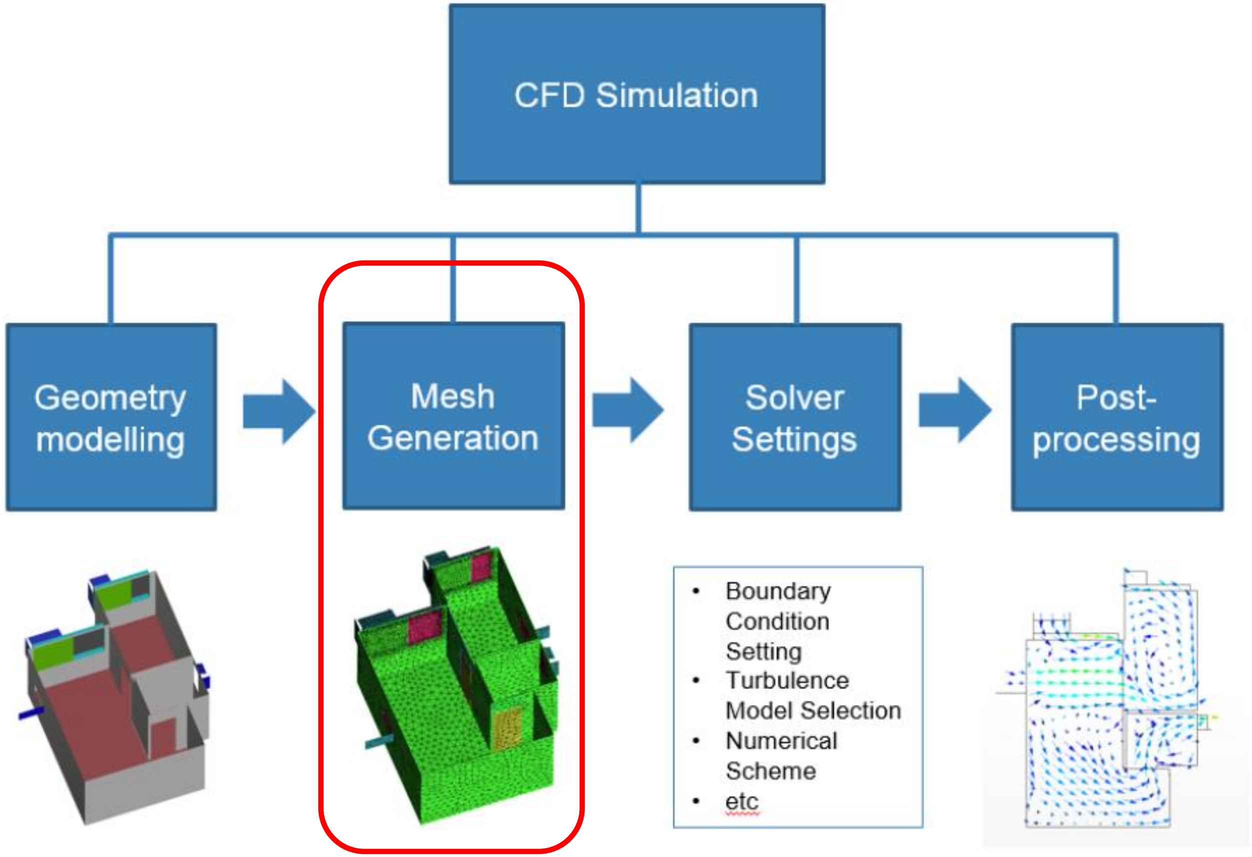
Regular Research Meeting

22/03/2021

Research Topic

- CFD application on natural ventilation design in a built environment



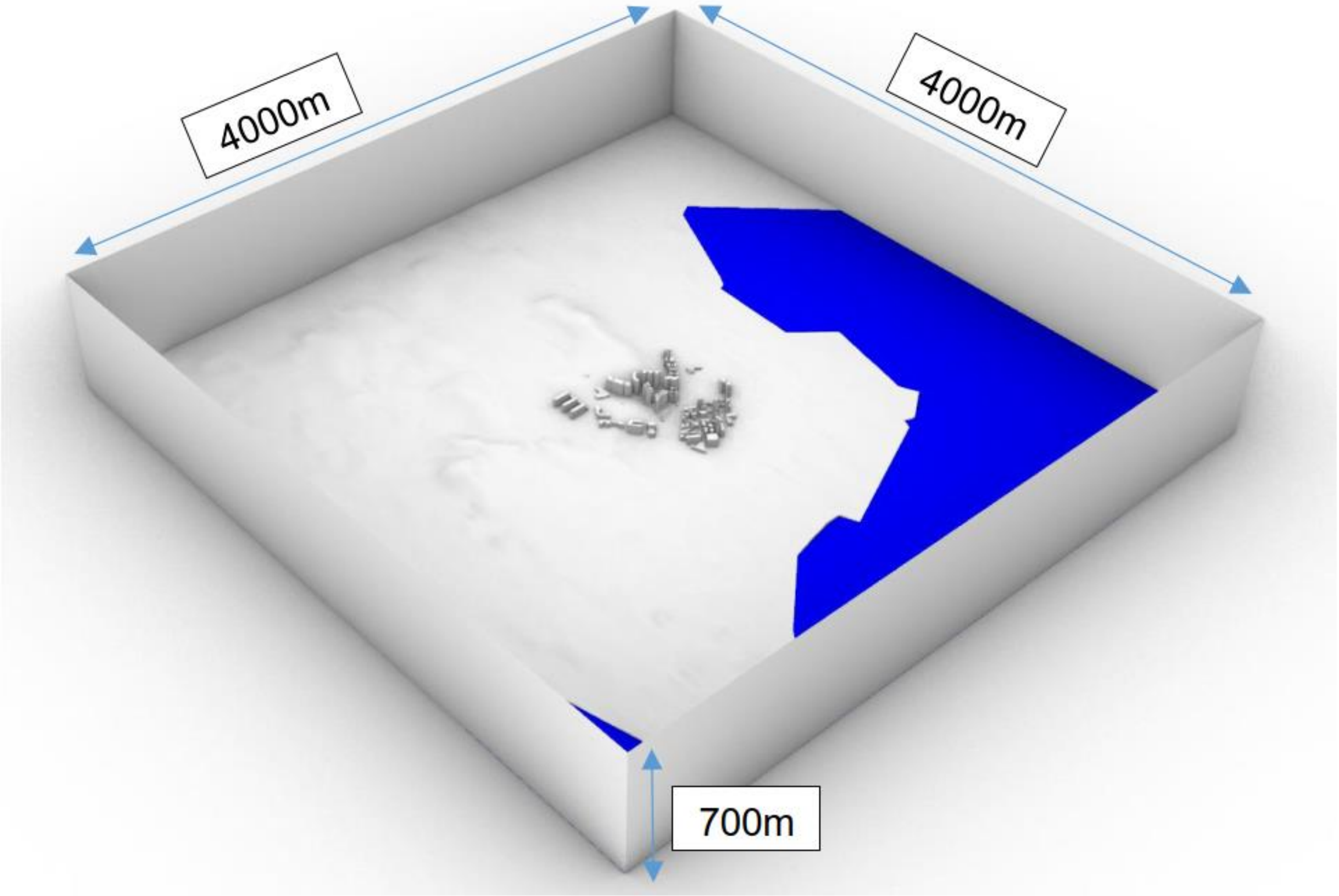
Demands from practitioners in the building industry:

