

### 3.5.12 Finance

`financialFoam` Solves the Black-Scholes equation to price commodities.

## 3.6 Standard utilities

The utilities with the OpenFOAM distribution are in the `$FOAM_UTILITIES` directory. The names are reasonably descriptive, *e.g.* `ideasToFoam` converts mesh data from the format written by I-DEAS to the OpenFOAM format. The descriptions of current utilities distributed with OpenFOAM are given in the following Sections.

### 3.6.1 Pre-processing

`applyBoundaryLayer` Apply a simplified boundary-layer model to the velocity and turbulence fields based on the 1/7th power-law.

`boxTurb` Makes a box of turbulence which conforms to a given energy spectrum and is divergence free.

`changeDictionary` Utility to change dictionary entries, *e.g.* can be used to change the patch type in the field and `polyMesh`/boundary files.

`createExternalCoupledPatchGeometry` Application to generate the patch geometry (points and faces) for use with the `externalCoupled` boundary condition.

`dsmcInitialise` Initialise a case for `dsmcFoam` by reading the initialisation dictionary system/`dsmcInitialise`.

`engineSwirl` Generates a swirling flow for engine calculations.

`faceAgglomerate` Agglomerate boundary faces using the `pairPatchAgglomeration` algorithm. It writes a map from the fine to coarse grid.

`foamSetupCHT` Sets up a multi-region case using template files for material properties, field and system files.

`mapFields` Maps volume fields from one mesh to another, reading and interpolating all fields present in the time directory of both cases.

`mapFieldsPar` Maps volume fields from one mesh to another, reading and interpolating all fields present in the time directory of both cases. Parallel and non-parallel cases are handled without the need to reconstruct them first.

`mdInitialise` Initialises fields for a molecular dynamics (MD) simulation.

`setFields` Set values on a selected set of cells/patchfaces through a dictionary.

`setWaves` Applies wave models to the entire domain for case initialisation using level sets for second-order accuracy.

`viewFactorsGen` View factors are calculated based on a face agglomeration array (`finalAgglom` generated by `faceAgglomerate` utility).

`wallFunctionTable` Generates a table suitable for use by tabulated wall functions.

### 3.6.2 Mesh generation

**blockMesh** A multi-block mesh generator.

**extrudeMesh** Extrude mesh from existing patch (by default outwards facing normals; optional flips faces) or from patch read from file.

**extrude2DMesh** Takes 2D mesh (all faces 2 points only, no front and back faces) and creates a 3D mesh by extruding with specified thickness.

**extrudeToRegionMesh** Extrude **faceZones** (internal or boundary faces) or **faceSets** (boundary faces only) into a separate mesh (as a different region).

**foamyHexMesh** Conformal Voronoi automatic mesh generator

**foamyQuadMesh** Conformal-Voronoi 2D extruding automatic mesher with grid or read initial points and point position relaxation with optional "squarification".

**snappyHexMesh** Automatic split hex mesher. Refines and snaps to surface.

### 3.6.3 Mesh conversion

**ansysToFoam** Converts an ANSYS input mesh file, exported from I-DEAS, to OpenFOAM format.

**cfx4ToFoam** Converts a CFX 4 mesh to OpenFOAM format.

**datToFoam** Reads in a **datToFoam** mesh file and outputs a points file. Used in conjunction with **blockMesh**.

**fluent3DMeshToFoam** Converts a Fluent mesh to OpenFOAM format.

**fluentMeshToFoam** Converts a Fluent mesh to OpenFOAM format including multiple region and region boundary handling.

**foamMeshToFluent** Writes out the OpenFOAM mesh in Fluent mesh format.

**foamToStarMesh** Reads an OpenFOAM mesh and writes a pro-STAR (v4) **bnd/cel/vrt** format.

**foamToSurface** Reads an OpenFOAM mesh and writes the boundaries in a surface format.

**gambitToFoam** Converts a GAMBIT mesh to OpenFOAM format.

**gmshToFoam** Reads **.msh** file as written by Gmsh.

**ideasUnvToFoam** I-Deas unv format mesh conversion.

**kivaToFoam** Converts a KIVA3v grid to OpenFOAM format.

**mshToFoam** Converts **.msh** file generated by the Adventure system.

**netgenNeutralToFoam** Converts neutral file format as written by Netgen v4.4.

**ccm26ToFoam** Reads CCM files as written by Prostar/ccm using ccm 2.6 (not 2.4)

`plot3dToFoam` Plot3d mesh (ascii/formatted format) converter.

`sammToFoam` Converts a Star-CD (v3) SAMM mesh to OpenFOAM format.

`star3ToFoam` Converts a Star-CD (v3) pro-STAR mesh into OpenFOAM format.

`star4ToFoam` Converts a Star-CD (v4) pro-STAR mesh into OpenFOAM format.

`tetgenToFoam` Converts `.ele` and `.node` and `.face` files, written by tetgen.

`vtkUnstructuredToFoam` Converts ascii `.vtk` (legacy format) file generated by vtk/paraview.

`writeMeshObj` For mesh debugging: writes mesh as three separate OBJ files which can be viewed with e.g. javaview.

