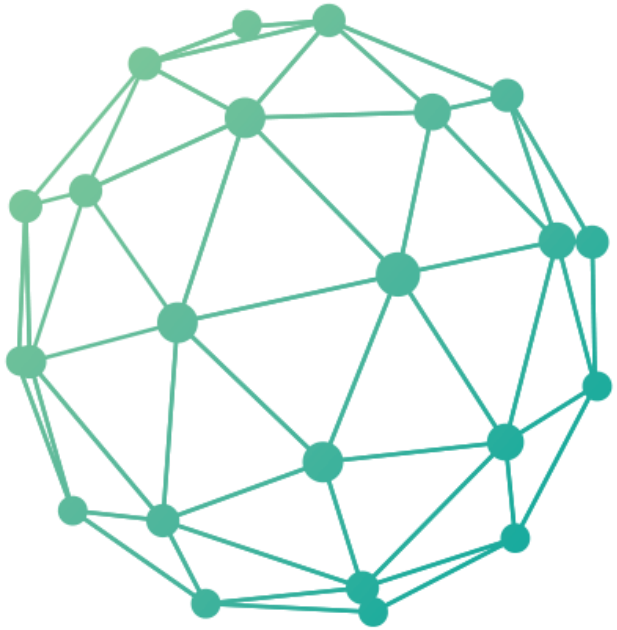
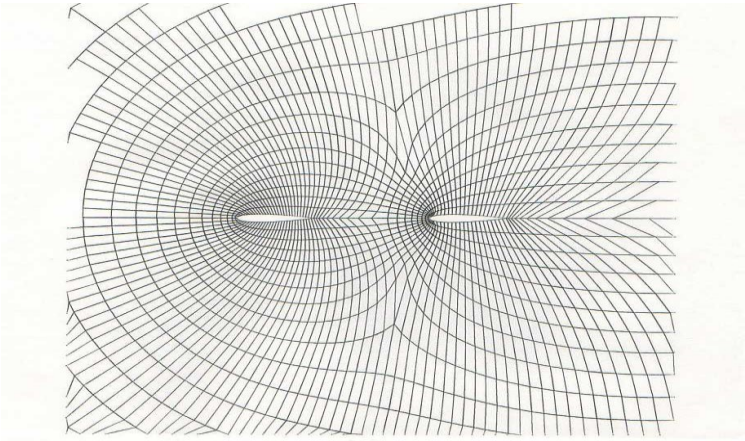
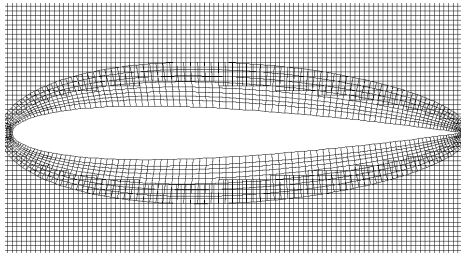


