Hands-On Training with OpenFOAM

External Aerodynamics: Ahmed Body

Hrvoje Jasak

`h.jasak@wikki.co.uk`

Wikki Ltd, United Kingdom

Outline

Summary of Objectives: Steady turbulent flow around a 3-D car-like geometry

- Basic mesh generation: `snappyHexMesh`
- Case preparation: mesh and fields; initial and boundary conditions
- Basic solver setup
- Adjusting discretisation and linear solver parameters
- On-the-fly data extraction: function objects
- Field visualisation
- Parallel processing: case preparation, parallel run, data reconstruction

Outline

Tutorial Steps

1. Generate mesh using OpenFOAM tools: `blockMesh`, `snappyHexMesh`
2. Prepare mesh for CFD simulation
3. Boundary conditions: air tunnel simulation
4. Turbulence model; transport properties; initial field
5. Run simulation; plot residual
6. Change discretisation: Laplacian and turbulence
7. Change linear solver: AMG
8. Add pressure sampling point
9. Add `minMaxField` function object on the fly
10. Field post-processing: ParaView
11. Parallel decomposition, file layout and decomposition visualisation tool
12. Basics of parallel operation of the solver
13. Data reconstruction and visualisation after a parallel run

Mesh Generation With Snappy

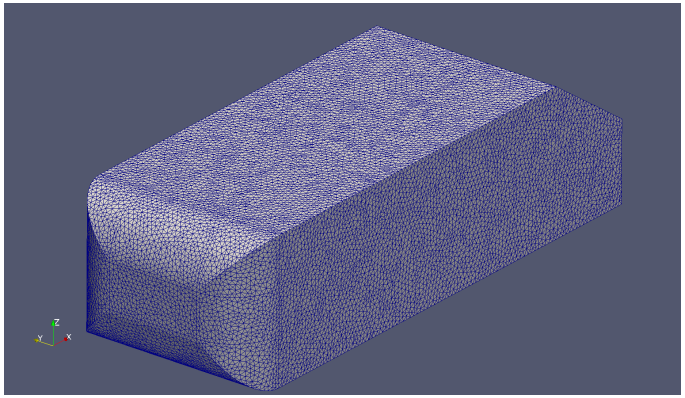
`snappyHexMesh`: Automatic Complex Geometry Mesher in OpenFOAM

- `snappyHexMesh` utility is available as a part of OpenFOAM
- Mesh generation fully automatic with many parameters allowing for the mesh quality control, local refinement and layered meshes
- The mesh generation algorithm proceeds in stages
  1. Creation of background (initial mesh)
  2. Cell splitting at edges and surfaces, cell removal, and local refinement
  3. Snapping to surfaces and layer creation
- Mesh generation algorithm is capable of handling complex surfaces
- Local feature detection (edges and surfaces) control is available: good representation of the original geometry
- Easy to control the transition from coarse to fine mesh zones: minimising the cell count
- Algorithm runs in parallel: creation of large meshes

Mesh Generation With Snappy

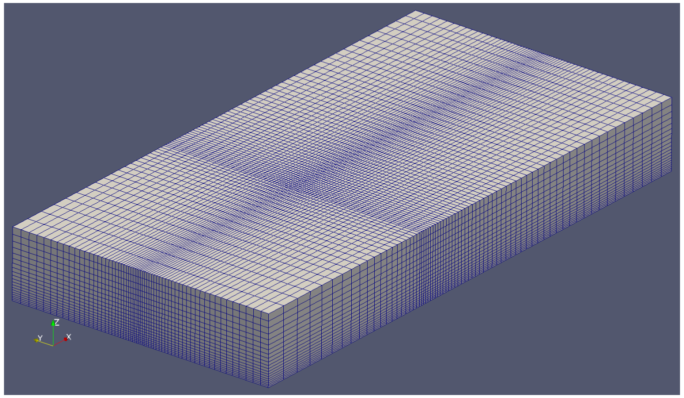
Surface Description: STL surface

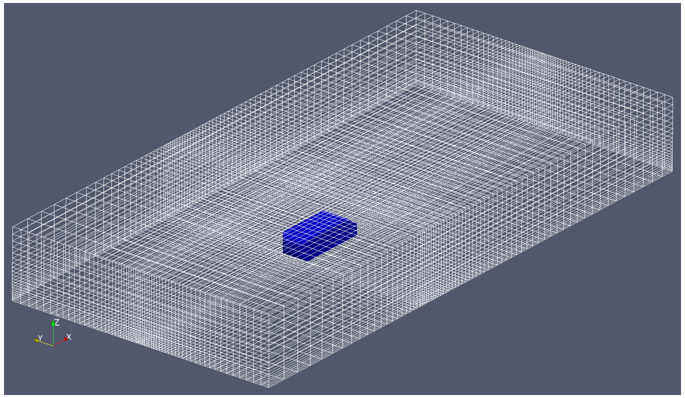
- Surface description for geometry obtained from external sources



