Liam Cassidy

Software Development for Engineering Research

Modules and Functions

Spring 2019

TO DO:

-add tests for known functions

-Look at exact requirements for OpenFOAM input files

-Figure out how to create the wood boundaries and mesh them/assign them properties.

-I suspect these will become blocks, and then assigned a surface condition of some sort.

-Determine feasibility of the boundary/initial condition user input module

-Determine feasibility of the solver/processor user input module

**-Postprocessing:** how to store data, how to assess relative fitness, how to iterate, and how to pretty much do anything.

-

Contents

# Module 1: import\_geometry

The goal of module 1, *geom\_import*, is exactly as it sounds; import a user-defined 2D cookstove geometry from a local excel document. The module will require the following functions

## Function 1: locate\_geometry

The *locate\_geometry* function will prompt the user with a dialog window to navigate to the formatted excel file on their local machine.

The function will output the directory, and the filename of the data sheet.

**Potential problems:** I am running windows, but the dialog box functionality might change on Mac or Linux machines. This could cause issues and should be tested.

#### Function 1 Test: test\_locate\_geometry

## Function 2: extract\_geometry

The *extract\_geometry* function will extract x, y, and z coordinates of the data points included in the geometry excel sheet.

#### Function 2 Test: test\_extract\_geometry

# Module 2: format\_geometry

The goal of the *format\_geometry* module is to take the geometry data from module 1, and convert the format to the required format of the blockMesh files for OpenFOAM simulations.

## Function 3: format\_vertices

The *format\_geometry* function will use the x,y,z vertices from the *extract\_geometry* function and will format the location of the point in a format consistent with blockMesh files for OpenFOAM input decks.

--LC more on this

#### Function 3 Test: test\_format\_vertices

## Function 4: create\_wood

This function will create the circular concentric regions where the wood will be meshed and modelled.

#### Function 4 Test: test\_create\_wood

## Function 5: create\_primary\_inlet:

The goal of the *identify\_boundaries* function is to identify which vertices define boundaries of the domain.

#### Function 5 Test: test\_create\_primary\_inlet

## Function 7: create\_second\_inlet

#### Function 7 Test: test\_create\_second\_inlet

## Function 8: create\_outlets

#### Function 8 Test: test\_create\_outlets

## Function 9: create\_blocks

#### Function 9 Test: test\_create\_blocks

## Function 10: create\_edges

#### Function 10 Test: test\_create\_edges

# Module 3: mesh\_geometry

## Function 11: mesh\_blocks

This function meshes the geometry previously assigned. For now, the meshing details will be prescribed. With further work, these could become user inputs.

#### Function 11 Test: test\_mesh\_blocks

## Function 12: show\_mesh

This function uses the paraview software to visualize the mesh the user has created.

#### Function 12 test: test\_show\_mesh

# Module 4: assign\_conditions (THIS MIGHT NOT HAPPEN)

The boundary conditions, initial condition will be specified in the excel data sheet as well. This module will look to assign the boundary and initial conditions to the OpenFOAM input files.

Note: this might not end up happening; due to the complexity of the postprocessing efforts, I might simply assign these conditions without allow user input (for now).

-This module would import and format initial and boundary conditions, write them to file.

# Module 5: assign\_models (THIS MIGHT NOT HAPPEN)

The computational model (physical approximations) would be in the excel sheet. The import, format, and assignment of these choices would be similar to the geometry module.

Note: this might not end up happening; due to the complexity of the postprocessing efforts, I might simply assign these conditions without allow user input (for now).

-This module would import and format solver details

# Module 6: run\_simulation

This module simply runs the simulation that was setup

## Function XX: simulate\_model

This function commands openfoam to

# Module 7: write\_results

# Module 8: compare\_results

# Module 9: modify\_geometry