Building designers want reasonable CFD result and short simulation time to optimize design schemes.

Meshing is the most important part of the simulation process.

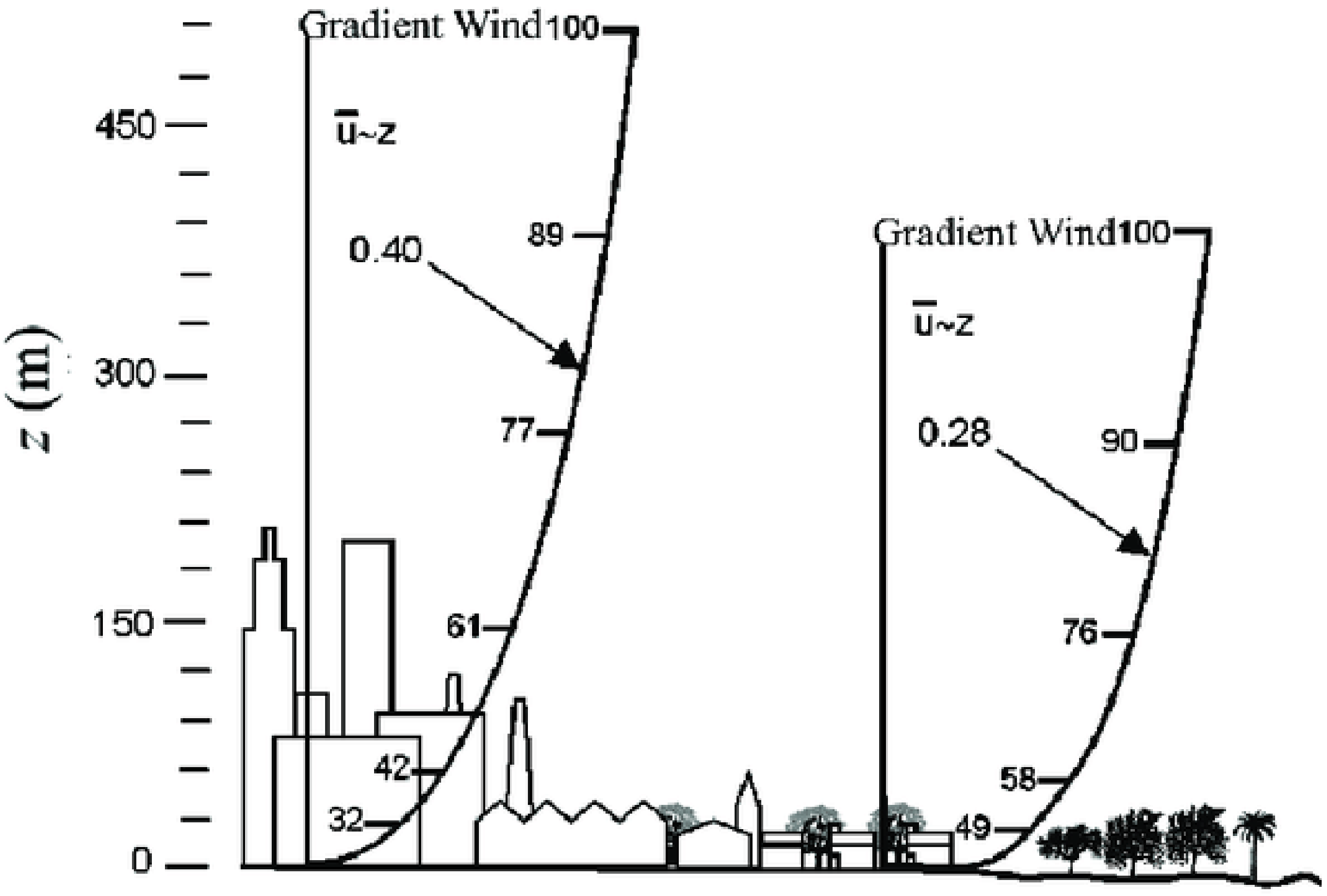
- Quality of mesh impacts the simulation accuracy.
- Number of mesh impacts the simulation time.

