NASA/TM—2014-218387

NASA One-Dimensional Combustor Simulation—User Manual for S1D_ML

Thomas J. Stueber and Daniel E. Paxson
Glenn Research Center, Cleveland, Ohio

October 2014

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include creating custom thesauri, building customized databases, organizing and publishing research results.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at *http://www.sti.nasa.gov*

- E-mail your question to *help@sti.nasa.gov*

- Fax your question to the NASA STI Information Desk at 443–757–5803

- Phone the NASA STI Information Desk at 443–757–5802

- Write to:
  STI Information Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076–1320

NASA/TM—2014-218387

NASA One-Dimensional Combustor Simulation—User Manual for S1D_ML

Thomas J. Stueber and Daniel E. Paxson
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

October 2014

This work was sponsored by the Fundamental Aeronautics Program at the NASA Glenn Research Center.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA Center for Aerospace Information
7115 Standard Drive
Hanover, MD 21076–1320

National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Available electronically at http://www.sti.nasa.gov

NASA One-Dimensional Combustor Simulation—User Manual for S1D_ML

Thomas J. Stueber and Daniel E. Paxson
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract

The work presented in this paper is to promote research leading to a closed-loop control system to actively suppress thermo-acoustic instabilities. To serve as a model for such a closed-loop control system, a one-dimensional combustor simulation composed using MATLAB software tools has been written. This MATLAB based process is similar to a precursor one-dimensional combustor simulation that was formatted as FORTRAN 77 source code. The previous simulation process requires modification to the FORTRAN 77 source code, compiling, and linking when creating a new combustor simulation executable file. The MATLAB based simulation does not require making changes to the source code, recompiling, or linking. Furthermore, the MATLAB based simulation can be run from script files within the MATLAB environment or with a compiled copy of the executable file running in the Command Prompt window without requiring a licensed copy of MATLAB. This report presents a general simulation overview. Details regarding how to setup and initiate a simulation are also presented. Finally, the post-processing section describes the two types of files created while running the simulation and it also includes simulation results for a default simulation included with the source code.

Introduction

The National Aeronautics and Space Administration (NASA) Glenn Research Center (GRC), has a long and distinguished history of aeronautical propulsion research involving combustor concept and development of combustor component technology since its early days as part of the National Advisory Committee for Aeronautics (NACA) (Ref. 1). Combustion research at GRC addressing civilian aero-propulsion environmental concerns date back to the late 1960s (Ref. 1). Current objectives for aircraft propulsion systems to increase performance and decrease emissions has led to advanced combustor design research employing lean-burning (low fuel-to-air ratio) direct injection (LDI) systems for spraying fuel directly into the flame zone. As the combustion process gets leaner, the LDI system requires better fuel atomization for quick and uniform mixing with air to enable low flame temperatures and NO_x formation. However, lean-burning combustors have an increased susceptibility to high-pressure oscillations—thermo-acoustic instabilities (Ref. 2). Thermo-acoustic instabilities are much like sound waves with potential for high-pressure oscillations that can fatigue the combustor components and downstream turbine blades; hence, significantly decreasing the safe operating life of a combustor. Active Combustion Control (ACC) research considers modulating the fuel flow into the combustor to assert pressure oscillations out-of-phase with the thermo-acoustic instability; thus, suppressing the thermo-acoustic instability. The work being addressed in this paper is to promote research leading to a closed-loop control system to actively suppress the thermo-acoustic instability. Such a control system will employ a dynamic pressure sensor for feedback, an affecter to promote out-of-phase pressure oscillations in the combustor through fuel flow modulation, and an algorithm to control the affecter with respect to signals from the sensor (Refs. 3 to 6). To streamline development of closed-loop control algorithms, a

combustor simulation, identified in this paper as the simulated one-dimensional MATLAB-language-based S1D_ML code, has been developed.

A precursor one-dimensional code for simulating combustors (Ref. 7) has been composed to assess the feasibility of various control algorithm schemes (Refs. 8 to 13) for suppressing the thermo-acoustic instability. The simulation is a process that is initiated and runs in the Command Prompt window. The simulation source files are formatted and written as FORTRAN 77 source code and they must be compiled and linked to create the simulation executable file. This simulation will be referred to as the simulated one-dimensional combustor FORTRAN code (S1D_F) version. Modifications can be made to the S1D_F simulation source code to capture specific geometry and operating conditions of advanced combustor prototype. The simulation is coded to include the relevant physical features of the combustor and test rig and it yields data documents that can be reduced to reports depicting an instability behavior (Ref. 14).

To simulate a new combustor prototype with the S1D_F combustor code, actual process-defining source code modifications are required. This task can be challenging in that it requires familiarity with the source code and FORTRAN formatting standards. The work reported in this document describes how to work with a code written in a higher level language that can be compiled into a process representative of the FORTRAN coded process. The following are the two primary objectives for this task: First, make available to end users the source code for the one-dimensional combustor simulation process composed in a higher level programming language—MATLAB. This version of the code is identified as the simulation for a one-dimensional combustor coded using MATLAB software (S1D_ML). MATLAB is a higher-level language that employs both graphical- and text-based coding techniques with many features and toolboxes that are readily accepted by engineers. Source code modification can be made to the S1D_ML simulation using MATLAB programming tools. The second objective is to make available to end users a compiled executable that reads an input setup file to define combustor geometry and boundary conditions. This second objective will alleviate the need to own a copy of MATLAB to make changes to the process-defining source code and for running the S1D_ML simulation.

This report describes a general simulation overview of the S1D_ML process. Next, this report will describe the general process for initiating a simulation. The document continues with an explanation of the simulation setup file used by the S1D_ML code. This setup file is very similar to the text document used with the FORTRAN code with a few extra variables to serve as process defining parameters. Next, is a listing of process control flags (PCFs) included in the code for operating convenience and to decrease simulation process computational time. Next, is a high level presentation of the simulation process followed by a post-processing section that presents instructions for report generating using the simulation resultant files. Finally, a summary giving an overview of the contents of this document is presented.

Simulation Overview

The S1D_ML includes a default simulation setup embedded within the compiled process code. The default simulation will be used as an example for discussing set-up and running the S1D_ML process. The setup process identifies a simulation period and an incremental time step. The index variable k will define the k^{th} incremental time step for the process.

The simulation models a combustor by dividing it into several zones. The number of zones is without limit except for limitations on computational resources and time available for running the code. Each zone is defined to have uniform cross sectional area and is further divided into cells of equal axial length. One of the zones is where the flame is held and recirculation exists. This zone is identified as the Flame-Hold zone. More information pertaining to the Flame-Hold zone is given in the Test Article Definition

subsection in the Simulation Setup File section below. Within each cell, the simulation numerically integrates governing equations for a calorically perfect gas using a second-order MacCormack scheme (Ref. 15). Considering all of the zones together; the simulation will include n computational cells, an upstream boundary condition cell, and a downstream boundary condition cell for a total of n+2 cells in the simulation. The first cell is the upstream boundary condition, the second cell is at the combustor inlet, cell n+1 is at the combustor exit, and cell n+2 is the downstream boundary condition. Cell index numbers incrementally increase across zones—cell numbering does not restart at the entrance of each zone. Cells 1 and n+2 represent the termination of the computing domain. The following descriptions pertaining to the calculations and considerations for the zones and for each cell are as described in the Paxson papers (Refs. 7 and 15). The zones may be defined as partially opened, fully opened, or choked (e.g., constant mass flux). In either case, the code anticipates the flow direction and applies appropriate states. If outflow is anticipated, only the static pressure is imposed. Information pertaining to density, velocity, and reaction fraction are obtained via characteristic jumps across the adjacent numerical cell. If inflow is anticipated, total pressure and temperature and reaction fraction are imposed. The velocity quantity is obtained through iteration and characteristic jumps across the adjacent numerical cell. Furthermore, the reactant fraction can also be specified at an inlet, allowing the possibility of simulating premixed combustion. This would require the fuel source term, s_4, to be set to 0.0. Computational cells 2 through n+1 represent the simulated combustor. The index variable i identifies the i^{th} cell of the simulation.

