To Do List and Open Issues

Javier Villarreal

1 Active Tasks

1.1 Data architecture and variables

The current task is to create all the variables necessary to hold the simulation data. So far, the only existing variables are those used to read in text file data.

Most of the C++ variables were defined as members in classes. The classes are

- Data
- FlowVars
- InternalBoundary
- Approximator
- Time
- ExternalBoundary
- SimFluid
- Domain
- GeneticAlgorithm

Only some classes are instantiated in the main code, though, with some only existing within other classes as members. The data structure is roughly

```
Data is never instantiated, serves as an "abstract" class
    Q[4] => air.[r,u,v,e]
Domain simRegion
    InternalBoundary body
    ExternalBoundary edge
SimFluid air
    FlowVars r,u,v,e
    Time t
Approximator solver
    GeneticAlgorithm ga
    dom => simregion
    air => air
```

where `=>` denotes pointer variables

A full listing of the classes and their respective class member variables (not member functions) can be found in `C++_classes.txt`. Of course, there are also variables defined within the scope of certain functions. Those are listed in `C++_functions.txt`.

So far, the truly necessary variables to fully describe compressible fluid flow are the primitive variables: ρ (density), u (x-velocity), v (y-velocity), E_t (total energy), and some dependent variables, like p (pressure), T (temperature), μ (dynamics viscosity), k (thermal conductivity). Because the code is non-dimensionalized, though, thermal diffusivity is instead represented by the Prandtl number, which is constant for calorically perfect gases.

1.1.1 Current Status

All the variables listed above have been defined, but rather than using a complex structure like that found in the C++ code, the variables and their derivatives are currently defined in simple arrays. Supplementary variables will be defined as needed.

1.2 Configuration File

In the C++ version of the code, parameters for how to run the code were defined in various places. Fluid properties were in SimulationValues.txt, genetic algorithm options were defined in the main program (SOMA.cpp), and the file directories and other constant parameters were defined in Data.cpp and .hpp. Rather than having the simulation configuration spread out over many different places, a single file will be used to house all necessary parameters.

A search of the literature will find that the CFD General Notation System (CGNS) exists for that purpose. It could potentially lead to a significant improvement towards the code's ease of use in the future.

For now, however, the configuration information will instead be created in a separate fortran module.

1.2.1 Note

There is no way for Fortran to know ahead of time whether a directory exists (e.g. trying to write output data into a not-yet-existent output directory). The only thing I can think to remedy this is to make sure the output directory exists while running the makefile, but that means the makefile has to be run with a command line argument to specify which geometry is being run ahead of time. Would that information then have to in turn be passed into the Fortran code?

2 Future Improvements

2.1 Build Options

Improvement could be made to the makefile (or a different build software) for having different builds in separate directories for debugg and release versions. There are also some softwares that automatically figure out dependencies, so they don't have to be explicitly stated in the makefile.

2.2 Pre-processing/Data structure

The way the data is stored in the input text files and in variables within the code is suboptimal. Changes could be made both within pre-processing (i.e. the Matlab codes that generate those files) and how the variables are declared in Fortran.

3 Completed Tasks

3.1 Build Options

The code is built with a basic Makefile. Compilation flags are hardcoded, so a `make clean` is needed anytime flags are changed.

3.2 Read in simulation data

The data files necessary to make the code run are different depending on whether the code is running a 2- or 3-dimensional problem. For 2D, geometries, the metadata files are:

- SimulationValues.txt (Mach, AOA, Re)
- Sizes.txt (# of domain, body, farfield, cloud, ghost, extrapolation, total nodes)

and the geometry data files are:

- x,y.txt (node coordinates)
- DX,DY.txt (DQ coefficients)
- EC.txt (extrapolation coefficients)
- Jd,Jb,Jf.txt (domain, body, farfield node indices)
- nxb,nyb.txt (body node unit normal vectors)
- nxf,nyf.txt (farfield node unit normal vectors)
- s11,s12,s21,s22.txt (Flow tangency matrices)