Software Development for Engineering Research 00 (2019) 1–6

StoveOpt: Biomass Cookstove Optimization Tool

Liam J. Cassidy

2000 SW Monroe Ave, 342 Rogers Hall, Covallis, OR 97331

Abstract

Keywords:
Software Development, Biomass, Cookstove, Computational Fluid Dynamics

~~1. Introduction and Motivation~~

In recent years, numerous research studies have identified a compelling case for desiging and distributing cleaner burning biomass cookstoves for low resource communities. Household air pollution, as a product of poor combustion efficiency of existing cooking technology, can cause chronic respiratory conditionsand has been linked to nearly four million premature deaths annually [REFERNCE THE CLEAN COOKING ALLIANCE PAGE–OPEN IN BROWSER]. Additionally, using biomass as a fuel source for existing inefficient cooking technology contributes significantly to global black carbon emission, which (LC PUT SOMETHING JUICY HERE FROM BOND REPORT). Identification of the pervasive lack of clean cooking technology has led to numerous investigations regarding the improvement of existing cooking technology.

Research studies in the recent past have investigated the state of clean cooking development from social, economic, and technological perspectives. BLURB ABOUT STATE OF SOCIAL. BLURB ABOUT STATE OF ECONOMICS. Within the clean cookstove industry, technical investigations focus primarily on understanding how to improve biomass combustion products and thermal efficiencies of cookstove technology.Research groups around the globe have contributed ~~to the library of engineering evaluations of existing~~ cookstoves by sharing model details and results of computational simulations. Many existing computational models, however, draw physical conclusions based on individual cookstove configurations leaving a vacancy in understanding for numerous different cookstove designs. The impact of producing a computational model of a biomass cookstove could be greatly improved by allowing designers to simulate a variety of cookstove configurations by way of a user-friendly open source software package. Many (ALL MAYBE?) existing

The intent of this paper is, first, to provide a review of existing computational modelling and software development efforts within the cookstove industry. Additionally, the paper presents version 1 (LC COME BACK HERE-is it v 1?) of the StoveOpt (LC MAKE SURE CITATIONS ARE SOLID WHEN REFERRING TO OWN SOFTWARE) software package; an open source biomass cookstove computational fluid dynamics (CFD) simulator with built in optimimzation functionality. Specifically, the paper will discuss on the methodoly used in the CFD simulations and the implementation of the software (design, layout, functionality, etc.). The paper will share results from a preliminary case, draw conclusions about the package, and establish future work. (THIS IS KIND OF WORDY AND RUINS THE NARRATIVE–LOOK BACK HERE AND REVISE HEAVIL)

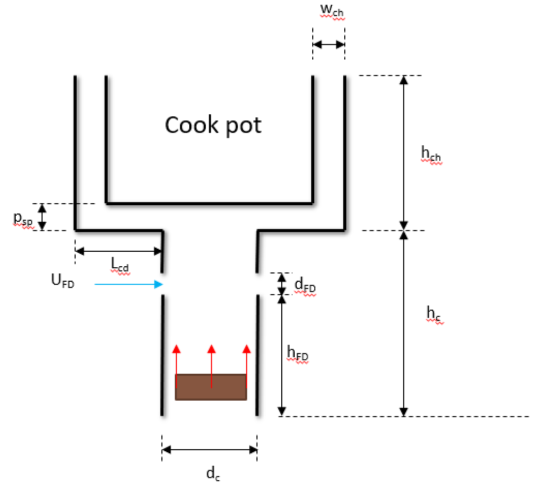
Email address: cassidyl@oregonstate.edu (Liam J. Cassidy)

2. Methodolgy

The StoveOpt software package convert user-defined geometric parameters to create input files compatible with OpenFOAM version 6, an open-source CFD package commonly used by engineers and scientists within academia and various other industries (LIAM GO REFERENCE THE SOFTWARE FROM OPENFOAM MAIN PAGE). Analyzing results of simulations of the user-defined geometry at various secondary air flow bulk velocites automates the creation of new text based OpenFOAM input files via a simple optimization approach. Cases are created and analyzed in an iterative fashion until an optimal secondary air flow velocity is recovered from the design space. A detailed discussion of the full methodology is as follows.

2.1. *Geometry*

The robust quality of the StoveOpt software is boasted by the abilty for a user to define any cookstove geometry, allowing for a potential interpretation and optimization of any stove design. The initial geometry definition is performed by editing a formatted Excel (REFERENCE EXCEL??) file existing within the *stovegeom* folder within the StoveOpt directory. The specific parameters required to fully define the geometry are presented below in Figure 1 and Table 1 (LEE FIGURE OUT HOW TO DO FIGURES BETTER).



