ESI OpenFOAM CONFERENCE
in Germany | OCT 15-17, 2019

A comparative assessment and benchmarking study of OpenFOAM® overset meshes capabilities

Joel Guerrero – Wolf Dynamics SRL

16/10/2019

esi get it right®

www.esi-group.com

Who am I?

- My name is Joel Guerrero and I am a researcher at the University of Genoa (Italy).

- I am also the CTO and technical curriculum developer 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 engineering.

- I have memorized 8 digits of π.

http://www3.dicca.unige.it/guerrero/
http://www.wolfdynamics.com

What is Wolf Dynamics?

- Wolf Dynamics is a spin-off of the University of Genoa.
- It was created to fill the gap between University and Industry in the Liguria region, Italy (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 also offer training services and serve as an incubator for new graduates looking to learn more about multiphysics simulations.

wolf dynamics

multiphysics simulations, optimization & data analytics

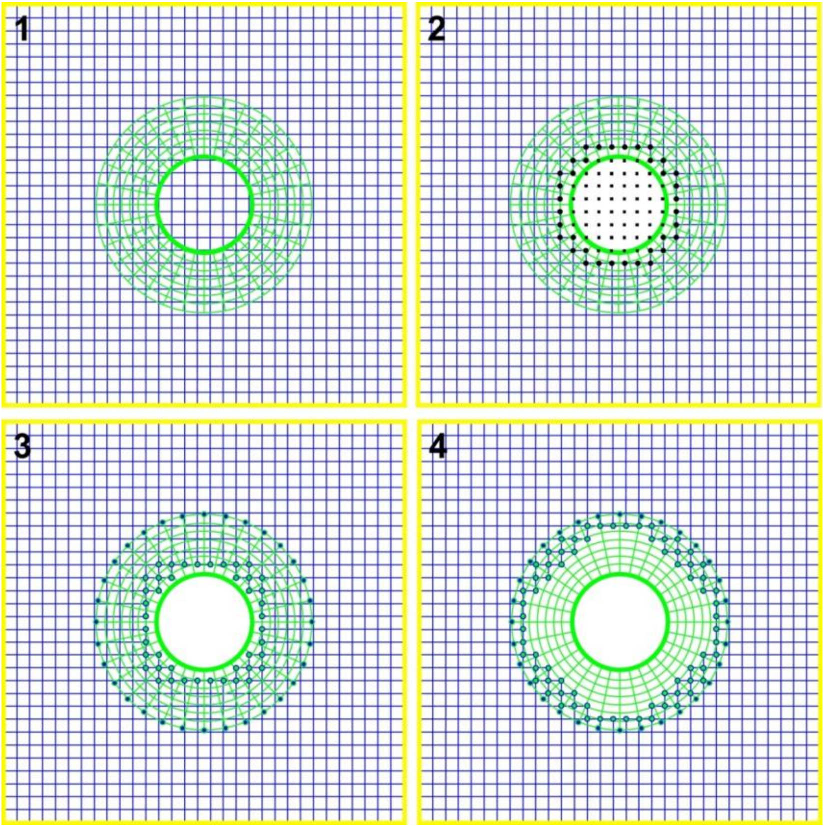
www.wolfdynamics.com

Agenda

1. Overview of overset meshes – Development timeline
2. The grammar of overset meshes in OpenFOAM®
3. Benchmarking cases
4. What is missing? What can be improved?
5. Some basic guidelines when working with overset meshes
6. Main takeaways

1. Overview of overset meshes – Development timeline

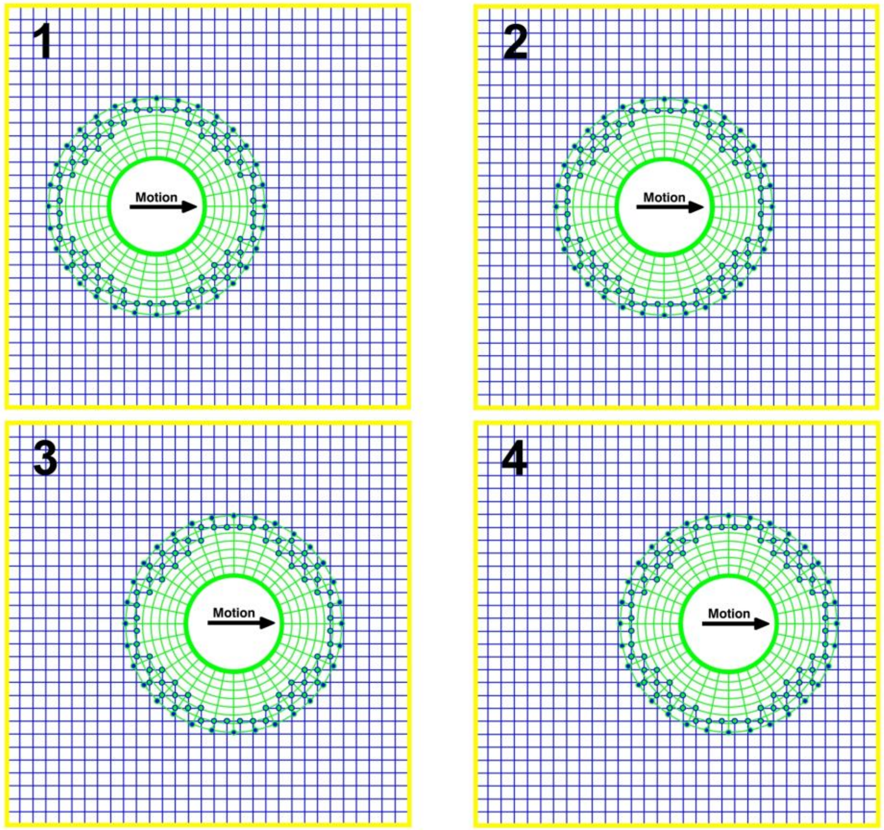
- The overset meshes (**OM**) method consists in generating a set of component meshes (**CM**) that cover the domain and overlap where they meet.
- Domain connectivity between the **CM** is obtained through proper interpolation in the overlapping areas.
- The **CM** can be structured or unstructured.
- In the CFD community, the **OM** method 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 and complex geometries (think structured meshes/solvers).
- **OM** are also known as overlapping grids, overset composite grids, composite overlapping meshes, chimera meshes, patches grids, composite grids.



1. Component meshes (**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 – Mesh set with the minimum overlap region.

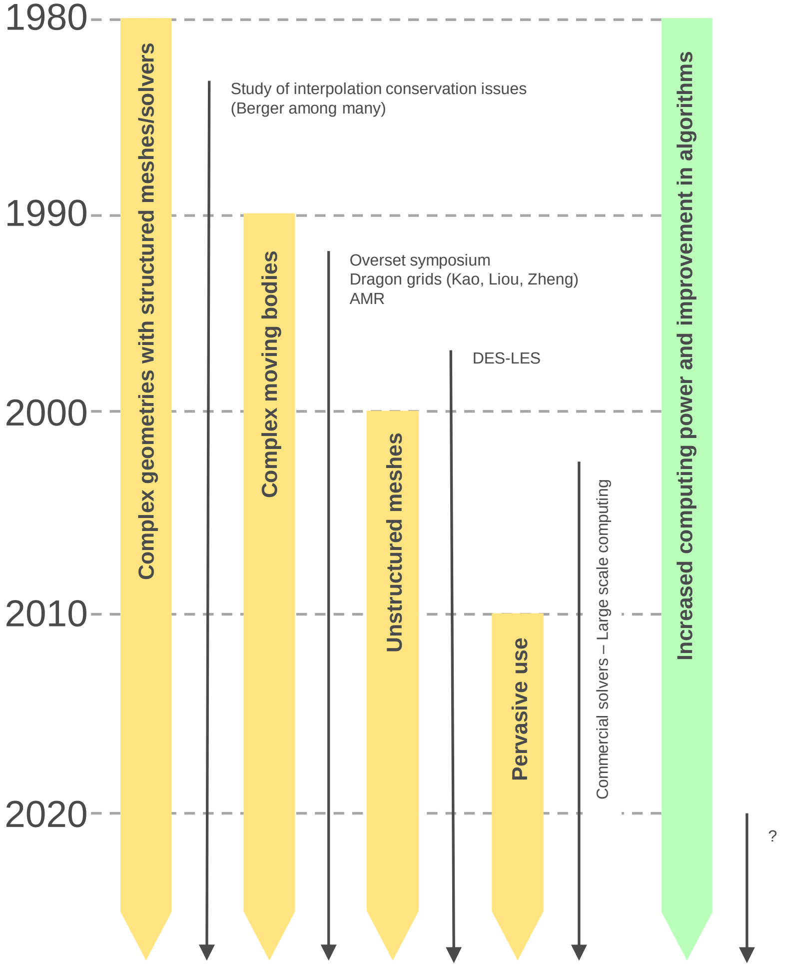
1. Overview of overset meshes – Development timeline

- If the **CM** are moving, overset connectivity information, such as interpolation stencils and unused points regions (Chimera holes), is recomputed each time-step.
- The motion of the **CM** may be a 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.
- **OM** can easily handle multiple bodies undergoing relative motion.
- They can even handle 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 a mesh set with the minimum overlap between component meshes.
- But sets with larger overlap regions can be used as well.

1. Overview of overset meshes – Development timeline



- Maybe the first use of overlapping grids was reported by Volkov in the late 1960's.
- The method was further developed and promoted by Starius and Kreiss in the late 1970's.
- It was 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 has been heavily used to deal with complex geometries and moving bodies (Benek, Boger, Bunning, Chan, Chesshire, Dougherty, Gomez, Henshaw, Meakin, Noack, Petersson, Rogers, Steger, Suhs, among many).
- Since 2000's, the use of overset meshes with unstructured meshes gained popularity.
- From 2010's most commercial CFD solvers and many open-source simulation frameworks use **OM**.
- Symposium on Overset Composite Grids and Solution Technology (http://oversetgridsymposium.org/).
    - Biyearly event.
    - First edition took place in 1992 – NASA Ames Research Center, California.
    - Next edition: 2020 – NASA Langley, Virginia.

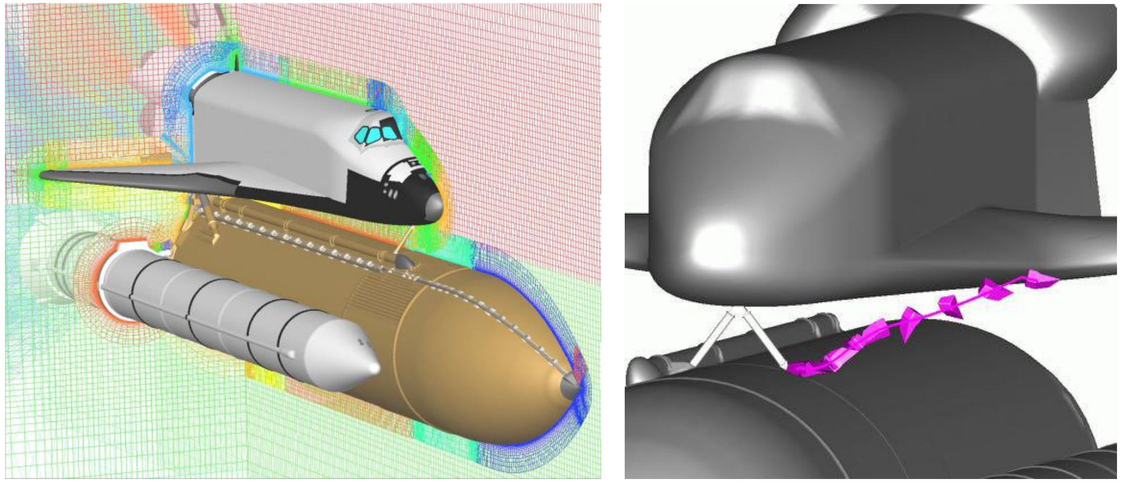
1. Overview of overset meshes – Development timeline

- **Incomplete list of overset solvers and libraries:**
  - Research solvers with overset meshes capabilities:
    - NASA → Overflow^S, CFL3D^S*, FUN3D^U, USM3D^U, TetrUSS^U, INS3D^S
    - US labs and academia → LLNL Overture^S*, Sandia NALU^U*, NIST Overset-HDG^U, CFDShip-Iowa^S
    - Europe → 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 assembling overset meshes (research and commercial):
    - SUGGAR++, DiRTlib, Chimera Grid Tools (CGT), BEGGAR, Ogen*, Maggie, Pegasus, TIOGA, Ronnie, Cassiopee, Pointwise.
  - In OpenFOAM® ecosystem:
    - ESI (1906)*, Foam-Extend*, Bellerophon, Opera, naoe-FOAM-SJTU, FoamedOVER (SUGGAR++), Caelus (SUGGAR++)

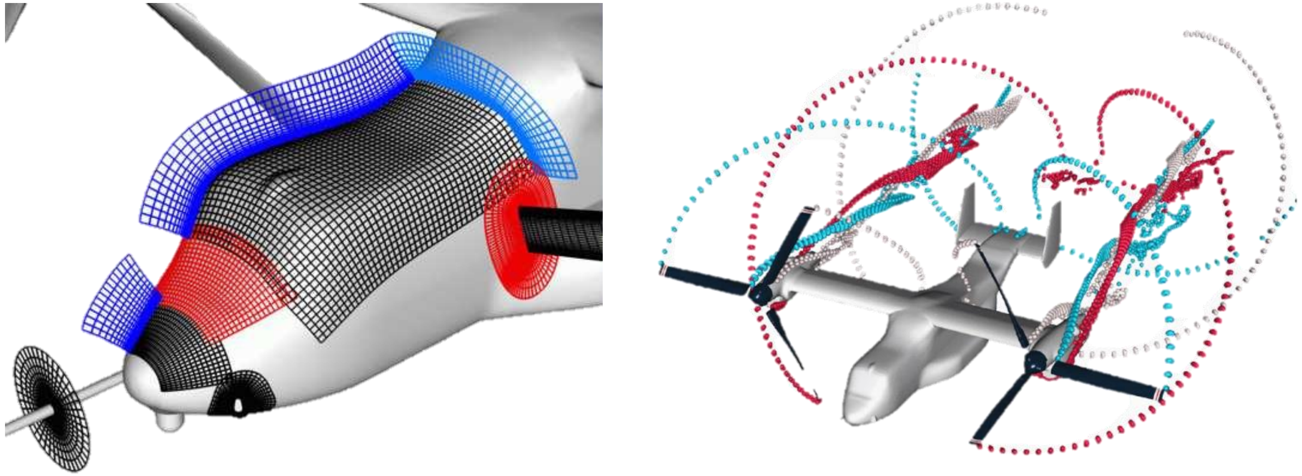
S → Structured, U → Unstructured, * → Solvers tested

1. Overview of overset meshes – Development timeline

- Overset meshes are used to solve the most challenging moving bodies problems.



Space shuttle
Figure credit: P. Buning, W. Chan, R. Gomez, S. Pandya.
Copyright on the images is held by the contributors. Apart from Fair Use, permission must be sought for any other purpose.



V-22 Osprey
Figure credit: W. Chan, R. Meakin, W. Wissink.
Copyright on the images is held by the contributors. Apart from Fair Use, permission must be sought for any other purpose.

2. The grammar of overset meshes in OpenFOAM®

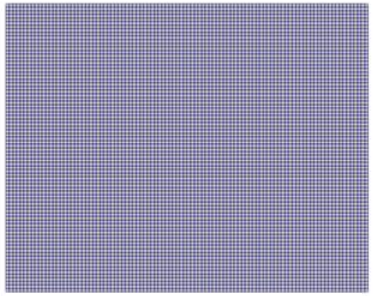
- The process of assembling overset meshes in OpenFOAM® is very straightforward.
- Surprisingly, very similar to that of Ansys Fluent (the second solver we will use for benchmarking).
- Four basic steps are involved:
    1. Generate component meshes and merge them together (done by the user).
    2. 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 library).

