The grammar of overset meshes in OpenFOAM®

Joel Guerrero
University of Genoa + Wolf Dynamics

Presented at the Fourth Midwest OpenFOAM User Group Meeting
October 4-5, 2019. Minneapolis, Minnesota, USA.
https://sites.google.com/view/mofug2019v1/home

Who am I?

- My name is Joel Guerrero and I am a researcher at the University of Genova (Italy).
- I am also the CTO of Wolf Dynamics.
- My main areas of research are multi-physics simulations, numerical optimization, exploratory data analysis, data analytics, and interactive data visualization.
- Lately, I have been evangelizing about cloud computing, visual storytelling, and agile simulations.

Who is Wolf Dynamics?

- Wolf Dynamics is a spin-off of the University of Genova (innovative start-up).
- It was created to fill the gap between University and Industry in the Liguria region (and the world).
- We work with SMEs to help them become agile, innovate, and more competitive by using numerical simulations.
- But we also work with LEs mainly offering validation services for assessing the transition from commercial software to open-source applications.
- We offer training services and serve as an incubator for new graduates looking to learn more about scientific computing.

wolf dynamics

multiphysics simulations, optimization & data analytics

www.wolfdynamics.com

The grammar of overset meshes in OpenFOAM® – In 5 chapters

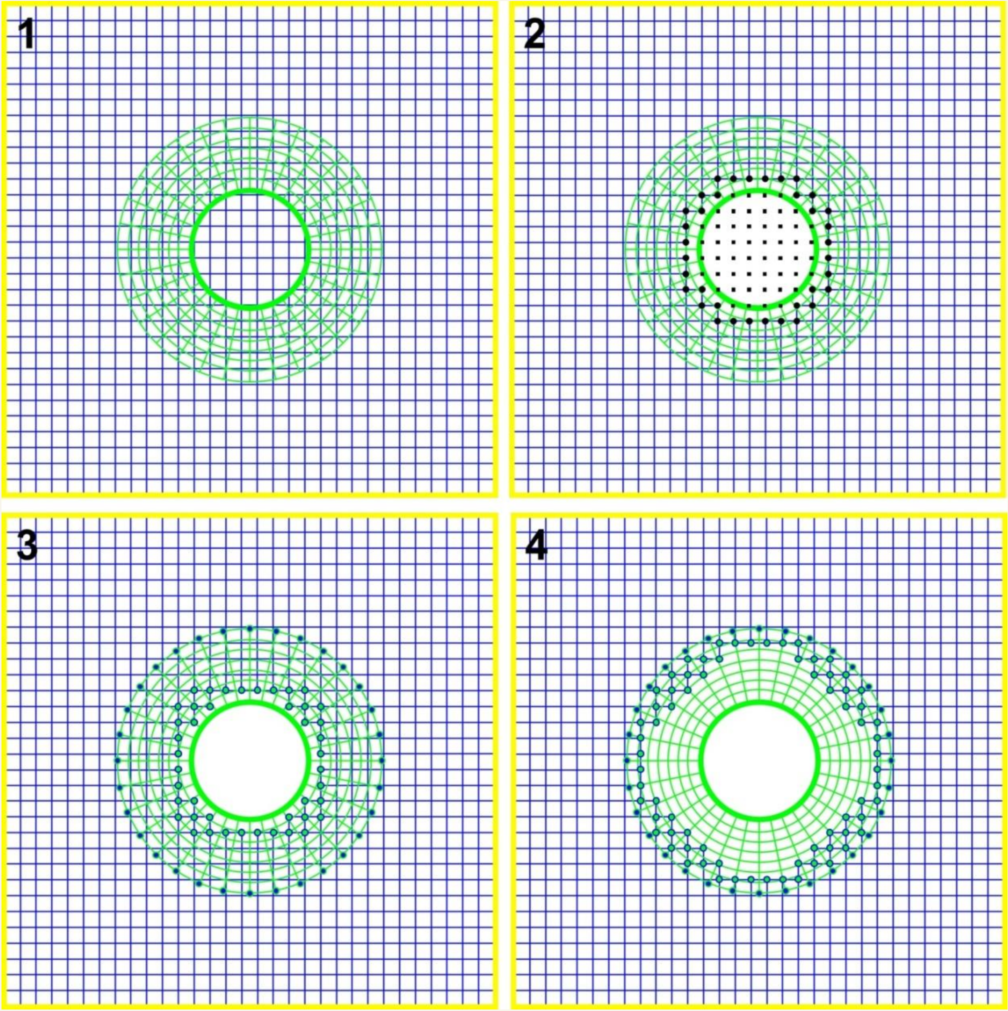
1. Overview of overset meshes
2. Overset meshes in OpenFOAM®
3. Applications – Overset meshes in action
4. Guidelines when working with overset meshes in OpenFOAM®
5. Main takeaways

Chapter 1

Overview of overset meshes

1. Overview of overset meshes

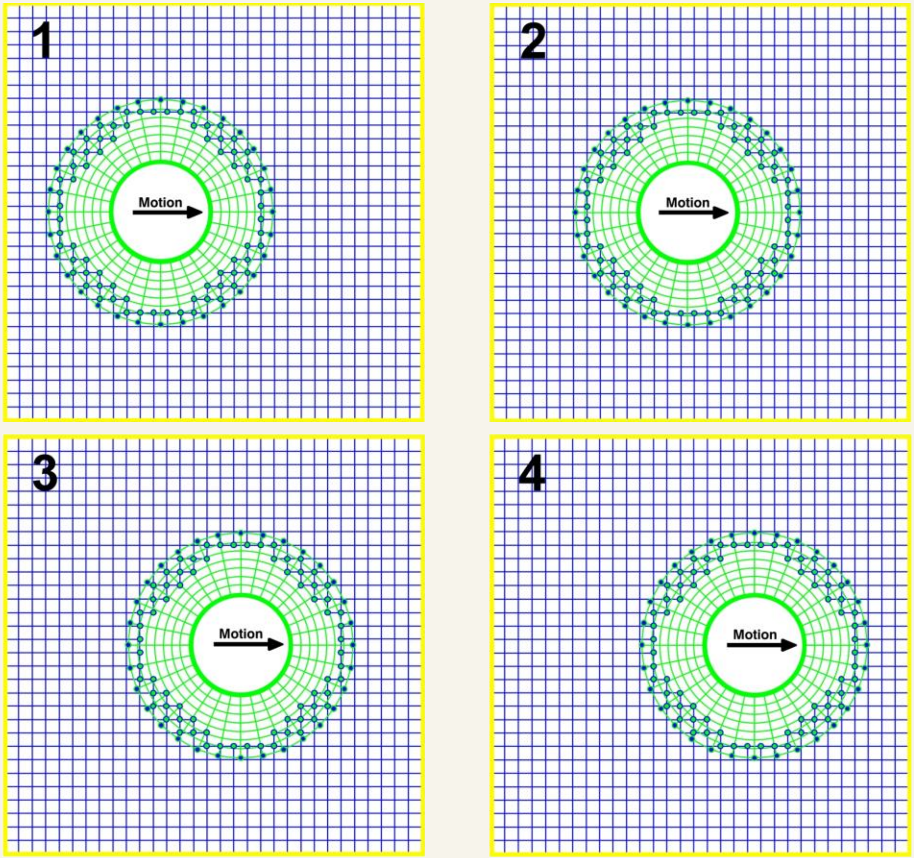
- The overset meshes method consists in generating a set of component meshes (structured or unstructured), that cover the domain and overlap where they meet.
- Then, domain connectivity is obtained through proper interpolation in the overlapping areas.
- Overset meshes (**OM**) are also know as overlapping grids, overset composite grids, composite overlapping meshes, chimera meshes, patches grids, composite grids.
- The **OM** method has been in use for quite some time.
- In the CFD community, it has been in use since the early 1980’s.
- It was then and it is now recognized as an attractive approach for treating problems with moving bodies, complex geometries (think structured meshes), resolve fine structures with AMR, and conduct parametrical studies.



1. Component meses (CM) – The CM are generated separately.
2. Hole cutting – Identification of unused points.
3. Identification of valid interpolation points (this is a valid mesh).
4. Optimized overset mesh – Overlap area minimization.

1. Overview of overset meshes

- If the component meshes are moving, overset connectivity information, such as interpolation stencils and unused points regions (Chimera holes), is recomputed each time-step or prescribed interval.
- The motion of the component meshes may be an user defined function, may obey the Newton-Euler equations for the case of rigid body motion or may be the boundary nodes displacement in response to the stresses exerted by the fluid pressure for the case of FSI problems.
- Overset meshes can easily handle multiple bodies undergoing relative motion.
- They can even collisions.
- Overset meshes guarantees high quality meshes even for very large displacements.



- Moving overset mesh.
- The interpolation stencil and Chimera holes are recomputed every time-step.
- The illustrated overset mesh corresponds to the optimized set, but the mesh with the maximum overset region can be used as well

1. Overview of overset meshes

- On the origin of **OMs** – Development timeline
  - Maybe the first use of operlapping grids was reported by Volkov in the late 1960’s.
  - Further developed and promoted by Starius and Kreiss in the late 1970’s.
  - Formally introduced into the CFD community in the early 1980’s by the pioneering work of Benek, Buning, Dougherty, Meakin, Steger, Suhs.
  - Since the 1990’s it is has been heavily used to deal with complex geomtries and moving bodies (Benek, Boger, Bunning, Chan, Chesshire, Gomez, Henshaw, Meakin, Noack, Petersson, Rogers, Steger, among many).
  - Since 2000’s, the use of **OM** with unstructured meshes gained popularity.
  - Since 2010’s most of commercial CFD solver and many open-source frameworks use **OM** with structured and unstructured meshes.
  - Symposium on Overset Composite Grids and Solution Technology (http://oversetgridsymposium.org/).
    - Biyearly event.
    - First edition took place in 1992 – NASA Ames Research Center, Moffett Field, California.
    - Next edition: 2020 – NASA Langley, Hampton, Virginia USA

1. Overview of overset meshes

- **Incomplete list of overset solvers and libraries:**

  - Research solvers with overset meshes capabilities (first row only NASA):
    - Overflow^{S}, CFL3D^{S}*, FUN3D^{U}, USM3D^{U}, TetrUSS^{U}, INS3D^{S}, LLNL Overture^{S}*, NIST Overset-HDG^{U}, DLR TAU-Code^{U}, Onera eLSA^{S}, SU2^{U}, CNR Xnavis^{S}*

  - Commercial solvers with overset meshes capabilities:
    - Ansys Fluent^{U}*, Star-CCM++^{U}*, ESI CFD-ACE+^{U}, Metacomp CFD++^{U}, MSC-Crandle scFLOW^{U}, ICFD++^{U}