The simulation variables are dimensionless in that they are normalized as described in Table 1. The notation for the variables in this document includes variables with prime marks to represent variables with dimensional units and variables with asterisk superscripts to represent normalization parameters. The length terms are normalized by the overall combustor length, L^* (ft). Velocity is normalized by a reference speed of sound value, a^* (ft/sec), which is based on a reference temperature, T^* (R). Likewise, pressure and density are normalized by reference values of p^* and ρ^*. Time is normalized by the characteristic wave transit time, L^*/a^*. The value for gamma was selected to be 1.3 for the default simulation because the mixture of gases in the combustion chamber is expected to result in a lower ratio of specific heats than pure air. The same argument is used when determining a suitable value for the characteristic gas constant.

TABLE 1.—DIMENSIONLESS TERMS AND REFERENCE VALUES THAT PERTAIN TO THE DEFAULT SIMULATION

[The value for a^* is a calculation based on the value of T^*, $\left(a^* = \sqrt{\gamma R' T^*}\right)$ where γ (1.3) is the ratio of specific heats, R' (53.5889 $\frac{\text{lbf ft}}{\text{lbm °R}}$) is the characteristic gas constant, and T^* is the reference temperature (1460 R) for the simulation.]

Dimensionless term	Dimensioned term	Physical narrative	Default normalization values	Relationship between dimensionless and terms with dimensions
p	P' (psia)	Pressure	p^* = 250 psia	$p' = pp^*$
T	T' (R)	Temperature	T^* = 1460 R	$T' = TT^*$
ρ	ρ' (lbm/ft^3)	Density	ρ^* = 0.460 lbm/ft^3	$\rho' = \rho\rho^*$
x	x' (ft)	Distance	L^* = 5.45 ft	$x' = xL^*$
u	u' (ft/s)	Velocity	a^* = 1809 ft/s	$u' = ua^*$
t	t' (s)	Time	----------	$t' = tL^*/a^*$
f	f' (Hz)	Frequency	----------	$f' = fu^*/L^*$
γ		Ratio	1.3	
	$R' \frac{\text{lbf ft}}{\text{lbm °R}}$	Gas const.	53.5889	

The non-dimensional equation of state is simply $p = \rho T$ and the non-dimensional speed of sound is $T^{1/2}$. Finally, the Reynolds number, Re^* is defined as $\rho^* a^* L^*/\mu$ where μ is molecular viscosity. The combustion (or reaction rate) model, R_t, is as described in Equation (1):

$$R_t = \begin{cases} K_0 \rho z\left(\zeta_1 - \zeta_2 z\right)\left(1 - \dfrac{T_{ign}}{T_i}\right); & T_i > T_{ign} \\ 0; & T_i < T_{ign} \end{cases} \tag{1}$$

Where K_0 is the reaction rate constant, ζ_1 and ζ_2 are constants defining the type of reaction. The following are appropriate values for defining (ζ_1, ζ_2): (1.0, 1.0) or (1.0, 0.0). T_{ign} is the ignition temperature and T_i is the dimensionless temperature in the i^{th} computational cell.

The set of governing equations that are iteratively solved are as described in the following source vector (Refs. 7 and 15):

$$\overline{S}\left(\overline{w}, x\right) = \begin{bmatrix} 0 \\ \dfrac{\partial}{\partial x}\left(\dfrac{\varepsilon_t}{\mathrm{Re}^*}\left(\dfrac{\partial u}{\partial x}\right)\right) + s_2 \\ \dfrac{\partial}{\partial x}\left(\dfrac{\varepsilon_t}{\mathrm{Re}^*}\dfrac{\partial}{\partial x}\left(\dfrac{u^2}{2} + \dfrac{T}{(\gamma - 1)\mathrm{Pr}_t}\right)\right) + q_0 R \\ \dfrac{\partial}{\partial x}\left(\dfrac{\varepsilon_t}{\mathrm{Re}^* \mathrm{Sc}_t}\left(\dfrac{\partial z}{\partial x}\right)\right) - R_t + s_4 \end{bmatrix} \tag{2}$$

Where ε_t is the turbulent viscosity ratio defined as the ratio of turbulent (eddy) to molecular viscosity, μ_t/μ. The values of Pr_t and Sc_t are the turbulent Prandtl and Schmidt numbers respectively. The z term is the mass fraction of reactant fluid containing chemical energy. The s_2 term is a "pink" noise source of momentum that is only implemented in the vicinity of the fuel injector. This term is intended to mimic the effects of shed vortices, shear layers, etc., that are seen on actual combustors and thought to be responsible for the typical "noise floor." The s_4 term represents additional fuel and is also only to be implemented at the fuel injector location. For this code, the addition of fuel is assumed to add no mass to the system. This point is consistent with the assumption that the mass flow rate of fuel is negligible compared to that of air. The q_0 term is defined as follows:

$$q_0 = \frac{h_f}{a^{*2}\left(a/f + 1\right)} \tag{3}$$

where the h_f and a/f terms in Equation (3) are the fuel lower heating value and the air-to-fuel ratio respectively. Finally, it is assumed in the code that turbulent diffusion is only significant in a region where combustion occurs. As such, the turbulent viscosity ratio distribution is prescribed such that it is zero outside of the Flame-Hold zone. Within the Flame-Hold zone, it rises rapidly from zero to a maximum. The eddy viscosity diffusion within the flame-Hold zone is described with more detail in the Simulation Setup File section and Test Article Definition subsection below.

The process is designed to periodically save information for post-processing report generation. The data saved can be reduced to create the following two types of graphical reports: Power Spectral Density (PSD) and a Wave pattern analysis (Wave). The PSD uses a Blackman Harris Fast Fourier Transform to illustrate the power spectral density of pressure at two select cells. The two select cells, identified as stations P1 and P2, are operator identified upon initiation of a simulation and the process for identifying P1 and P2 is discussed in the next section titled Initiating a Simulation. The Wave illustrates pressure and

velocity on contour plots with respect to location (cell number) and time. This second report type enables recognition of resonance patterns in the data set.

The next section will describe the necessary steps to initiate a simulation.

Initiating a Simulation

This document will present four methods for initiating the one-dimensional combustor simulation: The first three methods leverage the use of MATLAB and involve starting and running the simulation in the MATLAB environment. These methods require a licensed copy of MATLAB software. The code defining the S1D_ML process that is collected in files with .m extensions was composed using MATLAB version R2013b 64-bit. For running the code in MATLAB, version R2013b or later is recommended. The fourth method relieves the MATLAB license requirement by using a compiled copy of the code that can be started and run in the Command Prompt window.

Method one is for running the simulation, the S1D_ML.m script file, from within the MATLAB Command Window. This method requires all .m files supporting the S1D_ML.m file to be copied into the same directory as the S1D_ML.m file. The MATLAB environment permits a variety of ways to initiate and run a simulation defined by .m files. One such procedure is to navigate the MATLAB environment dialog window to the directory where the S1D_ML.m file resides, open the document, and then select "Run" to initiate the simulation. The process will start and present a "Key-in Process Information:" dialog box with a few fields populated with default responses. The operator can cancel, accept defaults, or make changes in the fields and then accept field entries to proceed with the simulation. Since the process running is identical regardless of initiation method, the process for populating and proceeding beyond this dialog box will be addressed after all four initiation methods have been presented.

Method two is very similar to the first method and includes an additional capability to employ process control flags (PCF). As was the case for the first method, this second method also requires a copy of the S1D_ML.m script file and the supporting .m function files. For this method, the process to start a simulation is to first navigate the MATLAB environment dialog window to the directory where the S1D_ML.m file resides. The second step is to key in "S1D_ML" at the MATLAB environment Command Window prompt. As mentioned above, an additional benefit to starting the simulation using this method is the capability to add PCFs to the simulation. For example, a save PCF can be included by entering the following line at the MATLAB Command Window prompt: "S1D_ML('Save')." The format includes everything between the double quotes, except for the end of sentence period. The 'Save' PCF will instruct the process to save the simulation results upon completion. More information pertaining to this feature and pertaining to other PCFs that are available is given below in the "Process Control Flags" section after the basic operation of the code has been presented. Upon simulation initiation, as was the case for method one, the process will present a "Key-in Process Information:" dialog box with a few fields populated with default responses.

