7.4 Sampling data

OpenFOAM provides a set of sampling function objects to sample field data, either through a 1D line for plotting on graphs or a 2D plane and 3D surfaces for displaying as images. Each sampling tool is specified in a dictionary either in the main functions dictionary of the controlDict file, or separate files in the case system directory. The data can be written in a range of formats including well-known graphing packages such as Grace/xmgr, gnuplot and jPlot.

The plateHole tutorial case in the $FOAM_TUTORIALS/stressAnalysis/solidDisplacementFoam directory also contains an example for 1D line sampling:

```
    setConfig
    {
        axis        y;
    }

    // Must be last entry
    #includeEtc "caseDicts/postProcessing/graphs/graph.cfg"
}

// ************************************************************************* //
```

Keyword	Options	Description
interpolationScheme	cell	Cell-centre value assumed constant over cell
	cellPoint	Linear weighted interpolation using cell values
	cellPointFace	Mixed linear weighted / cell-face interpolation
	pointMVC	Point values only (Mean Value Coordinates)
	cellPatchConstrained	As cell but uses face value on boundary faces
setFormat	raw	Raw ASCII data in columns
	gnuplot	Data in gnuplot format
	xmgr	Data in Grace/xmgr format
	jplot	Data in jPlot format
	vtk	Data in VTK format
	ensight	Data in EnSight format
	csv	Data in CSV format
surfaceFormat	null	Suppresses output
	foamFile	points, faces, values file
	dx	DX scalar or vector format
	vtk	VTK ASCII format
	raw	xyz values for use with *e.g.*gnuplotsplot
	stl	ASCII STL; just surface, no values
	ensight	EnSight surface format
	boundaryData	A form that can be used with timeVaryingMapped boundary conditions
	starcd	Nastran surface format
	nastran	
fields	List of fields to be sampled, *e.g.* for velocity U:	
	U	Writes all components of $\mathbf{U}$
sets	List of 1D sets subdictionaries — see Table 7.4	
surfaces	List of 2D surfaces subdictionaries — see Table 7.5 and Table 7.6	

Table 7.3: keyword entries for sampleDict.

The dictionary contains the following entries:

interpolationScheme
: the scheme of data interpolation;

sets
: the locations within the domain that the fields are line-sampled (1D).

surfaces
: the locations within the domain that the fields are surface-sampled (2D).

setFormat
: the format of line data output;

surfaceFormat
: the format of surface data output;

fields
: the fields to be sampled;

The `interpolationScheme` includes `cellPoint` and `cellPointFace` options in which each 多面体cell is decomposed into 特特拉赫德拉and the sample values are interpolated from values at the tetrahedra vertices. With `cellPoint`, the tetrahedra vertices include the polyhedron cell centre and 3 face vertices. The vertex coincident with the cell centre inherits the cell centre field value and the other vertices take values interpolated from cell centres. With `cellPointFace`, one of the tetrahedra vertices is also coincident with a face centre, which inherits field values by conventional interpolation schemes using values at the centres of cells that the face intersects.

The `setFormat` entry for line sampling includes a raw data format and formats for gnuplot, Grace/xmgr and jPlot graph drawing packages. The data are written into a sets directory within the case directory. The directory is split into a set of time directories and the data files are contained therein. Each data file is given a name containing the field name, the sample set name, and an extension relating to the output format, including .xy for raw data, .agr for Grace/xmgr and .dat for jPlot. The gnuplot format has the data in raw form with an additional commands file, with .gplt extension, for generating the graph. *Note that any existing* sets *directory is deleted when* sample *is run.*

The `surfaceFormat` entry for surface sampling includes a raw data format and formats for gnuplot, Grace/xmgr and jPlot graph drawing packages. The data are written into a surfaces directory within the case directory. The directory is split into time directories and files are written much as with line sampling.

The `fields` list contains the fields that the user wishes to sample. The sample utility can parse the following restricted set of functions to enable the user to manipulate vector and tensor fields, *e.g.* for `U`:

`U.component(`n`)`
: writes the nth component of the vector/tensor, $n = 0, 1 \ldots$;

`mag(U)`
: writes the magnitude of the vector/tensor.

The `sets` list contains sub-dictionaries of locations where the data is to be sampled. The sub-dictionary is named according to the name of the set and contains a set of entries, also listed in Table 7.4, that describes the locations where the data is to be sampled. For example, a `uniform` sampling provides a uniform distribution of `nPoints` sample locations along a line specified by a `start` and `end` point. All sample sets are also given: a `type`; and, means of specifying the length ordinate on a graph by the `axis` keyword.

Required entries

Sampling type	Sample locations	`name`	`axis`	`start`	`end`	`nPoints`	`points`
`uniform`	Uniformly distributed points on a line	●	●	●	●	●	
`face`	Intersection of specified line and cell faces	●	●	●	●		
`midPoint`	Midpoint between line-face intersections	●	●	●	●		
`midPointAndFace`	Combination of `midPoint` and `face`	●	●	●	●		
`cloud`	Specified points	●	●				●
`patchCloud`	Sample nearest points on selected patches	●	●				●
`patchSeed`	Randomly sample on selected patches	●	●				●
`polyLine`	Specified points (uses particle tracking)	●	●				●
`triSurfaceMeshPointSet`	Sample points on a triangulated surface	●	●				●

Entries	Description	Options	
`type`	Sampling type	see list above	
`axis`	Output of sample location	`x`	x ordinate
		`y`	y ordinate
		`z`	z ordinate
		`xyz`	xyz coordinates
		`distance`	distance from point 0
`start`	Start point of sample line	*e.g.*`(0.0 0.0 0.0)`	
`end`	End point of sample line	*e.g.*`(0.0 2.0 0.0)`	
`nPoints`	Number of sampling points	*e.g.*`200`	
`points`	List of sampling points		

Table 7.4: Entries within `sets` sub-dictionaries.

The `surfaces` list contains sub-dictionaries of locations where the data is to be sampled. The sub-dictionary is named according to the name of the surface and contains a set of entries beginning with the `type`: either a `plane`, defined by point and normal direction, with additional sub-dictionary entries specified in Table 7.5; or, a `patch`, coinciding with an existing boundary patch, with additional sub-dictionary entries specified in Table 7.6.

Keyword	Description	Options
`basePoint`	Point on plane	*e.g.*`(0 0 0)`
`normalVector`	Normal vector to plane	*e.g.*`(1 0 0)`
`interpolate`	Interpolate data?	`true`/`false`
`triangulate`	Triangulate surface? (optional)	`true`/`false`

Table 7.5: Entries for a `plane` in `surfaces` sub-dictionaries.

Keyword	Description	Options
`patchName`	Name of patch	*e.g.*`movingWall`
`interpolate`	Interpolate data?	`true`/`false`
`triangulate`	Triangulate surface? (optional)	`true`/`false`

Table 7.6: Entries for a `patch` in `surfaces` sub-dictionaries.