freeCappuccino

User's Guide

Nikola Mirkov

January 23, 2018

Chapter 1

Introduction

The freeCappuccino is a three-dimensional fully unstructured finite volume code for Computational Fluid Dynamics which comes in serial and parallel version.

Moreover, freeCappuccino is a fortran library for manipulation of discrete tensor fields, defined over polyhedral meshes.

Name 'Cappuccino' encapsulates the idea that it is CAFFA (Computer Aided Fluid Flow Analysis) with some FOAM (Field Operation and Manipulation).

freeCappuccino is free both as a free coffee and as free speech.

The purpose of the code is to enable experimentation in discretizations and mathematical models. Therefore we have included many discretization options and we are in the process of implementation of many turbulence models and scalar equations for different variables.

The code is based on the reference:

N Mirkov, B Rašuo, S Kenjereš, On the improved finite volume procedure for simulation of turbulent flows over real complex terrains, Journal of Computational Physics, Vol. 297 (2015), pp.18-45.

The code allows meshes in OpenFOAM® polyMesh format. Only slight changes are necessary in the 'boundary' file.

Discretized equations are written in CSR (Compressed Sparse Row) format, which allows easy interface with many linear solver libraries, such as LIS. Also, one can use linear solvers provided in the code such as, Incomplete Cholesky Conjugate Gradient (ICCG), ILU(0) preconditioned Bi-Conjugate Gradient Stabilised method for non-symmetric systems, etc.

1.0.1 Requirements

The code is written in modern fortran, so you will need fortran compiler (e.g. gfortran). Code may be built to use external library for solution of sparse linear systems LIS or be used without it, just using built in linear solvers.

1.0.2 Getting started

Clone the git repository or download the zip archive of the most recent source, compile and build the code, and checkout the examples such as the lid-driven cavity, Pitz-Daily backward facing step, channel, and many more to come. In following we describe the sequence of commands to get started.

Compile the code.

```
cd src
make
```

Update PATH environment variable to point to point to bin/ subfolder.

```
cd \~
gedit .bashrc
```

Add following line at the end of .bashrc file.

```
export PATH=/where/is/it/freeCappuccino/bin/:$PATH
```

Move to example case folder.

```
cd ../examples/cavity/
```

Unpack the archive with mesh and initial conditions.

```
tar -zxvf cavity-setup.tar.gz
```

Check input file using text editor, and make changes if needed.

```
gedit input
```

Run simulation script.

```
./run
```

During and after run, you can plot residuals.

```
gnuplot plotResiduals
```

Open results files in Paraview. Results are in VTK-date-time folder.

```
paraview
```

You can skip this for provided examples but in general when working with meshes in polyMesh format you need to create cell conectivity. Cell connectvity data is found in polyMesh/cells file, which is needed to write .vtu files during the run.

Compile cellConnectivity utility program.

```
cd utilities
make
```

Before you run the simulation, run cellConnectivity.

```
cd /location/of/case/folder
cellConnectivity
```