Cookstove Geometry

Geometry Parameter Table

Parameter	Variable
Combustion chamber diameter	d$_c$
Combustion chamber height	h$_c$
Secondary inlet height	h$_{FD}$
Secondary inlet diameter	d$_{FD}$
Channel width	w$_{ch}$
Channel height	h$_{ch}$
Cone deck length	l$_{cd}$
Pot spacing	p$_{[}sp$

2.2. *Mesh*

All mesh details for OpenFOAM CFD simulations are defined in the *blockMeshDict* file within the *system* folder of a case directory. OpenFOAM meshes are created by defining hexehedral blocks within using vertice definitions. The StoveOpt backend uses the user-defined geometry to create within the domain. The vertices written to the *blockMeshDict* file are used to create the various blocks of the cookstove model.

To validate the default level of mesh refinement, grid convergence studies are underway for the test case discussed in next sections. (LIAM REVISIT THIS AND POTENTIALLY DO THE ACTUAL GCI)

2.3. *Auxilary Conditions*

FIGURE XX below represents the computational domain and will be used to discuss boundary and intial conditions. The cookstove primary inlet is modelled as a constant bulk flow velocity of 2.5 m/s composed of room temperature air composed of 23 percent O_2 and 77 percent N_2 by volume; the primary inlet flow velocity was computed based on air mass flow rates predicted by existing CFD work (LEEE REFERENCE THE UW ARTICLE... CAN I THOUGH?). The fuel zone was assumed to have a constant release of 100 percent methane from each edge. Although this is an inaccurate representation of solid-fuel combustion pertaining to cookstoves, this assumption has far-less complexity than wood combustion, and was deemed reasonable for initial development. The secondary air had the same composition of the primary inlet, however, the bulk flow velocity, although a constant value for an individual simulation, was variable among the different simulation cases; the secondary air flow velocity is the design variable for the optimization and, therefore, is not held constant. For the purposes of this analysis work, the secondary air bulk flow velocity ranged from 25 m/s and 150 m/s; this range was chosen to include the theoretical air-fuel ratio for stoichiometric methane-air combustion. The cookstove outlets were each modelled as atmospheric pressure outlets. The entire domain is assumed to be at a uniform 300 K intially.

2.4. Physical Models

The study calls on an existing OpenFOAM solver called *reactingFoam*, a transient CFD solver that includes combustion chemisty, heat transfer for a compressible fluid flow. The various methods used to model the flow physics are as follows; note, the following information was obtained directly from the OpenFOAM API guide (v1812) (LIAM GO GET A LINK).

The reacting flow is assumed to be laminar throughout the computational domain. This was chosen to limit the complexity and runtime of the first attempts of the software.

The thermophysical model used to describe heat transfer due to the reaction is called *psiReactionThermo*, which is a model for a reacting mixture based on the compressibility of the mixture, Ψ, given by:

$$\Psi = (\mathrm{RT})^{-}1$$

Where R is gas constant, and T is the temperature of the mixture. This thermophysical model serves as the basis for many of the OpenFOAM combustion solvers. The mixture is assumed to behave as an ideal gas. The transport model used is based on Sutherland's law, which computes dynamic viscosity, μ, based on the absolute temperature of a mixture T, the Sutherland coefficient A_s, and Sutherland Temperature T_s.

$$\mu = (\mathrm{A}_s \sqrt{T})/(1 + T_s/T)$$

For the purposes of this work, Sutherland coefficient and Sutherland temperature were assigned values of$1.67x10^{-6}$ and $170.67K$ for each specie in the mixture. Specific heat capacity, c_p was computed as a function of temperature T, based on sets of coefficients from NIST JANAF tables for each of the species included (LC REFERENCE HERE).

$$c_p = R((((a_4T + a_3)T + a_2)T + a_1)T + a_0)$$

Chemistry is modelled using backward (implicit) Euler method. The method was chosen to maintain stability for the stiff problem. (LC MORE DEVELOPMENT NEEDED HERE) The combustion process is assumed to be laminar for the full course of the simulation (LC MORE DEVELOPMENT NEEDED HERE).

2.5. Temporal Schemes

The solver uses the first order implicit Euler method for moving forward in time,

$$(\partial(\phi))/(\partial(t)) = (\phi - \phi^o)/(\triangle(t))$$

Where ϕ represents the variable from the general transport equation, △(t) is the time time step, and ϕ^o is the general transport variable value from the previous time step.