- These steps are common for every CFD solver that uses overset meshes.
- The difference is the tools and methods used to merge meshes, assign zones, define grid priorities, compute stencils, and diagnosing the overset assembly.
- Let us illustrated these steps using an overset set with three component meshes. For this, we will use the classical cylinder case (Re = 200).

2. The grammar of overset meshes in OpenFOAM®

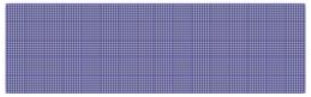
- Step 1 → Generate component meshes and merge them together (done by the user).

- The cell type can be any of the cells supported by OpenFOAM®.
- The meshes can be 2D and 3D.
- The meshes can be generated using any meshing utility (OpenFOAM® or third-party library).



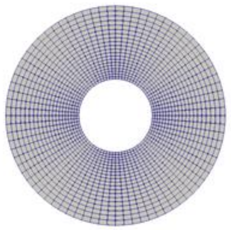
Component mesh 1 → **all**

\+



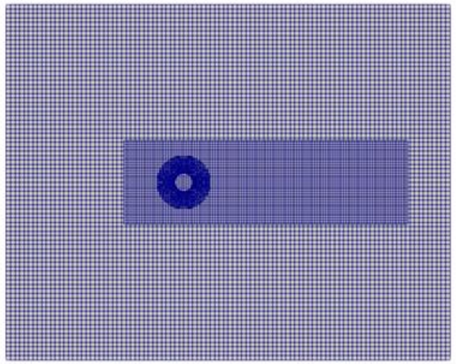
Component mesh 2 → **refinementZone**

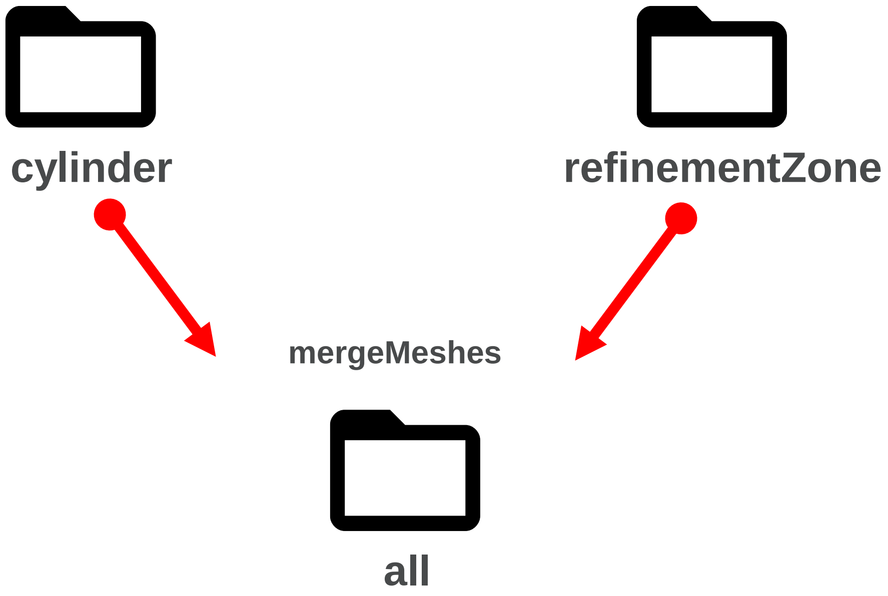
\+



Component mesh 3 → **cylinder**

=

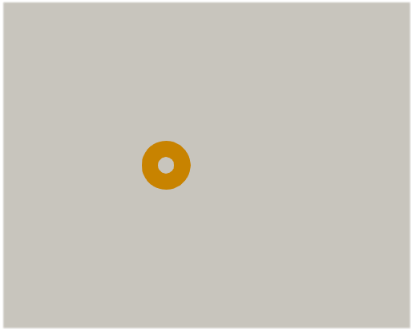




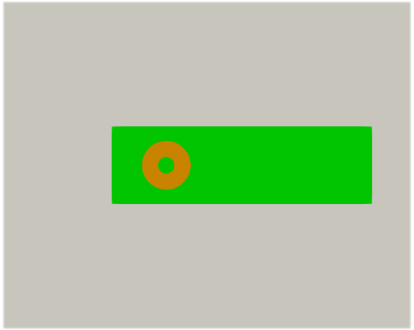
- Each **CM** is considered an individual case; therefore, they are generated in different directories.
- To assemble an overset mesh, you need to generate each **CM** in separated directories.
- Then, you merge them together using the utility **mergeMeshes**.
- You merge the meshes in a single directory. In this case, the component meshes **cylinder** and **refinementZone** are merged in the directory **all**.
- Notice that the directory **all** also contains a mesh (background mesh in this case).

2. The grammar of overset meshes in OpenFOAM®

- Step 1 → Generate component meshes and merge them together (done by the user).
  - **About the order of operations when merging meshes.**
    - In theory, it does not matter the order of the merge operations.
    - At the end, all **CM** should be merged into a single directory.
    - In this case, the **CM cylinder** and **refinementZone** are merged into the **CM all**.
    - The **zoneID** is assigned after merging the meshes.
    - It is highly recommended that the **oversetPatch** be the first one in the boundary file.



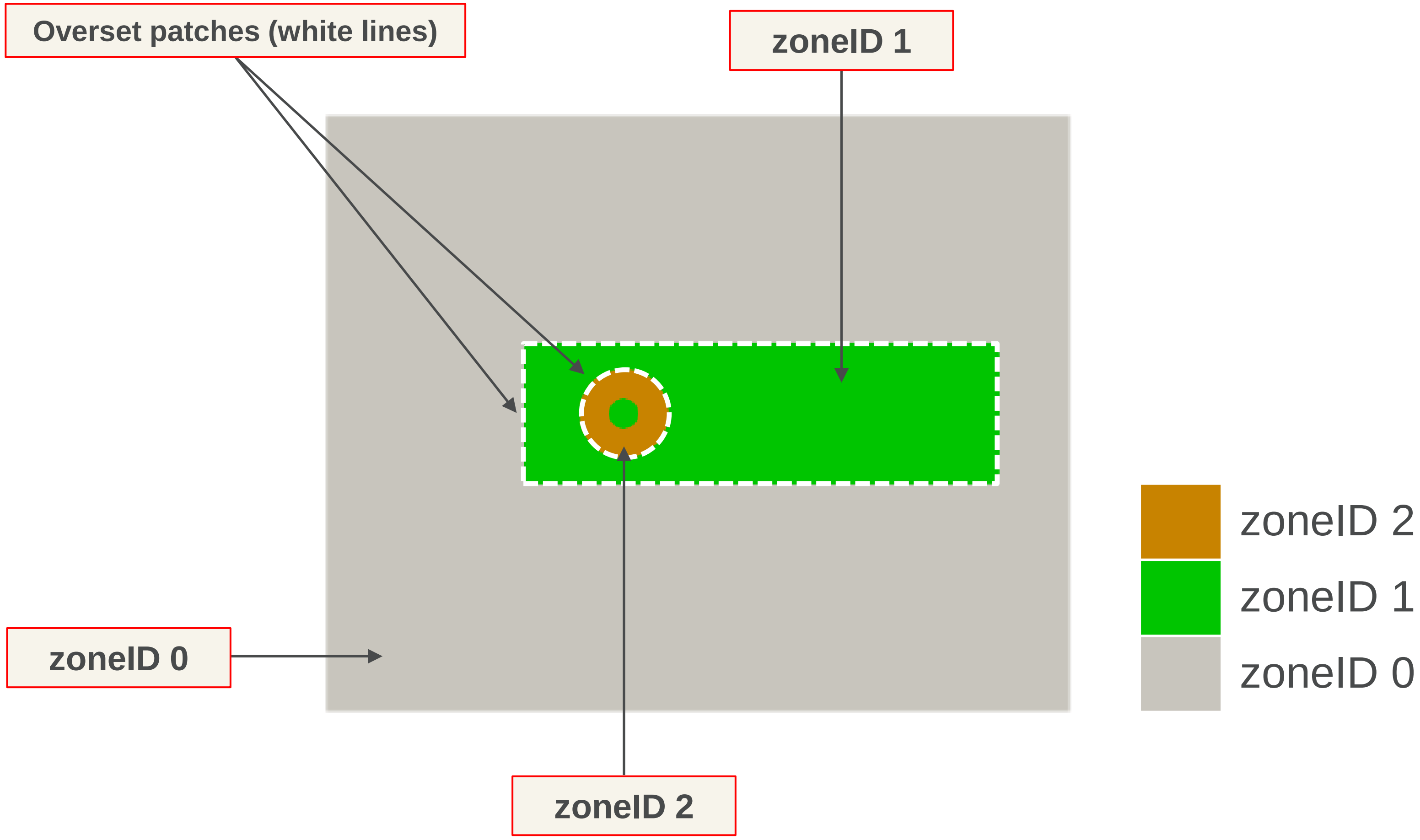
First merge operation → **cylinder** + **all**



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

2. The grammar of overset meshes in OpenFOAM®

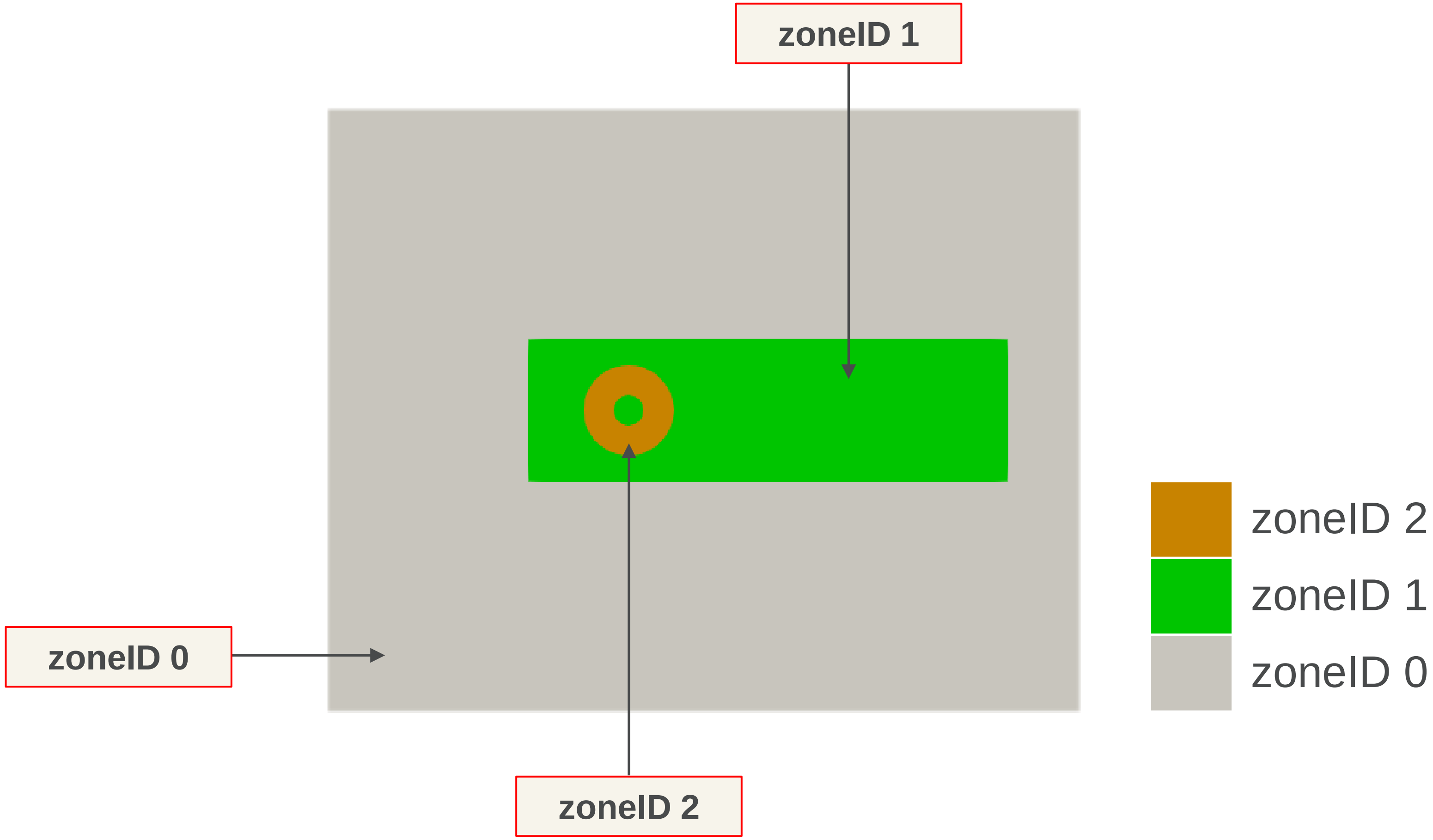
- Step 2 → Define overset patches (done by the user).



- The overset patches are defined by the user.
- They have the same name (defined by the user when generating the **CM**).
- And they are grouped together automatically when merging meshes.
- 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. The grammar of overset meshes in OpenFOAM®

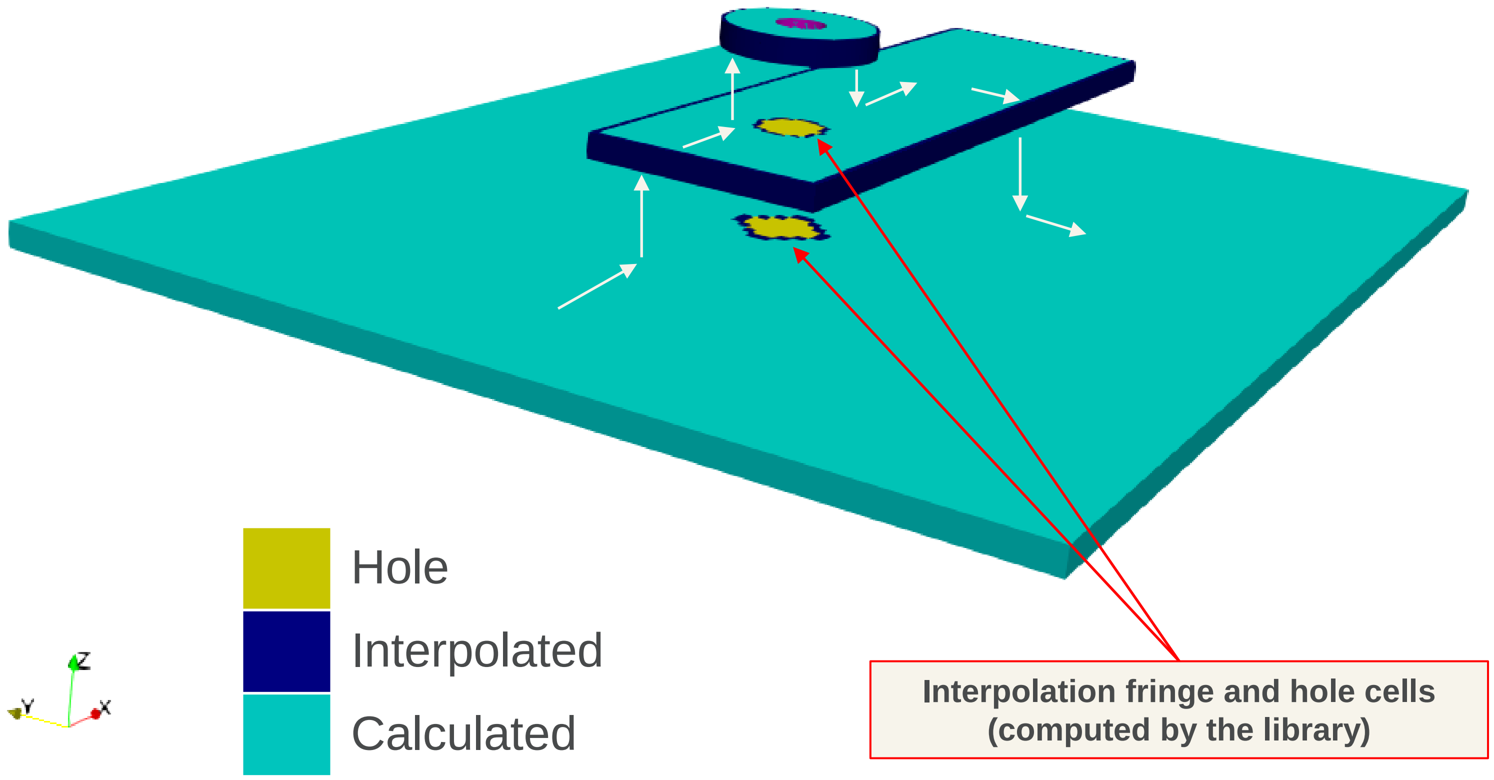
- Step 3 → Assign zones (done by the user).