Background Mesh With `blockMesh`

- Background mesh envelopes geometry; external boundaries may be kept





Block Mesh Generation

`blockMesh`: Simple Block-Structured Mesh Generator

- Generates a (matching) multi-block hexahedral mesh
- Operates on topologically supported structure
- Support for mesh grading and curved edges in blocks
- Boundary patches defined from block patches

Components

1. (Block) vertices: support for block structure
2. Hex blocks with number of cells and grading

```
hex (0 1 4 3 9 10 13 12)
(20 30 20)
simpleGrading (0.2 0.25 5)
```

3. Curved edge list

```
arc 0 1 (0.3 0.7 0)
```

4. Patch list: definition of patches external boundary, to support boundary conditions

Mesh Generation With Snappy

`snappyHexMesh`: Creating of Cut-and-Snapped Mesh

- Stages of Mesh Generation Process
  - Creation of background (initial) mesh using `blockMesh`
  - Cell splitting at edges and surfaces, cell removal, and local refinement
  - Snapping to surfaces and layer creation
- Structure of Mesh Control Files
  - `polyMesh` directory contains `blockMeshDict` for `blockMesh`
  - `triSurface` directory contains feature description and/or STL surface
  - `system` directory contains `snappyHexMeshDict`
- Main `snappyHexMeshDict` controls
  - `geometry`: primitive shapes, surfaces and refinement boxes
  - `castellatedMeshControls`: refinement parameters, location in mesh
  - `snapControls`: manage vertex motion in snapping
  - `addLayersControls`: surface cell layers controls

Mesh Generation With Snappy

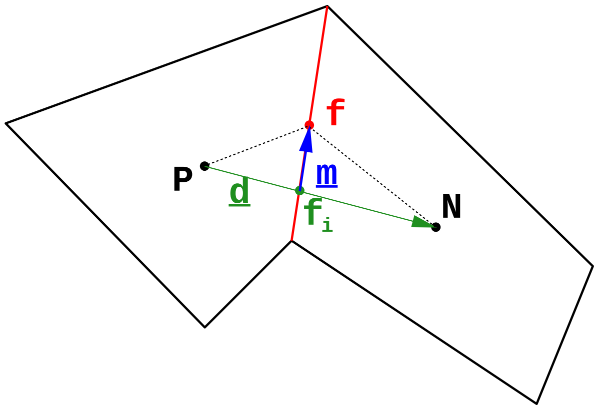
`snappyHexMesh`: Preparing a Mesh for CFD

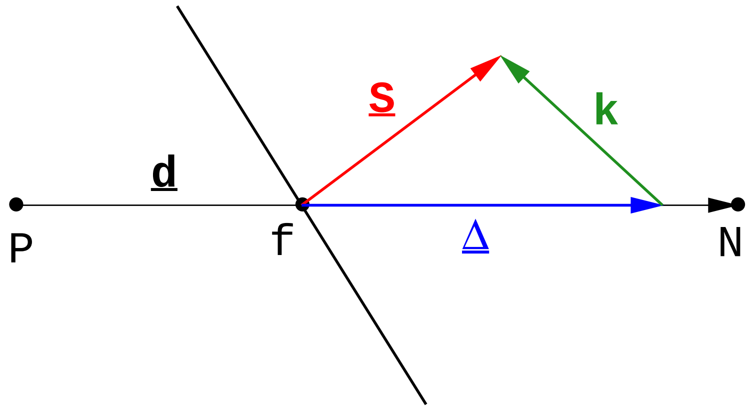
- `snappyHexMesh` generator operates in stages and stores intermediate meshes (on demand)
- We will use the final mesh only: copy mesh data from “mesh generation case” to “flow solution case”
- Check mesh dimensions, integrity and quality metrics: `checkMesh`
- Prepare setup of initial field and boundary conditions

Mesh Quality Metrics

Mesh Quality Metrics for the Finite Volume Method

- Quality of the mesh is closely connected with the underlying discretisation method: a cell away from "ideal" isotropic shape is not necessarily bad
- **Cell aspect ratio**. Defined as ratio of longest to shortest edge length. In many cases, this is desirable: align the cell with solution gradient
- **Cell size grading**. Usually with no consequences
- **Face non-orthogonality**. Defined as the angle between the face normal and $\overline{PN}$ vector α of $70 - 90^o$ increases solution cost and reduces accuracy; $\alpha > 90^o$ is fatal
- **Face skewness**. Defined as the distance between face centroid and face integration point. Reduces accuracy but without stability implications





Geometry and Flow Conditions

Case Setup: Ahmed Body

- Steady incompressible turbulent flow: `simpleFoam`
- Material properties: $\nu = 1.5 \times 10^{-5}\,\mathrm{m^2/s}$
- Inlet conditions:

$$\mathbf{u} = (40\,0\,0)\,\mathrm{m/s}$$

$$k = 6\,\mathrm{m^2/s^2}$$

$$\omega = 21.41\,1/\mathrm{s}$$