2.6. Spatial Schemes

In order to approximate cell-based quantities at cell interfaces, a variety of spatial interpolation schemes were used. Gradients terms within the general transport equation were evaluated using second order central differencing (LC MAKE AN EQUATION FROM THE CFD BOOK PROBABLY). For example, the approximations for gradients of ϕ_P with neighbors ϕ_E, ϕ_W, ϕ_N, and ϕ_S would be represented as,

$$\nabla(\phi_P) = (\phi_E - \phi_W)/(\triangle(x)) + (\phi_N - \phi_S)/(\triangle(y)$$

Where △x and △y are grid spacing in x-coordinate and y-coordinate directions, respectively.

Divergence terms from the general transport equation are approximated using OpenFOAM's *limitedLinear*, which uses a bounded second order central differencing method (LC MIGHT NEED TO ADD MORE HERE LATER ON).

The Laplacian terms, similar to the gradient terms, are approximated using second order central differencing of the form:

$$\nabla^2(\phi_P) = (\phi_E - 2\phi_P + \phi_W)/(\triangle(x)^2) + (\phi_N - 2\phi_P + \phi_S)/(\triangle(y)^2)$$

The spatial schemes are each second order accurate.

2.7. Optimization

The optimization process works to identify an optimal average secondary air flow velocity based on the average cook pot temperature. The optimization process, therefore, involves the secondary air flow velocity as the single design variable, and the average cook pot temperature as the single performance metric.

The optimization process was designed based on the assumption that the temperature of the cookpot as a function of secondary air flow velocity would behave qualitatively as the ambient temperature varies with respect to the air-fuel ratio for methane-air combustion. For the purpose of illustration, the optimization approach assumes qualitative behavior as depicted by FIGURE XX.

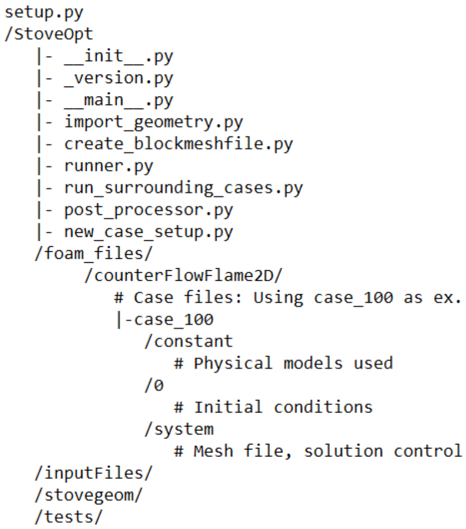
Adiabatic Flame Temperature

Given the results of the initial five simulation cases, the optimization algorithm results in four new cases. Because, theoretically, the optimal flow velocity might exist outside of the range of the first five cases, then the algorithm must have robust functionality. A schematic of the optimization algorithm is presented below in FIGURE XX.

3. Implementation

3.1. Software Layout

The StoveOpt software contains a collection of modules, which are called within a the main program script. Additionally, the package contains a test suite compatible with Pytest and an input file which is to be included as a required command line argument. FIGURE XX below shows a general structure of the software package. Table XX below presents responsibilities of the modules within the package, as well the contents of the most important subdirectories (referenced from FIGURE XX) within the package..



```
setup.py
/StoveOpt
   |- __init__.py
   |- _version.py
   |- __main__.py
   |- import_geometry.py
   |- create_blockmeshfile.py
   |- runner.py
   |- run_surrounding_cases.py
   |- post_processor.py
   |- new_case_setup.py
   /foam_files/
        /counterFlowFlame2D/
           # Case files: Using case_100 as ex.
           |-case_100
              /constant
                 # Physical models used
              /0
                 # Initial conditions
              /system
                 # Mesh file, solution control
   /inputFiles/
   /stovegeom/
   /tests/
```

Package Structure

PUT TABLE HERE

3.2. Functionality

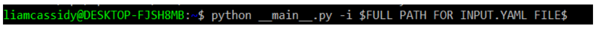
3.3. Dependencies

The StoveOpt software depends on various python packages, and additional software. For general navigation and file writing, os, sys, and shutil libraries are required. For array operations, Numpy is used. In order to use the input file and command line arguments, Argparse and Yaml packages are required. matplotlib is used for visualizing data. Lastly, the runner.py module, which is still in development stages, calls upon PyFoam, a python library used to control OpenFOAM simulations. In the current stage of the project, users must have a linux subsystem to run simulations; during development, a Ubuntu terminal was installed for this purpose.

