SOMA Documentation

Javier Villarreal

09/27/2021

1 Introduction

The purpose of this document is to keep track of the project to port the C++ SOMA code to Fortran.

2 C++ Code Architecture

The C++ code is broadly arranged in the following way:

1. Reads in simulation parameter text files and defines some constants.
2. constructs the objects that hold the variables: `Domain`, `SimFluid`, and `Approximator`
3. Reads in geometry-based data files to appropriate object variables.
4. Enforces boundary conditions and calculates derivatives on initial data.
5. Defines parameters for the genetic algorithm optimization code.
6. (optional) Reads initial values from text files and re-calculates BC's and derivatives.
7. Prints initial data to output text files.
8. Loops over time steps. (SOMA proper)
9. Prints latest values to output text files.

In the SOMA step, the code splits into one of two modes, explicit (using Runge-Kutta) or implicit (using RBF addition). Different mechanisms within the code based on convergence criteria or iteration counts switch the code between one mode or the other. Each one will be explaines in its own section.