Method three does not require a copy of the S1D_ML.m code or the supporting .m function files. To initiate a simulation, navigate the MATLAB environment dialog window to the directory that includes a copy of the compiled S1D_ML code. The compiled copy of the S1D_ML code has a .exe extension. Next, key in "!S1D_ML" next to the MATLAB prompt in the MATLAB environment Command Window. The format for this method includes everything between the quotes including the exclamation mark. This method can also include PCF strings. For example, the save PCF can be included by entering the following line at the MATLAB Command Window prompt: "!S1D_ML Save." This method does not require or accept the parenthesis or the single quotes that were presented with the second method. Upon simulation initiation, as was the case for the first and second methods, the process will present a "Key-in Process Information:" dialog box with a few fields populated with default responses.

Method four initiates the process in the Command Prompt window, requires a compiled copy of the S1D_ML code, and does not require a MATLAB license. For this method, if the user does not have a licensed copy of MATLAB, the compiled program must be linked with the MATLAB Compiler Runtime (MCR) library and the MCR must be available on the target computer where the executable is to be run. A copy of the MCR can be obtained and downloaded from the MathWorks website (Ref. 16[16]) and installed without requiring the procurement of a MATLAB license. Once that is completed, the process to initiate the simulation is to first navigate the Command Prompt window to the directory where the S1D_ML.exe file resides. Second, key in the S1D_ML characters next to the Command Prompt window prompt as described in the following example which includes employment of the save PCF: S1D_ML Save. The string of character entries for initiating the code for this method is very similar to the third method, less the leading exclamation point. Upon simulation initiation, as was the case for the first three methods, the process will present a "Key-in Process Information:" dialog box with a few fields populated with default responses.

Table 2 is included to summarize the four methods for initiating an S1D_ML simulation with the Save PCF feature employed for methods two through four—not available with method one as indicated in column four. The first column identifies each method type with a number. The second column identifies methods one through three requiring a licensed copy of MATLAB. The third column indicates method four can be employed with a licensed copy of MATLAB or with the use of the downloadable MCR. The fifth column identifies the environment the code operates in. Finally, the sixth column is an example for starting the process with two PCFs. Methods two through four do not require PCFs.

Regardless of the method used to initiate the S1D_ML simulation, the process starts with the following two steps: First, the process will present a "Key-in Process Information:" dialog box with a few fields populated with default responses as illustrated in Figure 1. The operator can cancel out of the program by using the mouse to select the "Cancel" button. Selecting "Cancel" will open another dialog box, "Exit Dialog," to confirm the instruction to quit the program. To confirm the intention to exit the simulation, use the mouse to select the "Yes" button; else, select the "No" button to revisit the "Key-in Process Information:" dialog box. Within the "Key-in Process Information:" dialog box, the operator can define the simulation time, output start time, perturbation level, data type, station P1, and station P2 by replacing the contents of the fields with desired values. The list of fields presented in the "Key-in Process Information:" dialog box are described in Table 3. Table 3 also reveals appropriate responses and a comment column that describes the sought after information and the general effect of the answers on the simulation. To proceed with the simulation, use the mouse button to select the "OK" button and the process will proceed to the second initialization step. For the second initialization step, the simulation will read process operating parameters that are either hard-coded into the process, a default simulation, or included in a separate document available for reading by the S1D_ML process.

TABLE 2.—FOUR METHODS FOR INITIATING THE S1D_ML SIMULATION
[The examples given in the last column for methods two through four also include use of PCFs. PCFs are not required for any method.]

Method	Requires MATLAB	Requires MATLAB or MCR	PCF available	Environment	Example with PCFs save and FI
1	x			MATLAB	Not Applicable
2	x		x	MATLAB	S1D_ML('Save_FI_90')
3	x		x	MATLAB	!S1D_ML Save_FI_90
4		x	x	Command Prompt	S1D_ML Save_FI_90

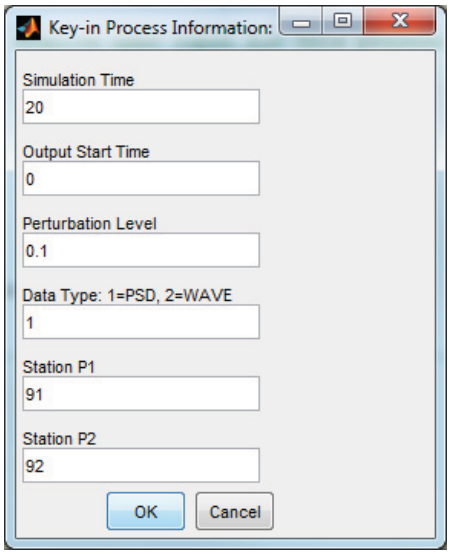

Figure 1.—Dialog box presented for operator input upon simulation initiation.

TABLE 3.—LISTING OF FIELDS IN "KEY-IN PROCESS INFORMATION:" DIALOG BOX FOR SETTING UP SIMULATION RUNTIME PROCESS

[Second column is the text located above the data entry fields, third column describes appropriate response to move forward with the simulation, and the fourth column includes comments regarding how the answered questions pertain to the simulation.]

Field number	Field text	Appropriate responses	Comments
1	Simulation Time	Numerical value greater than 0.0.	This value is T_{max}, which is the simulation time (dimensionless seconds). Dimensional term given in Table 1.
2	Output Start Time	Numerical value greater than or equal to 0.0 and less than the value entered for Simulation Time.	This value is T_{out}, which is the simulation output recording start time (dimensionless seconds).
3	Perturbation Level	Numerical value greater than or equal to 0.0.	Amplification factor applied to a ±1.0 pseudorandom number generator that changes the momentum coefficient of the terms in the momentum region during the simulation.
4	Data Type: 1=PSD, 2=WAVE	1 or 2	If a 1 (PSD) is entered, then the Station P1 and Station P2 fields are relevant. If a 2 (Wave) is entered, then the Station P1 and Station P2 fields are irrelevant.
5	Station P1	Integer ranging from 1 to n+2 (any cell in the simulation).	One of two cells in simulation where cell states are saved and reduced later to a power spectral density report.
6	Station P2	Integer ranging from 1 to n+2 (any cell in the simulation).	One of two cells in simulation where cell states are saved and reduced later to a power spectral density report.

Simulation Setup File

Overview

After receiving acceptable answers from the operator as described in Table 3, the process establishes simulation process parameters—initial conditions, boundary conditions, and process maintenance settings. This information can be found in any of the following three locations: hard-coded into the process (default), in a document formatted to open for reading and editing using Microsoft Excel software

(S1Dsetup.xlsx), or a binary file (S1Dsetup.s1d) formatted for fast reading by the S1D_ML process. Availability of licensed MATLAB software and the appropriate MATLAB .m text documents will enable making changes to the default simulation. The parameters defining the default simulation that are hard-coded into the S1D_ML process will be used if the "S1Dsetup.xlsx" document is not available in the root directory of the executable file. The benefit of having the .xlsx formatted document is that the values to initialize the simulation process can easily be modified using Microsoft Excel software to change numerical values in the spreadsheet cells. However, reading the .xlsx document is computationally very slow compared to the time required to read a binary document. Therefore, after the S1D_ML process reads an S1Dsetup.xlsx document, it will create a binary document composed of information read from the S1Dsetup.xlsx document that will be available for future simulation runs. Also included in the binary file is a time stamp value representative of the S1Dsetup.xlsx "Date modified" time stamp. When initiating a simulation, reading the binary file is much faster than reading the S1Dsetup.xlsx file; therefore, the process will read and use the data saved in the binary file if the binary file is current. If the value of the saved date variable in the binary file matches the S1Dsetup.xlsx "Date modified" time stamp; then, the binary file is assumed to be current. Therefore, any modifications to the .xlsx document will void usefulness of the .s1d document until the simulation has been run; when it will create a new copy of the .s1d document that captures the current S1Dsetup.xlsx time stamp. This section describes the initialization information that can be found by the S1D_ML process in any of the three locations by presenting the S1Dsetup.xlsx document.

