



Parallel curved mesh Subdivision for flow Simulation on curved Topographies

A. Gargallo-Peiró^{a*}, H. Owen^a, G. Houzeaux^a, X. Roca^a

*^aBarcelona Supercomputing Center
Carrer de Jordi Girona, 29-31, 08034 Barcelona, Spain*

Abstract

We present the implementation in the Alya code of a method to refine a mesh in parallel while preserving the curvature of a target topography. Our approach starts by generating a coarse linear mesh of the computational domain. Then, the former coarse mesh is curved to match the curvature of the target geometry. Finally, the curved mesh is given to the improved Alya code that now reads the curved mesh, partitions it, and sends the subdomain meshes to the slaves. The result is a finer linear mesh obtained in parallel with improved geometric accuracy. The main application of the obtained finer linear mesh is to compute a steady state flow solution on complex topographies.

Introduction

Standard commercial linear mesh generators (e.g. ANSYS, GridGen, or GiD,) are designed to approximate geometries for simulation. To do this, they feature boundary layer meshing, and local adaptivity. However, they are not ready to generate meshes suitable for petascale runs, mainly due to memory constraints. Furthermore, they only provide in general piecewise linear (straight-sided tetrahedra) or trilinear (ruled-sided hexahedra) approximations of the computational domain and therefore, subdivided meshes do not target to approximate curved boundaries. In this project, we have implemented a parallel uniform curved mesh subdivision method, by extending a parallel linear subdivision method [1, 2] developed at Barcelona Supercomputing Center on top of the Alya code.

The scientific case has been prompted by the interest of combining the research results of two of our ongoing projects, awarded in the Horizon 2020 framework of the European commission funded by the Energy oriented Centre of Excellence for computer applications (EoCoE, Project ID: 676629) and the Marie Skłodowska-Curie Actions (HiPerMeGaFlowS, Project ID: 658853) programs. The goal of this combination is to obtain mesh convergence results for the turbulent flow on real curved topographies, with applications to wind farm design.

Our approach starts by generating a coarse linear mesh. Then the coarse mesh is curved to match the boundaries of the computational domain [3]. Finally, the curved mesh is given to the improved Alya code [4,5] that now reads the curved mesh, partitions it, and sends the subdomain meshes to the slaves. The result is a finer linear mesh obtained in parallel with improved geometric accuracy. Then, a steady state flow solution is obtained on the finer linear mesh. This methodology adopts the coarse-to-fine paradigm we have proposed before to accelerate the mesh curving of moving domains [6] and to insert boundary layers in curved meshes [7].

* Correspondent author email address: abel.gargallo@bsc.es

The advantage of the proposed and implemented strategy is that the subdivision is performed and stored in parallel and therefore, there are no memory constraints. Furthermore, the finer linear mesh targets the curvature information described by the first curved mesh. Finally, if the original numbering is conserved, then the post-processing can be performed on the coarse curved mesh.

Methodology

In this section, we present the main steps and implementation details of our new method to refine a mesh in parallel while preserving the curvature of a target topography. Our approach consists of the following steps:

1. **Generating a coarse linear mesh.** We assume that a coarse linear mesh has been obtained and that the mesh features the required resolution for the current problem (CFD) such as boundary layer elements near the wall. Specifically, we generate the coarse linear mesh of the topography using WindMesh [8], an in-house mesh generator for Atmospheric Boundary Layer flows in complex topographies.
2. **Curving the coarse mesh.** The elements of the former coarse mesh are curved to match the boundaries of the computational domain. In this manner, the inherent curvature of the domain can be reproduced with more accuracy. To this end, we use a curved meshing method developed by the researchers at the BSC and implemented in WindMesh. The method uses piece-wise polynomial representations of the elements that can be curved to match the domain boundaries by minimizing the mesh distortion [3].
3. **Preserving curvature on parallel subdivision.** The coarse curved mesh is given to Alya, which is based on a master-slave strategy. The master reads the curved mesh, partitions it, and sends the subdomain meshes to the slaves. The slaves then refine their corresponding subdomains in parallel, by performing the subdivision of the curved elements in finer linear elements. During the process we have to ensure that the boundary nodes of the new elements match with the curved elements of the neighbouring subdomains. The result is a finer linear mesh obtained in parallel with improved geometric accuracy. For standard engineering applications, the computational mesh has between 10M and 100M of elements whereas the curved mesh that has to be partitioned and distributed has only between 10K and 100K of elements.

