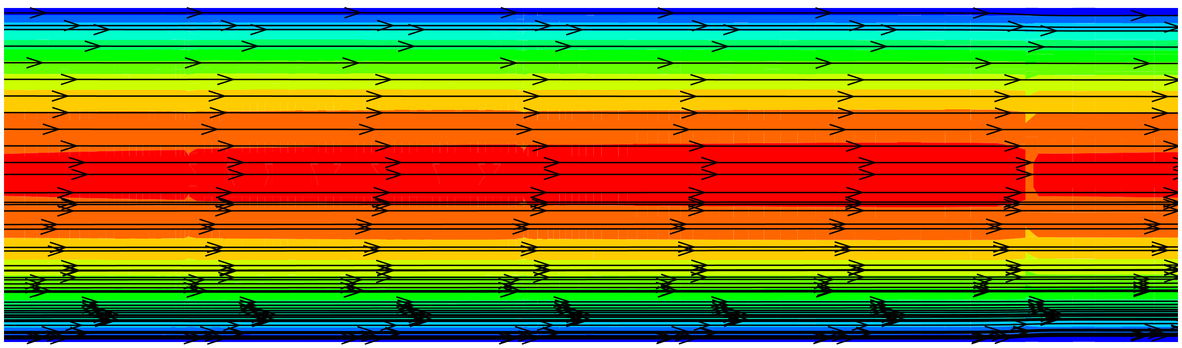
Unsteady flow with moving boundary



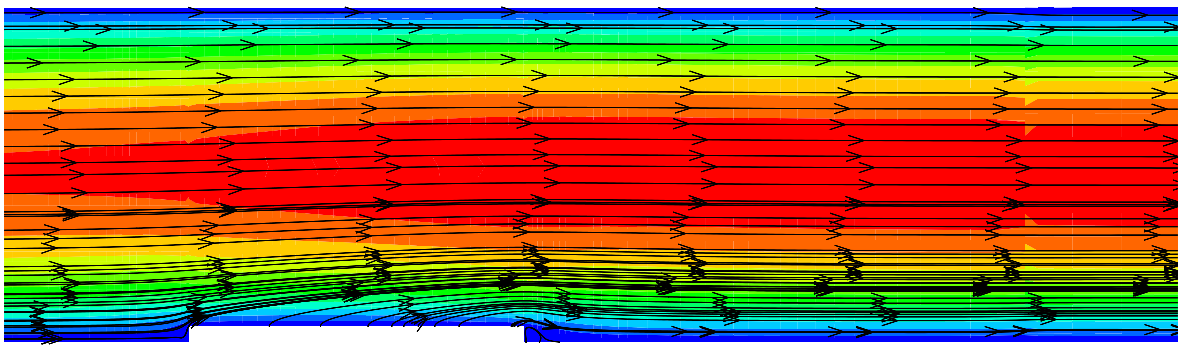
Unsteady flow with moving boundary



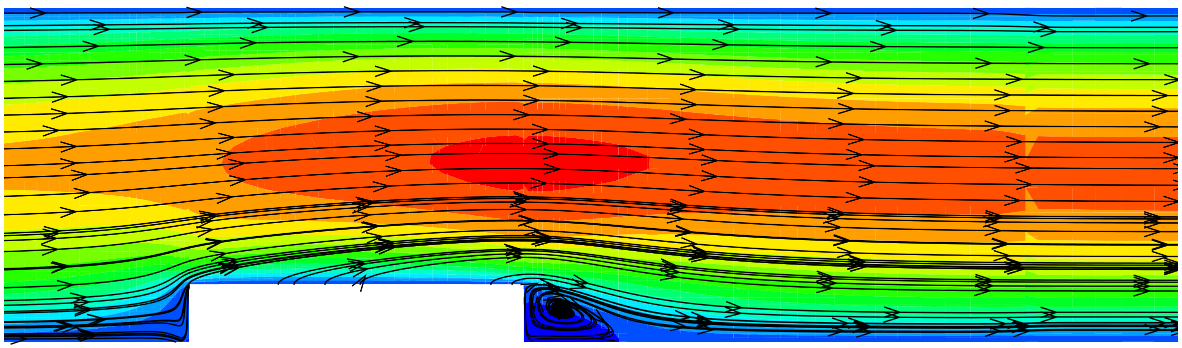
Unsteady flow with moving boundary



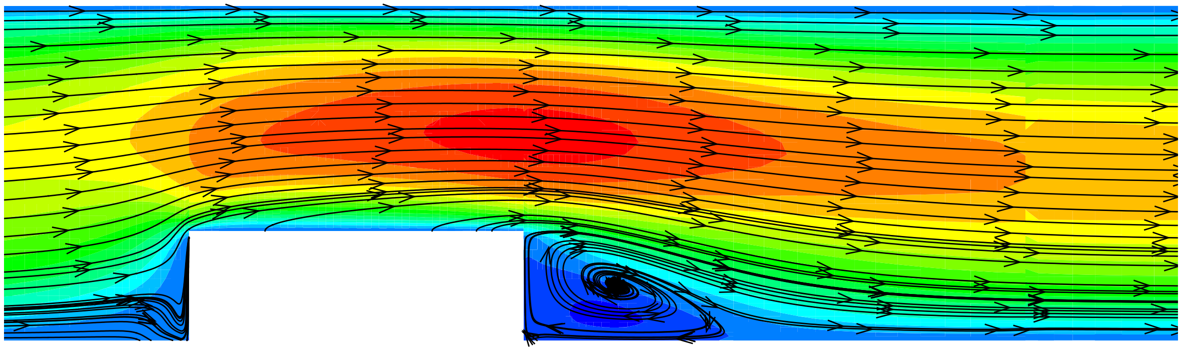
Unsteady flow with moving boundary



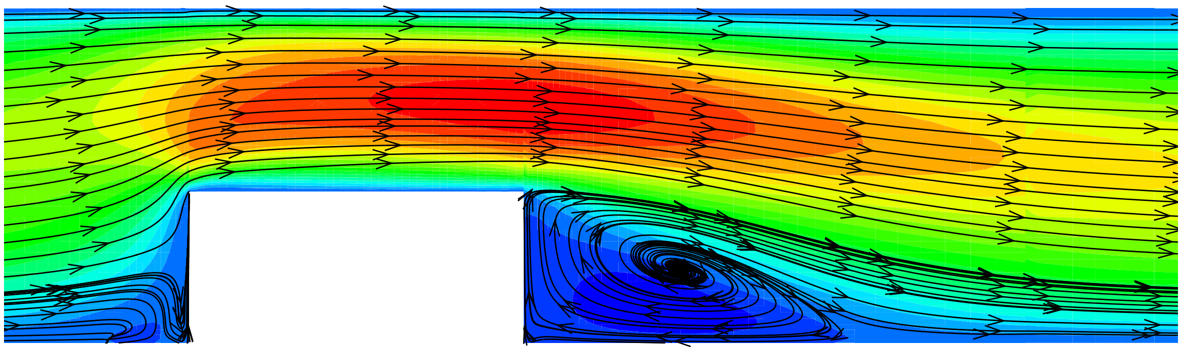