Opening the S1Dsetup.xlsx file using Microsoft Excel software reveals a spreadsheet that is segmented into six partitions with the following tags that should remain in column A and that should not be changed: This document provides initialization data for the S1D_ML.m simulation, Test Article Definition, Experiment Description, Filtering, Combustion Process, and Initial Conditions. Under the first tag, "This Document … simulation," is some notes with general information regarding the color coded scheme used to describe the contents of the document. The font color coding scheme is only for user reading convenience and the S1D_ML process does not use this information. All information included below the "Test Article Definition" tag defines the simulation—geometry, boundary conditions, and initial conditions. The tags are used by the simulation to identify reference row pointers for each of the document's other five partitions. When the simulation reads the S1Dsetup.xlsx document, it will search all the rows in column A for each of the tags. The value of a simulation variable identified as *pntr* is set to the row number for each tag. For example, when the process scans column A for the tag "Test Article Definition," the value of the *pntr* variable will be set to eight. Because the .xlsx document is set to have the "Num Zones" information located in the row immediately below the tag, the S1D_ML process will read the value located in row pntr+1 to identify the value for the number of zones in the simulation—for this variable, the value is expected to be in column C. Referenced row pointers, as opposed to dedicated row positions, were helpful for organizing the .xlsx document and it enables an opportunity to add variables in the future at the end of each partition without rewriting all the row positions.

The S1Dsetup.xlsx document includes the value of one inserted in the cell defined by the intersection of row one and column L (Excel cell identification nomenclature of L1). This numerical entry should not be changed or moved without careful consideration of the consequences. The numerical entry in cell L1 will instruct the simulation to read all the rows with populated cells from column A to column L. The value in cell L1 is of no significance so long as the entry is a numerical entry. If cell N1 has an entry, then columns M and N will also be read which will add time and consume memory resources. This part of the simulation can be improved by deleting the contents of L1 and inserting a value of one in cell H1—only read the contents of columns A through H instead of columns A through L.

To help explain some of the values entered into the .xlsx document, an illustration of the default simulation is given in Figure 2. Figure 3 is another illustration of the default geometry that also depicts the computational zones and state cells. The combustor illustrations in Figure 2 and Figure 3 include three

computational zones, an upstream boundary condition, and a final downstream boundary condition. The three computational zones are further segmented into state cells—state cells not illustrated in Figure 2. For example, the default simulation employs 200 state cells (n=200) and two boundary cells. The simulated combustor illustrated in Figure 2 has zone lengths of 28.4(in.) + 1.34(in.) + 35.7(in.) = 65.44 in. long. Each state cell will represent 65.44(in)/200 = 0.3272 (in./cell) of the combustor. Therefore, the first zone will include 28.4(in.)/ 0.3272 (in./cell) = 86.8 cells. The 86.8 value is rounded to identify the first zone as being defined by state cells numbered 2 to 88 (87+1). Likewise, the next zone will contain four cells numbered 89 to 92. The final zone will include the rest of the 200 state cells—93 through 201. The upstream boundary condition will be cell one and the downstream boundary condition will be cell 202. The flow state of the state cells are defined by the following six dimensionless parameters: Pressure (P), density (W), flow velocity (U), temperature (T), source vector (S), and reactant fraction (Z). All state cells within a given computational zone have a consistent cross-sectional area. A change in flow cross-sectional area may exist only between zones—only two cross-sectional area changes in this example simulation. The states of the upstream and final downstream cells are defined by the upstream and final downstream boundary conditions.

Information on the variables available to the user by use of the .xlsx document is captured in Table 4. The contents of Table 4 include default values for each user definable process variable, a corresponding PCF name (if one exists), and a brief description of the variable purpose. The values in Table 4 are dimensionless and the relationships between the dimensionless variables and dimensioned values are given in Table 1. The remaining five subsections describe the Table 4 entries and the information included in the S1Dsetup document under each tag.

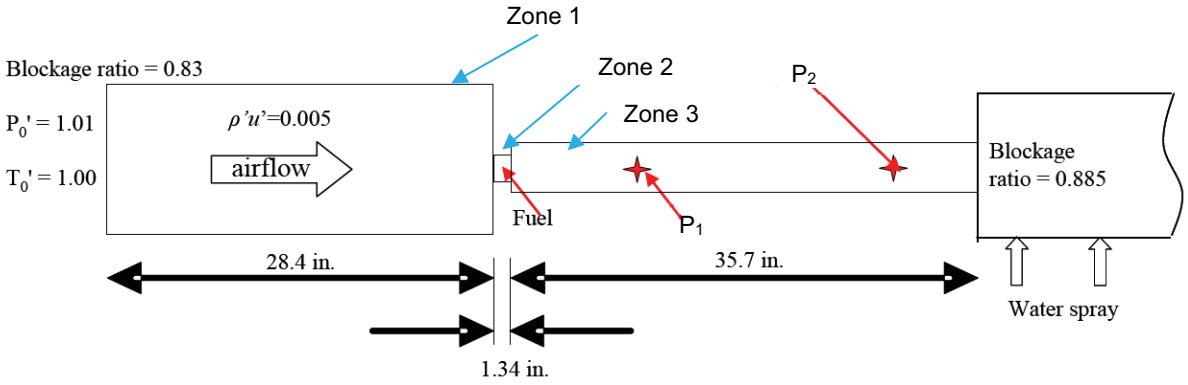

Figure 2.—Default Combustor Simulation Configuration. This illustration is not drawn to scale.

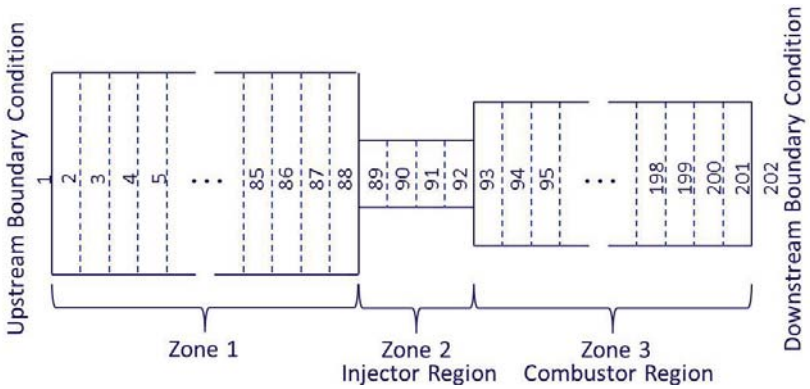

Figure 3.—Depiction of default simulation combustor segmented into three zones and further segmented into 202 cells of equal length. This illustration is not drawn to scale.

TABLE 4.—PROCESS PARAMETERS AVAILABLE FOR THE USER TO DEFINE IN THE .XLSX DOCUMENT
[Temperature and pressure values are set dimensionless by using the terms in Table 1.
The gray dividers are included to identify beginning and end of sections in the .xlsx document.]