Characteristics of mesh for natural ventilation application



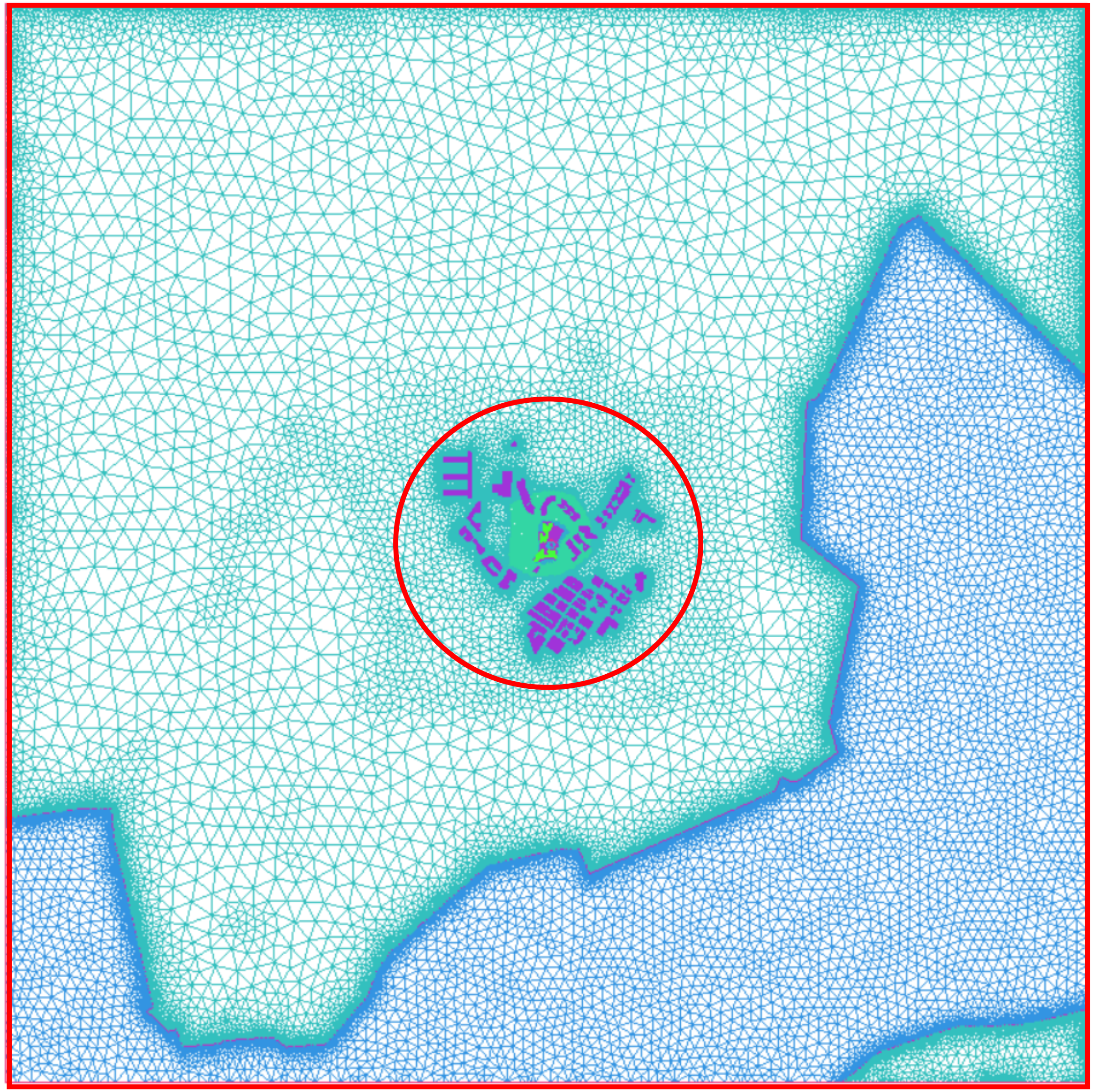
- Computational domain is very large in length, width and height;
- Wind profile of the atmosphere boundary condition is required;
- Coarse mesh is allowable for outer zone while fine mesh is required near building surfaces and ground surfaces;
- Boundary layer is required when studying the wind performance at the pedestrian level.

Characteristics of mesh for natural ventilation application



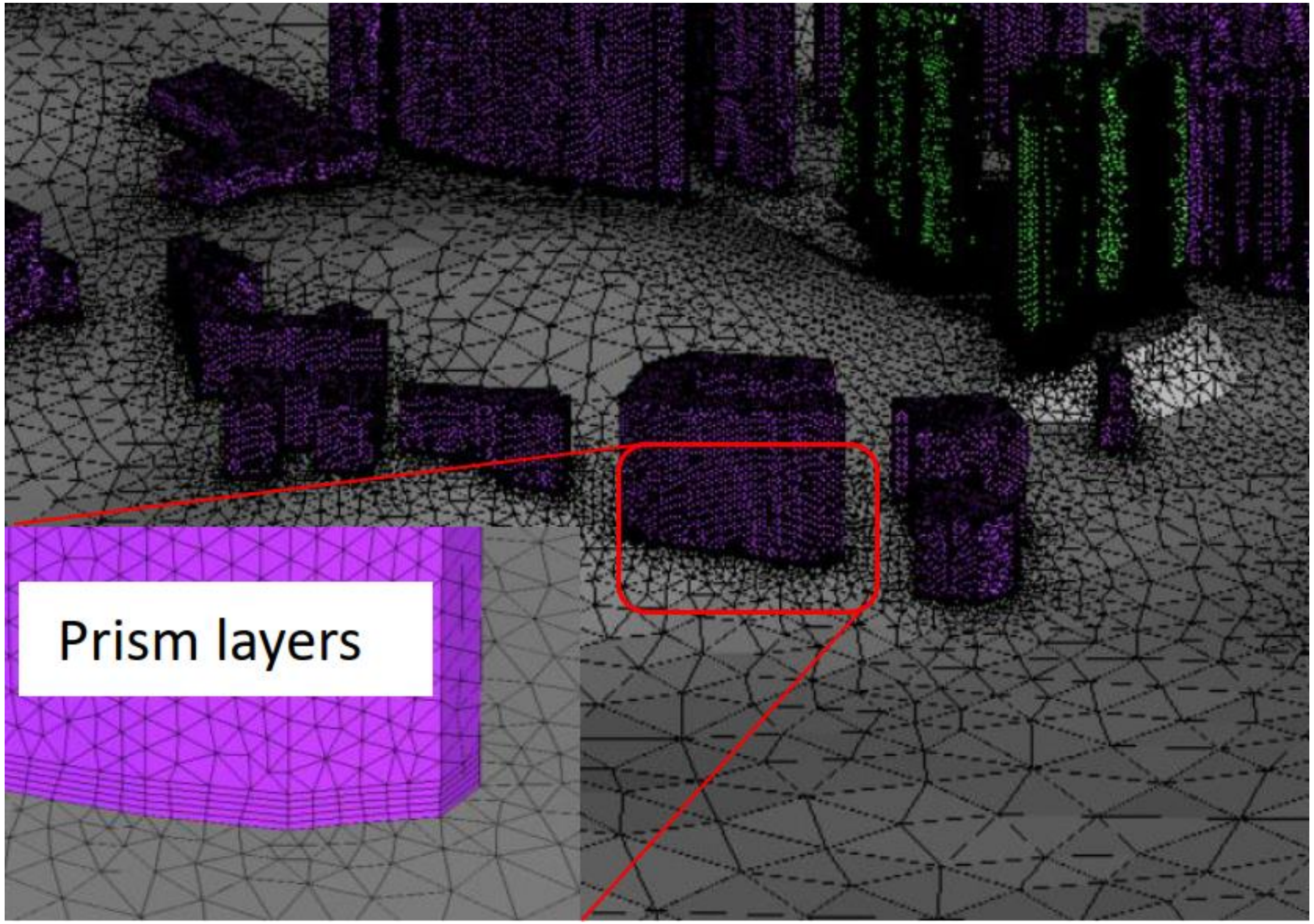
- Computational domain is very large in length, width and height;
- Wind profile of the atmosphere boundary condition is required;
- Coarse mesh is allowable for outer zone while fine mesh is required near building surfaces and ground surfaces;
- Boundary layer is required when studying the wind performance at the pedestrian level.