- A zone identification index (**zoneID**) is assigned to each component mesh after they have been merged.
- It is recommended to assign **zoneID** 0 to the background mesh (**all** in this case).
- The background mesh usually is the mesh that is not moving, the mesh with inlet and outlet patches, or the mesh that does not have overset patches.
- The **zoneID** index is used to establish the grid priorities.

2. The grammar of overset meshes in OpenFOAM®

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

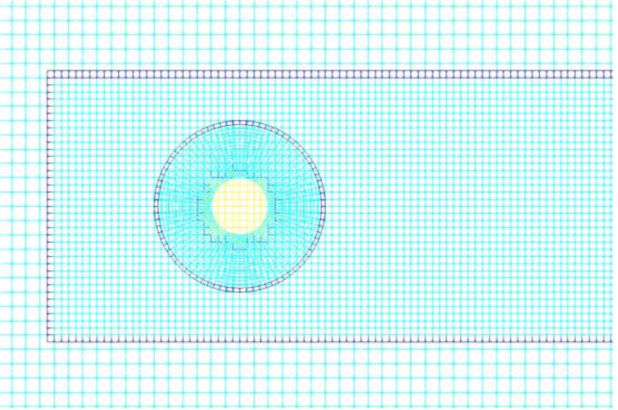


- The **overset patches** of each **CM** are defined by the user.
- The **interpolation fringe** close to the walls and the **hole cells** are computed automatically by the overset library.
- The cell types are defined as follows: **hole cells** (the solution is not computed), **interpolated cells** (the solution is interpolated from mesh-to-mesh), and **calculated cells** (the solution is computed).
- The **interpolated cells** can be classified as **acceptors** (receive information) and **donors** (send information).
  - The **donor** cells can be **interpolated** or **calculated** cells.

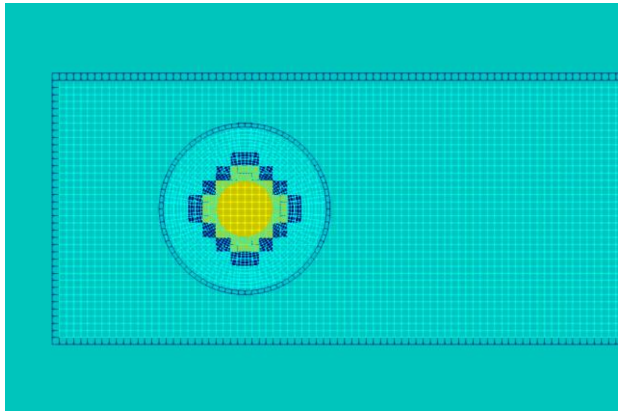
Cell type	Cell type index
Hole	2
Interpolated	1
Calculated	0

2. The grammar of overset meshes in OpenFOAM®

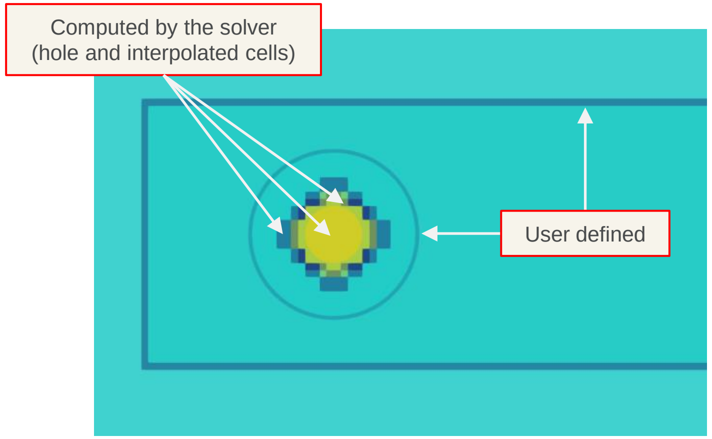
- Step 4 → Compute stencils and assign cell type (done by the overset library).



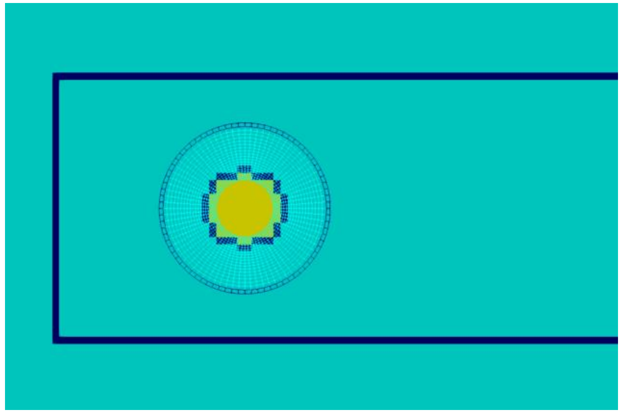
Wireframe visualization – All **CMs**



Wireframe visualization (**refinementZone** and **cylinder CM**)



Contour visualization with transparency – All **CMs**



Wireframe visualization (**cylinder CM**)

- Cells dimension close to interpolated cells should be of the same size to minimize interpolation errors.
- When computing the solution, an overset interpolation method must be chosen.
- Options available:
  - **cellVolumeWight**
  - **inverseDistance**
  - **trackingInverseDistance**
  - **leastSquares** (recommended by us)

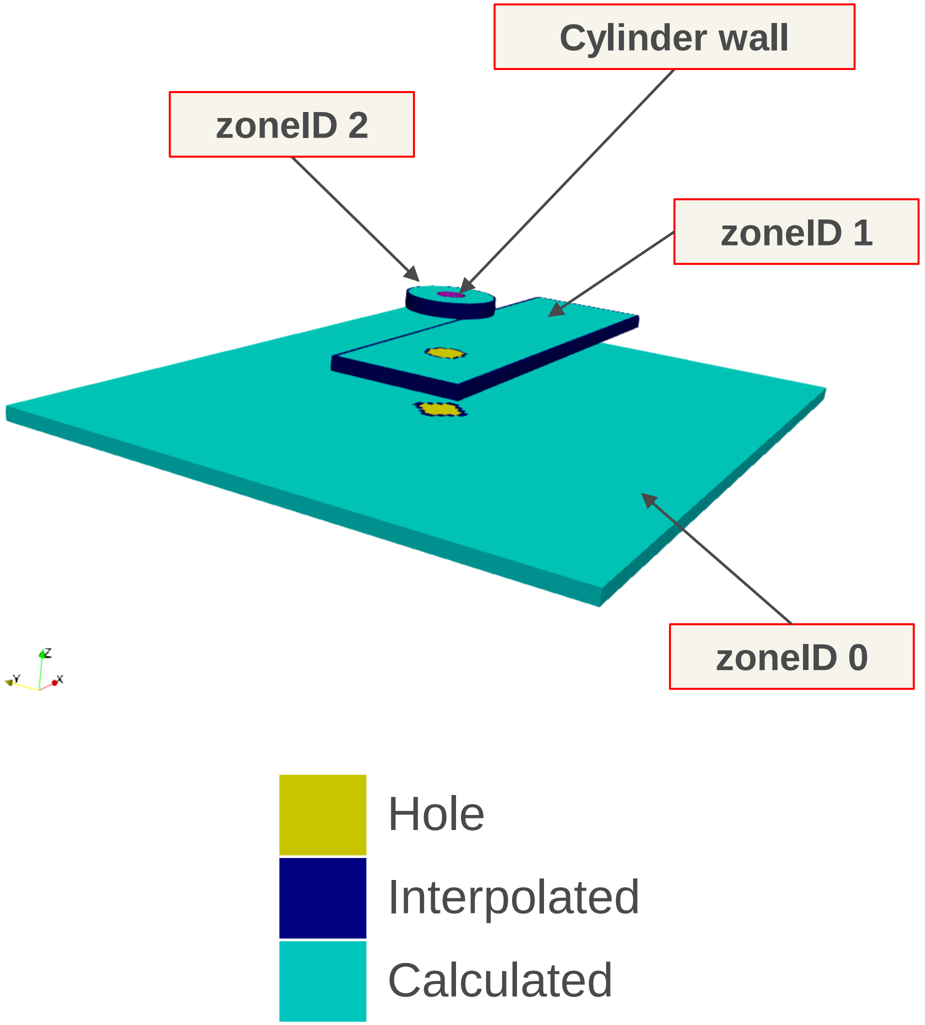
Hole

Interpolated

Calculated

2. The grammar of overset meshes in OpenFOAM®

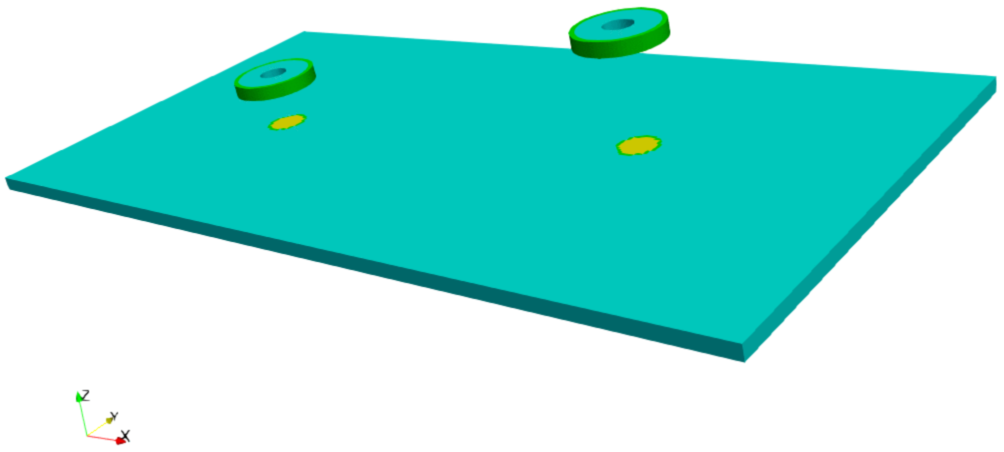
- **About the zoneID priority (or grid priority).**
  - The **zoneID** defines the order of the hole cutting operations on the component meshes.
  - High **zoneID** values, means high priority. That is, that **CM** will cut or imprint lower priority levels.
  - In this case, the **cylinder** mesh has 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 (the **wall**) will cut meshes **refinementZone** and **all**, the mesh **refinementZone** will cut the mesh **all** (if there are **walls**), and so on.
  - Different grid priorities will give you different overset assemblies and interpolation stencils, this must be carefully planned.
  - Remember, the Chimera holes are computed using walls, so if there are no walls, there are no holes.



2. The grammar of overset meshes in OpenFOAM®

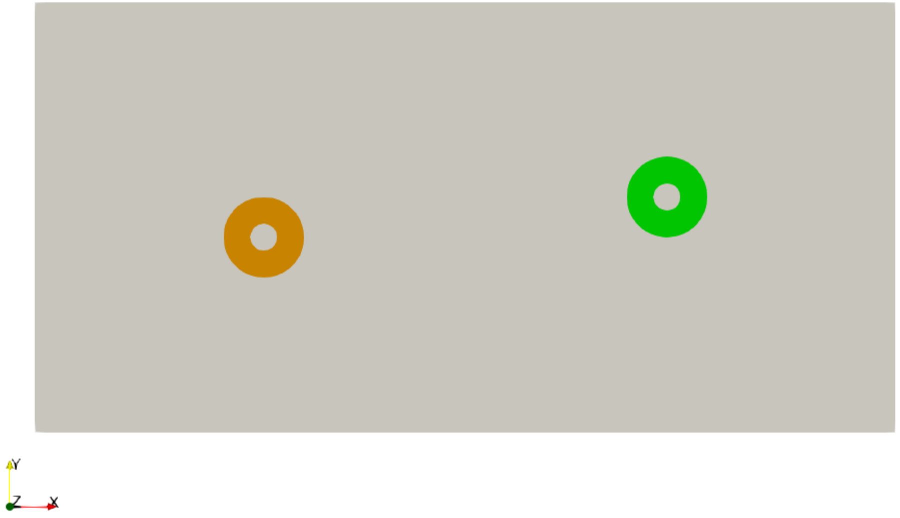
- Multiple bodies undergoing relative motion – Cell types (**cellTypes**) and zones identification (**zoneID**).
- The cell types are recomputed every time-step.

http://www.wolfdynamics.com//wiki/of_conf2019/f2.gif



cellTypes	Index
Hole	2
Interpolated	1
Calculated	0

http://www.wolfdynamics.com//wiki/of_conf2019/f1.gif

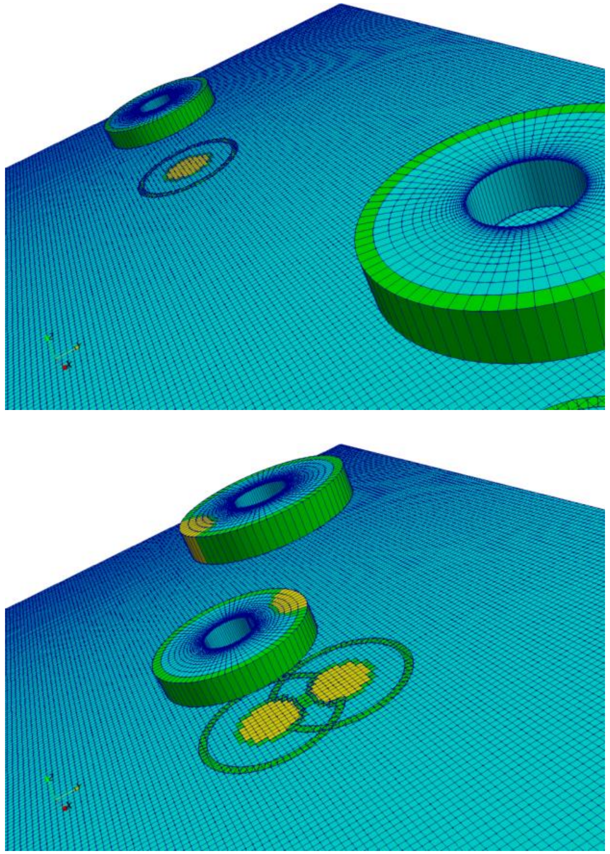


zoneID
2
1
0

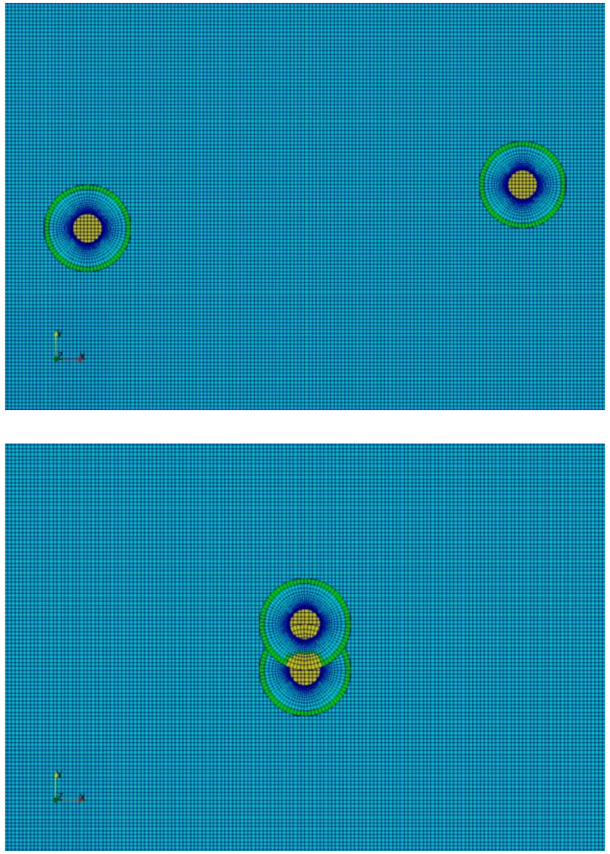
2. The grammar of overset meshes in OpenFOAM®