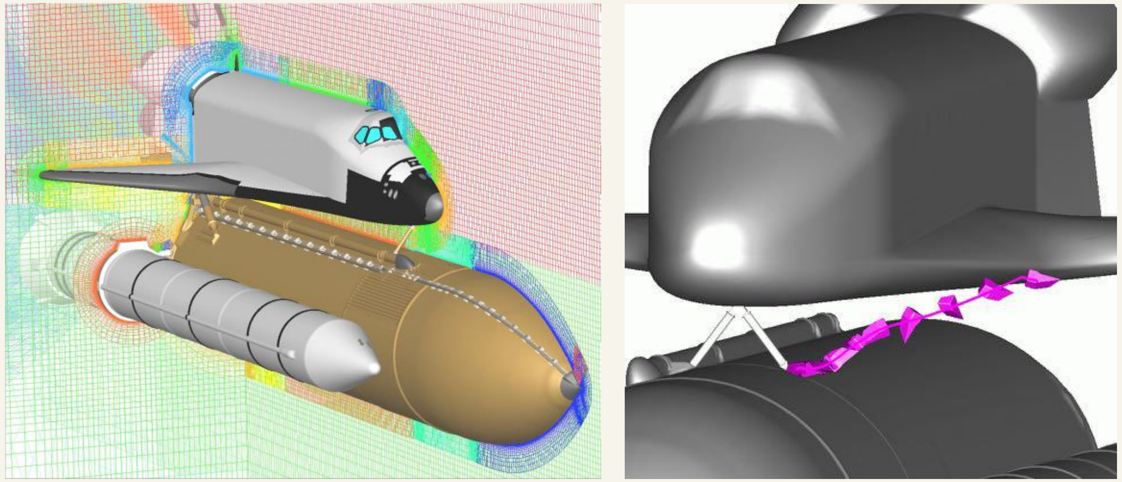
  - Libraries for assmbling overset meshes (research and commercial):
    - DiRTlib, SUGGAR++, Chimera Grid Tools (CGT), BEGGAR, Ogen*, Maggie, Pegasus, Ronnie, Cassiopee, Pointwise.

  - In OpenFOAM® ecosystem:
    - ESI (1906)*, Foam-Extend*, Bellerophon, Opera, FoamedOVER (SUGGAR++), Caelus (SUGGAR++)

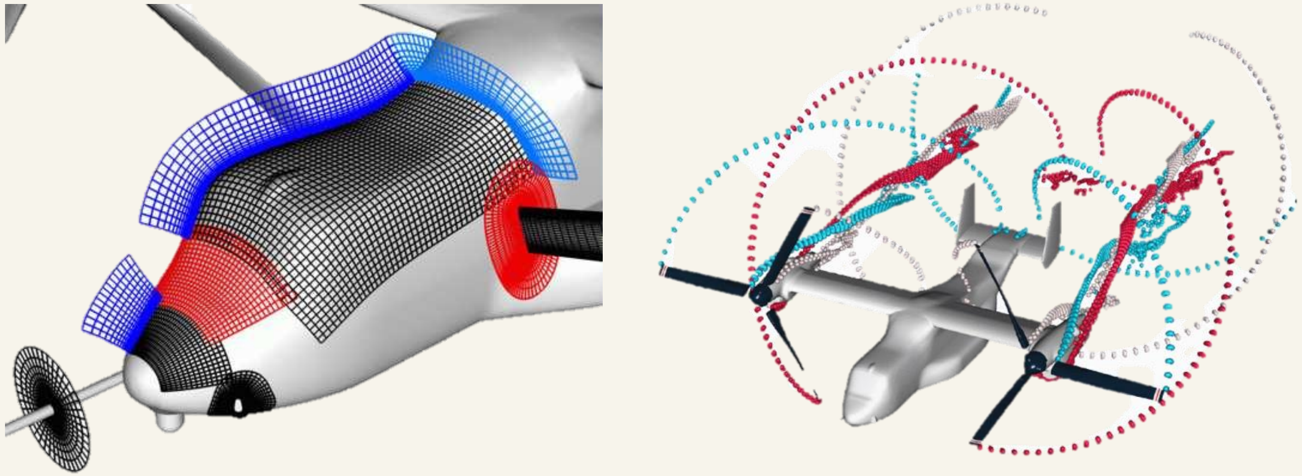
S → Structured, U → Unstructured,

1. Overview of overset meshes

- Some overset meshes applications.
- In todays world, overset meshes are often used to solve challenging moving bodies problems.



Space shuttle
Figure credit: P. Buning, W. Chan, R. Gomez, S. Pandya.



V-22 Osprey
Figure credit: W. Chan, R. Meakin, W. Wissink.

Chapter 2

Overset meshes in OpenFOAM®

2. Overset meshes in OpenFOAM®

- The process of assembling overset meshes in OpenFOAM® is very straightforward.
- Four basic steps are involved:
    1. Generate component meshes and merge then together (done by the user).
    2. Define overset patches (done by the user).
    3. Assign zones (done by the user).
    4. Compute stencils and assign cell type (done by the overset solver)

- These steps are common for every CFD solver that works with overset meshes.
- What it change is the tools and methods used for merging meshes, assigning zones, and computing stencils.

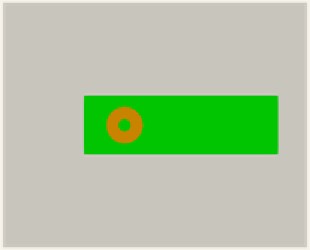
2. Overset meshes in OpenFOAM®

- Step 1 → Generate component meshes and merge then together (done by the user).

Component mesh located in the directory **all**

+

=

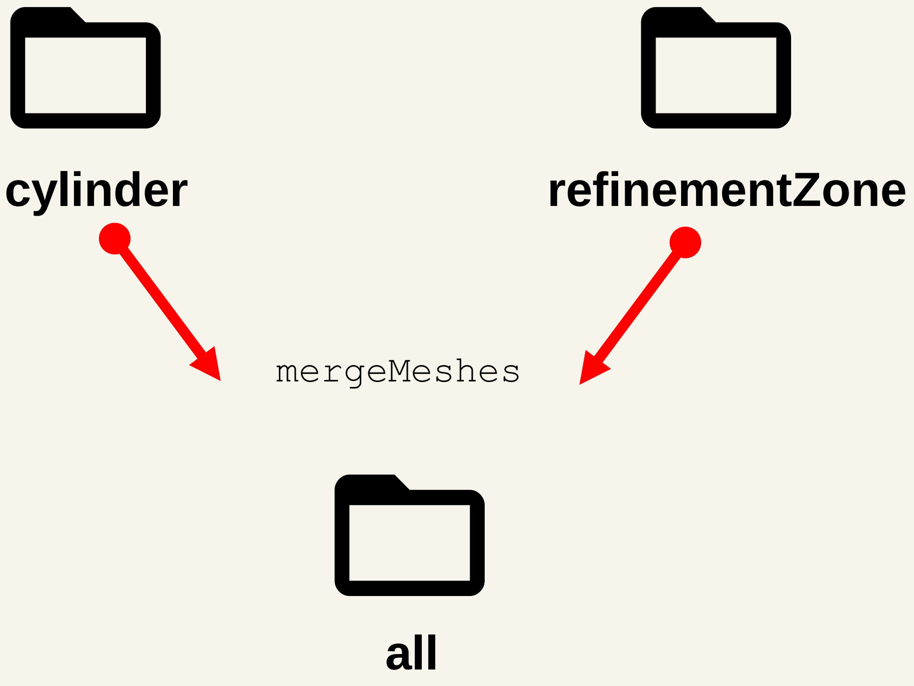


Component mesh located in the directory **refinementZone**

+

Component mesh located in the directory **cylinder**

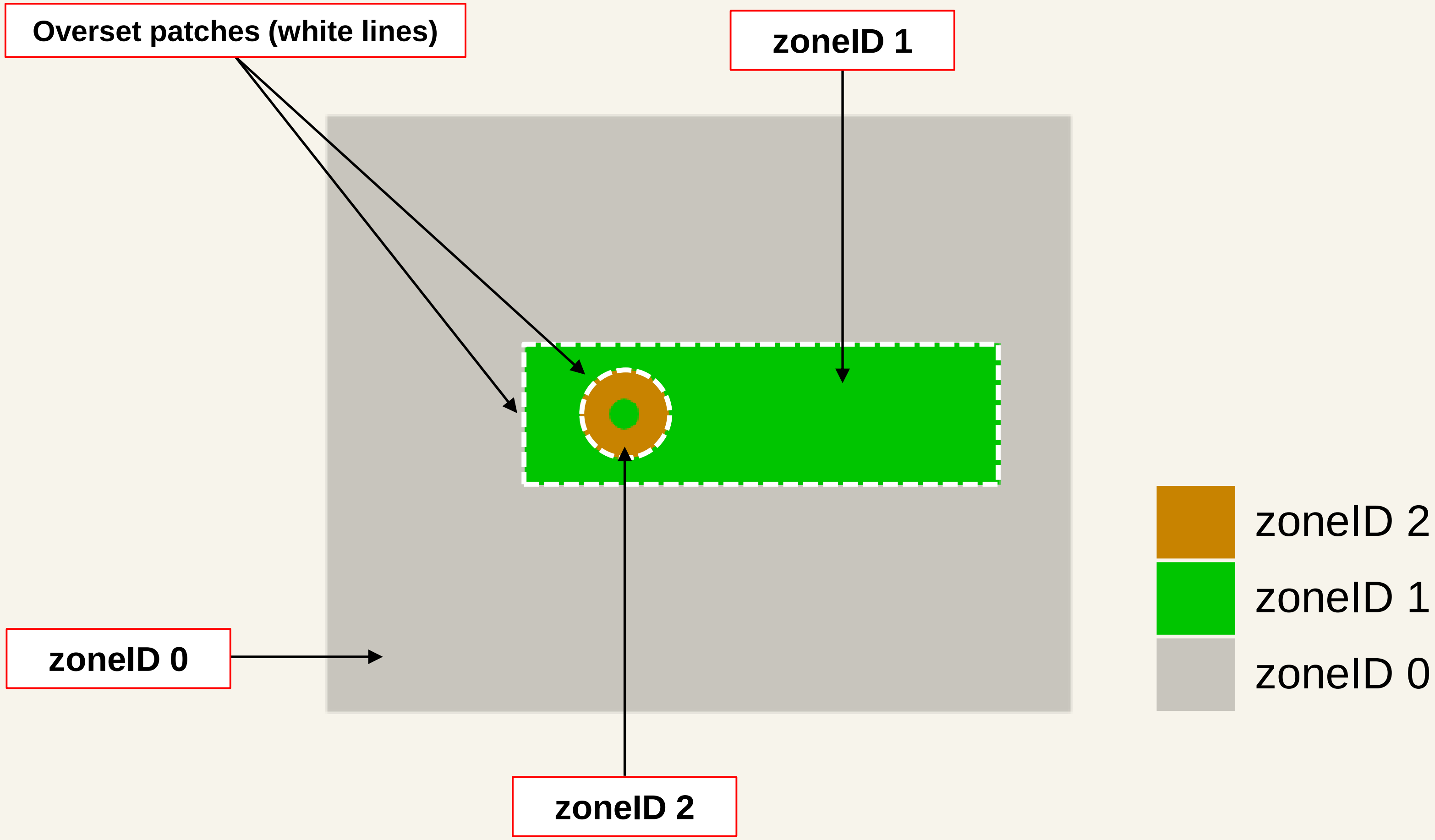
Note:
Component meshes are considered different cases; therefore, they are generated in different directories



- To assemble an overset mesh, you generate the meshes in separated directories, and then you merge the meshes using the utility `mergeMeshes`.
- You merge the meshes in a single directory, in this case we will merge the meshes **cylinder** and **refinementZone** in the directory **all**.

