

## FIRST STEPS TOWARD A PARALLEL, FLEXIBLE, AND CONFIGURABLE MESH DATABASE FOR UNSTRUCTURED MESH APPLICATIONS

Juan P. Saldía<sup>a,b</sup> and Sergio A. Elaskar<sup>a,b</sup>

<sup>a</sup>*Departamento de Aeronáutica, Facultad de Ciencias Exactas, Físicas y Naturales (FCEFN),  
Universidad Nacional de Córdoba (UNC), Argentina*

<sup>b</sup>*Instituto de Estudios Avanzados en Ingeniería y Tecnología (IDIT) FCEFN-UNC/CONICET,  
<http://www.inv.idit.efn.uncor.edu>*

**Keywords:** unstructured mesh library, mesh database, parallel.

**Abstract.** Unstructured mesh-based methods, have certain features which make them very attractive when geometric flexibility and/or solution-adaptive techniques are considered, as is common in numerical simulations in different domains of engineering. Unstructured mesh oriented systems require the maintaining of a data structure, generally known as mesh database, which main purpose is the management of general information regarding the topology, geometry and the (discrete) fields which are defined on the elements of the mesh. In general, this requirement impacts in the form of an exponential increment of software complexity of these systems in comparison to their structured mesh oriented counterpart, where the information regarding the mesh and fields is almost trivially obtained. The complexity of the mesh database is even more pronounced when parallel functionality is required and/or flexible and configurable capabilities are considered. Most scientific-computing oriented solutions in the literature relies on application-tailored mesh databases (libMesh, deal.II), which are not directly adaptable to general applications, or in more generic mesh databases (FMDB, ParFUM, DUNE), which nevertheless are not flexible enough in how the information stored in the mesh database is configured. In this work, we present our first steps towards the development of an open-source library with the aim of providing a mesh database with the following design goals in mind: generality, efficiency, freedom of configuration, ease of use, and complete functionality in parallel systems. In particular, we describe details of data structures and generic algorithms, which in combination with C++ template meta-programming techniques aims to provide efficiency, generality and freedom of configuration by the user. Some details of how the library achieves functionality on parallel systems through the MPI Standard, are also discussed. An object oriented interface which hides the details of the library implementation to the user and provides ease of access from current C++ codes and modern Fortran codes is briefly described, and a generic demonstration of the library usage is shown. Finally, we present an example of the incorporation of the library in an already functional in-house developed parallel unstructured mesh fluid dynamic solver for hypersonic nonequilibrium flows.