3.4. Installation

3.5. Running the Software

To run the software, first, the user must add the filename and directory of the user-defined stove geometry workbook to the *input.yaml* file within the *inputFiles* directory. Subsequently, users should open a command prompt, and navigate to the StoveOpt directory. The software can then be initiated by calling the main script with the path of the input file as a required argument:



```
liamcassidy@DESKTOP-FJSH8MB:~$ python __main__.py -i $FULL PATH FOR INPUT.YAML FILE$
```

Command Line Instruction

Following the intial run, five OpenFOAM case files will be written. At the current stage of the software, the user then must manually pre-process and run within a Linux-Ubuntu command window. To set up the mesh, users should navigate to a case directory within */foamfiles/counterFlowFlame2D*; running a *dir* command within a case directory should show *0*, *constant*, and *system* folders. Once at this location, the user should write the command *blockMesh* in the Ubuntu command window; this calls an OpenFOAM script to create the meshed file. Once the mesh is complete, the user will see a success message, and then should run the command *reactingFoam*. This begins the simulation as dictated by the control files. Running 2-3 cases in parallel is recomended to use computational time economically; multiple cases can be run by openning additional Ubuntu windows, and following the simulation process described above.

CONSIDER MAKING A LITTLE TABLE FOR INSTRUCTIONS

4. Test Case

A test case was run in order to evaluate the ability of the software package and understand necessary future work. The inputs used to run the case are shared below.

4.1. Geometry

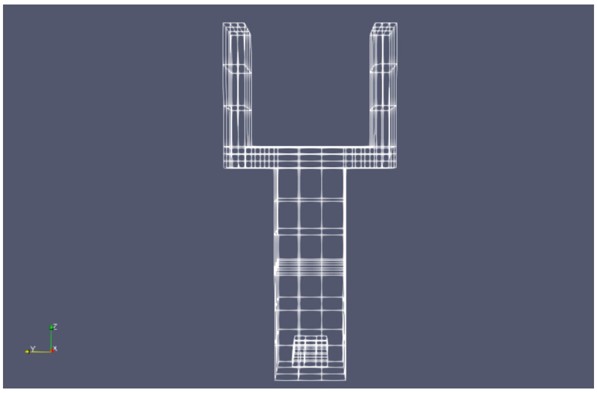
The geometry of the test case is presented below in TABLE XX.

Parameter	Value	Unit
Combustion chamber diameter	10	cm
Combustion chamber height	29.1	cm
Secondary inlet height	15.5	cm
Secondary inlet diameter	1	cm
Channel width	3	cm
Channel height	20	cm
Cone deck length	7	cm
Pot spacing	3	cm

Test Case Geometry

4.2. Mesh

A nonuniform mesh was created by assigning exactly three cells within each of the eleven blocks within the domain. The mesh resulted in 560 grid points and 297 cells. An image of the meshed test domain is shown below in FIGURE XX.



Test Case Mesh

4.3. Temporal Inputs

The time step was restricted to $10^{-3} secondstoprovidereasonabletemporalresolutionwithlowcomputati$

5. Results

The average pot temperatures for the first five simulations were computed and plotted as shown in FIGURE XX. The preliminary maximum average pot temperature was about 312 K, and was associated with a seconday air flow velocity of 50 m/s.

prelim_result.PNG

Pot Temperature vs Flow Velocity: Five Cases

Following the preliminary analysis, the post-processor file was run to identify the max and create the new cases for the purposes of the optimization study. The algorithm correctly identified the flow velocity of 50 m/s as the current maximum, and created four new surrounding cases for the next round of simulaitons.

The average pot temperature was plotted once again relative to the secondary air flow velocity; results are shown below in FIGURE XX.

second_runs.PNG

Pot Temperature vs Flow Velocity: Nine cases

After post-processing the second batch of simulations, a new optimal air flow rate was identified for the case of 52.8 m/s bulk flow velocity.

CREATE A TEMP CONTOUR OF THE 50 CASE FOR COMPARISON TO THE 52 CASE. THIS WOULD BE A GOOD RESULT TO SHARE

6. Conclusions

Reasonable qualitative results Algorithm correctly creates cases near maximum cases in the design space for a single design variable Firewood combustion not modelled Intensive computationally

7. Future Work

GCI Axisymmetric Optimization alg for multiple design vars Improve physical relevance (biomass, SS sim or extend time frame)