To implement the proposed approach, we have extended the existing parallel subdivision code in Alya in the following manner:

1. Implementing quadratic shape functions in Alya to allow isoparametric mappings of polynomial degree two to be represented. These elements feature an additional node in the middle of: the edges, the quadrilateral faces, and the hexahedral elements. The coordinates of these new nodes allow the curvature of the target geometry to be captured
2. Representing curved meshes in Alya as a standard linear mesh equipped with a vectorial elemental field. On the one hand, the linear mesh determines the topology of the curved mesh. On the other hand, the components of the vectorial element field are the coordinates of the nodes of the quadratic elements and therefore, determine the geometry of the curved mesh. This approach allows the ability of Alya to refine in parallel linear meshes to be exploited. Specifically, it allows the functionality that deals with the topology of the mesh to be re-used
3. Incorporating in Alya the possibility to read and write curved meshes represented with quadratical elements.
4. In the previous version of Alya, the coordinates of the new nodes in the middle of edges, faces and cells were not preserving the curvature of a target geometry. Accordingly, we have modified the code in such a manner that the coordinates of the new nodes interpolate the target geometry. To this end, the target geometry is represented by means of a coarse curved mesh composed by isoparametric elements of degree two.

Results

The PRACE preparatory access mechanism has facilitated the access to the MareNostrum supercomputer, required to obtain the results presented in this section. For the success of this proposal it was necessary to utilize all the requested computer time and disk storage. The performance of the initial Alya code was almost linear for a sufficient number of elements per subdomain, say 20,000. In the new version, we obtain the same performance results. However, in the new version, we are able to solve larger problems where the inherent curvature of the

topography is now preserved. This was not possible in the previous version of the code where after successive refinement, the geometric accuracy of the topography representation was not improved. The following examples illustrate this result.

With the previous version of the code, the target geometry is represented by a coarse linear mesh composed of 24K standard hexahedra, see **Figure 1 (a)**, and we refine it to obtain a finer mesh, see **Figure 1 (c)**. Accordingly, the refined mesh reproduces the straight-side geometry represented by the initial mesh. In the new approach, the target geometry is represented by a coarse curved mesh composed by 24K curved hexahedra, see **Figure 1 (b)**. After four consecutive subdivisions, with both approaches we obtain a finer distributed linear mesh composed by 98 304 000 elements ($24K \times 8 \times 8 \times 8 \times 8$ elements), few seconds later and completely stored in the memory. Now, the resulting refined linear mesh, see **Figure 1 (d)**, targets the curvature of the topography while in the previous version of the code this was not possible, see **Figures 1 (c)**.

With the previous version of the code, we are able to converge to a steady state of the flow however, the finer representation of the initial straight-sided geometry leads to artificial features in the solution, see **Figures 1 (a)** and **1 (c)**. These features are visible in those regions where the mesh features sharp edges characterized by large changes in the normal vector between contiguous faces. Using the finer mesh obtained with the new approach, we are able to converge to a steady flow state with an improved accuracy of the topography geometry and without artificial flow features, see **Figures 1 (b)** and **1 (d)**. This is because the mesh features a smoother variation of the normal vector between adjacent faces.

In **Figure 2**, we present a general and a detailed view of the initial straight-sided coarse mesh used in the previous version of the code. In addition, we include a section of the steady state flow solution obtained with the standard subdivision approach, see **Figures 2 (a)** and **2 (c)**, and with our subdivision approach, see **Figures 2 (b)** and **(d)**. When the standard subdivision approach is used, we observe that the artificial flow features are found next to the sharp edges of the initial straight-sided mesh. Instead, in **Figures 2 (b)** and **(d)** we observe that the flow solution is smoother and without artificial features when our subdivision approach is used. The blank space between the initial straight-sided mesh and the flow solution illustrates the error in the approximation of the target geometry when a standard subdivision approach is used.