- Multiple bodies undergoing relative motion – Cell types (**cellTypes**) and zones identification (**zoneID**).
- The cell types are recomputed every time-step.

http://www.wolfdynamics.com//wiki/of_conf2019/f4.gif



http://www.wolfdynamics.com//wiki/of_conf2019/f3.gif

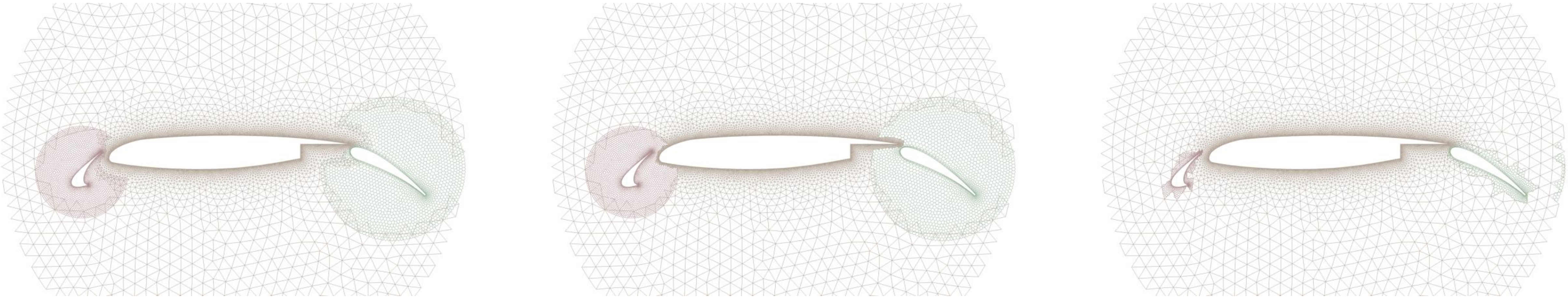


- In this case, the order of the **zoneID** or grid priorities does not make any difference as the cylinders **CM** are identical.
- But if the cylinders **CM** were different, the grid priorities will result in different chimera holes and interpolation stencils.
- The selection of the grid priorities should be planned in advanced. High grid priority means that the **CM** will cut or imprint lower priority grids.

	cellTypes	Index
	Hole	2
	Interpolated	1
	Calculated	0

2. The grammar of overset meshes in OpenFOAM®

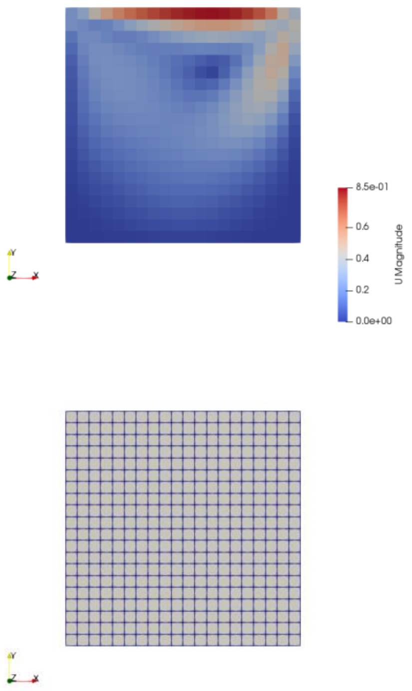
- **Influence of different grid priorities.**
  - The results shown were computed with Ansys Fluent with minimum overlap region.
  - OpenFOAM® and Ansys Fluent maximum overlap were similar.



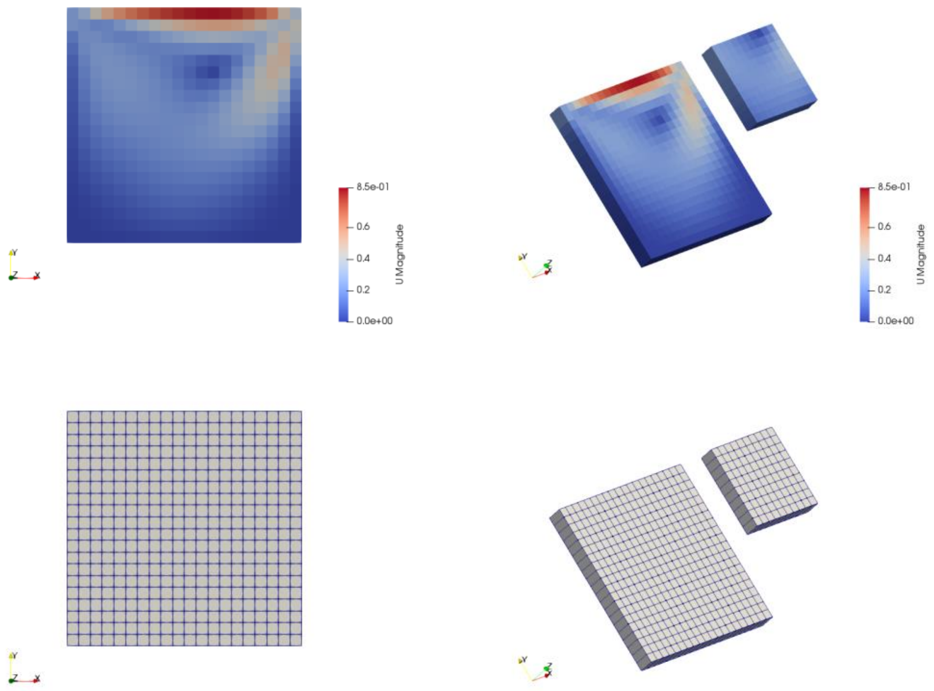
3. Benchmarking cases

- Driven cavity – Re = 100.
- Meshes generated using OpenFOAM® tools – Simulations conducted using OpenFOAM®.

Single body fitted mesh



Overset mesh – Perfectly matching meshes – No Chimera hole optimization

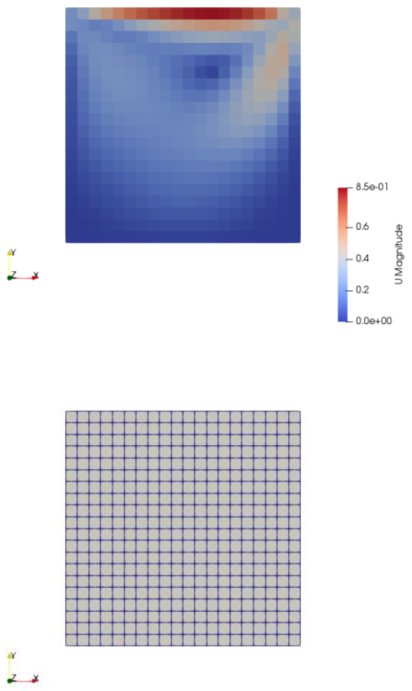


- When meshes are perfectly aligned, and have similar cell-size, the interpolation is one-to-one.
- Therefore, it is as close as possible to be conservative.
- But this is the exception, rather than the rule.
- All the overset interpolation methods implemented in OpenFOAM® 1906. will give good results in this case.

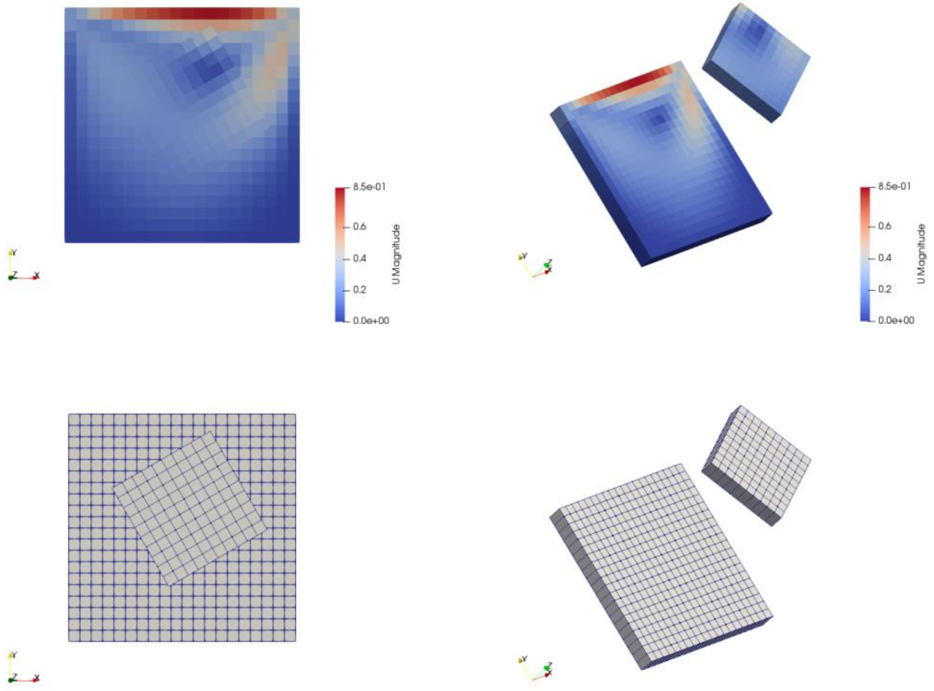
3. Benchmarking cases

- Driven cavity – Re = 100.
- Meshes generated using OpenFOAM® tools – Simulations conducted using OpenFOAM®.

Single body fitted mesh



Overset mesh – Component meshes misaligned – No Chimera hole optimization

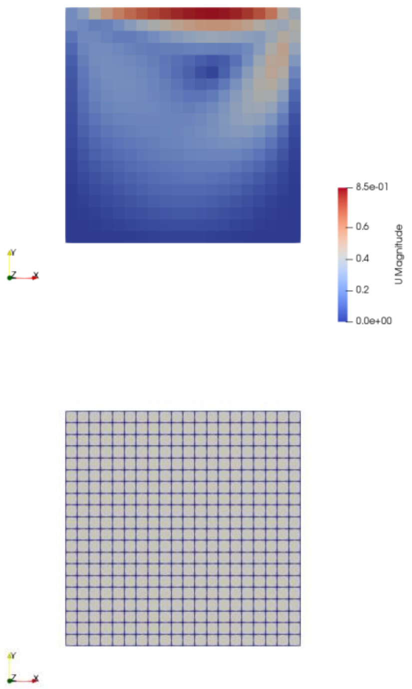


- As soon as we introduce some mismatching in the meshes, the overset interpolation starts to introduce errors in the solution.
- However, as we reach grid independent solutions, the errors introduced by the overset interpolation are reduced (as in many other things in CFD).
- Again, it is very important to have cells with similar sizes in the overset boundary.
- We have found that the best overset interpolations method is the **leastSquares**

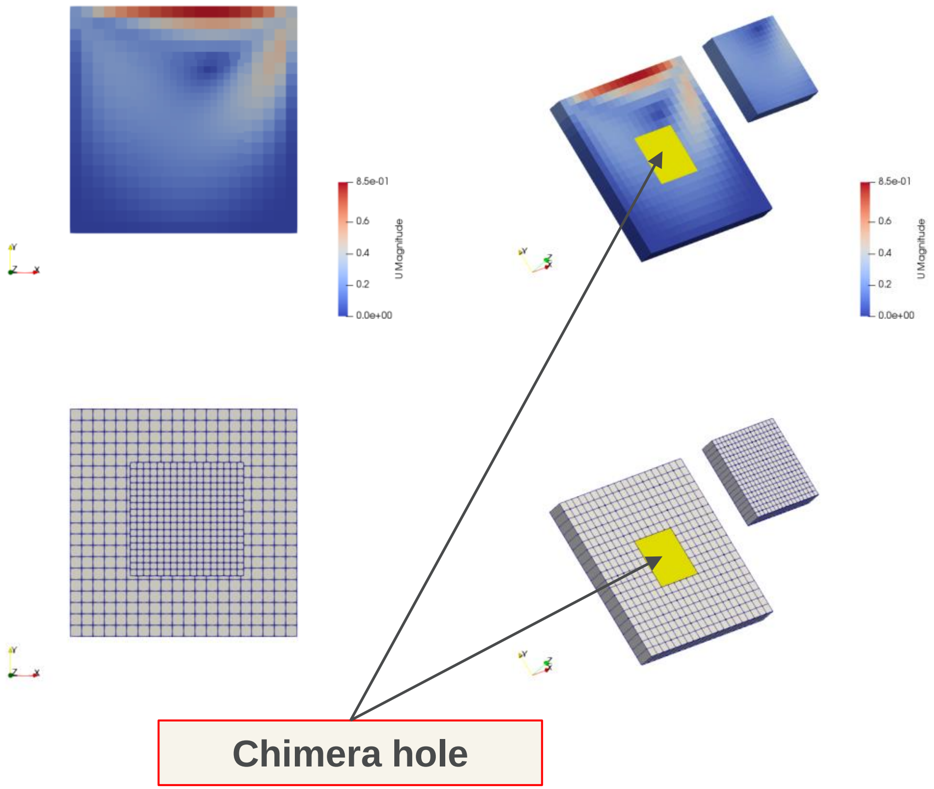
3. Benchmarking cases

- Driven cavity – Re = 100.
- Meshes generated using OpenFOAM® tools – Simulations conducted using OpenFOAM®.