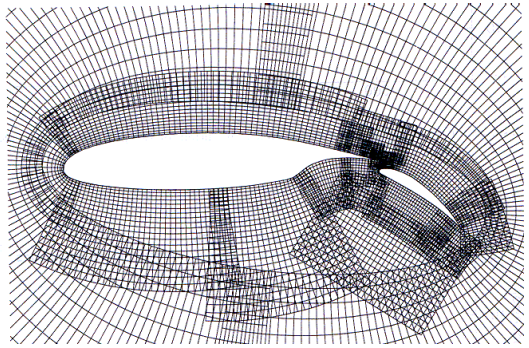
ELEMENT

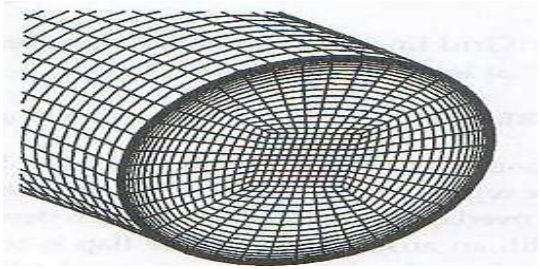
Parallel Mesh Generation

Mesh Types



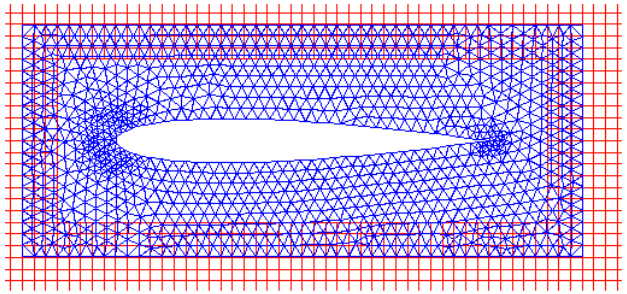






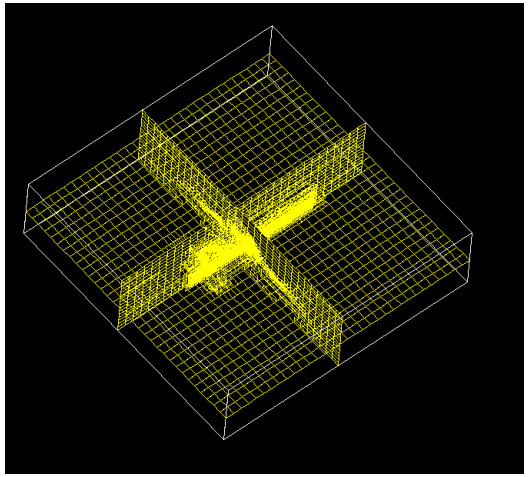
Multi-Block

Automation is difficult and not always topologically possible



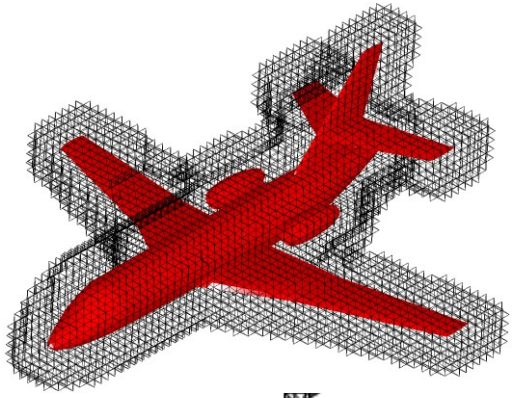
Overlapping Meshes

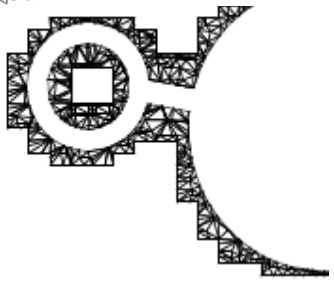
Interpolation could be problematic

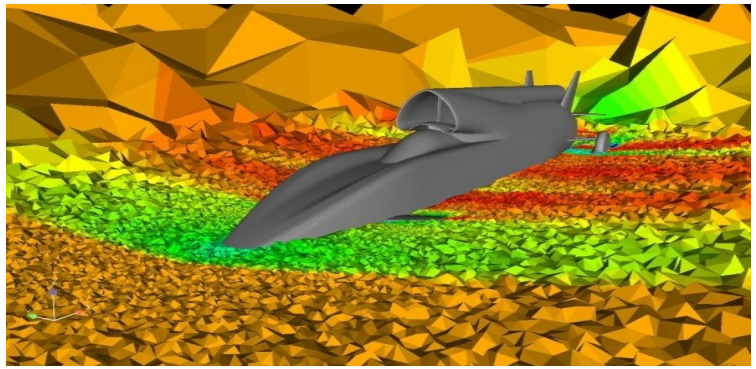


Octree/Unstructured meshes

Quality issues







Unstructured meshes

Memory and time

ELEMENT

Currently, meshes in the order of 100 million elements are regularly used in industries. Generation of such meshes require 4-6 hours of CPU time on a moderate PC and around 20-30Gb of memory.

Parallel unstructured mesh generators have been achieved by decomposing the three-dimensional problem into smaller sub-problems. The approaches can be classified as tightly coupled, decoupled or partially coupled.

Tightly coupled: implementation with AF and DT showed poor scalability beyond small number of processors.

De-coupled: implementation with AF and DT by insertion of a physical plane to partition the domain. Good scalability, no memory restriction. Partitioning may produce geometric restrictions that reduces quality. Load balancing is an issue.

Partially coupled: Implementation with AF and DT was based on modifying the point creation algorithm to refine around an imaginary plane. Good scalability. Enough memory to store the initial triangulation is required. Load balancing is an issue for the AF technique.

Above implementations have shown to generate meshes in the order 1 – 4 billions nodes utilizing up to ~4000 processors in 1-3 hours.

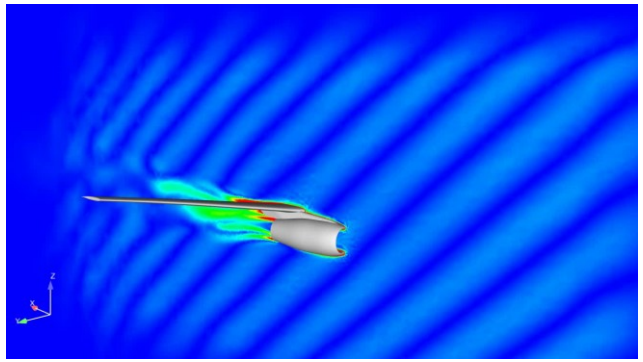
ELEMENT

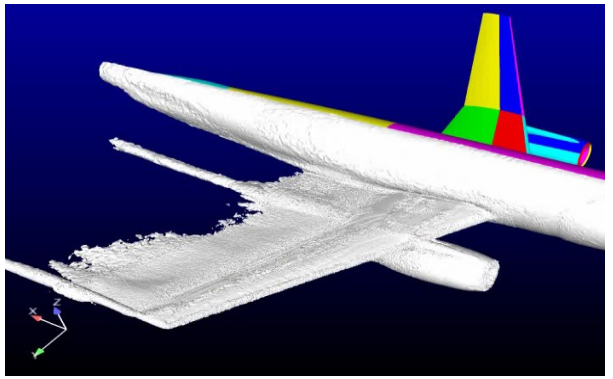
High Order Mesh Generation

Unsteady flow is of a great interest.