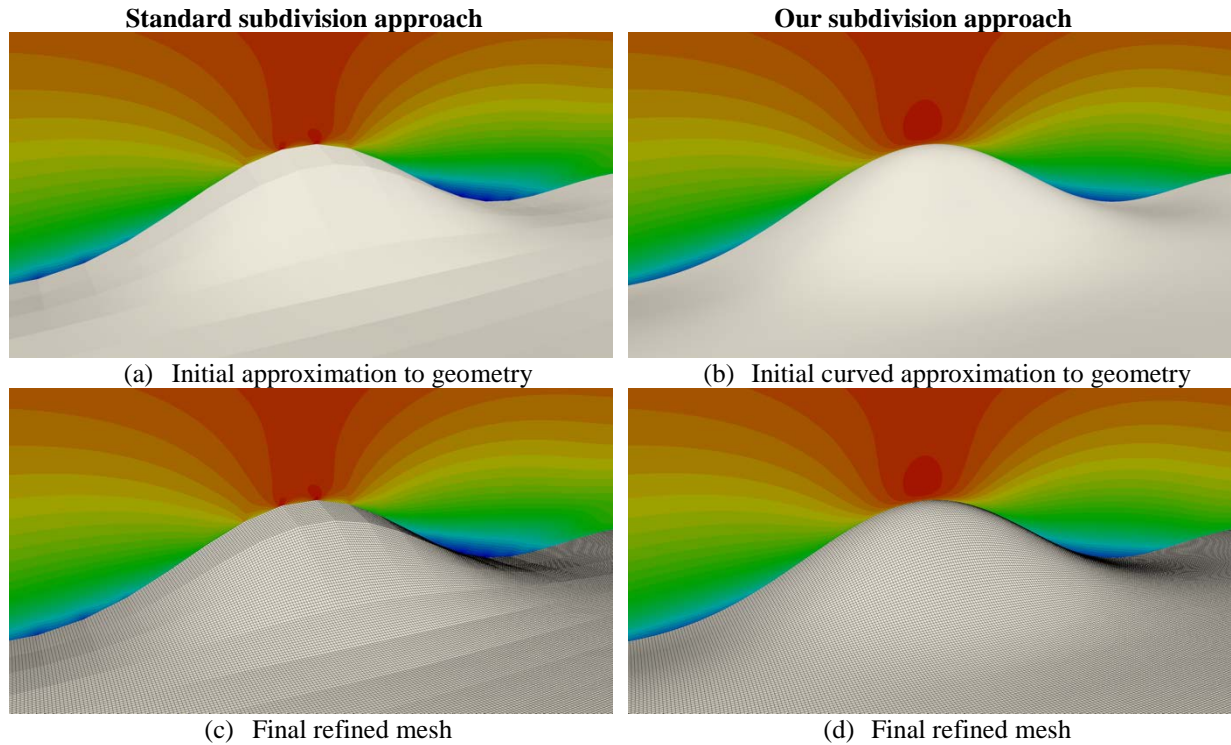


Figure 1. (a, b) Geometry and (c, d) meshes with the velocity magnitude of the steady state flow solution for: (left) standard subdivision approach and, (right) our subdivision approach.