Single body fitted mesh



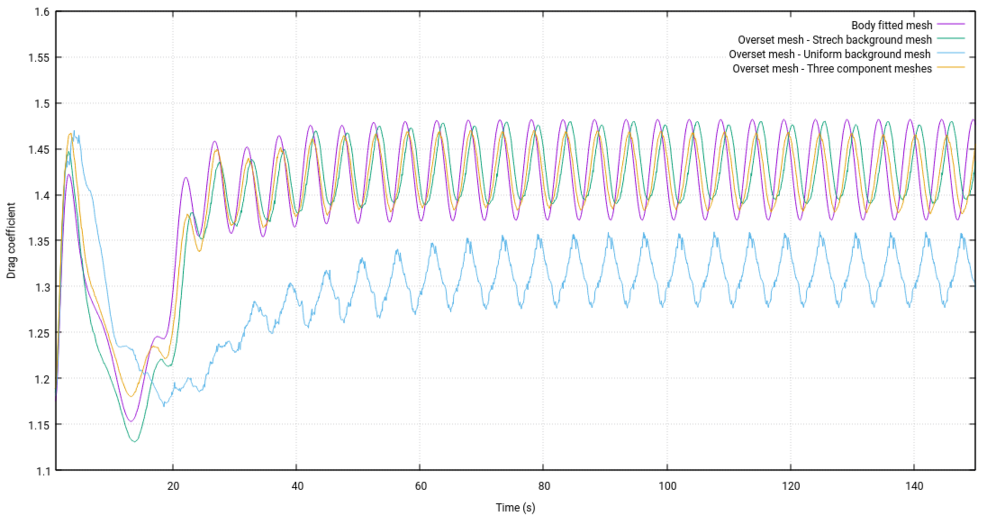
Overset mesh – Different component meshes – Chimera hole optimization

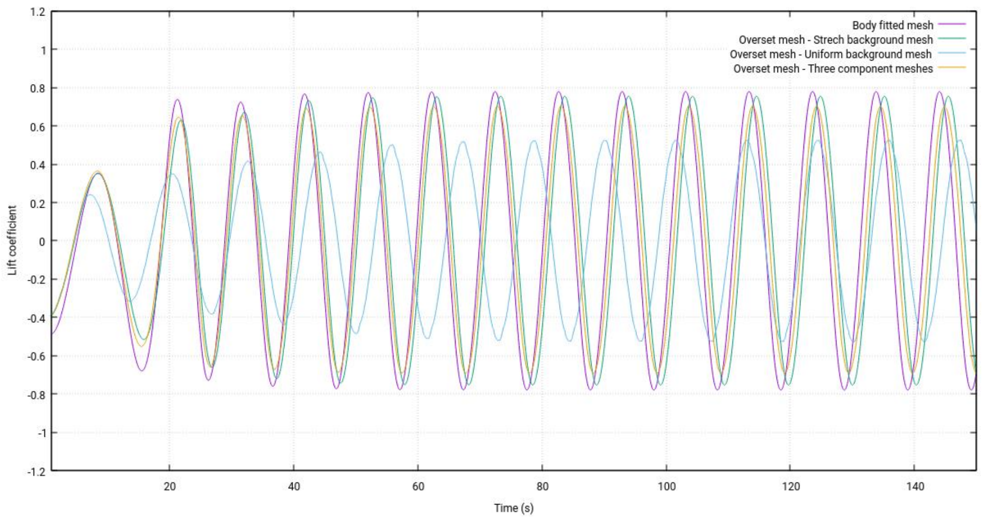


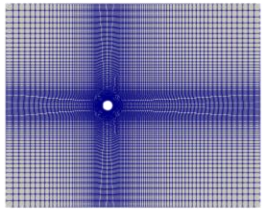
- In the previous cases, we did not use Chimera hole optimization.
- An optimized Chimera hole will minimize the overlap area.
- Therefore, it reduces the cell count by removing unnecessary cells.
- It also reduces interpolation errors related to the one-to-one interpolation.
- To our knowledge, Chimera hole optimization is not implemented in the overset library released with OpenFOAM® 1906.

3. Benchmarking cases

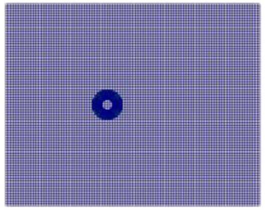
- Flow about a fixed cylinder using overset meshes – Re = 200.
- Meshes generated using OpenFOAM® tools – Simulations conducted using OpenFOAM®.



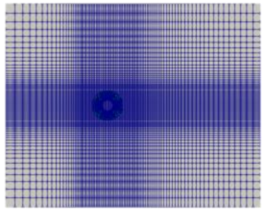




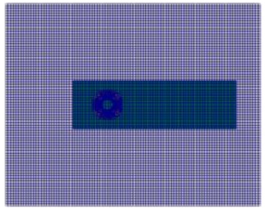
Single body fitted mesh



Overset mesh – Uniform background mesh – Two component meshes



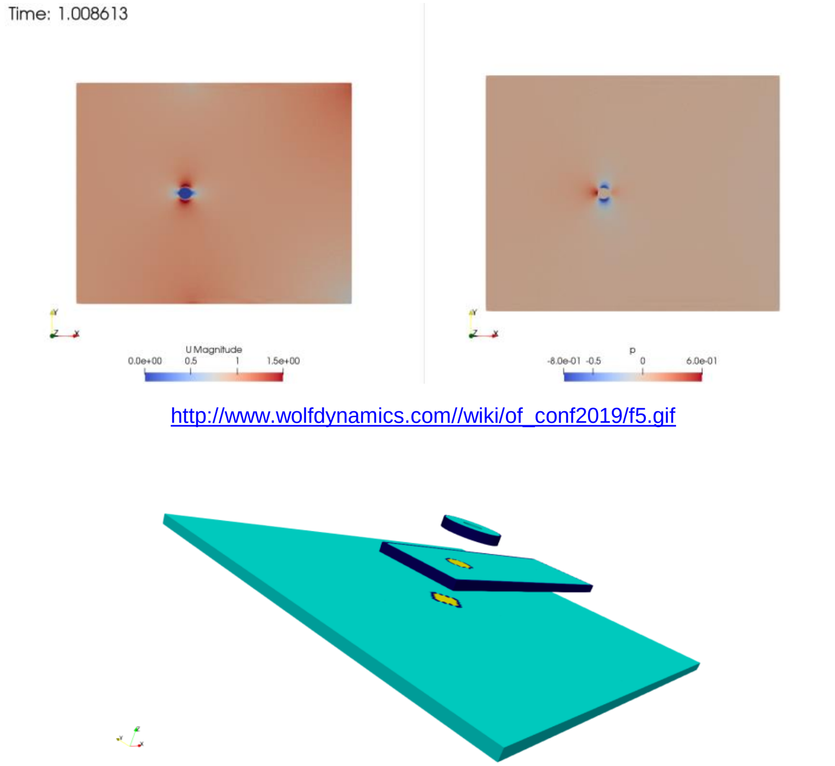
Overset mesh – Stretched background mesh – Two component meshes



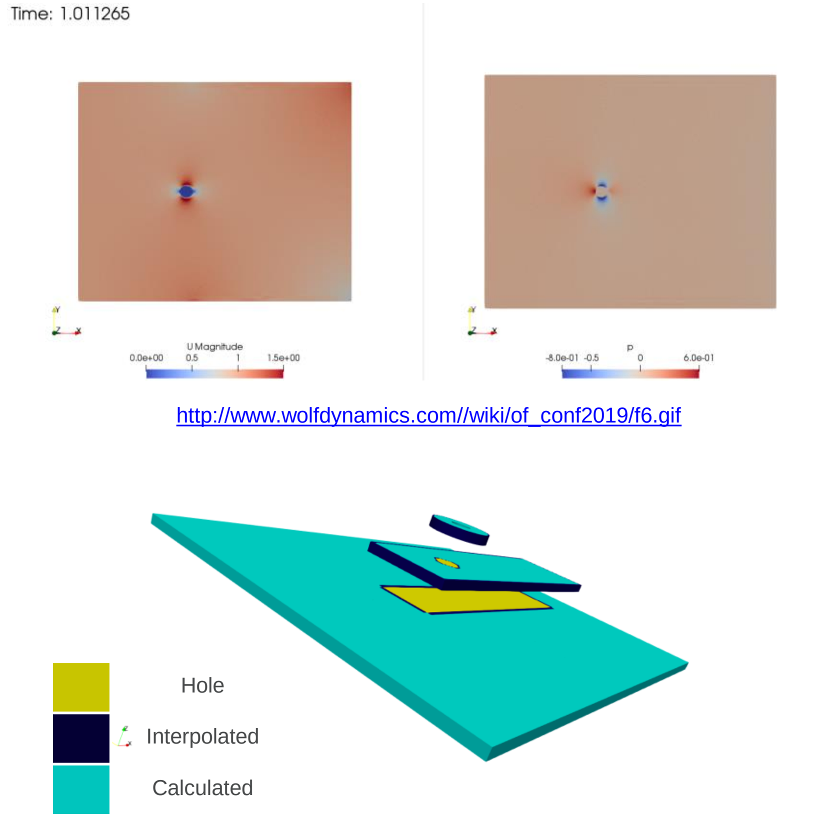
Overset mesh – Three component meshes

3. Benchmarking cases

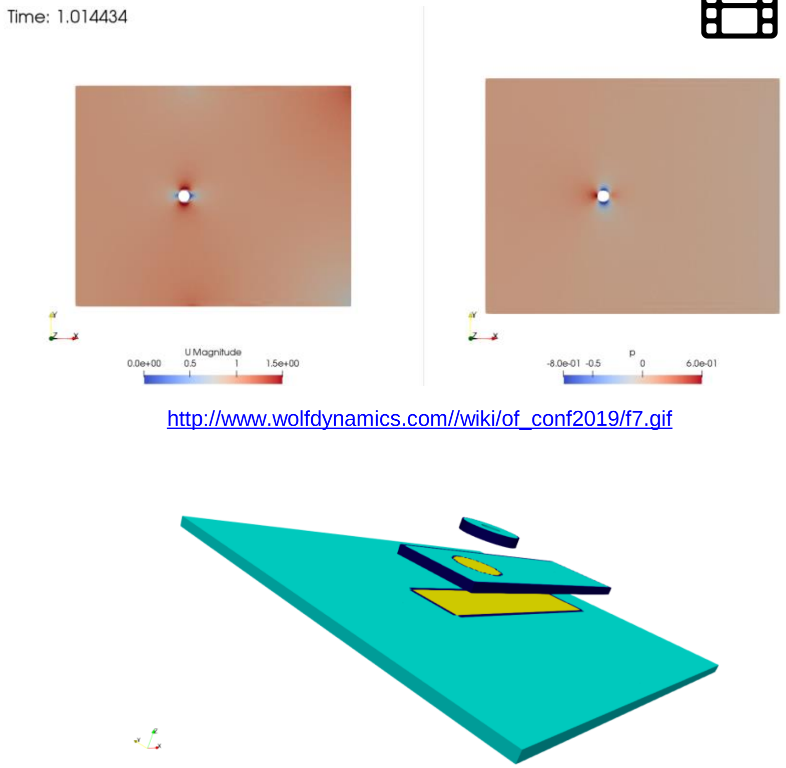
- Flow about a fixed cylinder using overset meshes – Re = 200.
- From a quantitative point of view these cases shown similar trends, qualitative speaking there are some differences.



http://www.wolfdynamics.com//wiki/of_conf2019/f5.gif



http://www.wolfdynamics.com//wiki/of_conf2019/f6.gif

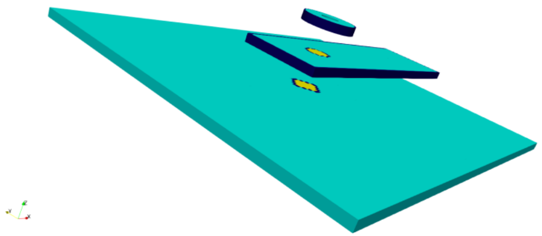


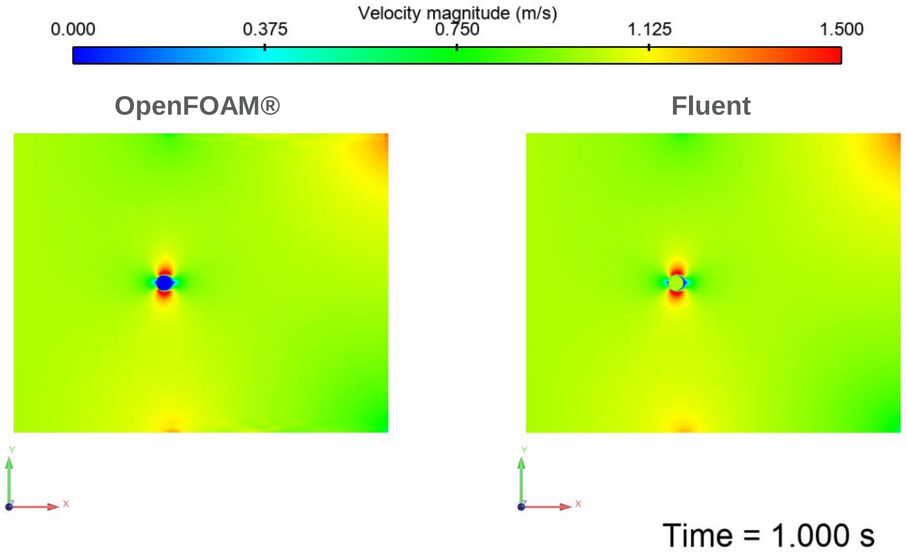
http://www.wolfdynamics.com//wiki/of_conf2019/f7.gif

NOTE: the optimized Chimera holes are not computed by the overset library

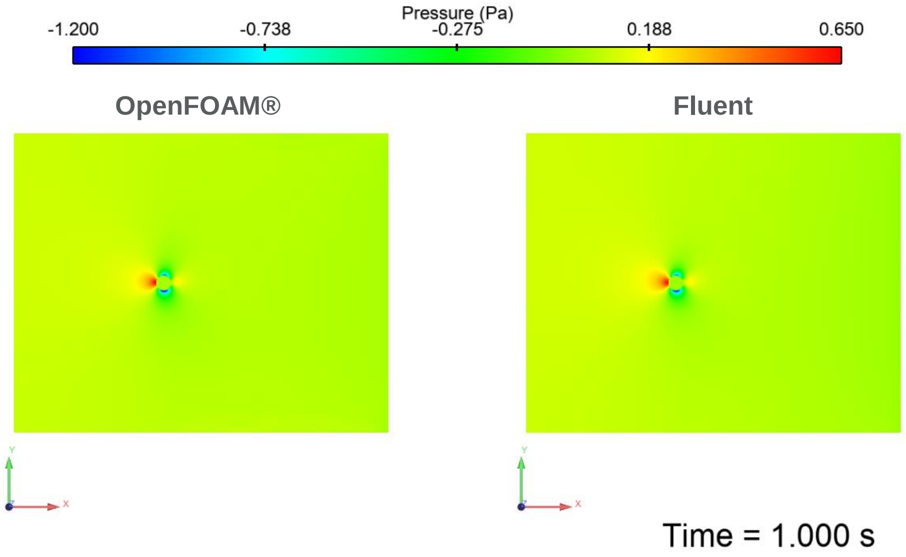
3. Benchmarking cases

- Flow about a fixed cylinder using overset meshes – Re = 200.
- No Chimera hole minimization.
- Simulations conducted using OpenFOAM® and Fluent.