Characteristics of mesh for natural ventilation application



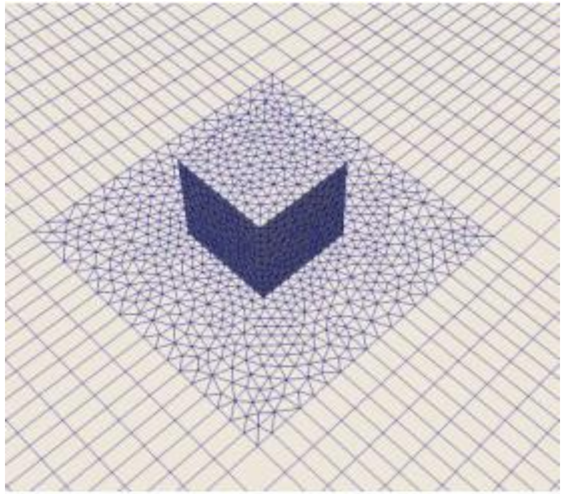
- Computational domain is very large in length, width and height;
- Wind profile of the atmosphere boundary condition is required;
- Coarse mesh is allowable for outer zone while fine mesh is required near building surfaces and ground surfaces;
- Boundary layer is required when studying the wind performance at the pedestrian level.

Characteristics of mesh for natural ventilation application

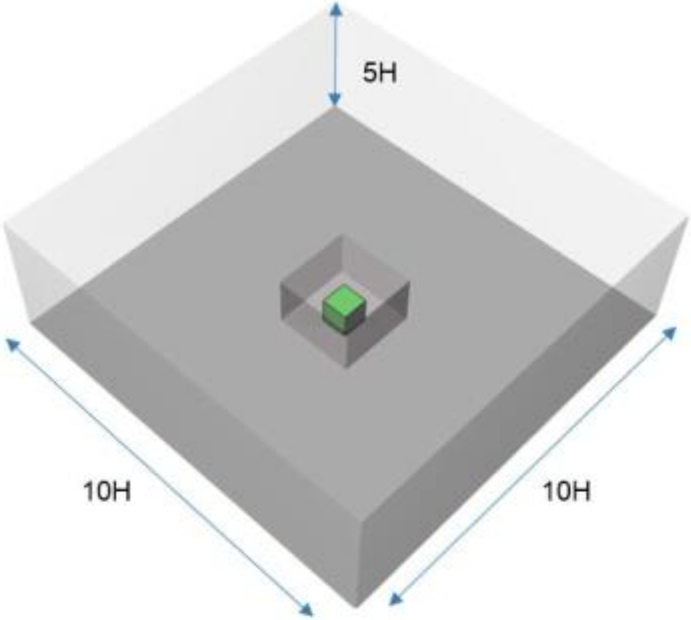


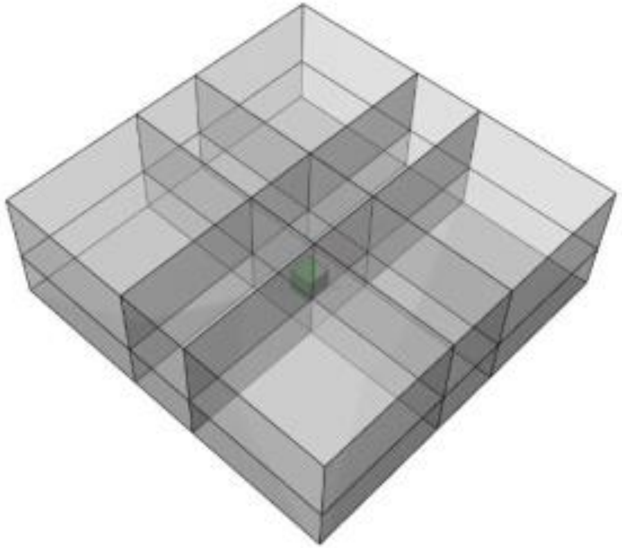
- Computational domain is very large in length, width and height;
- Wind profile of the atmosphere boundary condition is required;
- Coarse mesh is allowable for outer zone while fine mesh is required near building surfaces and ground surfaces;
- Boundary layer is required when studying the wind performance at the pedestrian level.