2. Overset meshes in OpenFOAM®

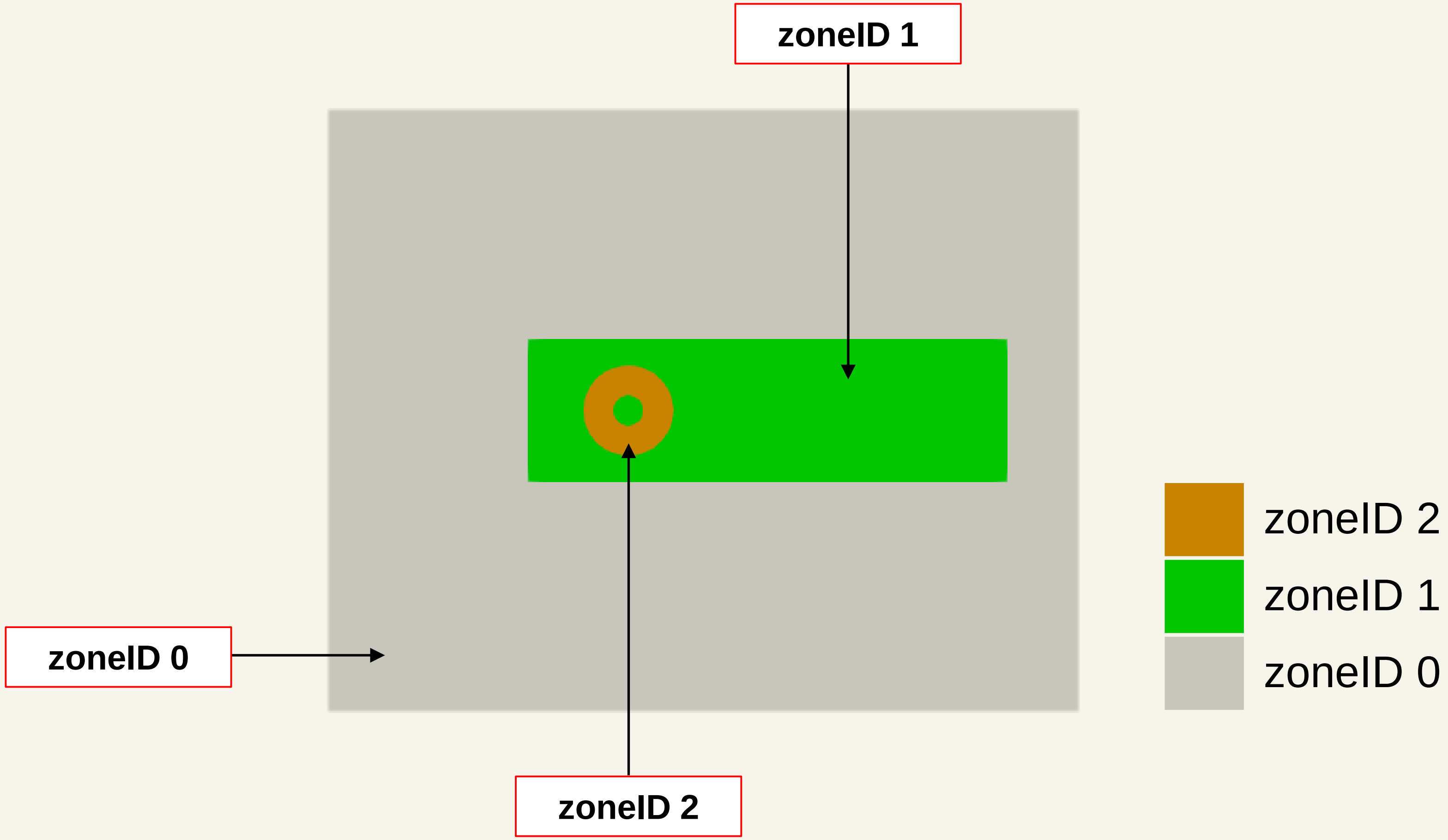
- Step 2 → Define overset patches (done by the user).



- The overset patches are defined by the user.
- Overset patches can intersect each other.
- They can also intersect other patches (walls).
- However, walls cannot intersect other walls (no collisions) or go out of the domain (escape).

2. Overset meshes in OpenFOAM®

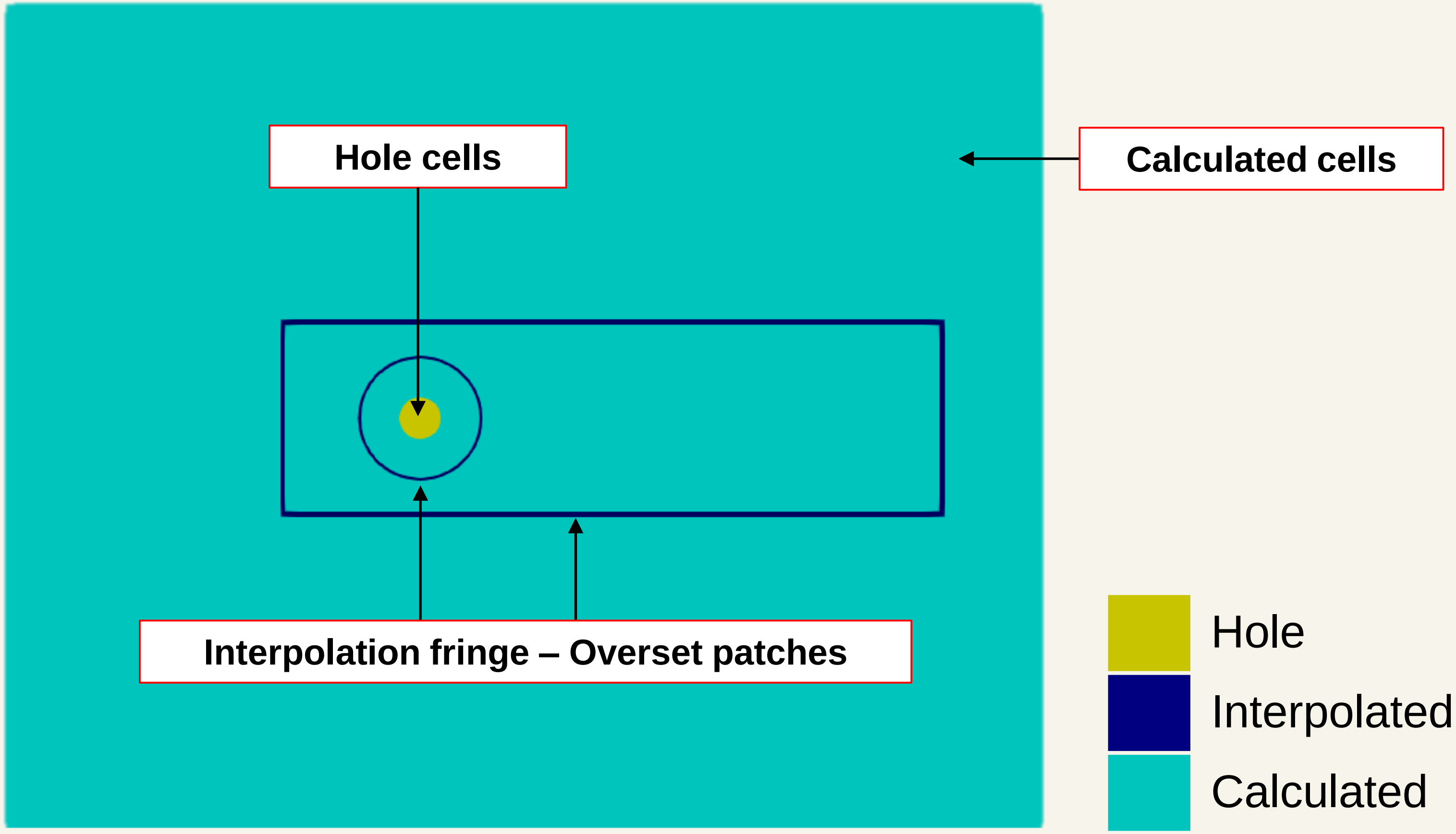
- Step 3 → Assign zones (done by the user).



- A zone identification (zoneID) is assigned to each component mesh after they have been merged.
- It is recommended to assign zoneID 0 to the background mesh.
- The background mesh can be the mesh that is not moving, the mesh with inlet and outlet patches, or the mesh that does not have overset patches.

2. Overset meshes in OpenFOAM®

- Step 4 → Compute stencils and assign cell type (done by the overset solver).

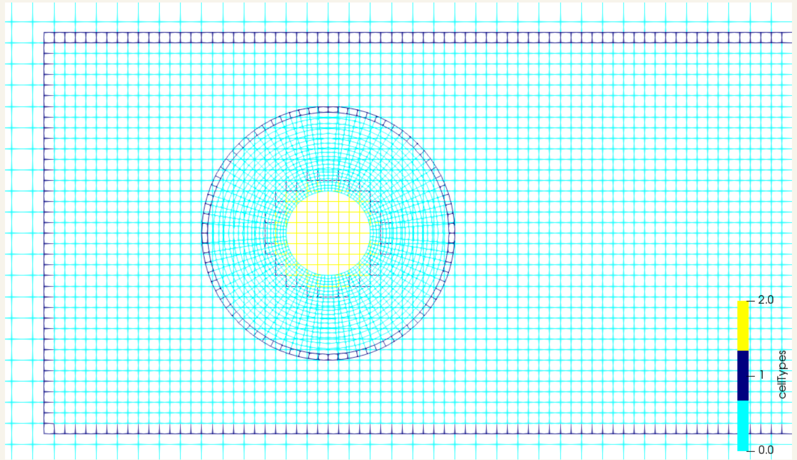


- The overset patches between the component meshes are set by the user, whereas, the interpolation fringe close to the walls, as well as the hole cells, are computed automatically by the overset solver.
- The cell types are defined as follows: **hole** cells (the solution is not computed), **interpolated** cells (the solution is interpolated from mesh-to-mesh), and **calculated** cells (the solution is computed).