### 3.6.4 Mesh manipulation

`attachMesh` Attach topologically detached mesh using prescribed mesh modifiers.

`autoPatch` Divides external faces into patches based on (user supplied) feature angle.

`checkMesh` Checks validity of a mesh.

`createBaffles` Makes internal faces into boundary faces. Does not duplicate points, unlike `mergeOrSplitBaffles`.

`createPatch` Utility to create patches out of selected boundary faces. Faces come either from existing patches or from a `faceSet`.

`deformedGeom` Deforms a `polyMesh` using a displacement field `U` and a scaling factor supplied as an argument.

`flattenMesh` Flattens the front and back planes of a 2D cartesian mesh.

`insideCells` Picks up cells with cell centre 'inside' of surface. Requires surface to be closed and singly connected.

`mergeMeshes` Merges two meshes.

`mergeOrSplitBaffles` Detects faces that share points (baffles). Either merge them or duplicate the points.

`mirrorMesh` Mirrors a mesh around a given plane.

`moveDynamicMesh` Mesh motion and topological mesh changes utility.

`moveEngineMesh` Solver for moving meshes for engine calculations.

`moveMesh` Solver for moving meshes.

`objToVTK` Read obj line (not surface!) file and convert into vtk.

`orientFaceZone` Corrects orientation of `faceZone`.

- polyDualMesh** Calculates the dual of a **polyMesh**. Adheres to all the feature and patch edges.
- refineMesh** Utility to refine cells in multiple directions.
- renumberMesh** Renumbers the cell list in order to reduce the bandwidth, reading and renumbering all fields from all the time directories.
- rotateMesh** Rotates the mesh and fields from the direction n1 to direction n2.
- setSet** Manipulate a cell/face/point/ set or zone interactively.
- setsToZones** Add **pointZones/faceZones/cellZones** to the mesh from similar named **pointSets/faceSets/cellSets**.
- singleCellMesh** Reads all fields and maps them to a mesh with all internal faces removed (**singleCellFvMesh**) which gets written to region "singleCell".
- splitMesh** Splits mesh by making internal faces external. Uses **attachDetach**.
- splitMeshRegions** Splits mesh into multiple regions.
- stitchMesh** 'Stitches' a mesh.
- subsetMesh** Selects a section of mesh based on a **cellSet**.
- topoSet** Operates on **cellSets/faceSets/pointSets** through a dictionary.
- transformPoints** Transforms the mesh points in the **polyMesh** directory according to the translate, rotate and scale options.
- zipUpMesh** Reads in a mesh with hanging vertices and zips up the cells to guarantee that all polyhedral cells of valid shape are closed.

### 3.6.5 Other mesh tools

- autoRefineMesh** Utility to refine cells near to a surface.
- collapseEdges** Collapses short edges and combines edges that are in line.
- combinePatchFaces** Checks for multiple patch faces on same cell and combines them. Multiple patch faces can result from e.g. removal of refined neighbouring cells, leaving 4 exposed faces with same owner.
- modifyMesh** Manipulates mesh elements.
- PDRMesh** Mesh and field preparation utility for PDR type simulations.
- refineHexMesh** Refines a hex mesh by 2x2x2 cell splitting.
- refinementLevel** Tries to figure out what the refinement level is on refined Cartesian meshes. Run BEFORE snapping.
- refineWallLayer** Utility to refine cells next to patches.

`removeFaces` Utility to remove faces (combines cells on both sides).

`selectCells` Select cells in relation to surface.

`splitCells` Utility to split cells with flat faces.

### 3.6.6 Post-processing

`engineCompRatio` Calculate the geometric compression ratio. Note that if you have valves and/or extra volumes it will not work, since it calculates the volume at BDC and TCD.

`noise` Utility to perform noise analysis of pressure data using the `noiseFFT` library.

`pdfPlot` Generates a graph of a probability distribution function.

`postChannel` Post-processes data from channel flow calculations.

`postProcess` Execute the set of `functionObjects` specified in the selected dictionary (which defaults to `system/controlDict`) or on the command-line for the selected set of times on the selected set of fields.

`particleTracks` Generates a VTK file of particle tracks for cases that were computed using a tracked-parcel-type cloud.

`steadyParticleTracks` Generates a VTK file of particle tracks for cases that were computed using a steady-state cloud NOTE: case must be re-constructed (if running in parallel) before use

`temporalInterpolate` Interpolate fields between time-steps e.g. for animation.

### 3.6.7 Post-processing data converters

`foamDataToFluent` Translates OpenFOAM data to Fluent format.

`foamToEnight` Translates OpenFOAM data to EnSight format.

`foamToEnightParts` Translates OpenFOAM data to EnSight format. An EnSight part is created for each `cellZone` and patch.

`foamToTetDualMesh` Converts `polyMesh` results to `tetDualMesh`.

`foamToVTK` Legacy VTK file format writer.

`smapToFoam` Translates a STAR-CD SMAP data file into OpenFOAM field format.

### 3.6.8 Surface mesh (e.g. OBJ/STL) tools

`surfaceAdd` Add two surfaces. Does geometric merge on points. Does not check for overlapping/intersecting triangles.