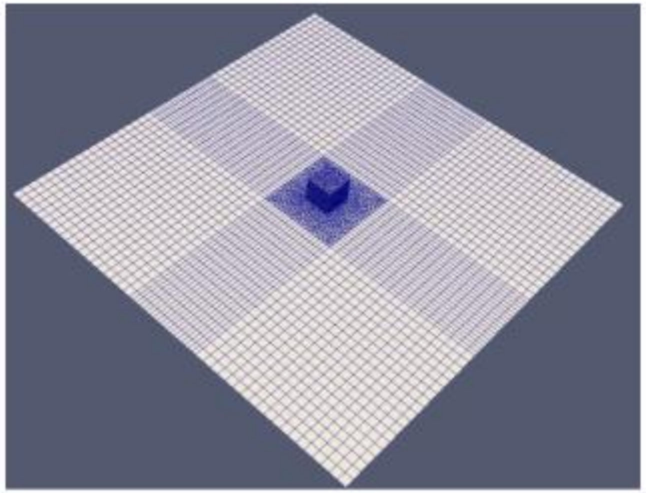
Hybrid mesh



A Solution to the problem:
Hybrid mesh – reduce the number of mesh and keep mesh quality
Outer zone – multiblock and structured grid
Inner zone – tetrahedral grid to capture building features



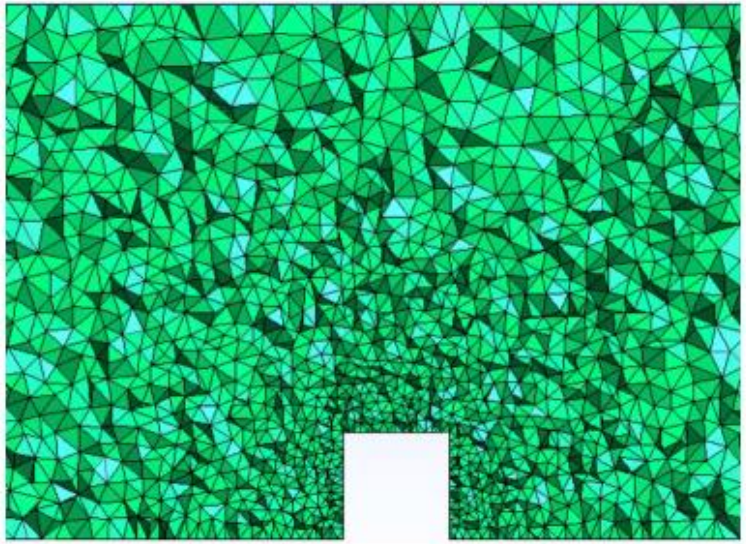


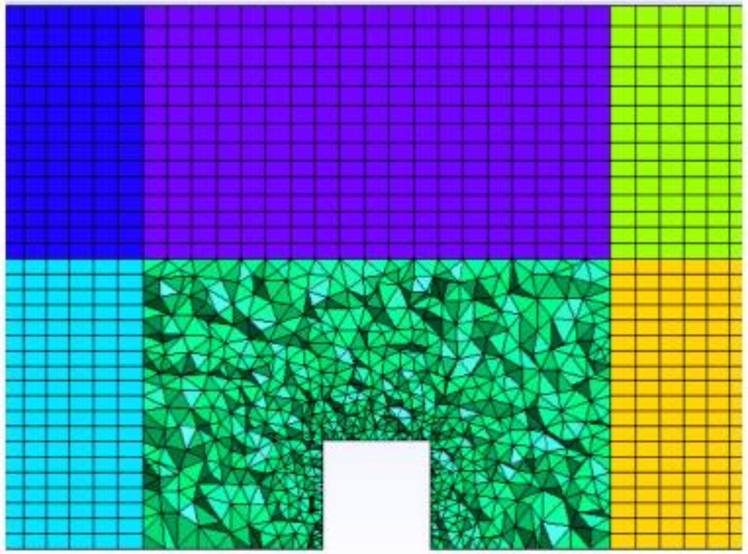


Hybrid mesh

A Solution to the problem:
Hybrid mesh – reduce the number of mesh and keep mesh quality
Outer zone – multiblock and structured grid
Inner zone – tetrahedral grid to capture building features

	Unstructured Mesh Scheme	Hybrid Mesh Scheme
Number of Tetrahedron	378,624	81,893
Number of Hexahedron	0	66,038
Number of Pyramid	0	1,805
Total Number of Mesh Element	378,624	149,736
Reduction of Mesh Elements	60.5%	
Total Computational Time	1,947 s (equivalent to 32 min)	403 s (equivalent to 6.7 min)
Reduction of Time	79.3%	





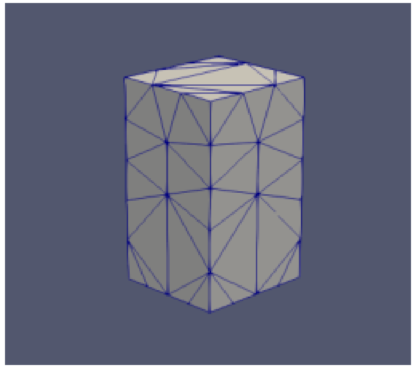
Octree-based hybrid mesh generation algorithm