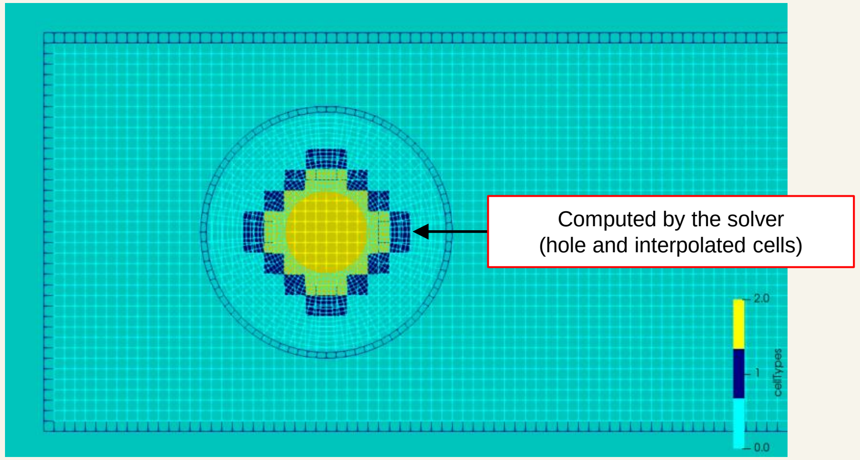
Note:
Not all interpolation fringes are visible in the figure

2. Overset meshes in OpenFOAM®

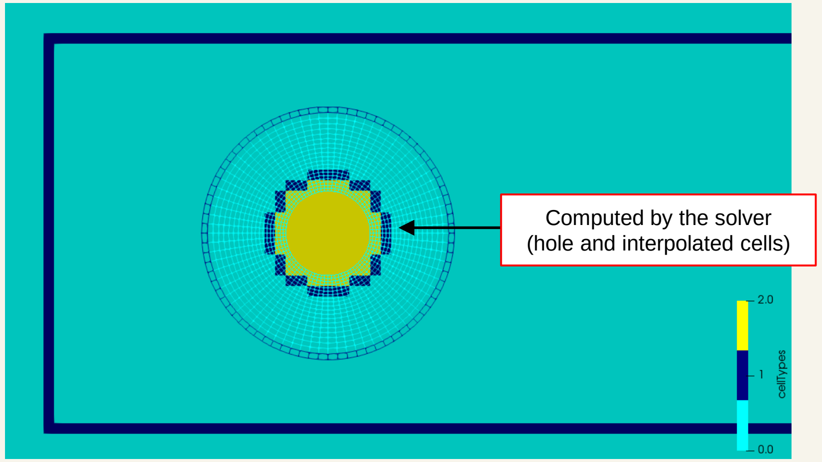
- Step 4 → Compute stencils and assign cell type (done by the overset solver).



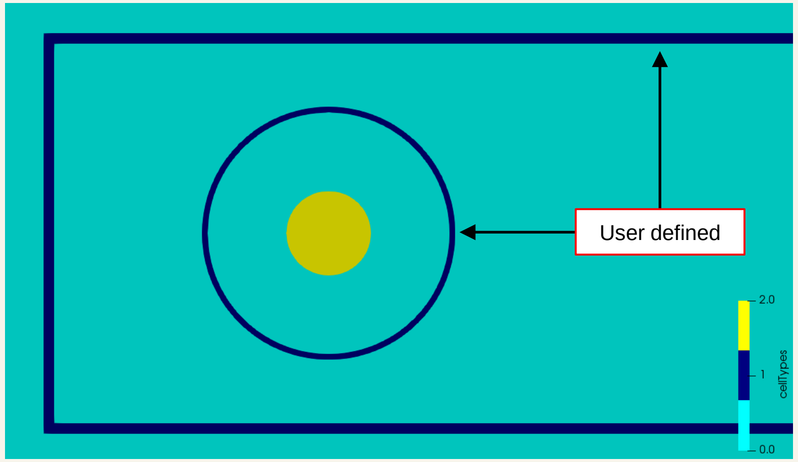
Wireframe visualization – All component meshes



Wireframe visualization – Component mesh 2 and 3



Wireframe visualization – Component mesh 3



Contour visualization – All component meshes

- Cells size close to interpolated cells should be of the same size to minimize interpolation errors.
- When computing the solution, an interpolation method must be chosen.
  - cellVolumeWight
  - inverseDistance
  - trackingInverseDistance
  - leastSquares (recommended)

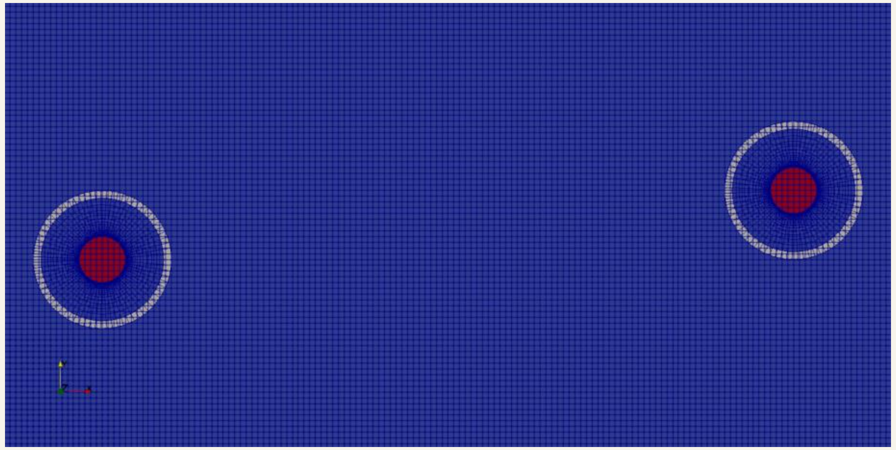
Hole

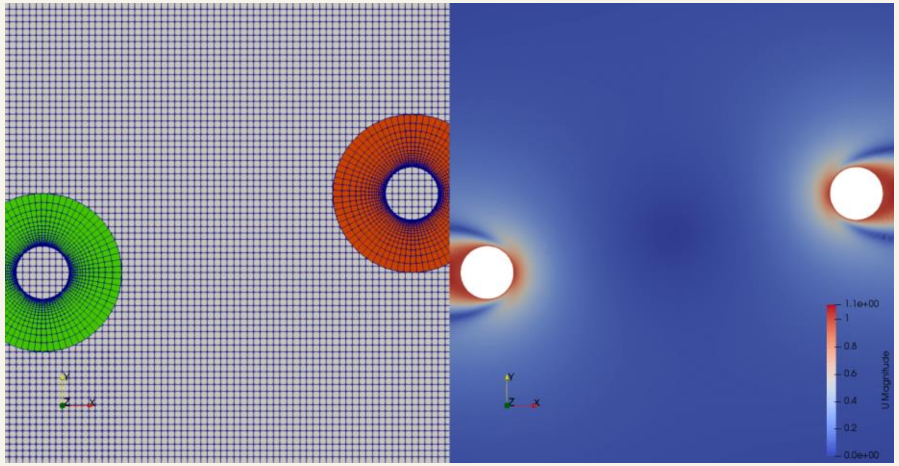
Interpolated

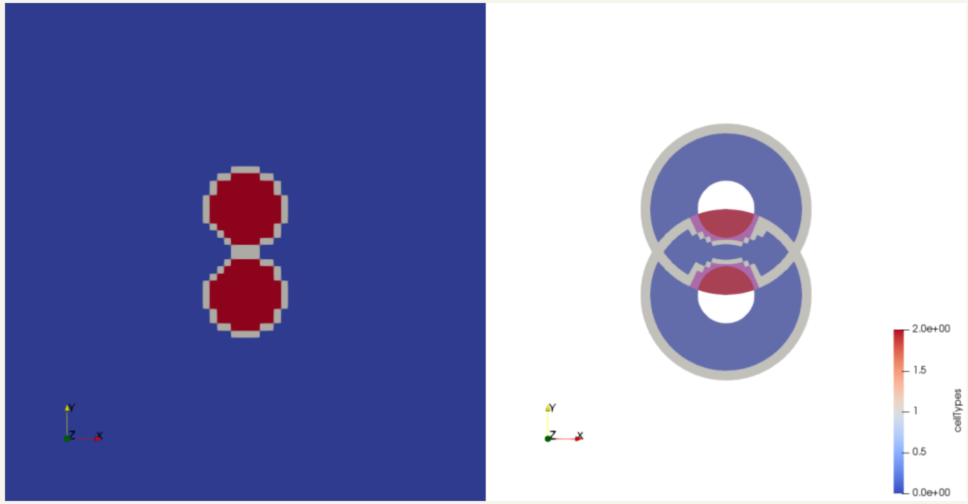
Calculated

2. Overset meshes in OpenFOAM®

- Moving overset mesh.
- The interpolation fringes are recomputed every single time-step.





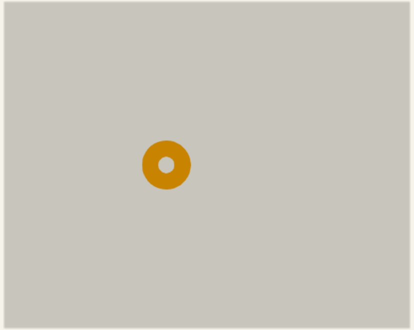


	Cell type value
Hole	2
Interpolated	1
Calculated	0

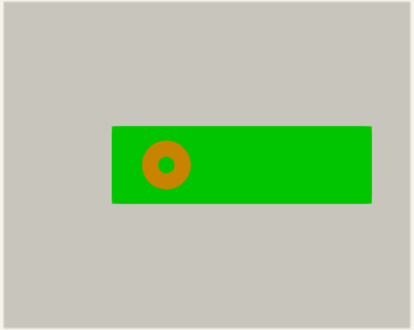
2. Overset meshes in OpenFOAM®

- About the order of the operations when merging meshes.
  - In theory, it does not matter the order of the merge operations.
  - At the end, all component meshes should be merged into a single component mesh.
  - In this case, the component meshes **cylinder** and **refinementZone** are merger into component mesh **all**.
  - The zoneID is assigned after merging the meshes.
  - It is highly recommended that the **oversetPatch** be the first one in the boundary file.

