

Chapter 4

OpenFOAM cases

This chapter deals with the file structure and organisation of OpenFOAM cases. Normally, a user would assign a name to a case, *e.g.* the tutorial case of flow in a cavity is simply named `cavity`. This name becomes the name of a directory in which all the case files and subdirectories are stored. The case directories themselves can be located anywhere but we recommend they are within a *run* subdirectory of the user's project directory, *i.e.* `$HOME/OpenFOAM/${USER}-8` as described at the beginning of chapter 2. One advantage of this is that the `$FOAM_RUN` environment variable is set to `$HOME/OpenFOAM/${USER}-8/run` by default; the user can quickly move to that directory by executing a preset alias, `run`, at the command line.

The tutorial cases that accompany the OpenFOAM distribution provide useful examples of the case directory structures. The tutorials are located in the `$FOAM_TUTORIALS` directory, reached quickly by executing the `tut` alias at the command line. Users can view tutorial examples at their leisure while reading this chapter.

4.1 File structure of OpenFOAM cases

The basic directory structure for a OpenFOAM case, that contains the minimum set of files required to run an application, is shown in Figure 4.1 and described as follows:

- A **constant directory** that contains a full description of the case mesh in a subdirectory *polyMesh* and files specifying physical properties for the application concerned, *e.g.* *transportProperties*.
- A **system directory** for setting parameters associated with the solution procedure itself. It contains *at least* the following 3 files: *controlDict* where run control parameters are set including start/end time, time step and parameters for data output; *fvSchemes* where discretisation schemes used in the solution may be selected at run-time; and, *fvSolution* where the equation solvers, tolerances and other algorithm controls are set for the run.
- The **'time' directories** containing individual files of data for particular fields, *e.g.* velocity and pressure. The data can be: either, initial values and boundary conditions that the user must specify to define the problem; or, results written to file by OpenFOAM. Note that the OpenFOAM fields must always be initialised, even when the solution does not strictly require it, as in steady-state problems. The name of each time directory

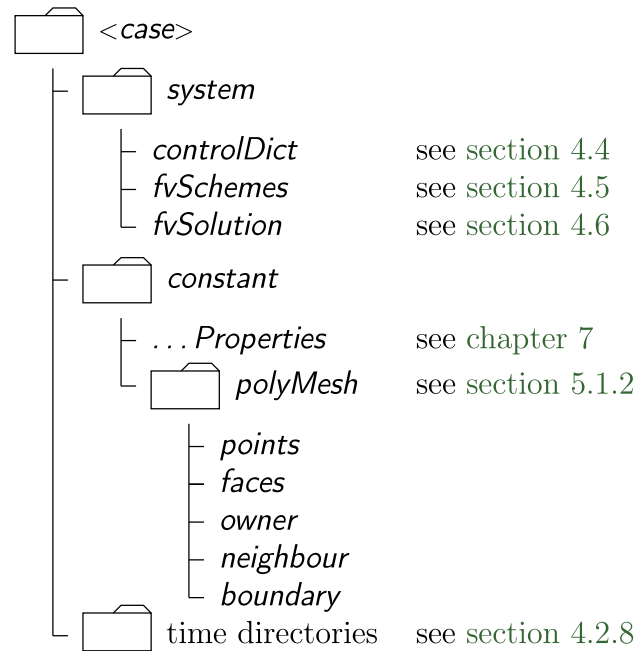


Figure 4.1: Case directory structure

is based on the simulated time at which the data is written and is described fully in section 4.4. It is sufficient to say now that since we usually start our simulations at time $t = 0$, the initial conditions are usually stored in a directory named 0 or $0.000000e+00$, depending on the name format specified. For example, in the `cavity` tutorial, the velocity field \mathbf{U} and pressure field p are initialised from files $0/U$ and $0/p$ respectively.

4.2 Basic input/output file format

OpenFOAM needs to read a range of data structures such as strings, scalars, vectors, tensors, lists and fields. The input/output (I/O) format of files is designed to be extremely flexible to enable the user to modify the I/O in OpenFOAM applications as easily as possible. The I/O follows a simple set of rules that make the files extremely easy to understand, in contrast to many software packages whose file format may not only be difficult to understand intuitively but also not be published. The OpenFOAM file format is described in the following sections.

4.2.1 General syntax rules

The format follows some general principles of C++ source code.

- Files have free form, with no particular meaning assigned to any column and no need to indicate continuation across lines.
- Lines have no particular meaning except to a `//` comment delimiter which makes OpenFOAM ignore any text that follows it until the end of line.
- A comment over multiple lines is done by enclosing the text between `/*` and `*/` delimiters.