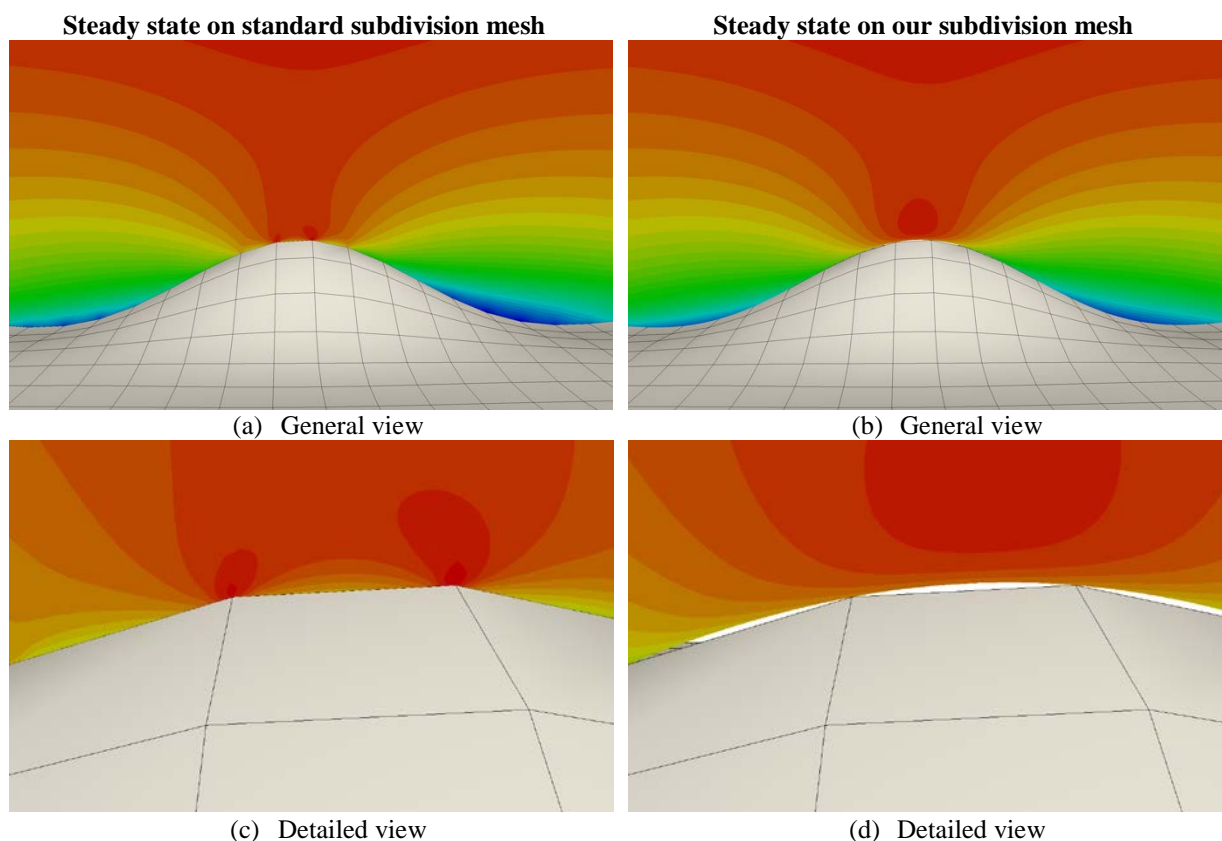


Figure 2. (a, b) General view and (c, d) detailed view of the standard subdivision mesh and a section of the velocity magnitude for the flow steady state on a mesh obtained with: (left) the standard subdivision approach and (right) our subdivision approach.

Concluding remarks

We have proposed and implemented a subdivision strategy performed and stored in parallel and therefore, there are no memory constraints. Furthermore, the finer linear mesh targets the curvature information described by the initial curved mesh. The mesh can be used to obtain the steady state of a flow solution of interest in simulating the generated power of a wind farm. In the near future, this approach could be used to perform the visualization of the results in the initial coarse mesh just by preserving the original mesh numbering.

References

- [1] Houzeaux G., de la Cruz R., Vázquez M., *Parallel uniform mesh subdivision in Alya*, PRACE whitepaper. 2011.
- [2] Houzeaux G., de la Cruz R., Owen H., Vázquez M., *Parallel uniform mesh multiplication applied to a Navier–Stokes solver*, *Computers & Fluids*, 80:142-51, 2013
- [3] Gargallo-Peiró A., Roca X., Peraire J., Sarrate J., *Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes*, *International Journal for Numerical Method sin Engineering*, 103 (5): 342-363, 2015
- [4] Houzeaux G., Vázquez M., Aubry R., Cela J., *A massively parallel fractional step solver for incompressible flows*. *J. Comput. Phys.*, 228(17):6316 – 6332, 2009

- [5] Houzeaux G., Eguzkitza B., Vázquez M., *A variational multiscale model for the advection-diffusion-reaction equation*. Commun. Numer. Meth. En. 25(7):787–809, 2009
- [6] Chaurasia H., Roca X., Persson P.O., Peraire J., *A coarse-to-fine approach for efficient deformation of curved high-order meshes*, 21st International Meshing Roundtable, 2012
- [7] Gargallo-Peiró A., Roca X., Peraire J., Sarrate J., *Inserting curved boundary layers for viscous flow simulation with high-order tetrahedra*, 22nd International Meshing Roundtable, 2013
- [8] Gargallo-Peiró, A., Avila M., Owen H., Prieto L., and Folch A., *Mesh Generation for Atmospheric Boundary Layer Simulation in Wind Farm Design and Management*. Procedia Engineering 124, 239-251, 2015

Acknowledgements

This work was financially supported by the PRACE project funded in part by the EU's Horizon 2020 research and innovation programme (2014-2020) under grant agreement 653838. The work of the second author has been partially supported by the Energy oriented Centre of Excellence for computer applications (EoCoE) program of the EU's Horizon 2020 framework under the contract H2020-EINFRA-2015-1. The work of the corresponding author has been partially supported by the Marie Skłodowska-Curie Actions program of the EU's Horizon 2020 framework under the Individual Fellowship contract HiPerMeGaFlows.