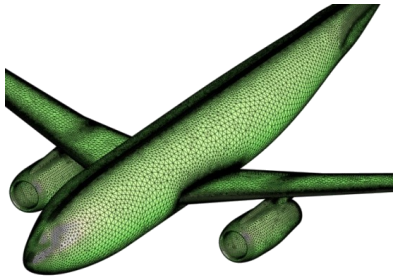
Low order methods suffer from dispersion and dissipation, hence, fine meshes are required.

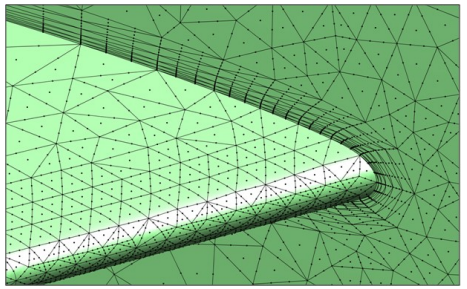
High order techniques offer a solution to reduce the required number of points per wavelength

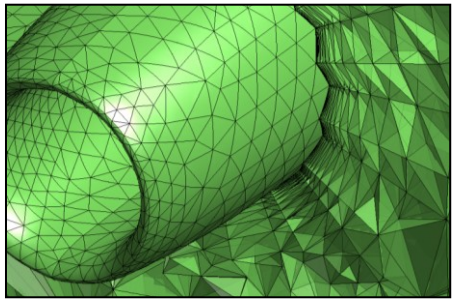




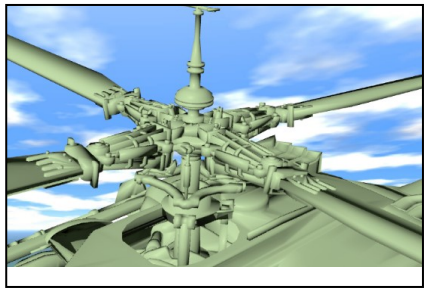
Methods that are based on modifying an existing linear mesh have shown to be robust and efficient.

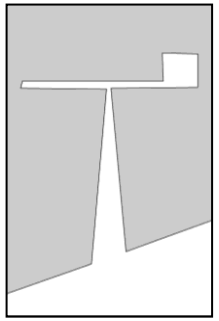


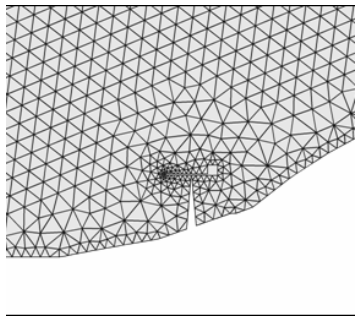


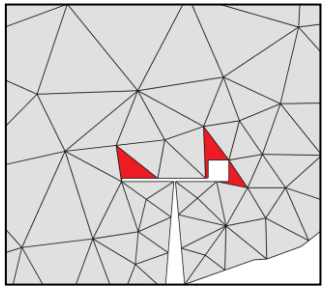


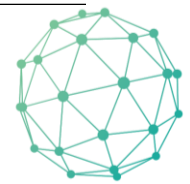
The need of de-featuring small geometric entities in a watertight geometry can be avoided if the generated elements carry the NURBS definition of the surface.











ELEMENT

- Software that is able to effectively leverage the spectrum of the current HPC hardware.
  - Mesh generation are often considered part of pre-processing that need to be run locally prior to using its output on main frame computers.
  - It is essential that these codes make use of the available modern computer architecture, hence, fine grain tuning of these codes that maintain portability is essential.
  - Addressing issues that would arise from generating meshes that have more that 6 order of magnitude's variation in the spacing distribution functions.

- Provides a much higher degree of automation.
  - Steady state simulations that need meshes in the order of 10 billion cells and require hours on 100000+ processors must be as close to optimum as possible. Previous knowledge should be encapsulated and utilised for the generation of such meshes. How can previous modelling results be used to generate an appropriate initial mesh?

- Produces robust and high quality meshes of all types.
  - Higher order elements of optimal quality need to be developed.
  - The development of mesh generation capabilities that combine established meshing techniques to create meshes that many not have optimum number of degree of freedom but can be executed in optimum time on HPC platforms

- Caters for the mesh requirements of the targeted physics.
  - A feature-independent mesh generation paradigm with a new element type that is devised to encapsulate all the geometric features when generating the mesh. This leads to:
    - Remove inaccuracy that results from geometric approximation.
    - Enable efficient P adaptivity without the need to carry the geometry description