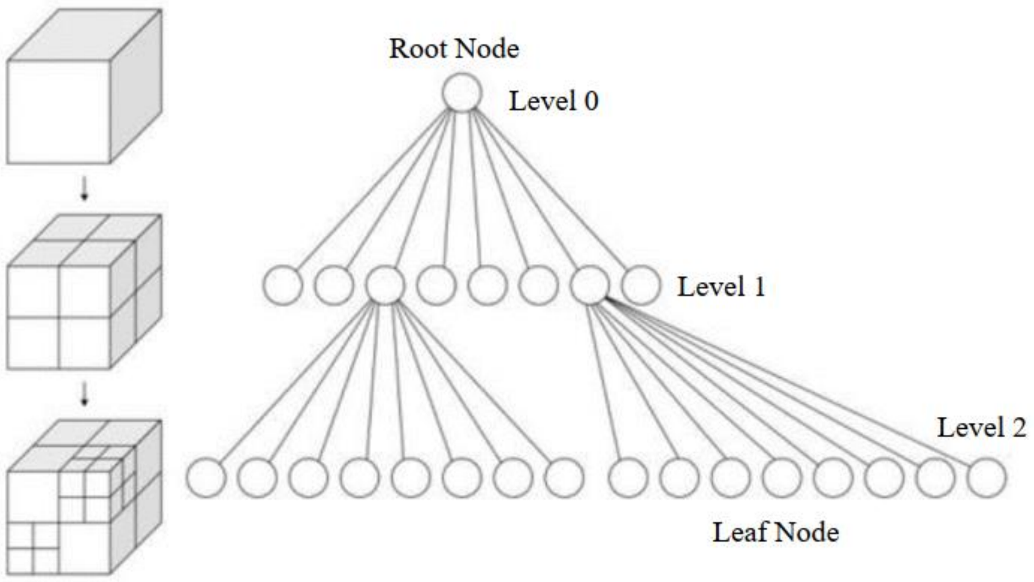
Further improvement on the hybrid mesh:

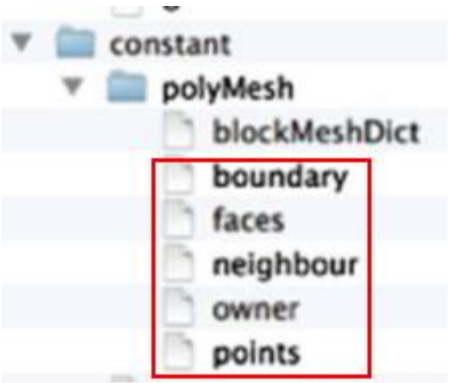
- Fully automatic meshing process
- Flexible mesh density control

Octree-based hybrid mesh generation algorithm

Read STL geometry → Mesh density setting → Automatic meshing → Output OpenFoam format mesh data







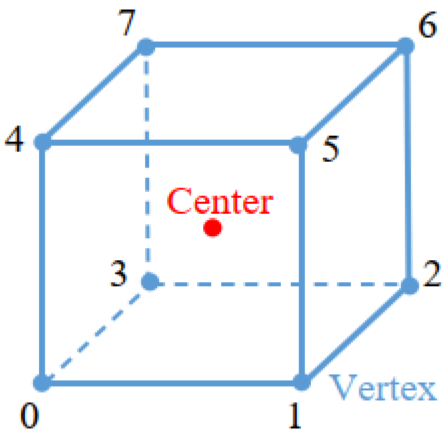
Octree-based hybrid mesh generation algorithm

Algorithm Development in C++

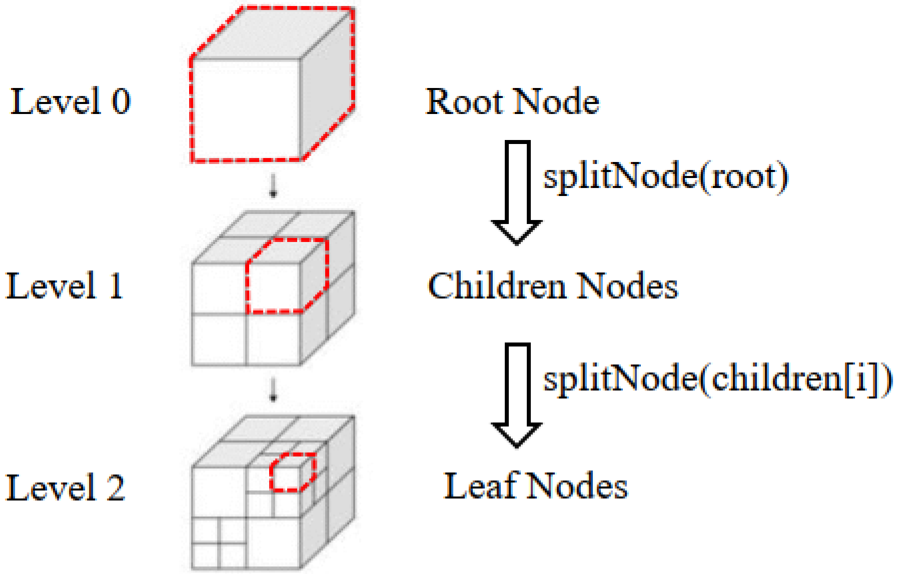
```
Class OctNode
{
string nid;
vector<double> center;
int level;
OctNode* parent;
vector<OctNode*> children;
…
void addChildNode(string _nid, int _level, vector<double> _center);
…
};
```

```
Class Octree
{
OctNode* root;
…
void define_root();
void splitNode (OctNode* node);
…
};
```