First merge operation → **cylinder** + **all**

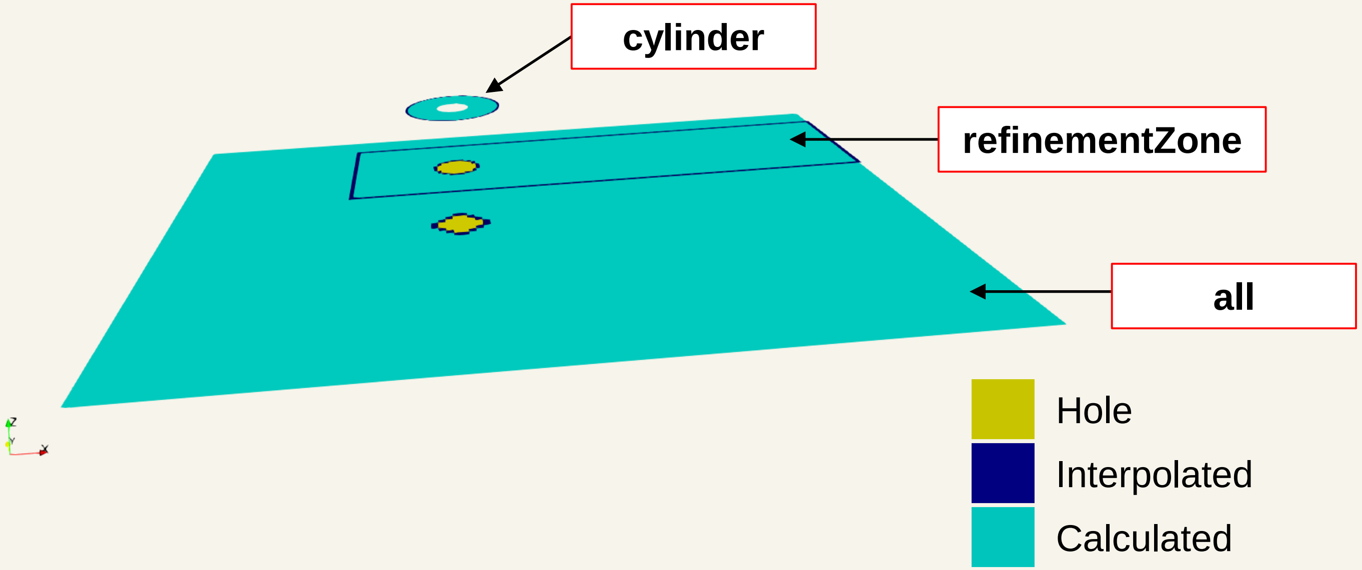


Second merge operation → **refinementZone** + **previous merged mesh**



2. Overset meshes in OpenFOAM®

- About the **zoneID** priority (or **grid priority**).
  - The **zoneID** defines the order of the hole cutting operations on the component meshes.
  - High **zoneID** values, means high priority. That is, the wall on that mesh will cut lower priority levels.
  - In this case, the **cylinder** mesh has a **zoneID** equal to 2, the **refinementZone** mesh has a **zoneID** equal to 1, and the **all** mesh (background) has a **zoneID** equal to 0.
  - Therefore, the **cylinder** mesh will cut meshes **refinementZone** and **all**, the **refinementZone** will cut the mesh **all**, and so on.

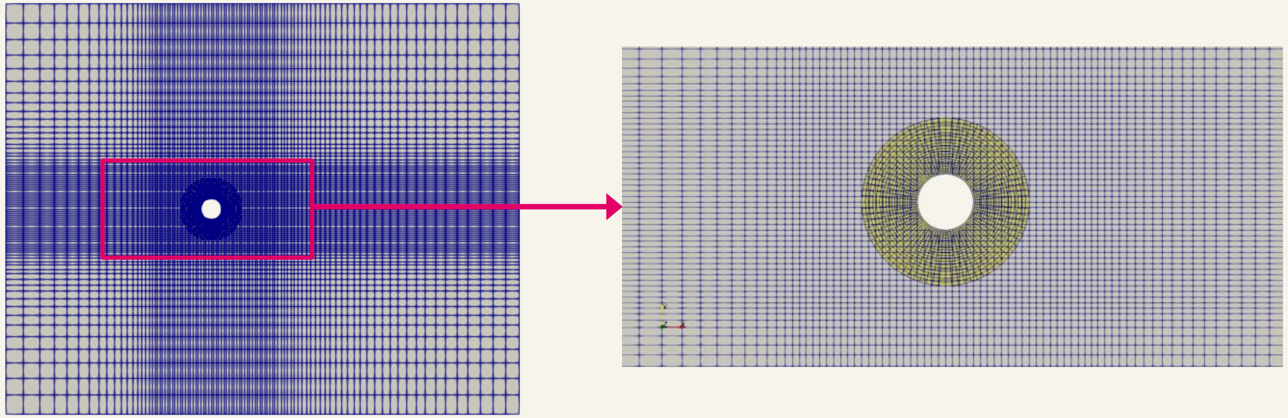


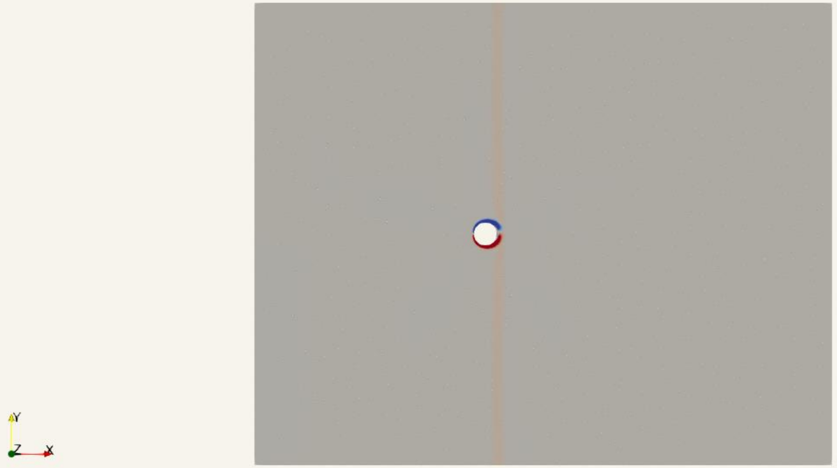
Component mesh	zoneID (grid priority)
cylinder	2
refinementZone	1
all	0

- Different grid priorities will give you different overset assemblies
- Remember, the holes are computed using walls, so if there no walls, there are no holes.

2. Overset meshes in OpenFOAM®

- Overset meshes simulation workflow in OpenFOAM®





Step 1 – Generate and merge component meshes
Done by the user

Step 2 – Define overset patches
Done by the user

Step 3 – Assign zones (grid priorities)
Done by the user

Step 4 – Set numerics for overset meshes (overset interpolation, interpolation type, corrections, and so on)
Done by the user

Step 5 – Compute stencils and assign cell type
Done by the solver

Step 6 – Compute and interpolate solution
Done by the solver

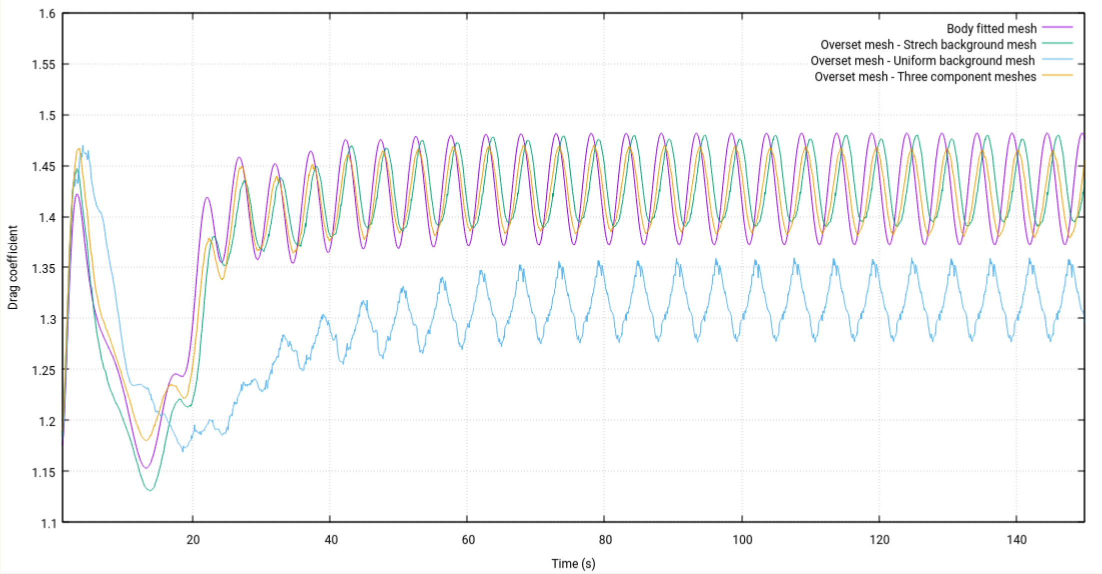
Step 7 – Postprocessing (which is more tedious that working with single meshes)
Done by the user

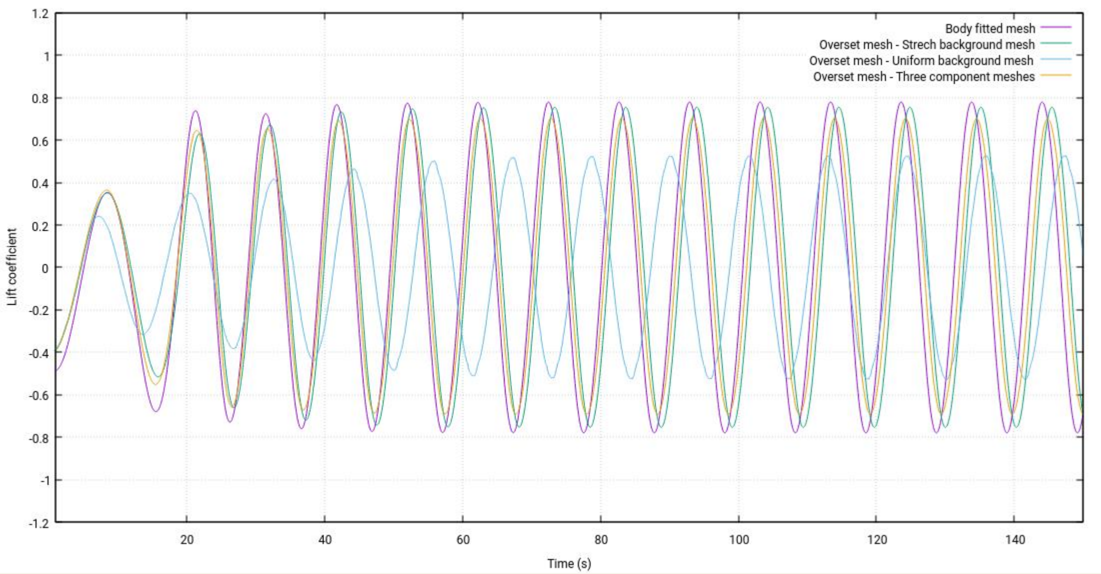
Chapter 3

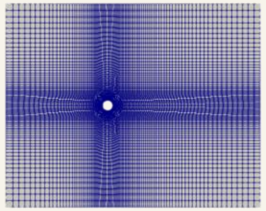
Applications – Overset meshes in action

3. Applications – overset meshes in action

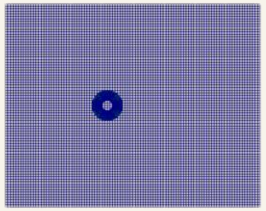
- Flow about a fixed cylinder using overset meshes – Re = 200



