

# 1 star4ToFoam

This section describes how to convert a mesh generated for STAR-CD with pro-STAR version 4 into a form that can be read by OpenFOAM mesh classes.

## 1.1 General advice on conversion

It is strongly recommended that the user run the pro-STAR mesh checking tools before attempting the conversion. After conversion, the `checkMesh` utility should be run on the newly converted mesh. Problematic cells and boundaries should be checked and fixed before attempting to use the mesh with OpenFOAM. Note that an invalid mesh will not run with OpenFOAM, although it may run in another environment that does not impose similar validity criteria ??.

## 1.2 Limitations

The converter does not support sliding interfaces.

## 1.3 Changes from pro-STAR version 3.x

Since the pro-STAR file formats, and some concepts, have changed substantially from pro-STAR version 3.x, older formats cannot be read with the converter. Legacy pro-STAR files can, however, be re-exported with newer pro-STAR versions.

### 1.3.1 Cells Shapes – Trimmed and Polyhedral Cells

All three-dimensional pro-STAR cell shapes are supported: hexa, tetra, prism, pyramid and N-sided polyhedra. With the recent addition of polyhedral cells to Star-CD, trimmed cells have become unnecessary.

### 1.3.2 Couples

Arbitrary and integral couples are no longer supported by the converter (nor by the STAR-CD solver itself). Coupled cells should instead be transformed into polyhedral cells using the following pro-STAR command:

```
cptrans all
```

Alternatively, the OpenFOAM mesh topology changing operations can be used to join particular boundary patches, but this is beyond the scope of this section.

### 1.3.3 Baffles

Baffles elements (*i.e.*, zero-thickness elements inserted into the fluid domain) are now supported by the converter. When considering baffles, it is important to note that they are not cells *per se*, but a means of preventing internal connectivity between two cells in favour of specifying a different relationship between them. A baffle boundary patch is thus a special case of a coupled boundary patch and organized similarly:  $0 \rightarrow N/2$  for the baffle side1 faces and  $N/2 \rightarrow N$  for the baffle side2 faces. The baffle connectivity is currently documented by the converter in `constant/polyMesh/interfaces` for possible future use.

For the simplest and most common baffle boundary condition (thin walls with slip/no-slip conditions), the standard wall boundary types can be used. Boundary patch types for more specialized applications, e.g., porous baffles or localized heat injection, have not yet been implemented.

#### 1.3.4 Cell Tables

The star4ToFoam converter retains the pro-STAR cell type information. A description of this and its use can be found in 1.8.

### 1.4 Preparation for pro-STAR export

Couples must be translated into polyhedral cells with the following pro-STAR commands:

```
cptrans all
```

Any dangling boundaries must be purged with the following pro-STAR commands:

```
cset all
bset news cset
bset invert
bdel bset
bcomp all$yes
```

Extraneous vertices should be removed with the following pro-STAR commands:

```
cset all
vset news cset
vset invert
vdel vset
vcomp all$yes
```

Compacting the cell numbering will also help reduce the conversion memory requirements, but is not strictly required:

```
ccomp all$yes
```

It is, however, imperative that the model be checked with the pro-STAR command:

```
check all
```

which may reveal some unforeseen error(s). Also, take note of the model scale. By default the star4ToFoam utility assumes millimeters have been used within pro-STAR and scales to meters accordingly. Use the ' -noscale ' option to suppress this behaviour.

### 1.5 Writing out the mesh data

The pro-STAR file menu option "Save As Coded..." will save the model configuration ( .inp file), and save the mesh geometry information in three separate ASCII files: boundaries ( .bnd file), cells ( .cel file) and vertices ( .vrt file). Although the model configuration ( .inp file) is not used directly by the star4ToFoam converter, other utilities can extract useful information from this file.

After saving the three files, note the sub-models, material and fluid properties used – these will be needed when preparing the OpenFOAM case files.

## 1.6 Converting the mesh to OpenFOAM format

The translator utility `star4ToFoam` can now be run to create the points, cells and boundaries files necessary for a OpenFOAM run:

```
star4ToFoam <root> <caseName> <meshFilePrefix> [-noscale]
```

where `meshFilePrefix` is the name of the prefix of the mesh files, including the full or relative path. The option `'-noscale'` can be used to suppress the default millimeter to meter scaling. After the utility has finished running, OpenFOAM boundary types should be specified in the usual way with FoamX or editing by hand.

## 1.7 Default boundary conditions

By default STAR-CD assigns any boundary faces that are not explicitly associated with a boundary region to the “Default\_Boundary\_Region” with the assigned boundary type 0. The `star4ToFoam` converter follows a similar logic – any missing boundary faces will be added to the final patch “Default\_Boundary\_Region”. This patch should be carefully examined for inadvertently collected faces.