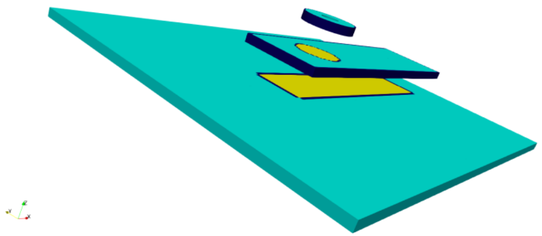
http://www.wolfdynamics.com//wiki/of_conf2019/f8.gif

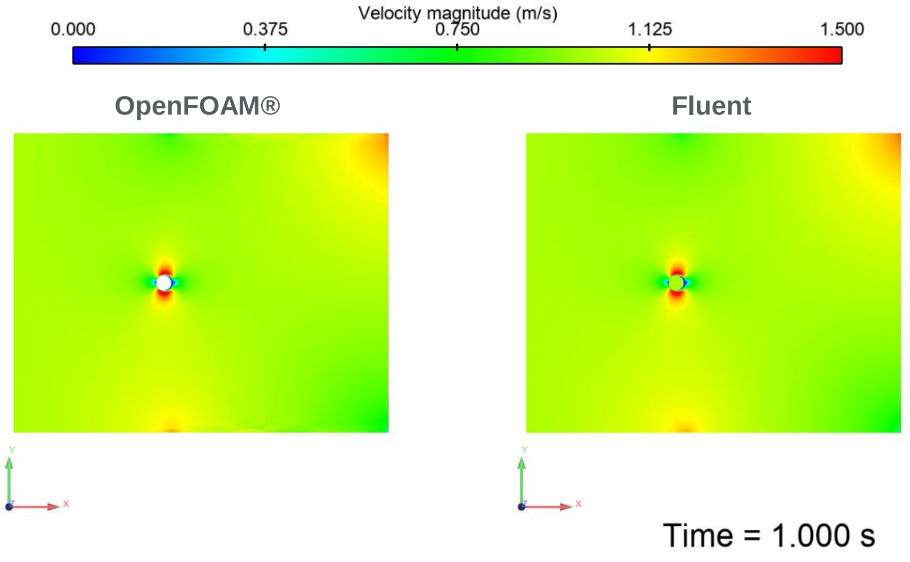


http://www.wolfdynamics.com//wiki/of_conf2019/f9.gif

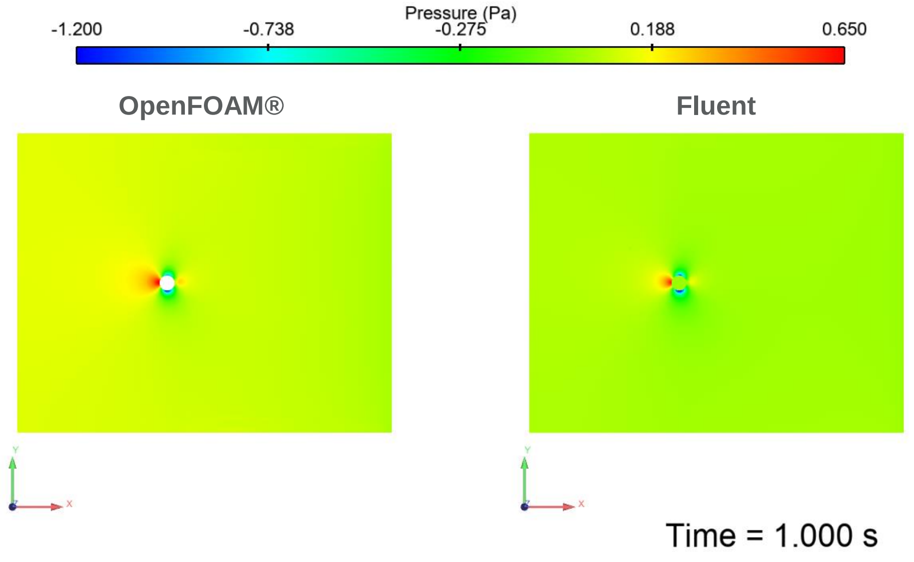
3. Benchmarking cases

- Flow about a fixed cylinder using overset meshes – Re = 200.
- Chimera hole minimization (minimum overlap region).
- Simulations conducted using OpenFOAM® and Fluent.





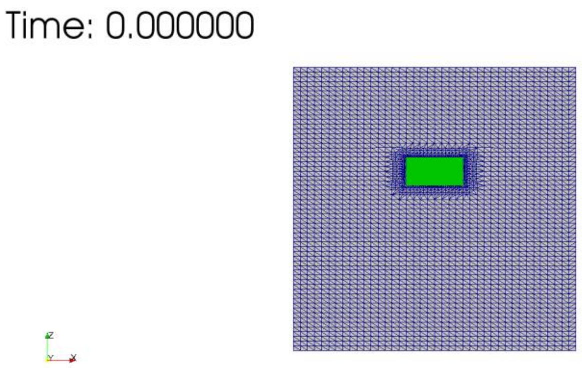
http://www.wolfdynamics.com//wiki/of_conf2019/f10.gif



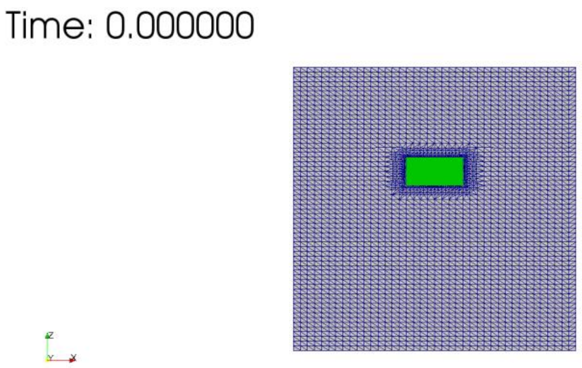
http://www.wolfdynamics.com//wiki/of_conf2019/f11.gif

3. Benchmarking cases

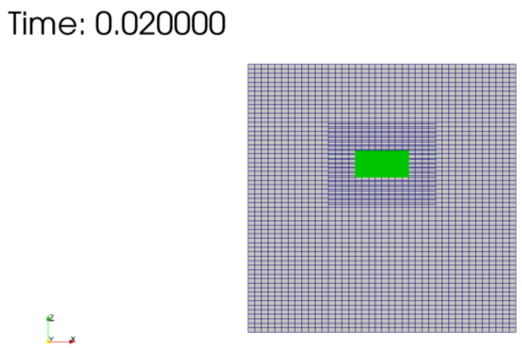
- Falling and floating body – Rigid body motion – Comparison of different meshing techniques in OpenFOAM®.
- Meshes generated using OpenFOAM® tools.



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

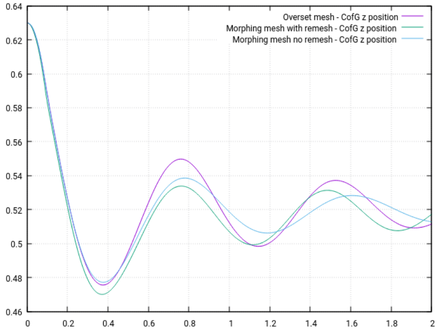


Overset meshes
http://www.wolfdynamics.com/training/dynamicMeshes/overset_rbm1.gif

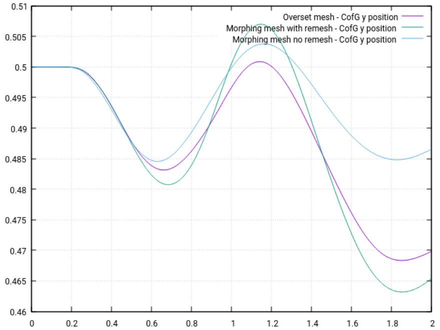
When the quality is low, you stop the simulation, extract body position, generate a new mesh, and map the solutions. Requires some level of automation.

3. Benchmarking cases

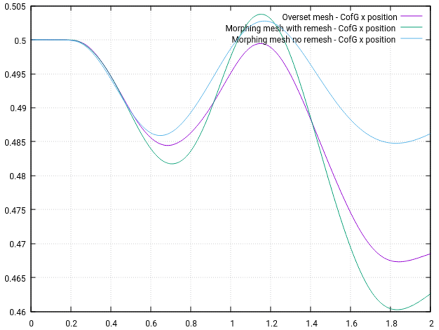
- Falling and floating body – Rigid body motion – Comparison of different meshing techniques.
  - Comparison of the body dynamics using three different approaches to deal with the rigid body motion.



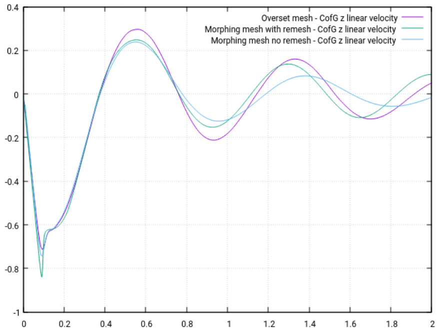
CofG z position vs. Time



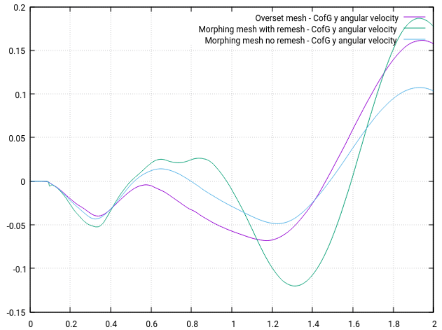
CofG y position vs. Time



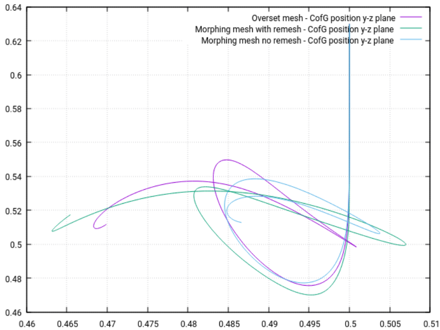
CofG x position vs. Time



CofG z linear velocity vs. Time



CofG angular velocity about axis y vs. Time

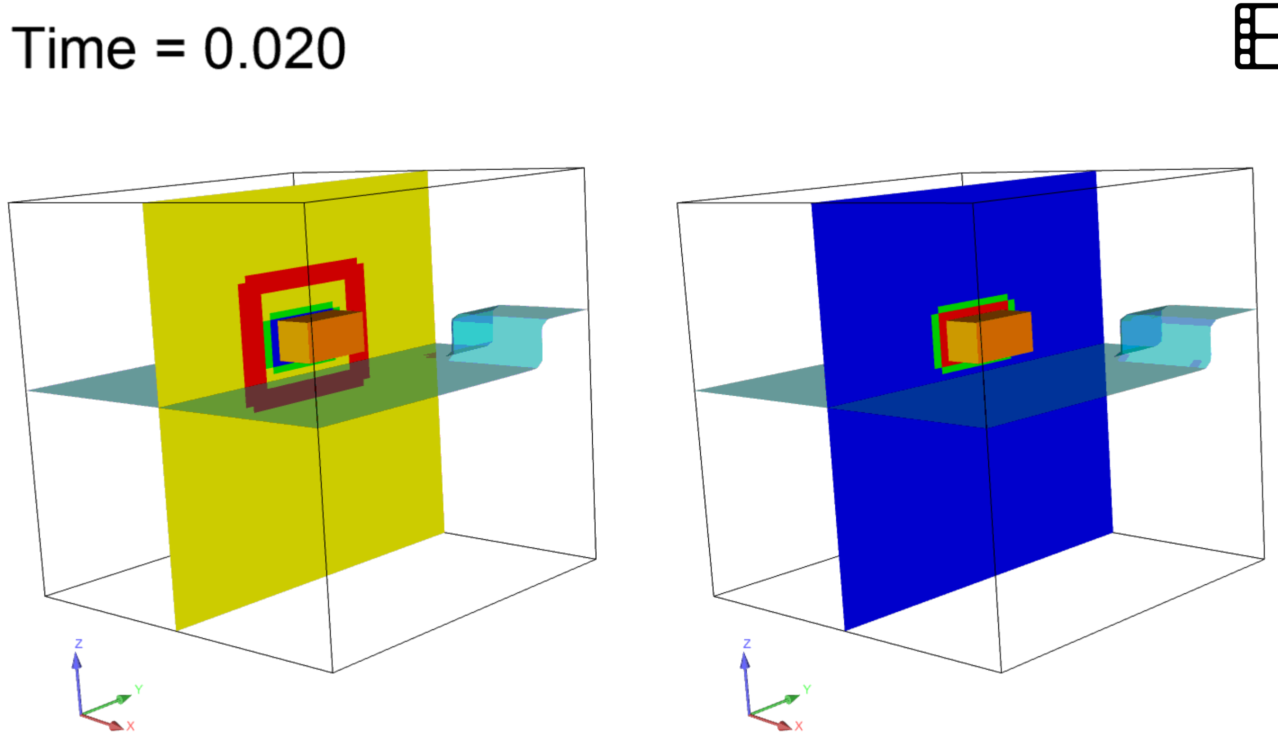


CofG position in the plane y-z vs. Time

3. Benchmarking cases

- Falling and floating body – Rigid body motion.
- Simulations conducted using OpenFOAM® and Fluent.

- The computed stencils are very similar.
- There were no orphan cells in both cases.
- The computing time and cell count for the cutting hole and stencil computation were very similar for both solvers.
- Both solvers were tested using the body position and orientation obtained from the CFD simulations.
- OpenFOAM® overset library performed very well in this case.



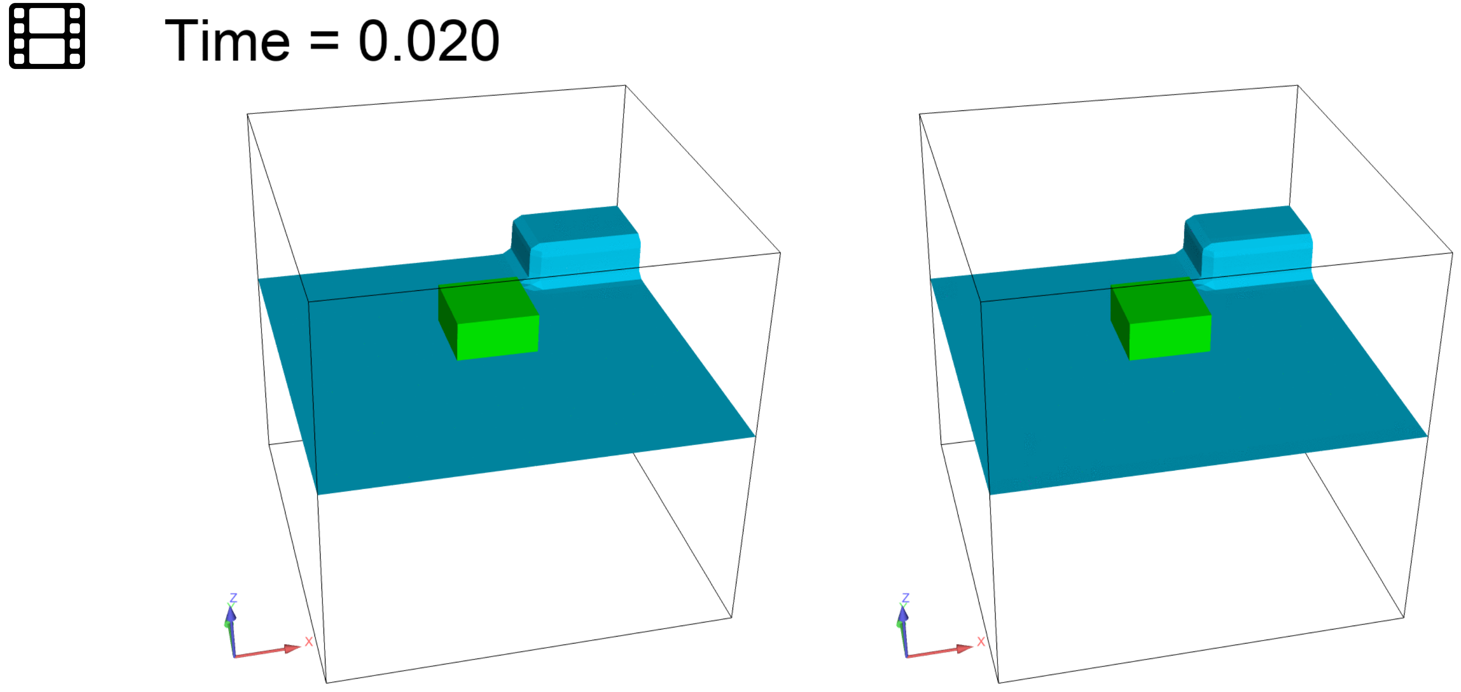
Fluent **OpenFOAM®**

http://www.wolfdynamics.com//wiki/of_conf2019/f12.gif

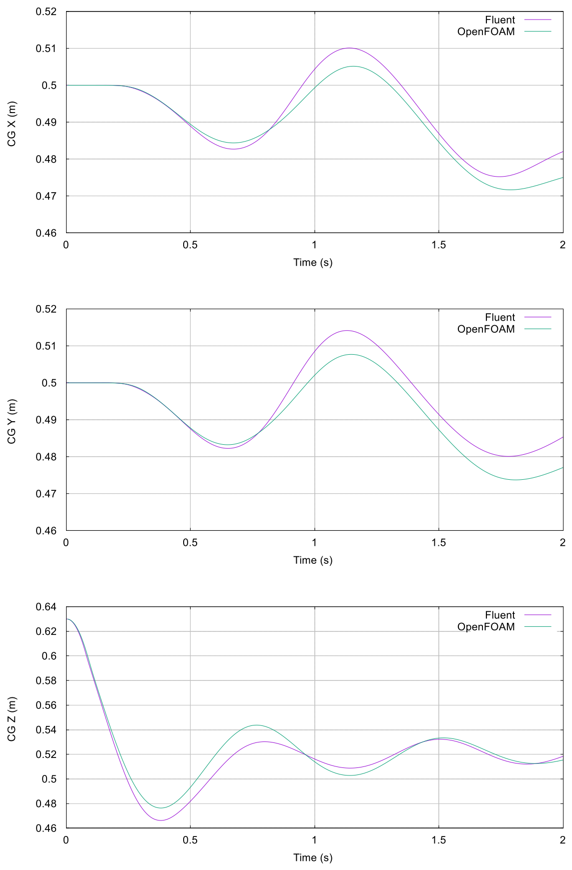
3. Benchmarking cases

- Falling and floating body – Rigid body motion – Fluent vs. OpenFOAM®.
  - Surprisingly, the computing time for both codes was very similar.