OctNode Structure

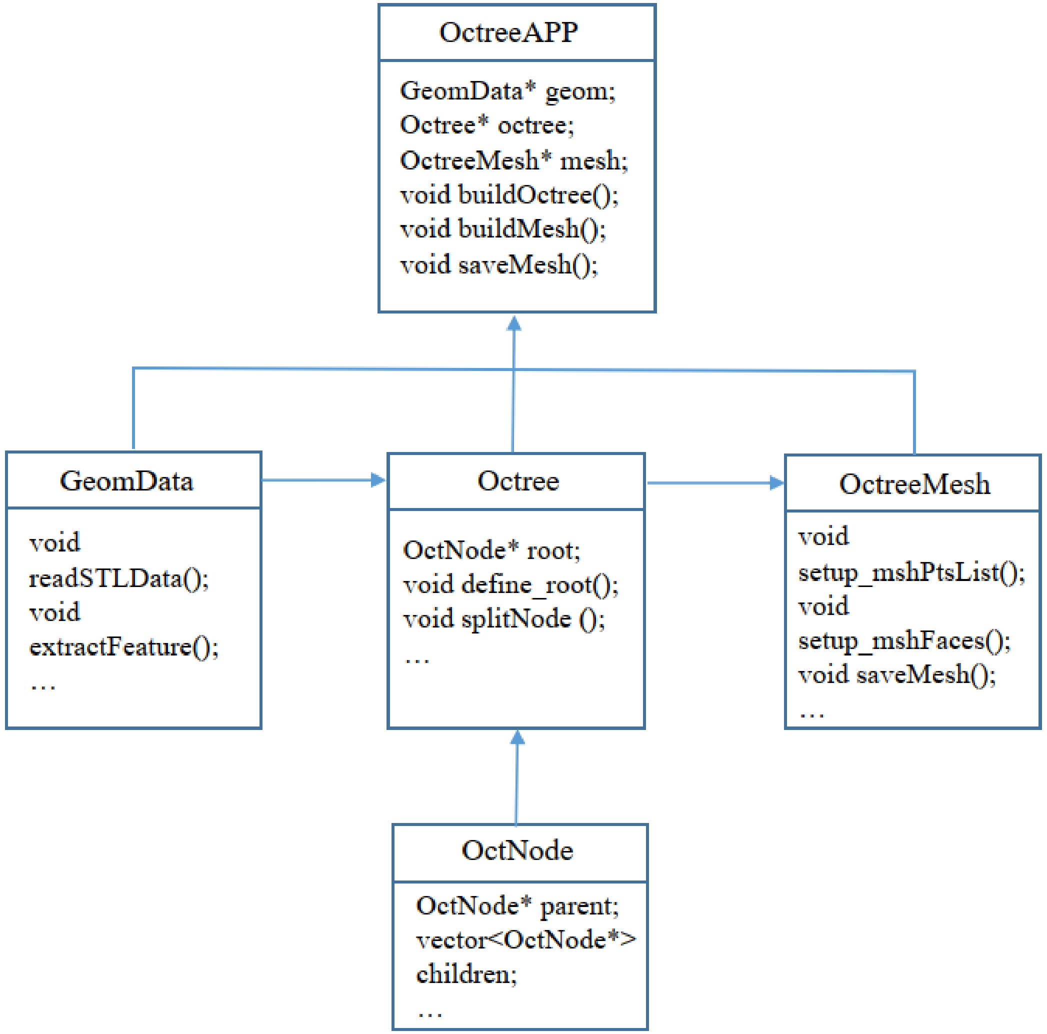


Octree Structure



Octree-based hybrid mesh generation algorithm

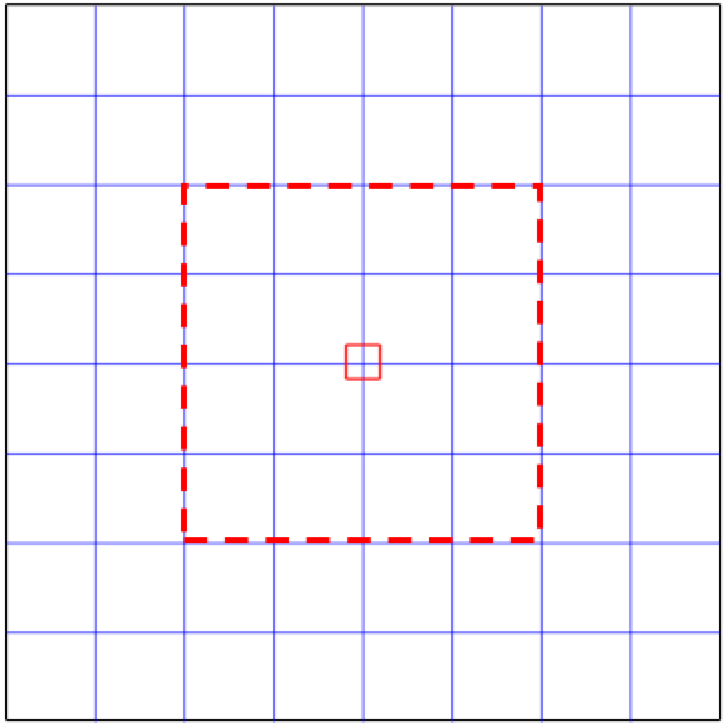
Algorithm design and implementation



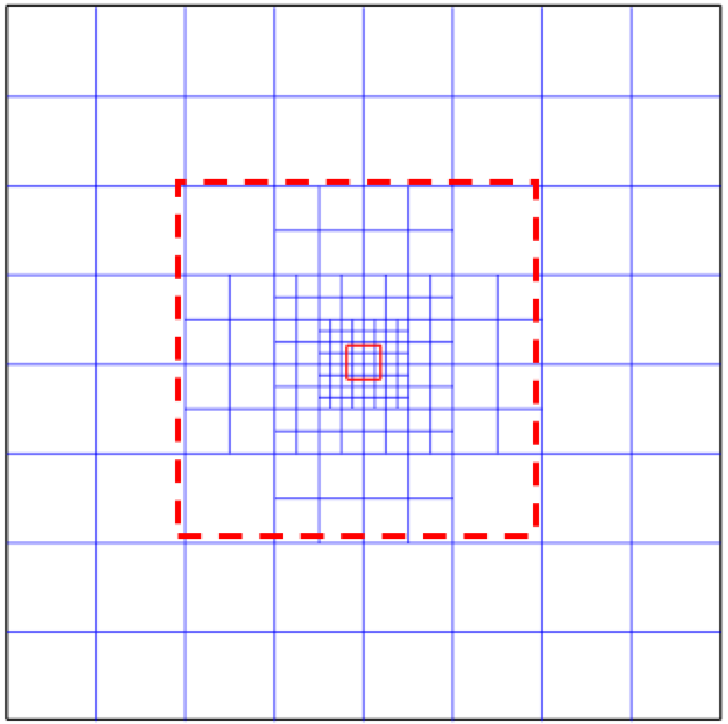
C++ Class	Function
OctreeAPP	Initiate the mesh generation application and control the operation process
GeomData	Read, process and store the input geometry data, e.g. the list of points and triangles from the STL file
Octree	Initiate root node and split nodes to generate required octree data structure
OctreeMesh	Convert the octree data structure to a specific CFD mesh format and output the mesh data for a CFD solver.