Simulation variable	PCF name	Default value	Definition
Num Zones	NSN	5	Three mid zones and single cells upstream and downstream.
Lengths (in.)		28.4	Actual measurements for the lengths of combustor zone 1.
		1.34	Actual measurements for the lengths of combustor zone 2.
		35.7	Actual measurements for the lengths of combustor zone 3.
Area (in^2)		113.0973	Cross-sectional area for simulated combustor zone 1.
		3.332916	Cross-sectional area for simulated combustor zone 2.
		16	Cross-sectional area for simulated combustor zone 3.
Momentum Region		89	Cells where computational noise is introduced
Flame Hold	FH	98	Entry identifies the Flame-holding cell. The Flame-Hold zone is identified as the zone that includes the Flame-holding cell and further defines the recirculation region.
Fuel Injection	FI	90	Cell number location for simulated fuel injection
Frequency (Hz)		5000	Data output frequency
A* (ft/sec)		1809	Speed of sound
Convergence		0.00001	Acceptable convergence error
gamma		1.3	Ratio of specific heats
DTI		0.0025	Dimensionless simulation incremental numerical time step
S14		0.25	Constant for artificial viscosity.
LSSC	LSSC	35.8	Reaction mean flow rate if fuel is injected other than at inlet
df_H		-0.9977	Denominator filtering coefficients for high-pass filter.
cf_H		0.9988	Numerator filtering coefficients for high-pass filter.
df_L		0.8882	Denominator filtering coefficients for low-pass filter.
cf_L		0.0559	Numerator filtering coefficients for low-pass filter.
EddyT	ET	0.0015	Ratio of ε_t/Re* in differential equation two.
Schmit		1	Turbulent Schmidt Number
Prntt		1	Turbulent Prandtl Number
Zeta1		1	ζ_1 in calculation for R_t
Zeta2		0	ζ_2 in calculation for R_t
Q0	Qnot	4.2	q_0 in Equation (5)
K0	Knot	30	Reaction Rate Constant
TC0		1.3	Ignition Temperature (T_{ign})
Pstage	Pstage	0.78534	Static pressure at the outlet of the combustor
Tstage	Tstage	2.3	Temperature at the outlet of the combustor. Only used when inflow occurs.
ZE		0	Reaction fraction at right (outlet) end of combustor. Only used when inflow occurs
Pstagl	PstagL	1.005	Total pressure at left end of the combustor for open end boundary condition. Mass flux for constant flux boundary condition
Tstagl	TstagL	1	Total temperature at left end of combustor. It is assumed as a stagnation value
ZI		0	Reaction fraction at left (inlet) end of combustor
ARE		0.115	Open area of outlet end of combustor divided by nominal cross section.
ARI		0.17	Open area at inlet end of combustor divided by nominal cross sectional area.
ZZP1		0	Fuel flow rate variation from mean
P			Array of initial pressure values for each cell
W			Array of initial density values for each cell
U			Array of initial flow velocity values for each cell
Z			Array of initial reactant fractions for each cell

Test Article Definition

For reading the values pertaining to these variables, the value of the *pntr* variable is set to the row number with the "Test Article Definition" tag in column A. This section is further partitioned into paragraphs identified with bold italicized text tag labels as found in the .xlsx document in column A.

Num Zones: As per the illustrations in Figure 2 and Figure 3, the number of zones (Num Zones) in the combustor test article is five—three zones and both upstream and downstream boundary conditions. A value of five is entered into the spreadsheet cell identified by the intersection of column C and the row *pntr+1*.

Lengths (in.): The *pntr+2* row, "lengths (in)," describes the lengths of each zone. Since the default simulation has five zones, the lengths of the middle three zones are entered into this row of cells in columns C, D, and E—28.4 (in.), 1.34 (in.), and 35.7 (in.) as described in Figure 2. If the simulation had six zones, then a value of six would be entered into the *pntr*+1 row column C and four columns in row *pntr*+2 would be populated with zone length information—columns C, D, E, and F.

Area (in^2): The simulation cross-sectional areas, "Area (in^2)," associated with each zone is entered into the spreadsheet cells located in the row, *pntr+3*. This information is not depicted in the Figure 2 or Figure 3.

Momentum Regions: The *pntr*+4 row pertains to friction applied to the simulation at select cells. The select cells are identified in row *pntr*+4 "Momentum Regions" in Columns C through E. With a value of one entered in cell L1, spreadsheet cells C through L can be populated with state cell numbers (10 state cells). Moving the constant one from L1 to M1 will make available identification of eleven state cells with momentum (columns C through M).

Momentum Coef.: The contents of column C and row *pntr*+5, "Momentum Coef.," is the value applied to all state cells identified as being a momentum source cell on row *pntr*+4. This term is the "pink" noise source of momentum identified as s_2 in Equation (2)—should only be implemented in the vicinity of the fuel injector.

Flame Hold: The value in row *pntr*+6 identifies the Flame-holding cell. The Flame-Hold zone is defined to be the zone that includes the Flame-holding cell. The impact of the Flame-holding cell on the simulation is not shown per se in the governing equations. It is manifested as a fixed spatial gradient, occurring over 5 percent of the combustor length, whereby the value of the diffusion coefficient ε_t/Re^* changes from 0.0 to the maximum value specified by the EddyT term located under the Combustor Process tag. The exact location for defining the value for EddyT is discussed below in the Combustor Process sections. The Flame-Hold zone includes a recirculation region that ranges from 3 percent of the number of simulation cells upstream of the identified flame-holding cell, less one for the identified cell, to 2 percent of the number of simulation cells downstream of the flame-holding cell. Following the default as an example, the recirculation region ranges from computational cell number 92 (98 – 0.03*200) through cell number 102 (98+0.02*200). All cells upstream of and including cell number 92 will have an ε_t/Re^* value of 0.0. Cells 92 through 102 will have values for ε_t/Re^* increasing from 0.0 to a final value following a $\sin^2$ gradient. Values assigned for the ε_t/Re^* term to cells beyond the recirculation region (downstream from cell 102 in the

example) is dependent on the combustor geometry. If another zone exists between the Flame-Hold zone and the combustor exit, this term will decrease back to zero following a $\sin^2$ gradient that results in 0.0 at the exit cell of the Flame-Hold zone. Else, the remaining cells downstream from the recirculation region will be assigned a value equal to the final value assigned to the cell identifying the downstream end of the recirculation region (cell 102 in the example)

Fuel Injection: The value in row *pntr*+7 identifies the cell where fuel is injected into the simulation if fuel is injected in a location other than the inlet. This value dictates which cell is affected by the s_4 term in the source vector. The actual value of s_4 applied to the process is defined by the LSSC term. The exact location in the setup document of the value for LSSC is discussed below in the next section "Experiment Description."

Experiment Description

For reading the values pertaining to these variables, the value of the *pntr* variable is set to the row number with the "Experiment Description" tag in column A.

Frequency (Hz): The value in row *pntr*+1 is the data output frequency in hertz units. For Type 1, PSD, simulations; this value determines the sampling period between data saves. This term is not used for Type 2, Wave, simulations. As indicated in Table 1, the dimensionless value for frequency will be this value multiplied by the ratio of the combustor length to the reference speed of sound. For the example case, $f = f' L'/a^* = 5000(5.45/1809) = 15.06$. This value is used in conjunction with the DTI term located in row *pntr*+5 to essentially employ a low-pass filter for data saving. The DTI term is the simulation step size; therefore, the simulation cycle frequency is 1/DTI in dimensionless units. For the example case, 1/DTI = 1/0.0025 = 400. To save 15.06 data points from every 400 data points available will result in a save the latest simulation states process after every 27 (400/15.06) simulated dimensionless time steps.

A* (ft/sec): The value in row *pntr*+2 is the reference velocity (1809 ft/sec) which is based on the reference temperature of 1460 R, the ratio of specific heats of 1.3, and the characteristic gas constant of 53.5889 lbf ft/lbm/R.

Convergence: The value in row *pntr*+3 is the value of the Convergence term used by the process to determine when the change in values for an iteratively derived answer is small enough to determine a suitable solution has converged; else, iterative looping would proceed indefinitely. The author recommends this value to not be altered.

gamma: The value for gamma, identified in row *pntr*+4, was selected to be 1.3 for the default simulation because the mixture of gases in the combustion chamber is expected to result in a lower ratio of specific heats than pure air.

DTI: The value of DTI available in row *pntr*+5 is used for determining when to record data to file for both Type 1 and Type 2 data sets. As previously described, this is a dimensionless term describing the interval between simulation steps. The use of

this term as it pertains to identifying data to save for Type 1 data output simulations is described above along with the discussion concerning the data save frequency term in row *pntr*+1. For Type 2 simulations, the process is hard coded to acquire at most only 533 data sets regardless of the simulation span. For small $T_{max} - T_{out}$ simulation spans, every data set calculated will be saved—this is the case if $T_{max} - T_{out} \leq$ DTI*533. If the simulation span is greater than DTI*533, then the simulation save interval will be $(T_{max} - T_{out})$/DTI/533.

S14: The S14 term, defined in row *pntr*+6, is a constant for artificial viscosity. The artificial viscosity has been added to dampen the non-physical oscillations in the vicinity of strong gradients such as those brought about by combustion process. The value for S14 may be lowered until 'wiggles' are small near large gradients.