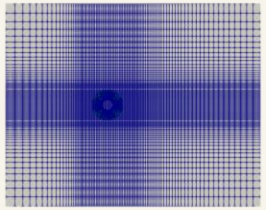




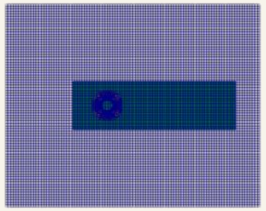
Single body fitted mesh



Overset mesh – Uniform background mesh – Two component meshes



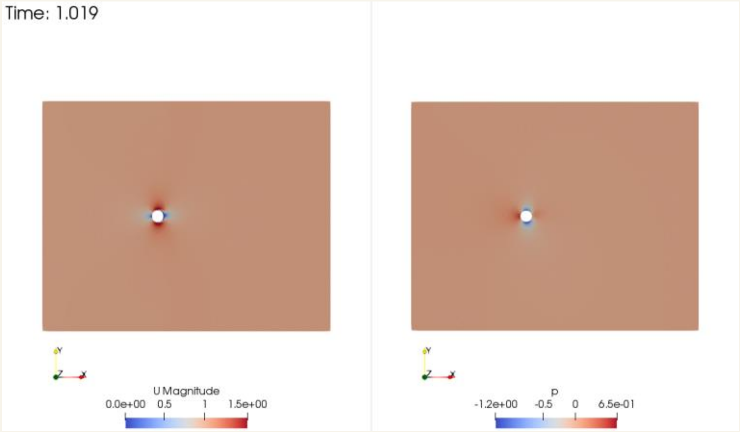
Overset mesh – Stretched background mesh – Two component meshes



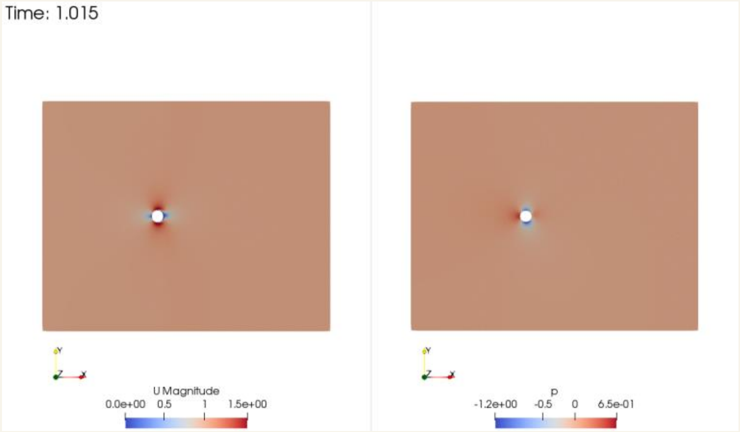
Overset mesh – Three component meshes

3. Applications – overset meshes in action

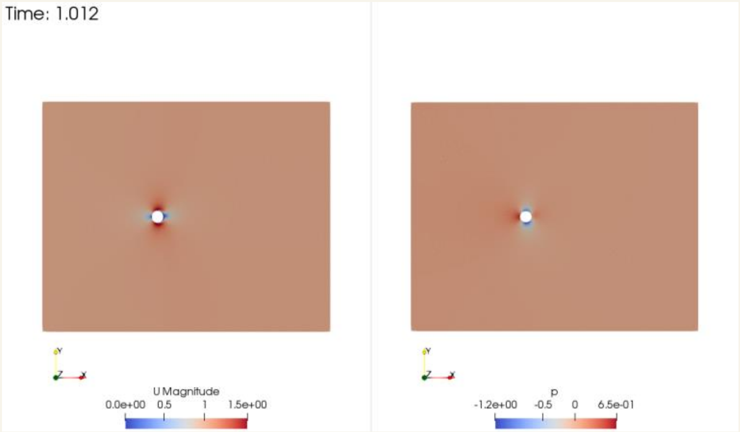
- Flow about a fixed cylinder using overset meshes – Re = 200



Two component meshes – No stretching in the background mesh
http://www.wolfdynamics.com/training/dynamicMeshes/overset6.gif



Two component meshes – Stretching in the background mesh
http://www.wolfdynamics.com/training/dynamicMeshes/overset5.gif

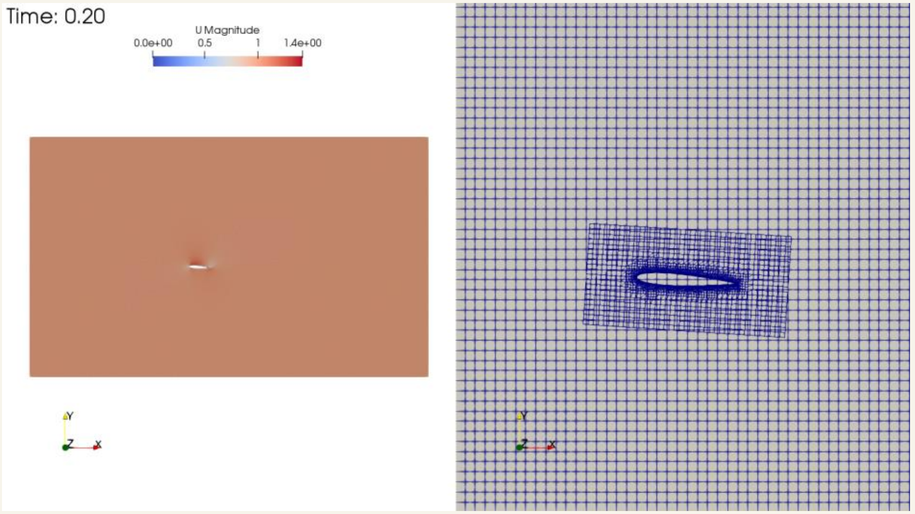


Three component meshes – Stretching in the background mesh and refinement component mesh
http://www.wolfdynamics.com/training/dynamicMeshes/overset7.gif

3. Applications – overset meshes in action

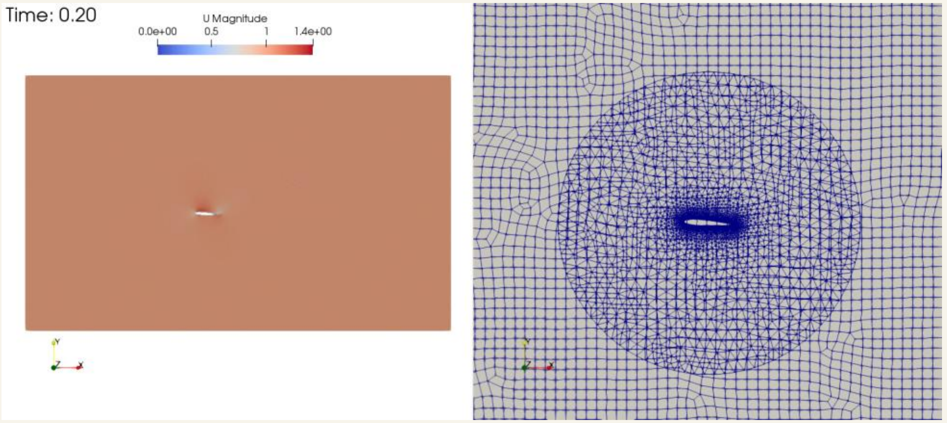
- Flapping airfoil undergoing prescribed heaving and pitching motion
- The solvers can handle mixed elements.

Mesh generated with snappyHexMesh



http://www.wolfdynamics.com/training/dynamicMeshes/overset4.gif

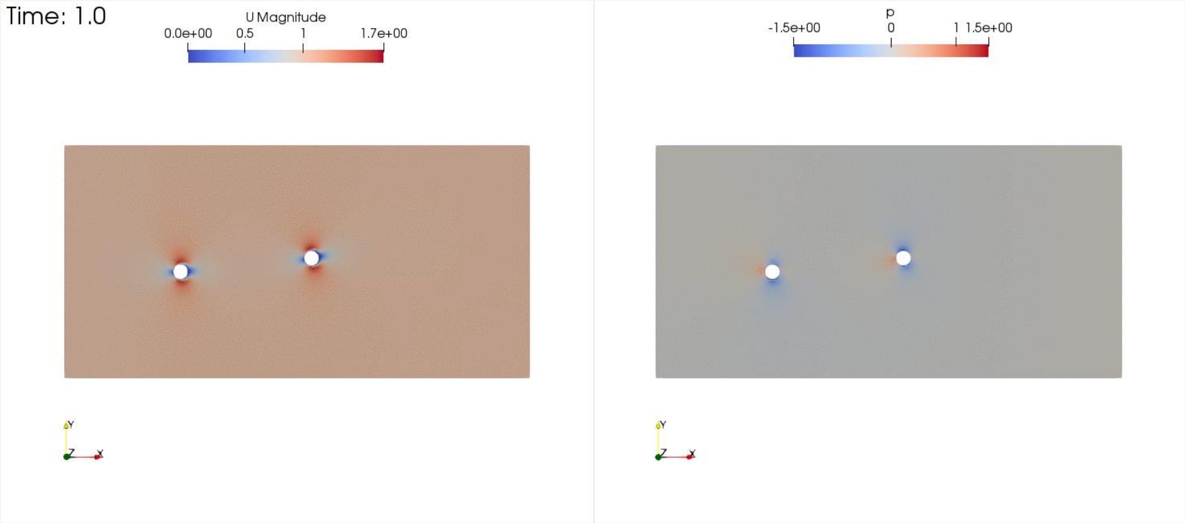
Mesh generated with Ansys mesher



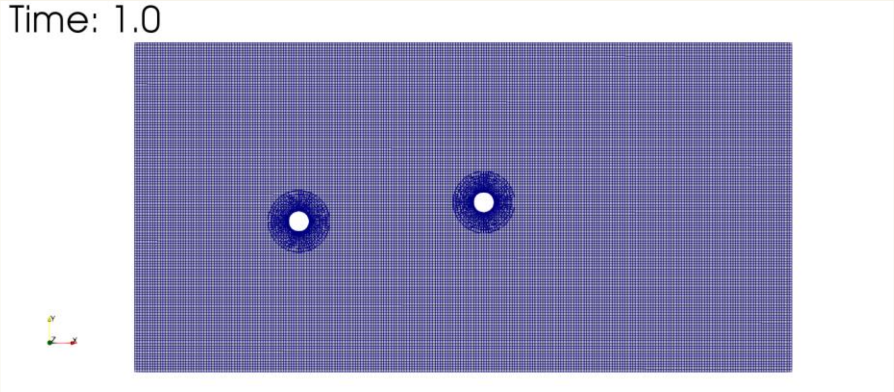
http://www.wolfdynamics.com/training/dynamicMeshes/overset3.gif