`surfaceAutoPatch` Patches surface according to feature angle. Like `autoPatch`.

`surfaceBooleanFeatures` Generates the *extendedFeatureEdgeMesh* for the interface between a boolean operation on two surfaces. Assumes that the orientation of the surfaces is correct.

`surfaceCheck` Checks geometric and topological quality of a surface.

`surfaceClean` Removes baffles - collapses small edges, removing triangles. - converts sliver triangles into split edges by projecting point onto base of triangle.

`surfaceCoarsen` Surface coarsening using 'bunnylod':

`surfaceConvert` Converts from one surface mesh format to another.

`surfaceFeatureConvert` Convert between `edgeMesh` formats.

`surfaceFeatures` Identifies features in a surface geometry and writes them to file, based on control parameters specified by the user.

`surfaceFind` Finds nearest face and vertex.

`surfaceHookUp` Find close open edges and stitches the surface along them

`surfaceInertia` Calculates the inertia tensor, principal axes and moments of a command line specified `triSurface`. Inertia can either be of the solid body or of a thin shell.

`surfaceLambdaMuSmooth` Smooths a surface using lambda/mu smoothing.

`surfaceMeshConvert` Converts between surface formats with optional scaling or transformations (rotate/translate) on a `coordinateSystem`.

`surfaceMeshConvertTesting` Converts from one surface mesh format to another, but primarily used for testing functionality.

`surfaceMeshExport` Export from `surfMesh` to various third-party surface formats with optional scaling or transformations (rotate/translate) on a `coordinateSystem`.

`surfaceMeshImport` Import from various third-party surface formats into `surfMesh` with optional scaling or transformations (rotate/translate) on a `coordinateSystem`.

`surfaceMeshInfo` Miscellaneous information about surface meshes.

`surfaceMeshTriangulate` Extracts surface from a `polyMesh`. Depending on output surface format triangulates faces.

`surfaceOrient` Set normal consistent with respect to a user provided 'outside' point. If the `-inside` option is used the point is considered inside.

**surfacePointMerge** Merges points on surface if they are within absolute distance. Since absolute distance use with care!

**surfaceRedistributePar** (Re)distribution of **triSurface**. Either takes an undecomposed surface or an already decomposed surface and redistributes it so that each processor has all triangles that overlap its mesh.

**surfaceRefineRedGreen** Refine by splitting all three edges of triangle ('red' refinement). Neighbouring triangles (which are not marked for refinement get split in half ('green' refinement). (R. Verfurth, "A review of a posteriori error estimation and adaptive mesh refinement techniques", Wiley-Teubner, 1996)

**surfaceSplitByPatch** Writes regions of **triSurface** to separate files.

**surfaceSplitByTopology** Strips any baffle parts of a surface. A baffle region is one which is reached by walking from an open edge, and stopping when a multiply connected edge is reached.

**surfaceSplitNonManifolds** Takes multiply connected surface and tries to split surface at multiply connected edges by duplicating points. Introduces concept of - **borderEdge**. Edge with 4 faces connected to it. - **borderPoint**. Point connected to exactly 2 **borderEdges**. - **borderLine**. Connected list of **borderEdges**.

**surfaceSubset** A surface analysis tool which sub-sets the **triSurface** to choose only a part of interest. Based on **subsetMesh**.

**surfaceToPatch** Reads surface and applies surface regioning to a mesh. Uses **boundaryMesh** to do the hard work.

**surfaceTransformPoints** Transform (scale/rotate) a surface. Like **transformPoints** but for surfaces.

### 3.6.9 Parallel processing

**decomposePar** Automatically decomposes a mesh and fields of a case for parallel execution of OpenFOAM.

**reconstructPar** Reconstructs fields of a case that is decomposed for parallel execution of OpenFOAM.

**reconstructParMesh** Reconstructs a mesh using geometric information only.

**redistributePar** Redistributes existing decomposed mesh and fields according to the current settings in the **decomposeParDict** file.

### 3.6.10 Thermophysical-related utilities

**adiabaticFlameT** Calculates the adiabatic flame temperature for a given fuel over a range of unburnt temperatures and equivalence ratios.

**chemkinToFoam** Converts CHEMKINIII thermodynamics and reaction data files into OpenFOAM format.

`equilibriumCO` Calculates the equilibrium level of carbon monoxide.

`equilibriumFlameT` Calculates the equilibrium flame temperature for a given fuel and pressure for a range of unburnt gas temperatures and equivalence ratios; the effects of dissociation on O<sub>2</sub>, H<sub>2</sub>O and CO<sub>2</sub> are included.

`mixtureAdiabaticFlameT` Calculates the adiabatic flame temperature for a given mixture at a given temperature.

### 3.6.11 Miscellaneous utilities

`foamDictionary` Interrogates and manipulates dictionaries.

`foamFormatConvert` Converts all IOobjects associated with a case into the format specified in the `controlDict`.

`foamListTimes` List times using `timeSelector`.

`patchSummary` Writes fields and boundary condition info for each patch at each requested time instance.