http://www.wolfdynamics.com//wiki/of_conf2019/f13.gif

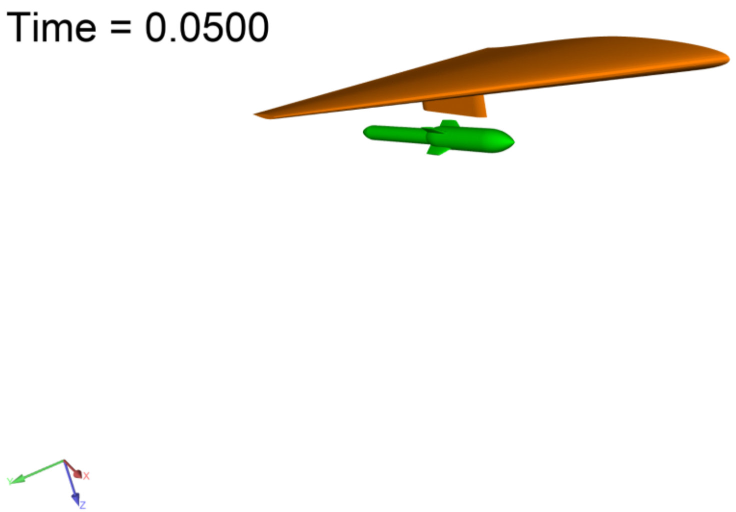


Fluent **OpenFOAM®**



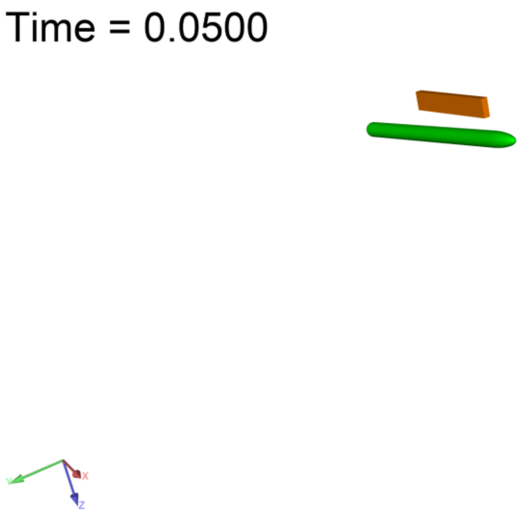
3. Benchmarking cases

- Wing/Pylon/Finned Store separation – Simplified geometry.
  - Hereafter, we only measured the performance of the overset library (stencils computation and hole cutting).
  - We did not compute the flow; we simply applied previously computed/experimental store trajectories to the simplified model.
  - The meshes were generated using Fluent tools (we had terrible problems with OpenFOAM® meshing tools).



Original geometry

http://www.wolfdynamics.com//wiki/of_conf2019/f14.gif

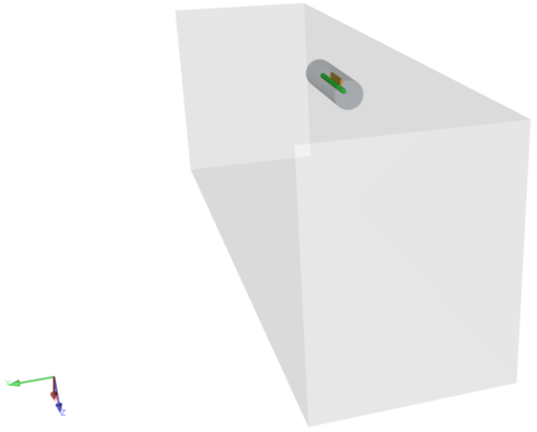


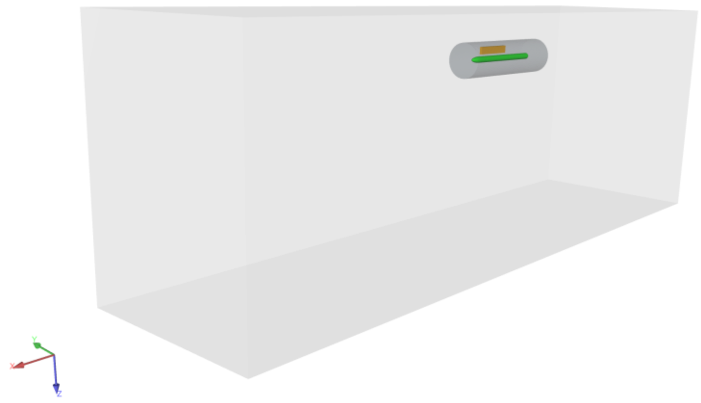
Simplified geometry

http://www.wolfdynamics.com//wiki/of_conf2019/f15.gif

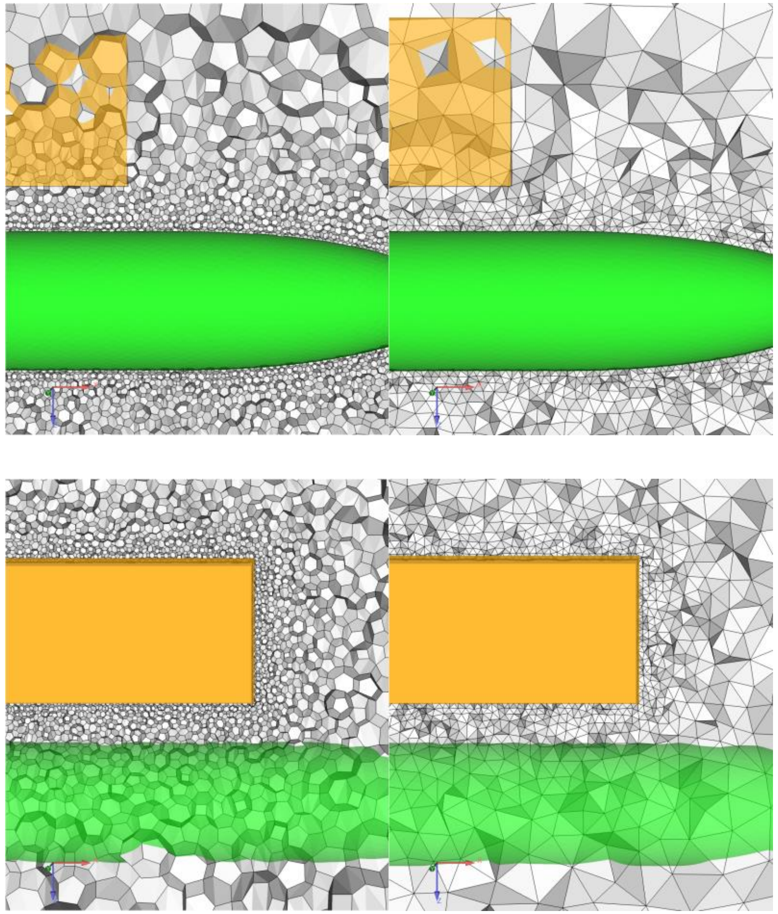
3. Benchmarking cases

- Wing/Pylon/Finned Store separation – Simplified geometry.





Component mesh 2 – Store mesh



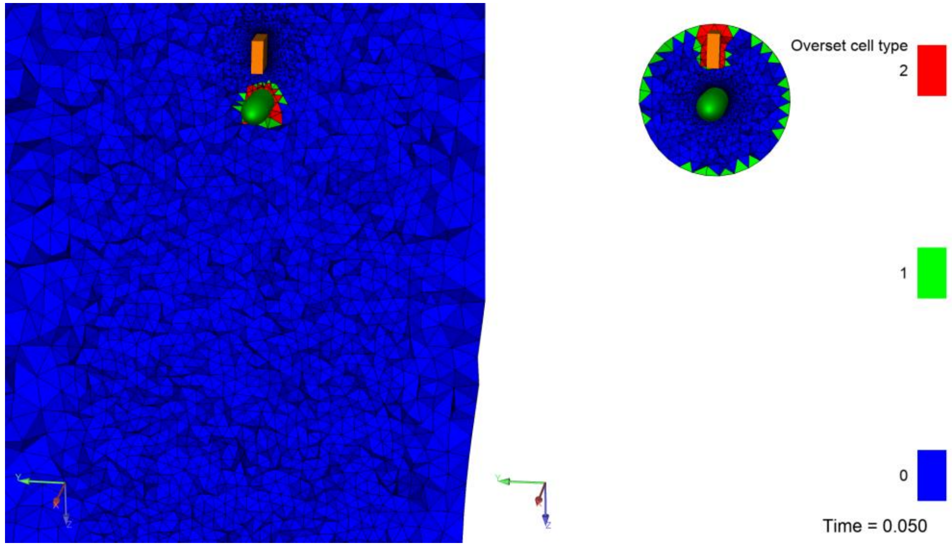
Component mesh 1 – Background mesh

- Assembling and running simulations with overset meshes requires careful planning at the geometry and mesh levels.
- In walls that are close together, there should be enough cells in order to compute a valid interpolation stencil.
- In our experience, at least 5 or more cells are required in OpenFOAM®. Polyhedral cells might require more elements.
- Cell size between component meshes should not be too dissimilar.
- If a component mesh is moving, try to have cells with similar sizes along the trajectory of meshes exchanging information (specially in the overset boundaries).

3. Benchmarking cases

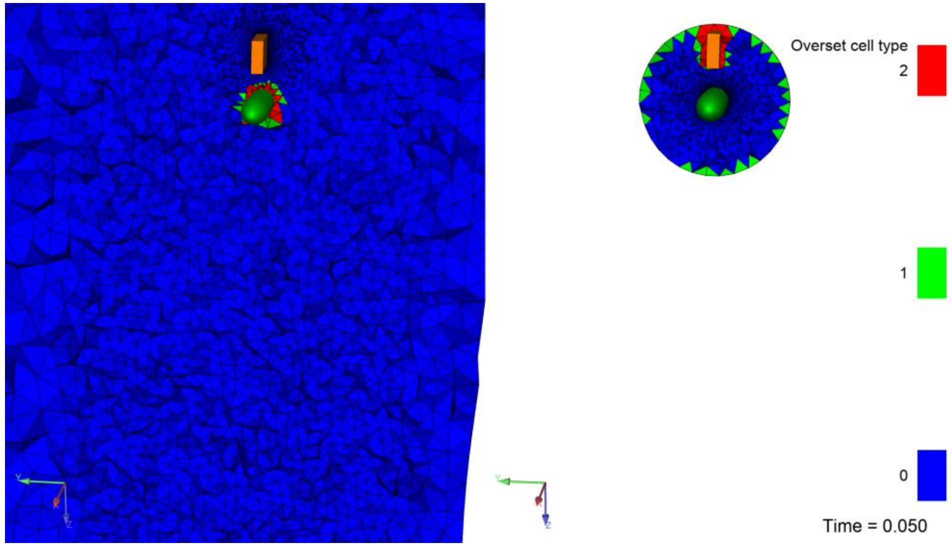
- Wing/Pylon/Finned Store separation – Simplified geometry – OpenFOAM® computations.
  - The time-step must be small enough to accommodate for a sequential change of the cell type from unused to interpolated and then calculated.
  - To avoid orphan cells and for good accuracy and stability of the solution, the mesh motion CFL number should be kept below 1.

http://www.wolfdynamics.com//wiki/of_conf2019/f16.gif



Some figures of merit
Time-step = 0.05; CPU time (4C) = 506

http://www.wolfdynamics.com//wiki/of_conf2019/f17.gif



Some figures of merit
Time-step = 0.005; CPU time (4C) = 4958

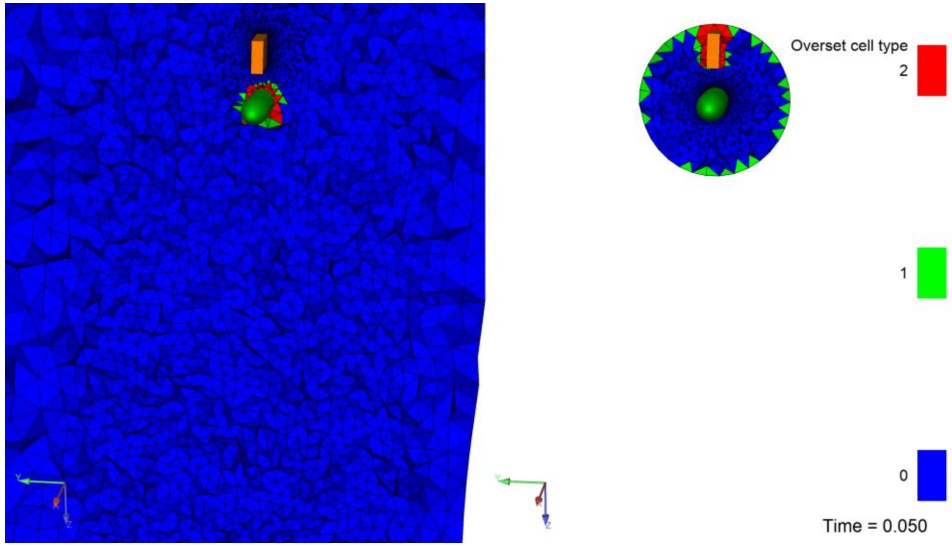
Mesh size ≈ 1.5 millions cells

3. Benchmarking cases

- Wing/Pylon/Finned Store separation – Simplified geometry.
  - OpenFOAM® – Fluent comparison (many optimization flags were disabled in Fluent).
  - The cell count of the stencils were very similar. OpenFOAM® does not provide orphan information.