## 1.8 Cell tables and cell types

STAR-CD uses a so-called “cell table” to organize and assign material and model properties (*e.g.*, fluid/solid, porosity, ...) to specific groups of cells. The `ccm` and `star` converters use two files to preserve this information:

- the file `0/cellTableId` is a volume scalar field containing the index of each cell in the cell table. Regardless of the control dictionary settings, this file is written in ASCII format. For most cell tables, this will be compacter than binary and helps ease processing with external programs.
- the file `constant/cellTable` contains dictionaries specifying the characteristics of each cell table entry.

The combination of both files allows `cellSets` and `cellZones` to be created for any combination of properties (*e.g.*, porosities and solids) for subsequent use in OpenFOAM. Additionally, the reverse converters (`foamMeshToStar` and `foamMeshToCcm`) use this information when exporting an OpenFOAM mesh to one of the STAR-CD formats. The keywords used for the dictionary entries (`Label`, `MaterialType`, *etc.*) are chosen to closely match those used by the `ccm`-format. Here is an example of a `constant/cellTable` sub-dictionary entry:

```
<word>                                     // unique identifier
{
    Id          <int>;                      // mandatory unique index
    Label       <word>;                     // optional
    MaterialType <word>;                     // optional (fluid | solid)
    MaterialId  <int>;                       // optional
    PorosityId  <int>;                       // optional
    GroupId     <int>;                       // optional
    SpinId      <int>;                       // optional
}
```

While the ccm26ToFoam converter can automatically extract this information from the ccm file, the pro-STAR formats restrict the star4ToFoam converter to extracting the Id and MaterialType entries.

### 1.8.1 FOAM/Star (fStar) utilities

As previously mentioned, the cell table information saved in 0/cellTableId and constant/cellTable can be used to define new cellSets. Assuming that the star4ToFoam converter has been used, the information saved in constant/cellTable is, however, rather incomplete. More useful information can be extracted from the pro-STAR .inp file using the fStar-cellTable utility.

**fStar-cellTable <file.inp>** Extract STAR-CD cell table information from the input file into a form suitable for creating a constant/cellTable file.

**fStar-cellSets [property1 . propertyN]** Use the contents of 0/cellTableId and constant/cellTable to create constant/polyMesh/sets/. . . cellSets. The requested property names are reformatted for correct capitalization.

**fStar-coordinates <file.inp>** Extract STAR-CD coordinate system information from the input file into a form suitable for creating a constant/coordinateSystems file for future use (*e.g.*, with porosities or rotating systems).

## 1.9 foamMeshToStar

There can be several reasons for wishing to export an OpenFOAM mesh into another format:

- importing OpenFOAM mesh geometries in third-party tools that require a cell-based storage scheme.
- the need for data exchange and/or benchmarking with an existing CFD package.

The newer pro-STAR file formats are fortunately not only relatively simple to read and write, they also include support for polyhedral cells. This makes them an ideal, lossless meta-format for a cell-based representation of OpenFOAM meshes. Using the star4ToFoam and foamMeshToStar converters, OpenFOAM support can be quickly added for many other formats. For example, import/export for I-deas universal files can be realized with fewer than 100 lines of Perl code for each direction.

When creating the pro-STAR cells (.cel file), the foamMeshToStar converter uses the contents of the 0/cellTableId and constant/cellTable files to determine the cell table index and the material type. If these files are missing, the cellZones (if any) are used. By default the foamMeshToStar utility assumes that millimeters are to be used within pro-STAR and scales the mesh accordingly. Use the '-noscale' option to suppress this behaviour. The foamMeshToStar converter can also be invoked with a '-surface' option. In which case, the boundary faces are written as two-dimensional shell elements and only the points used by these faces are written. The written points retain their original index. The extracted shell elements are assigned the index from the boundary face, and assume the cell table index from their associated volume cell. This allows morphing of the mesh surface in an external package and subsequent re-importing or re-reading of the modified coordinate positions in OpenFOAM.

### **1.10 ccm26ToFoam / foamMeshToCcm / foamDataToCcm**

In progress - brief description: ccm26ToFoam imports the joint ccm format used by STAR-CD and starccm+. The conversion uses the same cell tables semantics as the star4ToFoam conversion. Baffles are supported, but due to the way that the ccm format is generated this aspect requires much closer attention. When the model includes porosities or solids, the calculation domain is automatically separated into individual regions. The interfaces specification then includes not only baffles, but also the connection between the regions. Reorganizing and re-joining the mesh is thus necessary. Depending on how OpenFOAM treats multi-physics, it could be useful to maintain the separate regions and rework the interfaces to provide the appropriate coupling between regions (e.g., solid/fluid).

The foamMeshToCcm is similar to foamMeshToStar, but does not include surface extraction. The foamDataToCcm utility transcribes primitive OpenFOAM fluid fields to ccm format.