LSSC: The value for the LSSC term is defined in row *pntr*+7. This value is the reaction mean flow rate for fuel injected in a cell other than at the inlet. The process uses this value for the s_4 term in the source vector identified in Equation (2).

Filtering

For reading the values pertaining to these variables, the value of the *pntr* variable is set to the row number with the "Filtering" tag in column A.

dfH, cfH, dfL, and cfL: The terms identified in rows *pntr*+1 and *pntr*+2 are coefficients for a high-pass filter polynomial. Likewise, the terms identified in rows *pntr*+3 and *pntr*+4 are coefficients for low-pass filter. These values are determined by exercising the MATLAB 'Butter' function as described below for obtaining coefficients for a low-pass filter:

$$\left[c_L, d_L\right] = \text{butter}\left(1, \frac{f_L}{f_N}, 'low'\right) \tag{4}$$

Where, the 1 is to specify coefficients for a first order Butterworth filter, f_L is the non-dimensional low-pass filter frequency, f_N is the non-dimensional Nyquist frequency, and the 'low' string is low-pass indicator for the butter command. Determining the coefficients for the high-pass filter similarly uses Equation (4); except with f_H value and using the 'high' script. For these simulations, the f_L value used was a calculation corresponding to 2500 Hz—2500 * L^*/a^* (7.69). The value for f_H was from a calculation corresponding to 50 Hz—50* L^*/a^* (0.15). The non-dimensional Nyquist frequency of 200, being the same for both filters, is defined as ½(1/DTI). Combining the results of both butter operations will result in a Butterworth band-pass filter from 50 to 2500 Hz as described in Equation (5).

$$\begin{aligned} F_{SSC}(k) = &-(df_{H1} + df_{L1})F_{SSC}(k-1) df_{H1}\, df_{L1}\, F_{SSC}(k-2) \\ &+ cf_{H0}\, cf_{L0}\, R_{and}(k) \\ &+ c(f_{H0}\, cf_{L1} + cf_{H1}\, cf_{L0})\, R_{and}(k-1) \\ &+ cf_{H1}\, cf_{L0}\, R_{and}(k-2) \end{aligned} \tag{5}$$

Where the $F_{SSC}(k)$ is the value applied to the s_2 variable in the source vector identified in Equation (2) and the R_{and} term is a product of a pseudorandom number generator within the range of ± the Perturbation Level identified from the keyed answer to question number three in Table 3. Perturbations are applied to the simulation by use of pseudorandom number generator that changes the momentum coefficient of the momentum region during the simulation.

Combustion Process

For reading the values pertaining to these variables, the value of the *pntr* variable is set to the row number with the "Combustion Process" tag in column A.

EddyT: The value in row *pntr*+1 is applied to the μ_t variable for determining the value of ε_t in the source vector as defined in Equation (2).

Schmit and ***Prandtl***: The values in rows *pntr*+2 and *pntr*+3 are the turbulent Schmidt (Sc_t) and Prandtl (Pr_t) numbers applied to the source vector as defined in Equation (2).

Zeta1 and ***Zeta2:*** The values in rows *pntr*+4 and *pntr*+5 are the values used for ζ_1 and ζ_2 respectively in Equation (1).

Q0: The value in row *pntr*+6 is the value of the q_0 variable applied to Equation (2). This term is the reactant heat of reaction. Generally, this term is determined using Equation (3).

K0: The value in row *pntr*+7 is the value of the reaction rate constant, K_0, that is applied to Equation (1).

TC0: The value in row *pntr*+8 is the value for the T_{ign} variable in Equation (1). This variable defines the ignition temperature for the simulation.

Pstage, Tstage, and ***ZE:*** The values in rows *pntr*+9, *pntr*+10, and *pntr*+11 are simulation values for the variables Pstage, Tstage, and ZE. These terms are the non-dimensional static pressure, non-dimensional temperature, and reaction fraction at the exit end of the combustor. The Tstage and ZE terms are only used if inflow occurs.

Pstagl, Tstagl, and ***ZI:*** The values in rows *pntr*+12, *pntr*+13, and *pntr*+14 are simulation values for the variables Pstagl, Tstagl, and Zl. These terms are the non-dimensional static pressure, non-dimensional temperature, and reaction fraction at the entrance end of the combustor. The Pstagl is used for open end boundary conditions with a constant mass flux and the Tstagl is a stagnation temperature value.

ARE: The value in row *pntr*+15 is the open area of the right end of the combustor divided by the nominal cross section. The nominal cross section for the default combustor is $4^2 = 16$ in^2. The right side of the combustor is a contrived open area dimension that considers exit restrictions also known as exit blockage ratio. The default value is set to 1.0 – 0.885 = 0.115. Where the 1.0 is the normalized combustor cross-sectional area and the 0.885 is the exit blockage ratio.

ARI: The value in row *pntr*+16 is the open area at the left end of the combustor divided by the nominal cross section. The nominal cross-sectional area in the inlet area is $6^2\pi$. The left end of the combustor is the open area of the perforated upstream plate. This value can be determined based on a blockage ratio term. For

the default simulation, the blockage ratio is 0.83. Therefore, the default value for this variable is set to 1.0 – 0.83=0.17.

Initial Conditions

The last partition of the document is data written in populated cells below the "Initial Conditions" script. These values represent the initial values for the interior numerical cells of the simulation. Data in this partition starts at *pntr*+2 and located in columns B, D, F, and H. Each row of data includes state values for each computational cell. The number of computational cells, n, in the simulation is dictated by the number of data rows in this partition.

Process Control Flags

Changing any value in the S1Dsetup.xlsx document and saving the change will result in a new "Date modified" time stamp associated with the S1Dsetup.xlsx document. The new time stamp is surely not going to match the time stamp saved in the binary file; therefore, the process will read the .xlsx document and populate a new binary file. Another way to make simulation changes without editing the .xlsx document is to include specific process control flags next to the calling function—see methods two through four in the "Initiating a Simulation" section of this report. The process control flags, PCFs, define values for select operating variables to enable the code to run without a need for reading the .xlsx document. Specifically, instead of making a change in the .xlsx document to change the value for a process variable; use a PCF flag to avoid establishing a new "Date modified" time for the .xlsx document. The information in Table 5 identifies the available PCFs and how they affect the process setup.

TABLE 5.—PROCESS CONTROL FLAGS (PCFS) AVAILABLE FOR THE USER TO DEFINE SELECT PROCESS PARAMETERS WITHOUT EDITING THE .XLSX DOCUMENT
[The PCF format is the PCF title, underscore, and a numeric value as illustrated in the third column.]

Process Control Flag	Affect	Example
ET_x	Ratio of Et/Re*. x is the real number value.	ET_0.0015
FH_x	Flame-holding cell. x is a whole number	FH_98
FI_x	Fuel injector cell. x is a whole number	FI_90
Knot_x	Reaction rate constant (K0). x is the real number value.	Knot_30.1
LSSC_x	Reaction mean flow rate. x is the real number value.	LSSC_35.8
MatLab	Read and write using MATLAB format (.mat files). The filename for the new data output document will correspond with the time that it is created: S1dData_yyyMMdd_hhmm_ssss_TYPE.mat. Where y is the year, M is the month, d is the day, h is the hour, m is the minute, s is milliseconds, and TYPE is either PSD or Wave.	MatLab
NSN_x	Number of zones. x is a whole number.	NSN_5
Pstage_x	Static pressure at output. x is the real number value	Pstage_0.78534
PstagL_x	Total pressure at left (entrance). x is the real number value.	PstagL_1.0
Qnot_x	Q_0 value. x is the real number value.	Qnot_4.2
Save	Will create a new spreadsheet document to hold the results of the current simulation. The filename for the new document will correspond with the time that it is created: S1Dsetup_yyyyMMdd_hhmm_sssss.xlsx. Where y is the year, M is the month, d is the day, h is the hour, m is the minute, and s is milliseconds.	Save
Tstage_x	New temperature at outlet. x is the real number value.	Tstage_2.3
TstagL_x	Total temperature at left (inlet). x is the real number value.	1.0

TABLE 6.—EXAMPLE FOR EMPLOYING MULTIPLE PROCESS CONTROL FLAG PARAMETERS

Initialization	Format
MATLAB environment	S1D_ML('Save_FI_87_FH_99');
Command Prompt Window	S1D_ML Save_FI_87_FH_99

Multiple PCFs can be defined by cascading the alphanumeric characters for each flag along with an underscore separator between flags. Note that flag entry does not need to be in alphabetical order. However, the flag value must immediately follow the flag name—with an underscore separating the name from the value. The information identified in Table 6 is an example for defining the simulation to have the fuel injector cell at 87, the flame-holding cell at 99, and to save the results in an .xlsx formatted document suitable for future reading using Microsoft Excel software. Notice the difference in format between the MATLAB environment and the Command Prompt Window.