3. Applications – overset meshes in action

- VIV of two cylinders – Rigid body motion with overset meshes



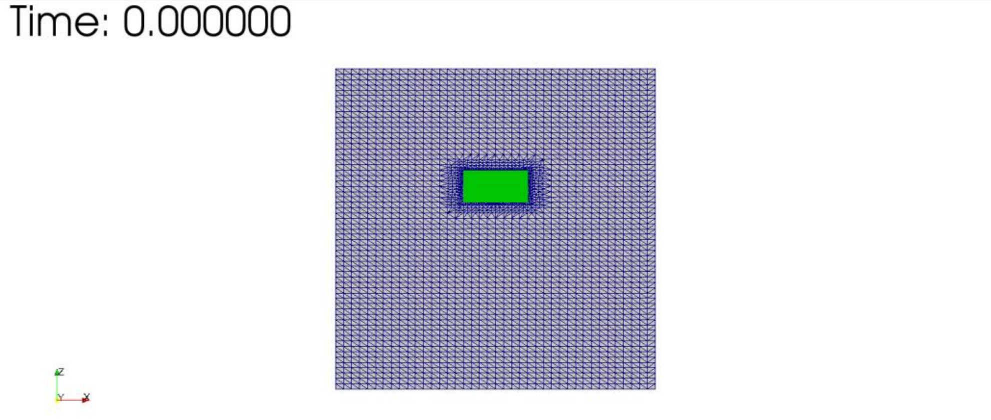
http://www.wolfdynamics.com/training/dynamicMeshes/viv1.gif



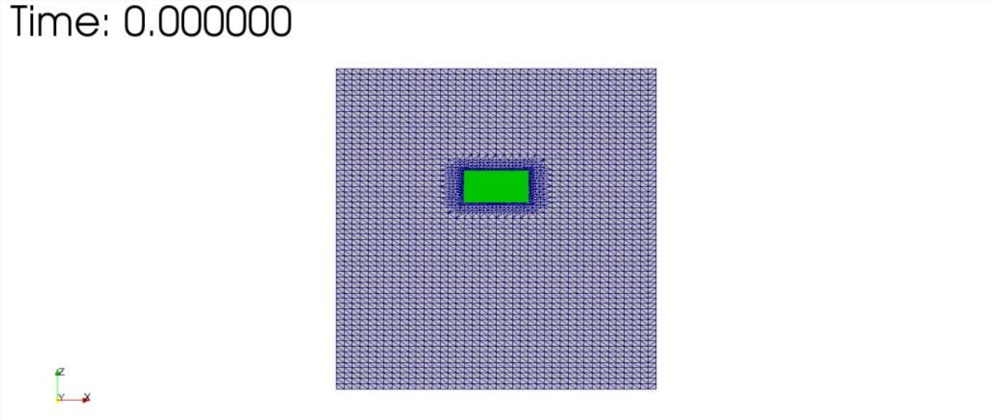
http://www.wolfdynamics.com/training/dynamicMeshes/viv2.gif

3. Applications – overset meshes in action

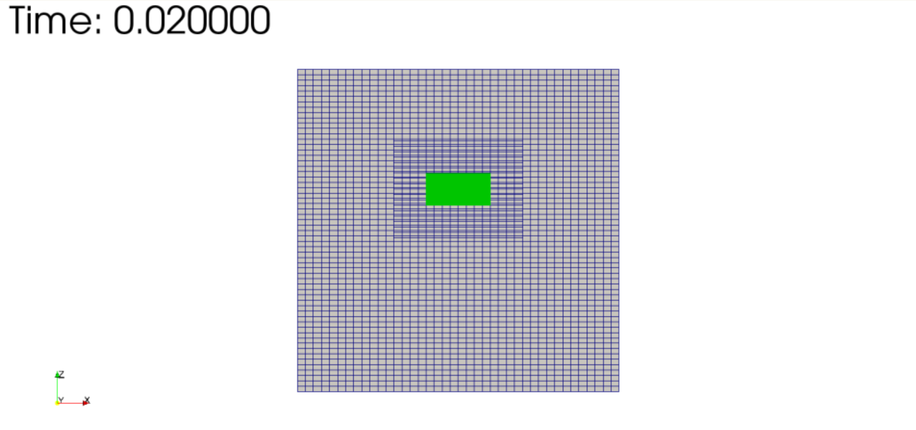
- Falling and floating body – Rigid body motion – Comparison of different meshing techniques.



Morphing meshes – Body fitted mesh
http://www.wolfdynamics.com/training/dynamicMeshes/dof1.gif



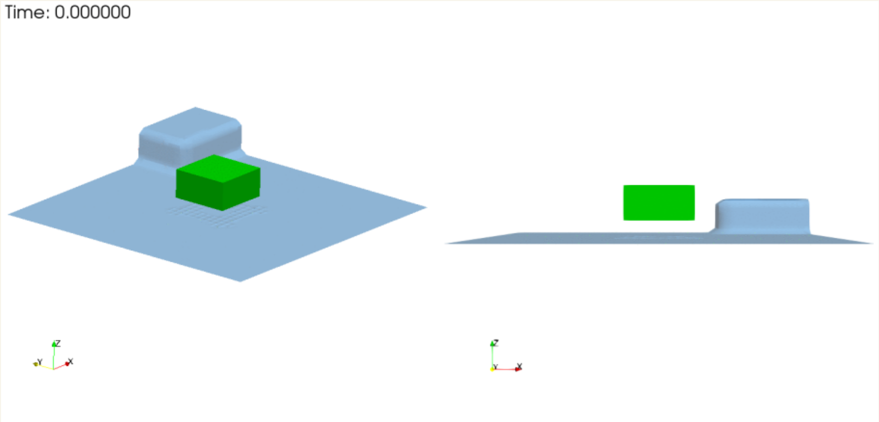
Morphing meshes – Body fitted mesh with remeshing
http://www.wolfdynamics.com/training/dynamicMeshes/dof3.gif



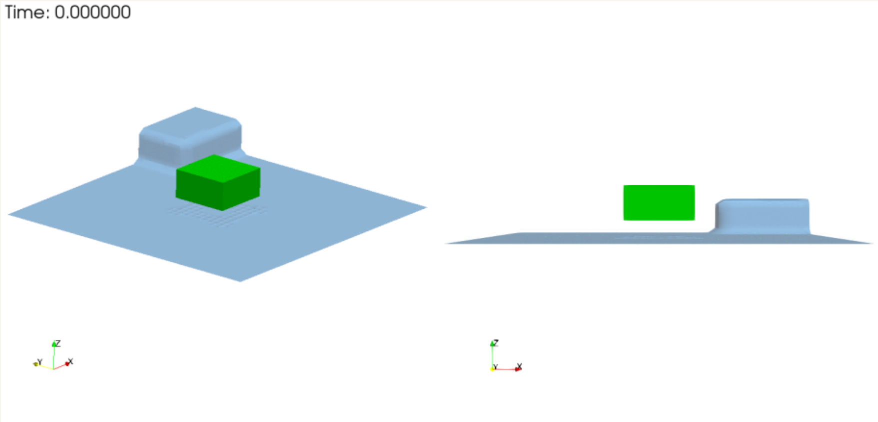
Overset meshes
http://www.wolfdynamics.com/training/dynamicMeshes/overset_rbm1.gif

3. Applications – overset meshes in action

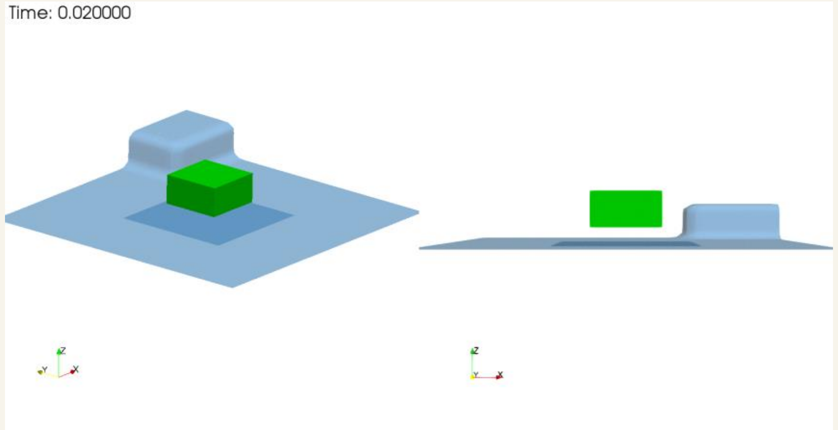
- Falling and floating body – Rigid body motion – Comparison of different meshing techniques.