Octree-based hybrid mesh generation algorithm

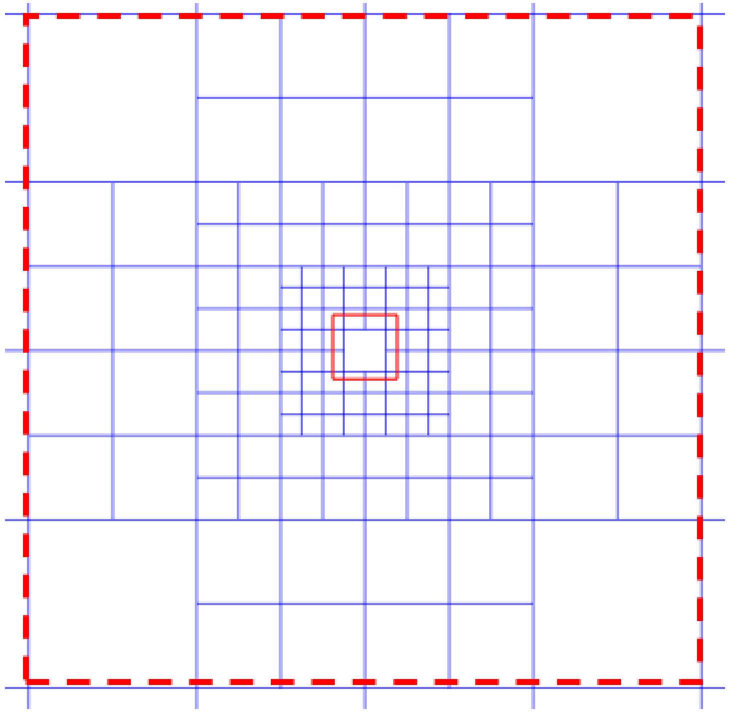
Example



a) Root node defined first to cover the whole domain and then split node until reaching the minimum node level.



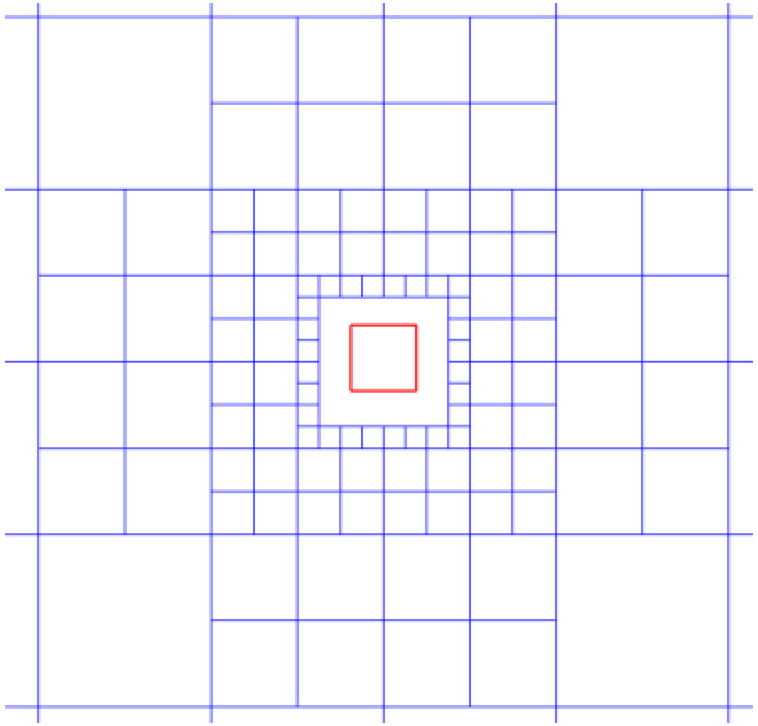
b) Split node adaptively to capture the geometry feature meanwhile keeping the octree balanced



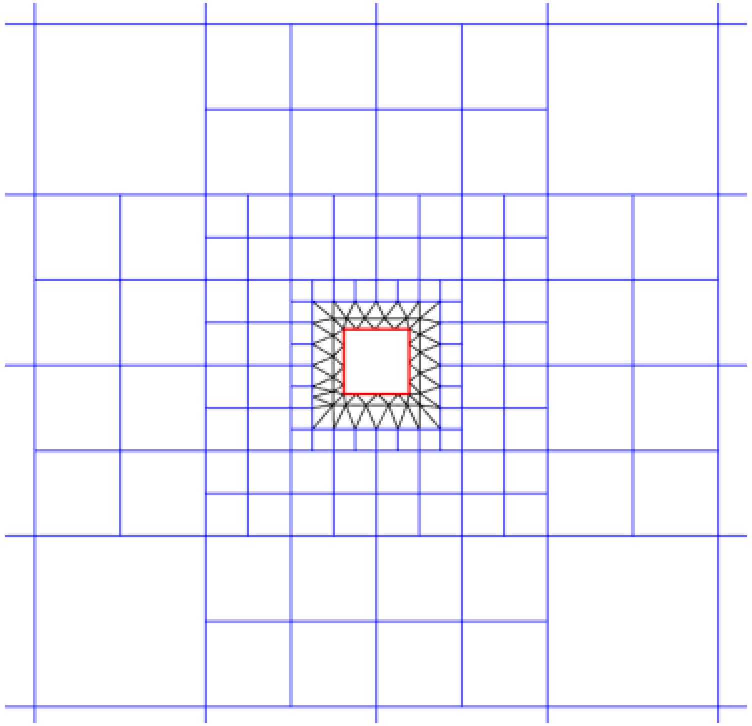
c) Exclude nodes within the geometry from the octree.

Octree-based hybrid mesh generation algorithm

The way forward



d) Delete nodes within the buffer zone offset from the geometry surfaces.



e) Reconstruct with tetrahedral mesh element