Simulation Process

This section presents the process used to solve the one-dimensional flow solver used in the S1D_ML simulation process (Ref. 7). The discussion in this section is with respect to the default combustor geometry illustrated in Figure 2 and Figure 3 and the default values as entered in Table 4. The motivation for including this section in the document is to illustrate how the variables defined in Table 4 affect the simulation process.

The default simulation combustor with Num Zones (NSN) set to five is suitably divided into three one-dimensional zones (NSN less one upstream zone less one downstream zone leaves three computational zones). The computational zones are further subdivided into cells representative of equal axial length. For this discussion, the .xlsx document included 200 rows (n = 200) of initial condition data; therefore, the simulation will have n+2=202 computational cells. Within each of the n cells, numerical integration is performed on a dimensionless governing equation with the assumption of a calorically perfect gas and the results are dimensionless values (Ref. 7).

Upon startup, the process will initialize the states of the simulation cells. If a S1Dsetup.s1d binary file exists, the process will open the file and compare the time stamp to the directory time stamp of the S1Dsetup.xlsx document. If the comparison reveals a match, the process continues by reading the rest of the binary file. Reading the binary file is much faster than reading the .xlsx document. If the comparison reveals a mismatch, the process will read the S1Dsetup.xlsx document and create a new S1Dsetup.s1d document that includes the time stamp of the S1Dsetup.xlsx document. Next, the process will overwrite values read from the documents with employed PCF values.

The simulation process includes multiple inner loops and a main outer loop. The outer loop incrementally progresses from time 0.0 to the end time described in the answer to question 1 in incremental non-dimensional time steps defined by the variable DTI. The default value for DTI is 0.0025 as defined in Table 4. For each time step, the process determines a new pink noise term to apply to s_2 in Equation (2) arrived from a band-pass filtered pseudo random number generator. Next, the simulation will progress through seven (2*NSN – 3) inner loops based on the NSN default value of five as defined in Table 4. The first inner loop establishes the state of the up-stream, entrance, boundary condition. The process to establish the state of cell number one is dependent on the value for ARI—ARI is defined in Table 4. The second inner loop establishes the state of the down-stream, or exit, boundary condition. The process to establish the state of cell number n+2 is dependent on the value for ARE—ARE is defined in Table 4. If the ARI or ARE ratio is less than one, the end is considered to be partially open; whereas, if the ratio is equal to one, the end is considered to be fully open. If ARI is greater than one, the inflow is choked and the end boundary condition value is a constant mass flux (i.e. the value of the density and velocity product is constant). This simulation does not consider the case when the right boundary

condition is choked (ARE > 1). The next two (NSN – 3) inner loops establish the boundary conditions between the zones illustrated in Figure 3. Between each zone are an up-stream boundary condition and a down-stream boundary condition. Up-stream or down-stream depends on the perspective of the zone. The two (NSN-3) loops establish the following boundary conditions: downstream boundary condition for zone 1, upstream boundary condition for zone 2, downstream boundary condition for zone 2, and the upstream boundary condition for zone 3. Although the zone 1 downstream boundary condition is located at the same place as the upstream boundary condition for zone 2, the states of these two boundary conditions are not necessarily equivalent. Same argument pertains to zone 2 downstream and zone 3 upstream states. The calculations for determining these boundary conditions are dependent on the change in cross-sectional area between each zone—no change, expanding, or contracting.

The rest of the inner loops (NSN – 2) are for calculating new states for the cells in each zone based on the previous cell state and the up- and down-stream boundary conditions for each zone.

The frequency of data logging corresponds with the frequency identified in the data sheet (5 kHz was selected for the default simulation). After completing all inner loops for multiple outer loop cycles, simulation data useful for report generating is saved. The number of multiple outer loop cycles between data saves corresponds to the desired data output frequency. For Type 1 simulations where PSD analysis was determined based on the answer to question 4, the cell state pressure values are saved. Only the values associated with the pressure calculations in the cells identified by the answers to questions 4a and 4b are logged. For Type 2 simulations (Wave), seven state variables for all of the cells are logged.

After satisfactorily completing the simulation for the designated period, the process will terminate after populating a data output file. The naming convention for the output file that is populated with data that can be reduced to PSD or Wave reports is S1dData_yyyyMMdd_hhmm_ssss_TYPE.style. Where y is the year, M is the month, d is the day, h is the hour, m is the minute, s is milliseconds, TYPE is either PSD or Wave, and style is either MATLAB .mat format or text format. The simulation can pick-up where it left off by renaming the file that is created when the “Save” PCF is employed. Since the simulation initiates looking for a S1Dsetup.xlsx file, the newly created file will need to be renamed accordingly. To avoid losing the starting point for the previous simulation, it is advisable to first rename the S1Dsetup.xlsx. The names and contents of the output files are dependent on the use of, or lack of, the MATLAB or Save PCF flags as described in Table 7.

After simulation completion, the process resultant data file S1dData_yyyyMMdd_hhmm_ssss_TYPE can be reduced to graphical reports.

TABLE 7.—NAMING CONVENTIONS OF THE OUTPUT FILES ARE DEPENDENT ON THE EMPLOYED PROCESS CONTROL FLAG

[The first row indicates that a populated output file that can be reduced to power spectral density or wave reports will be the result of the simulation. The format of that output file being .mat or .txt is determined by whether or not the MatLab PCF is employed. The second row suggests using the “Save” PCF to start the next simulation from the ending of a previous simulation.]

PCF text	Resultant File	Comments
MatLab	S1dData_yyyyMMdd_hhmm_ssss_TYPE.mat. Where y is the year, M is the month, d is the day, h is the hour, m is the minute, s is milliseconds, and TYPE is either PSD or Wave. This file will include data saved in .mat format for generating PSD or Wave reports using MATLAB software tools.	If the MatLab PCF is not employed, a .txt file is populated with the same naming convention except .txt instead of .mat. This file will include data saved in text format for generating PSD or Wave reports.
Save	S1Dsetup_yyyyMMdd_hhmm_ssss.xlsx. Where y is the year, M is the month, d is the day, h is the hour, m is the minute, and s is milliseconds. This file is a copy of the initiating S1Dsetup.xlsx file with the values of variables that changed as a result of the simulation being replaced.	A corresponding binary file is also created with the extension .s1d instead of .xlsx.

Post Processing

Two types of graphical reports can be readily created using one of two processes: Spectrum or WaveReader. The Spectrum and WaveReader processes are available as MATLAB .m scripts or as compiled executables. Again, the compiled executables do not require a MATLAB license. The Spectrum process will generate a power spectral density report for PSD type data files. Likewise, the WaveReader process will create a wave pattern report for Wave type data files. To run the report generators, initiate the reporting process by selecting the desired data file via the graphical user interface (GUI) that is immediately presented. The GUI will present all the data files in the current root subdirectory with filenames appropriately named. From the GUI, navigate to the location of the desired data file, select the data file, and then select the "Open" button. Selecting "Cancel" in the GUI or closing the GUI without selecting a file with the "Open" button will exit the report generating process. The illustrations in Figure 4, Figure 5, and Figure 6 are example plots from the Spectrum (Figure 4 and Figure 5) and WaveReader process respectively based on reduced data from the default simulation with the following entries into the fields of the "Key-in Process Information:" dialog box: Simulation Time 20.0, Output Start Time 0.0, Perturbation Level 0.1, Data Type 1, Station P1 91, and Station P2 92. The PSD simulation was completed in 96.3 sec and the Wave simulation, using similar answers to the process initialization questions, completed in 96.7 sec.