OpenFOAM®

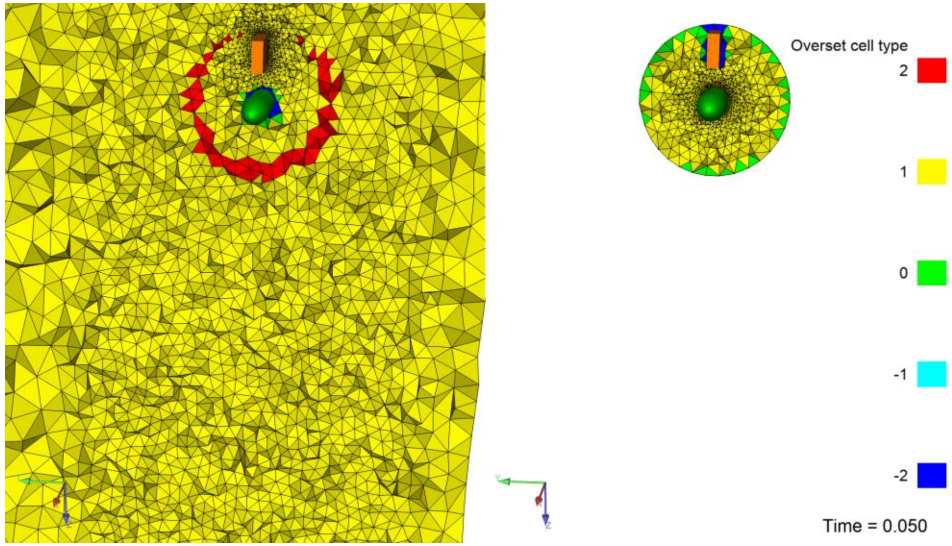
http://www.wolfdynamics.com//wiki/of_conf2019/f16.gif



Some figures of merit
Time-step = 0.05; CPU time (4C) = 506
Time-step = 0.005; CPU time (4C) = 4958

Fluent

http://www.wolfdynamics.com//wiki/of_conf2019/f18.gif



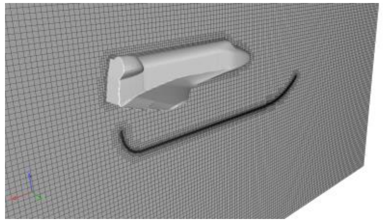
Some figures of merit
Time-step = 0.05; CPU time (4C) = 426
Time-step = 0.005; CPU time (4C) = 3250

Mesh size ≈ 1.5 millions cells

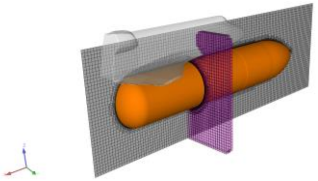
3. Benchmarking cases

- Space shuttle – Booster release

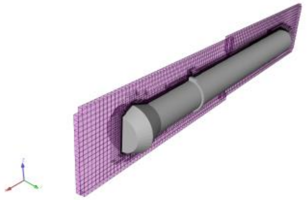
Component mesh 1



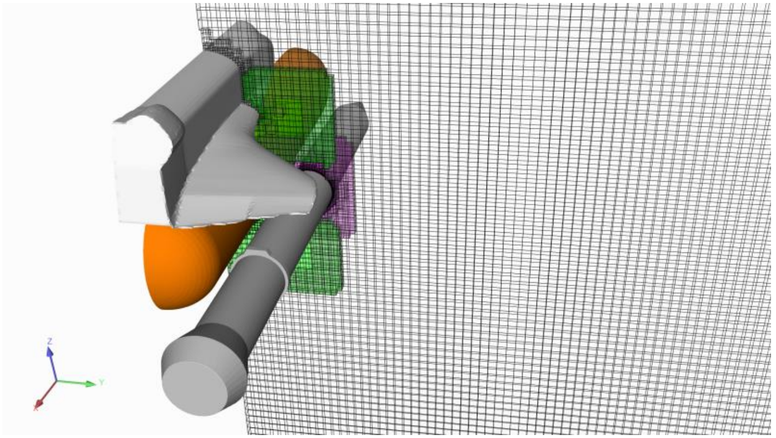
Component mesh 2



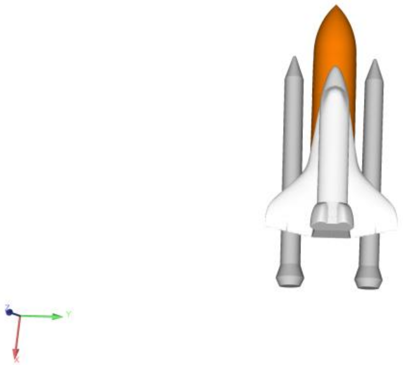
Component mesh 3



http://www.wolfdynamics.com//wiki/of_conf2019/f19.gif



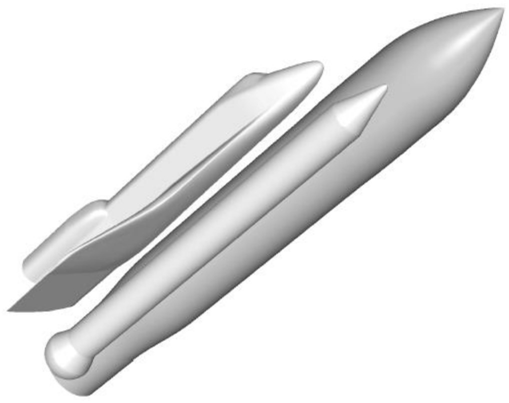
http://www.wolfdynamics.com//wiki/of_conf2019/f20.gif



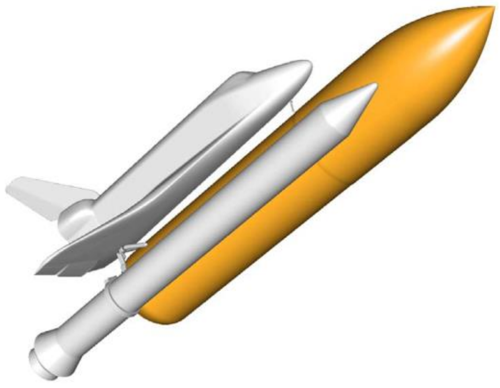
- The meshes were generated using OpenFOAM® meshing tools.
- The overset mesh assembly contains approximately 4.5 millions cells.
- We only simulated the kinematics.
- It was extremely difficult to get this case running. Way beyond practical use.
- To get 200 iterations it took approximately 16 hours in 4 cores, plus all the man-hours required to assembly the overset mesh and fix issues.
- This case evidences issues related to the use of collar grids, diagnosing the overset assembly and reporting of orphan cells.
- In comparison, it took approximately 2.5 hours to compute 200 iterations using Fluent.

3. Benchmarking cases

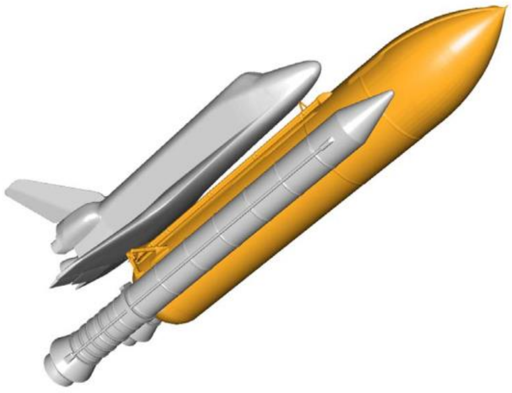
- Space Shuttle Launch Vehicle (SSLV) Grid System Evolution



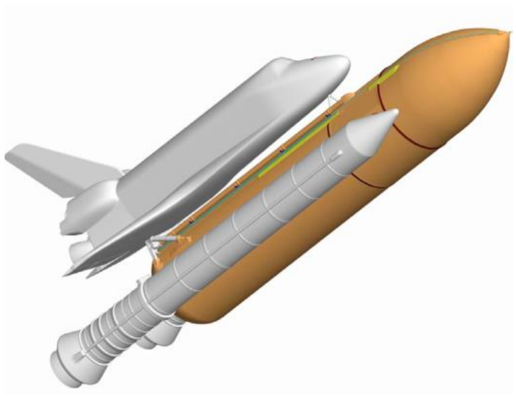
Early 80’s grid system
3 Component grids
10k surface points
0.3 million volume points



Late 80’s grid system
14 Component grids
35k surface points
1.6 million volume points



Early 90’s grid system
113 Component grids
268k surface points
16.4 million volume points



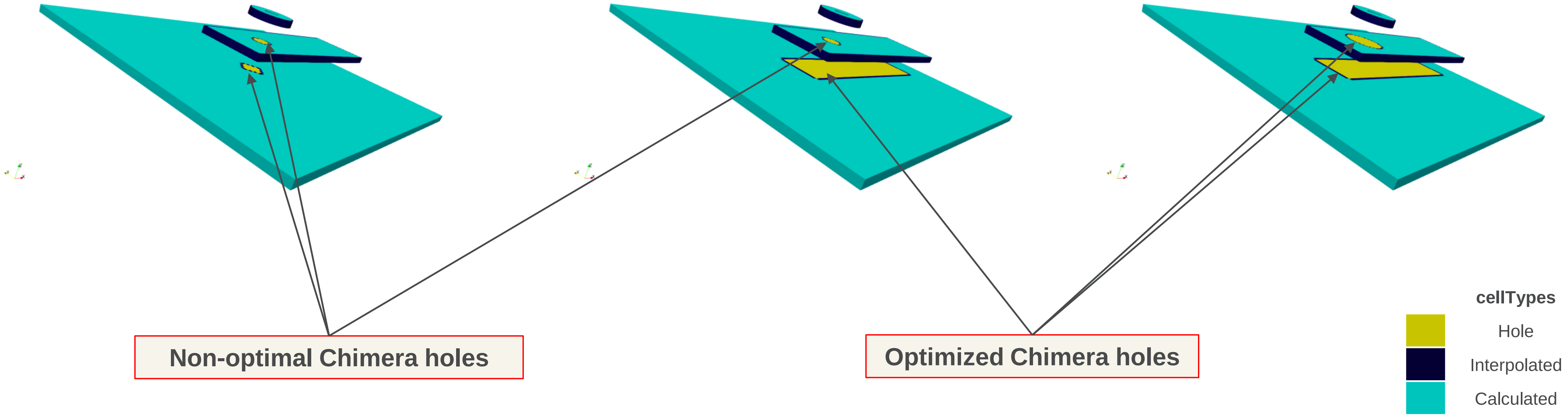
2004 grid system
267 Component grids
636k surface points
34.8 million volume points

20+ Years of Chimera Grid Development for the Space Shuttle

Figure credit: P. Buining, R. Gomez. 10th Symposium on Overset Composite Grid and Solution Technology. September 20-23, 2010, Moffett Field, CA

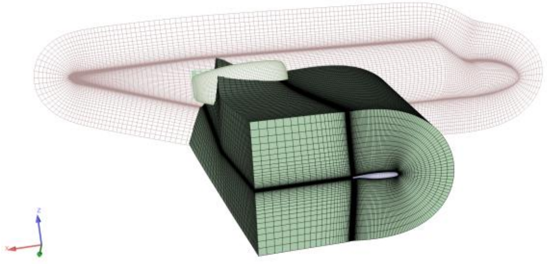
4. What is missing? What can be improved?

- Critical issues – Personal opinion.
  - Optimized Chimera hole – Cut hole minimization.
  - Cut hole algorithm – Currently it uses a voxel mesh, not necessary is the best method.
  - Better diagnosing of mesh assembly, reporting of orphan cells, and more control when assembling the sets.

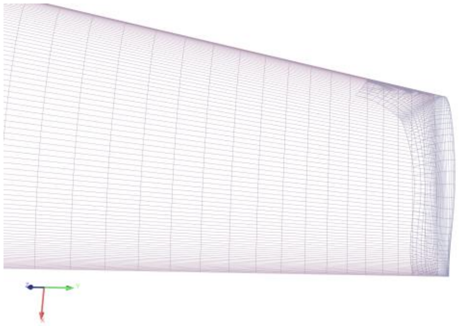


4. What is missing? What can be improved?

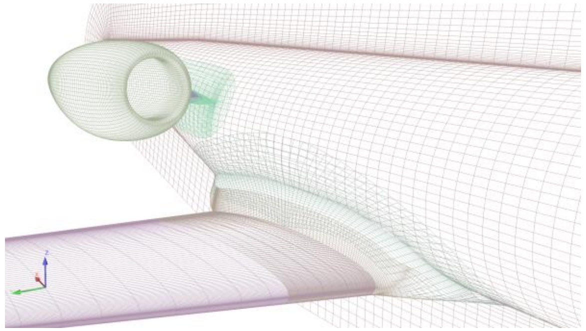
- Critical issues – Personal opinion.
  - Collar grids and caps support.
  - Algebraic, Laplacian or hyperbolic mesh marching.
  - Better control on grid priorities.



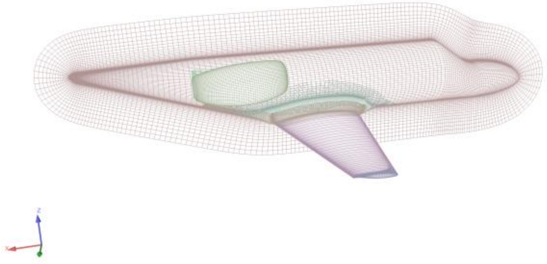
Hyperbolic mesh marching



Grid caps



Collar grids



Control on overset mesh assembly and grid priorities

4. What is missing? What can be improved?

- Warmly welcome improvements:
  - Collision detection.
  - AMR.
  - Dynamic load balancing.
  - Fixing sampling issues.
  - Easier postprocessing – Scripts for paraview (this is not only an OpenFOAM® problem).
  - DRYRUN option for overset solvers.
  - Efficient geometric multigrid, for example, using MGridGen (this is not only an OpenFOAM® problem).
  - Lagrange interpolation scheme.
  - LES/DES.

5. Some basic 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 5 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**). 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.
- Use iterative marching for the P-V coupling (PIMPLE in OpenFOAM®) and do at least five iterations (in our personal experience).

5. Some basic 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.
- 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).

6. Main takeaways

- By using overset meshes, simulations involving complex motion (prescribed, 6DOF, or FSI) of single or multiple bodies that were extremely difficult or impossible to simulate using traditional moving meshes methods (mesh smoothing, layering, or remeshing), are now easier to solve.

- If you are working with unstructured meshes and there are no moving bodies, it makes no sense paying the extra computational cost inherent to overset meshes.

- Have in mind that overset meshes can add numerical diffusion to the solution, not to mention that the interpolation is non-conservative. If you are conducting SRS simulations, be careful.

- Do not take overset meshes as a silver bullet. Simulations using overset meshes requires careful planning.

6. Main takeaways

- **Current state of the overset method in OpenFOAM® (1906):**
  - It is there, it works.
  - However, it can be improved.
  - Also, be sure to follow good standard practices to avoid beginner errors and small issues present in the overset library.

Overset CFD Technology Development and Application at the Boeing Company

- Boeing relies heavily on overset grid CFD methods to design and analyze virtually all of the air and space vehicles we build.
- Because this technology is so important to us, we have invested, and continue to invest, in the development of tool and process improvements and application validation.
- Concurrently, we remain active in the overset CFD community, and work to share and leverage as much externally developed technology as possible.
    - We truly appreciate the incredible amount of technology that is currently being developed within the overset CFD community, and particularly at NASA.
    - We have been, and will remain, open to collaborating with external organizations in a mutually beneficial manner to advance the state-of-the-art in overset CFD technology.

Jeffrey Slotnick
Boeing Research & Technology, Huntington Beach, CA, USA
11th Symposium on Overset Composite Grids and Solution Technology. Dayton, OH, USA. October 15–18, 2012

FYI – Useful links

- You can download the working cases at the following link (GitHub):
    - https://github.com/wolfDynamics/ESI_2019_OF_conference

- 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

wolf dynamics

multiphysics simulations,
optimization & data analytics

www.wolfdynamics.com

Be collaborative, be innovative, be cloud

guerrero@wolfdynamics.com

www.wolfdynamics.com