Morphing meshes – Body fitted mesh – Water surface visualization
http://www.wolfdynamics.com/training/dynamicMeshes/dof2.gif



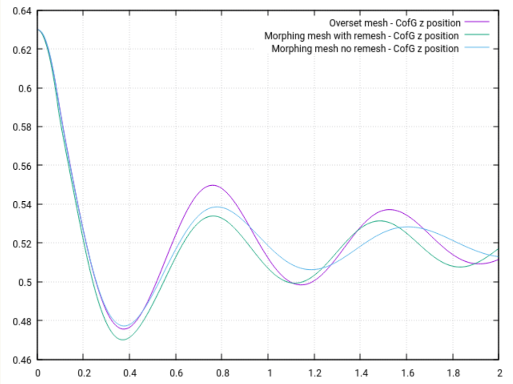
Morphing meshes – Body fitted mesh with remeshing– Water surface visualization
http://www.wolfdynamics.com/training/dynamicMeshes/dof4.gif



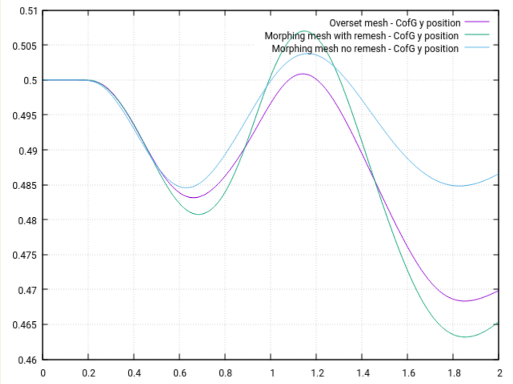
Overset meshes – Water surface visualization
http://www.wolfdynamics.com/training/dynamicMeshes/overset_rbm2.gif

3. Applications – overset meshes in action

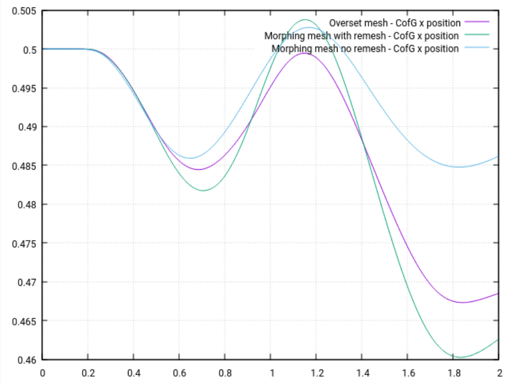
- Falling and floating body – Rigid body motion – Comparison of different meshing techniques.
  - Comparison of the body dynamics using three different approaches to deal with the rigid body motion.



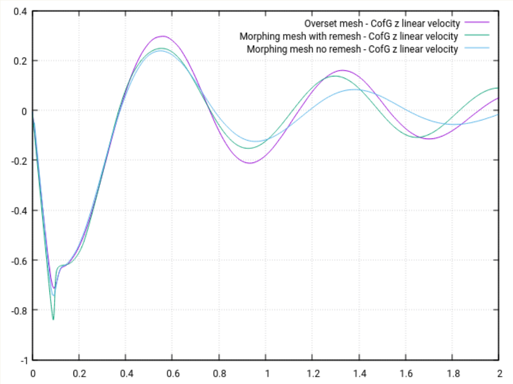
CofG z position vs. Time



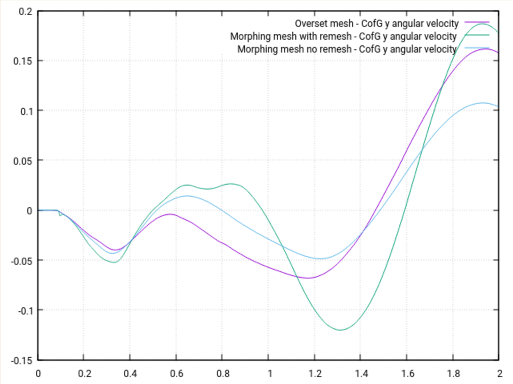
CofG y position vs. Time



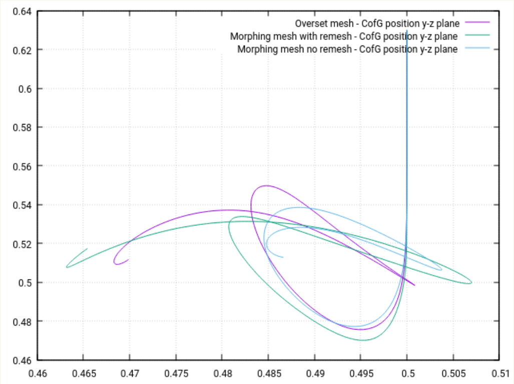
CofG x position vs. Time



CofG z linear velocity vs. Time



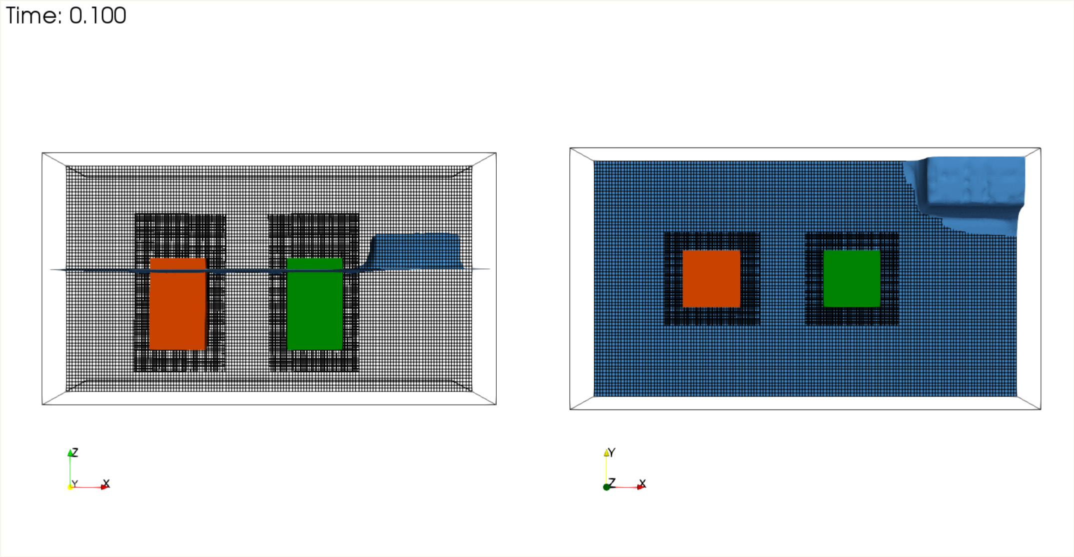
CofG angular velocity about axis y vs. Time

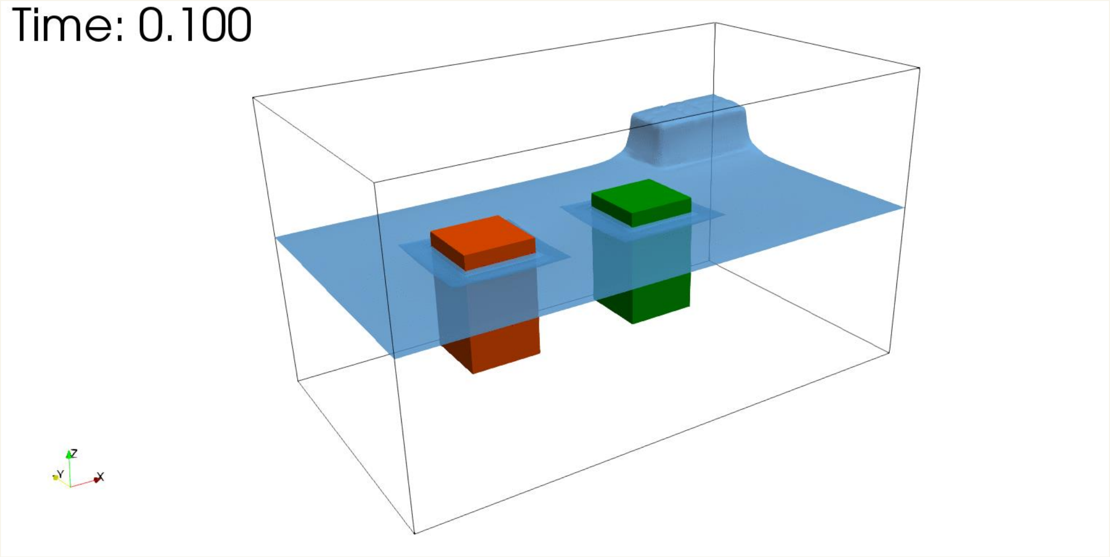


CofG position in the plane y-z vs. Time

3. Applications – overset meshes in action

- Rigid body motion with multiple bodies – Free surface computation (VOF)





http://www.wolfdynamics.com/training/dynamicMeshes/floating_overset1.gif

http://www.wolfdynamics.com/training/dynamicMeshes/floating_overset2.gif

Chapter 4

Guidelines when working with overset meshes in OpenFOAM®

4. Guidelines when working with overset meshes

- In overset meshes, cell size close to overset patches should be similar to minimize interpolation errors (the coarser mesh determines the error level).
- There should be at least 4 or more cells between body patches in order to construct a good interpolation stencil. Cells next to a patch are blocking the flow and cells next to the overset patch are used to interpolate the solution.
- It is a good practice to monitor the mesh courant number (**checkMeshCourantNo yes;**). Usually, the mesh CFL number is more restrictive than the flow CFL number.
- For good accuracy and stability, the mesh motion CFL number should be kept below 1.
- The time-step must be small enough to accommodate for a sequential change of the cell type from blocked to interpolated and then calculated.

4. Guidelines when working with overset meshes

- As we usually use overset meshes with moving bodies, it is recommended to use a robust, accurate and stable numerics, with a flow CFL number below 1.
- Use a robust and accurate interpolation method, the leastSquares implementation is a good choice.
- Place the overset interface appropriately, preferably where the field variables do not change much. Avoid strong pressure gradient at the overset patches.
- Overset region walls can not exit the background mesh. Also, bodies are not allowed to collide.
- Plan well the grid priorities assignment (zoneID). Different combinations can give different overset assemblies with unexpected outcomes. As a general rule, always assign zoneID 0 to the background mesh (usually the mesh that it is not moving or the mesh holding the external boundary conditions).

Chapter 5

Main takeaways

5. Main takeaways

- Thanks to overset meshes, a wide range of applications that were extremely difficult or impossible to simulate using traditional meshes (body-fitted single meshes) or using costly remeshing techniques are now easier to solve.
- Simulations involving complex motion (prescribed, 6DOF or FSI) of single or multiple bodies can be carry out in an efficient way.
- Overset meshes can also be used to conduct parametrical studies that previously required continuous remeshing.
- However, do not take overset meshes as a silver bullet. Simulations using overset meshes requires careful planning.
- Current state of overset meshes in OpenFOAM® (1906):
    - It is working with industrial cases. However, it can be improved.

FYI – Useful links

- A few of the cases presented are shared publicly.
    - Fixed cylinder.
    - Flapping foil.
    - Falling and floating body.

- You can download the working cases from GitHub:
    - https://github.com/wolfDynamics/midwest_OF_2019_overset/

- Feel free to visit our YouTube channel, where you will find more videos showing how to use overset meshes in OpenFOAM. You can find us on YouTube at the following link,

https://www.youtube.com/channel/UCNNBm3KxVS1rGeCVUU1p61g

Thank you very much for your attention

Be collaborative, be innovative, be cloud

wolf dynamics
multiphysics simulations, optimization & data analytics
www.wolfdynamics.com

guerrero@wolfdynamics.com

www.wolfdynamics.com