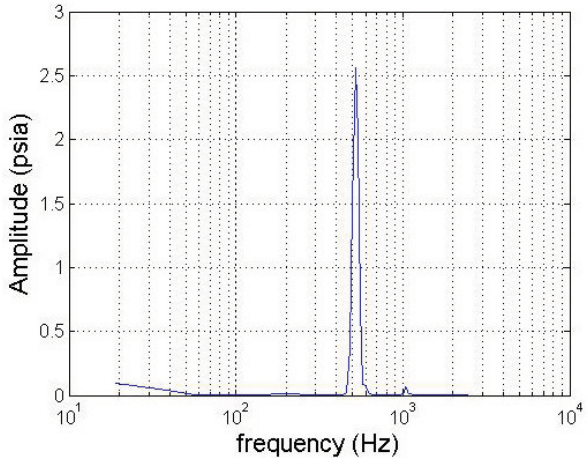

Figure 4.—Sample plot from Spectrum data reduction process illustrating the power spectral density at cell 91.

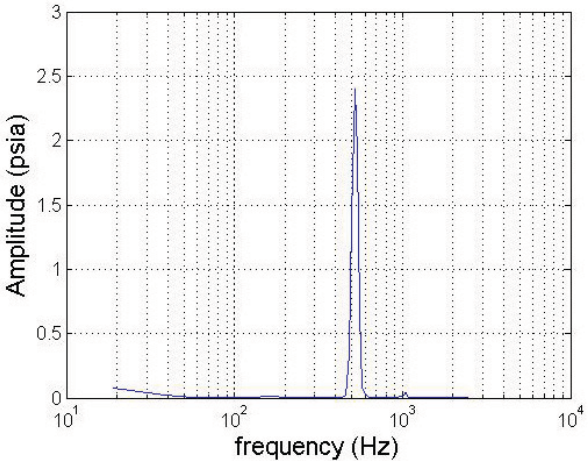

Figure 5.—Sample plot from Spectrum data reduction process illustrating the power spectral density at cell 94.

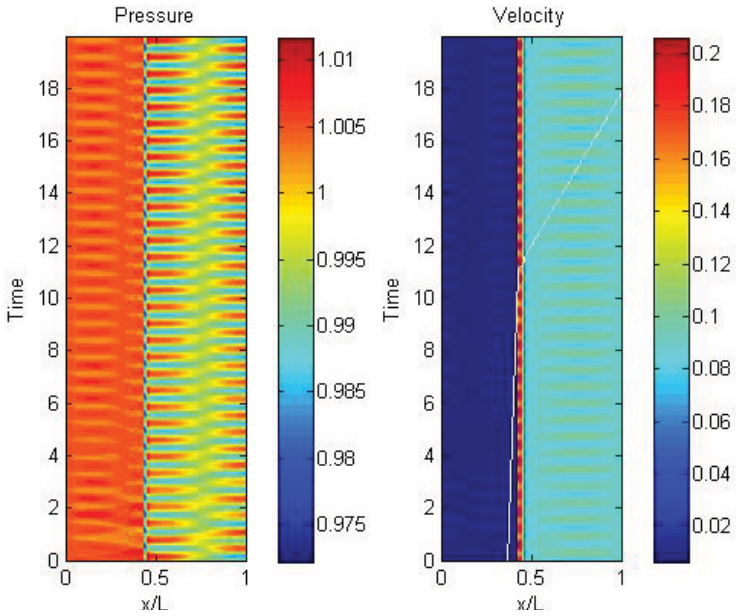

Figure 6.—Sample plot from WaveReader data reduction process illustrating the contours of pressure and velocity over 20 units of simulation time. The right illustration includes a particle path-line.

The illustrations in Figure 4 and Figure 5 clearly depict instability at a little more than 500 Hz. The illustration in the Figure 6 wave analysis illustrates the instability with respect to time. For the Figure 6 illustration, simulation cell states initiate at the bottom and simulation time increases and rises on the chart. The abscissa is the pressure and velocity profiles across the entire simulated combustor—magnitude is depicted with colors. Furthermore, the Velocity illustration includes a particle path-line.

Comparing the illustration in Figure 4 to Figure 6 against simulation experiments using the FORTRAN coded process revealed the same instability frequency and a similar wave pattern over the span of 20 units of simulation time (Ref. 15).

Summary

A one-dimensional, computational fluid dynamics based combustor simulation had been developed previous to this work, coded in FORTRAN, and documented that exhibits self-exciting, thermo-acoustic oscillations in premixed combustor geometries that typically have large, abrupt changes in cross sectional area. This process has been replicated as described in this document using the higher level programming MATLAB language. The source code for this simulation can be opened and modified at users risk using MATLAB tools. Furthermore, the code has been written and compiled to an executable that will allow users without a MATLAB license to also run the process. This simulation includes initialization questions for the user regarding general startup information—similar to the original FORTAN simulation. This document also introduces the availability of process control flags that can be used to quickly setup a simulation without redefining fields in a setup document that is formulated via Microsoft Excel software. Based on the information included in this document together with a copy of the S1D_ML process, reasonably accurate one-dimensional combustor simulation experiments can be conducted suitable for control strategy studies.

References

1. Chang, C.T., et al., "NASA Glenn Combustion Research for Aeronautical Propulsion," Journal of Aerospace Engineering, Volume 26, No. 2, April 2013.
2. Lieuwen, T.C., "Unsteady Combustor Physics," pp. 176-186, Cambridge University Press, 2012
3. DeLaat, J.C., Breisacher, K.J., Saus, J.R., and Paxson, D.E., "Active Combustion Control for Aircraft Gas Turbine Engines," NASA/TM—2000-210346, July 2000.
4. DeLaat, J.C. and Chang, C.T., "Active Control of High Frequency Combustion Instability in Aircraft Gas-Turbine Engines," NASA/TM—2003-212611, September 2003.
5. Kopasakis, G., "Systems Characterization of Combustor Instabilities with Controls Design Emphasis," NASA/TM—2004-212912, February 2004.
6. DeLaat, J.C., Kopasakis, G., Saus, J.R., Chang, C.T., and Wey, C., "Active Combustion Control for Aircraft Gas-Turbine Engines—Experimental Results for an Advanced, Low-Emissions Combustor Prototype," NASA/TM—2012-217617, July 2012.
7. Paxson, D.E., "A Sectored-One-Dimensional Model for Simulating Combustion Instabilities in Premix Combustors," NASA/TM—1999-209771, December 1999.
8. Kopasakis, G. and DeLaat, J.C., "Adaptive Instability Suppression Controls in a Liquid-Fueled Combustor," NASA/TM—2002-211805, August 2002.
9. Kopasakis, G., "High Frequency Adaptive Instability Suppression Controls in a Liquid-Fueled Combustor," NASA/TM—2003-212535, July 2003.
10. Le, D.K., DeLaat, J.C., and Chang, C.T., "Control of Thermo-Acoustics Instabilities: The Multi-Scale Extended Kalman Approach," NASA/TM—2003-212536, August 2003.
11. Kopasakis, G., DeLaat, J.C., and Chang, C.T., "Validation of an Adaptive Combustion Instability Control Method for Gas-Turbine Engines," NASA/TM—2004-213198, October 2004.
12. Le, D.K., DeLaat, J.C., Chang, C.T., and Vrnak, D.R., "Model-Based Self-Tuning Multiscale Method for Combustion Control," NASA/TM—2006-213855, February 2006.
13. Kopasakis, G., DeLaat, J.C., and Chang, C.T., "An Adaptive Instability Suppression Controls Method for Aircraft Gas Turbine Engine Combustors," NASA/TM—2008-215202, May 2008.
14. DeLaat, J.C. and Paxson, D.E., "Characterization and Simulation of the Thermoacoustic Instability Behavior of an Advanced, Low Emissions Combustor Prototype," NASA/TM—2008-215291, July 2008.
15. Paxson, D.E. and Mehta, V.R., "Using the NASA GRC Sectored-One-Dimensional Combustor Simulation," NASA TM—2014-216623, February 2014.
16. http://www.mathworks.com/products/compiler/mcr/. Accessed October 20, 2014.