Unsteady flow with moving boundary



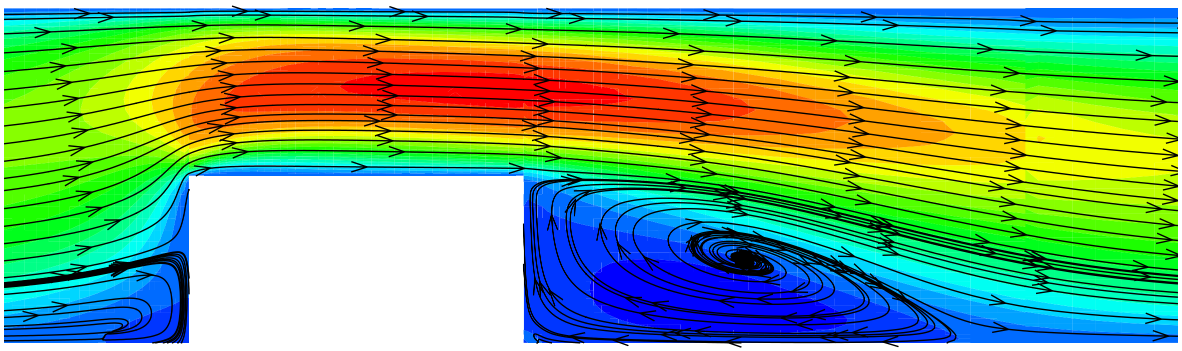
Unsteady flow with moving boundary



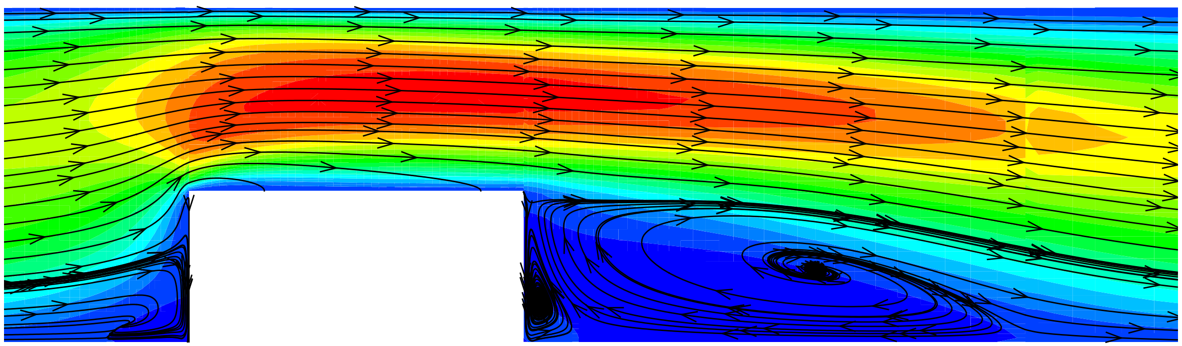
Unsteady flow with moving boundary



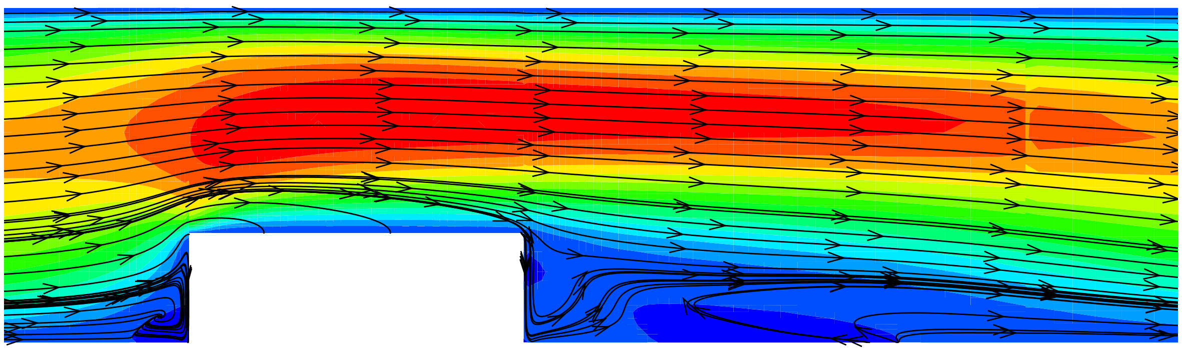
Unsteady flow with moving boundary



Unsteady flow with moving boundary



Unsteady flow with moving boundary



Conclusion

- An algorithm for the treatment of the interface between 3D unstructured meshes has been developed
- Independent multi-block meshes can be use to simulate fluids flows with moving boundaries
- Parallel implementation is not done. An interface must be on one single processor
- Possibility to treat the interface between solid and fluid meshes in fluid structure interaction problems