



iREVEAL User Manual

Version 2016.02.00

February 2016



This Material was produced under the DOE Carbon Capture Simulation Initiative (CCSI), and copyright is held by the software owners: ORISE, LANS, LLNS, LBL, PNNL, CMU, WVU, et al. The software owners and/or the U.S. Government retain ownership of all rights in the CCSI software and the copyright and patents subsisting therein. Any distribution or dissemination is governed under the terms and conditions of the CCSI Test and Evaluation License, CCSI Master Non-Disclosure Agreement, and the CCSI Intellectual Property Management Plan. No rights are granted except as expressly recited in one of the aforementioned agreements.

Revision Log

Version Number	Release Date	Description
Version 2014.10.0	10/31/2014	2014 October IAB Release – Initial Release of iREVEAL integrated with FOQUS
Version 2016.02.0	2/29/2016	Enabled non-conventional solid species

Table of Contents

1. Introduction	1
1.1. Motivating Example	1
1.2. Overview.....	1
1.3. Supported Features	3
1.4. Restrictions.....	3
2. Tutorial	4
2.1. Configuring ROM and Generating Input Samples	6
2.2. Building and Exporting ROM.....	14
2.3. Visualize Response Surface (Optional).....	16
2.4. Integrating ROM with ACM and Aspen Plus.....	23
3. Usage Information	30
3.1. Support.....	30
3.2. Restrictions.....	30
4. Advanced Features.....	30
5. Debugging	30
5.1. How to Debug.....	30
5.2. Known Issues	30
5.3. Reporting Issues.....	33
6. Terms	33
7. JSON Configuration File Format.....	34
8. References.....	52

List of Figures

Figure 1: Reduced Order Modeling Process	2
Figure 2: iREVEAL Home Window Under FOQUS.....	5
Figure 3: Create a Flowsheet From Configuration JSON File.....	7
Figure 4: Add New Ensemble Window for Sampling Input Space	8
Figure 5: Input Parameters and Ranges	9
Figure 6: Window for Sampling Scheme and Number of Samples	10
Figure 7: Dialog Window for Previewing Input Samples.....	11
Figure 8: 2-D Plot of 20 Sampled Points	12
Figure 9: Monitoring Execution Process during ROM Generation	15
Figure 10: Visualize Response Surface: Export and Import Flowsheet Results.....	16
Figure 11: Visualize Response Surface: Create New Ensemble 1	18
Figure 12: Visualize Response Surface: Create New Ensemble.....	19

Figure 13: Visualize Response Surface: Select New Ensemble	20
Figure 14: Visualize Response Surface: Analyze Ensemble	21
Figure 15: Visualize Response Surface: Validation Plot	22
Figure 16: Visualize Response Surface: Response Surface.....	23
Figure 17: ROM Model Displayed in ACM Window	25
Figure 18: Compiling ACM Source Code and Exporting DLL.....	26
Figure 19: Manage Model Libraries Under Customize Menu.....	27
Figure 20: Activating ACM Models in Manage Libraries Window.....	28
Figure 21: Aspen Plus Flowsheet With Feed and Product Streams Connected to ROM Block.....	29
Figure 22: Select Programs in Control Panel.....	31
Figure 23: Select “Make a file type always open in a specific program”	31
Figure 24: .py is associated with Notepad, that’s bad.....	32
Figure 25: Select Python.exe	32
Figure 26: Python now associated with .py	33

List of Tables

Table 1: Names and Types of the Data in the iREVEAL Configuration File.....	35
Table 2: Descriptions of Object Types Used in the iREVEAL Configuration File.....	36

To obtain support for this package, please send an email to
ccsi-support@acceleratecarboncapture.org.

1. INTRODUCTION

The iREVEAL framework is a generic reduced order modeling tool that provides a complete workflow to build a reduced order model (ROM), given a high-fidelity model. For CCSI, the tool is configured with capabilities to process any Computational Fluid Dynamics (CFD) model (MFIx, Barracuda, Fluent®). It can also be applied to other steady-state high-fidelity models that contain boundaries for the inputs and outputs of mass and energy such as the hybrid boiler model developed by CCSI (Ma et al., 2016). The generated ROM can be integrated into a commercial process simulation software such as Aspen Custom Modeler (ACM) and Aspen Plus. The generic reduced order modeling process is discussed for completeness, followed by the details on each step, and the features available in the iREVEAL toolkit in the following sections.

1.1. Motivating Example

A user trying to integrate a high-fidelity model into a larger system level simulation can use this tool to create an approximate surrogate model that speeds up the simulation by several orders of magnitude. The generated surrogate model is in a compliant ACM or Cape-Open format and can be directly integrated into the system level simulation.

Another use case for iREVEAL is to simply study a model under varying conditions. For example, a typical CFD model takes a week to run and reach steady state. If the user has already studied this model under 10 different conditions before and has the simulation results available, the user can use the results to create a ROM using iREVEAL and use the ROM, rather than the CFD model, to get a reasonable prediction in other input conditions, which could save a significant amount of simulation time.

1.2. Overview

The basic concept of model reduction assumes that the input-output relationship from a computationally expensive simulation can often be well approximated by a much lower dimensional, computationally inexpensive model that gives nearly the same output response. Generating a ROM requires generating and analyzing multiple instances of the simulation under varying conditions, so the overall behavior of the system can be reasonably approximated. The design of the iREVEAL framework is deeply driven by the generic workflow required to build a ROM, which is discussed next.

The ROM creation is a multi-stage process, driven by a scientist or engineer who has a deep understanding of the high-fidelity computational model. As shown in Figure 1, the steps are as follows:

1. Select a base case simulation model and then specify which input parameters and output values of the high-fidelity model should be represented in the ROM.

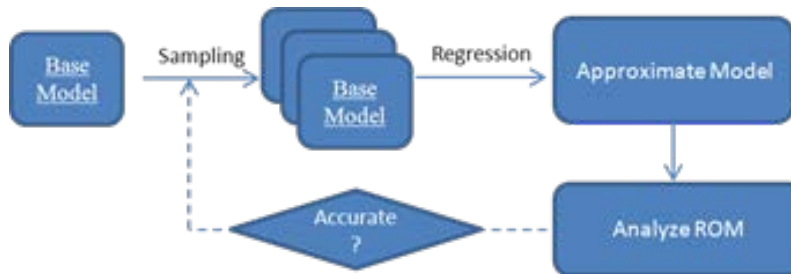


Figure 1: Reduced Order Modeling Process

2. Select a sampling method that generates N possible values for all of the selected input parameters within the ranges specified by the user. This produces N distinct input files for the simulation, each with a different set of values for the input parameters of interest.
3. Execute an ensemble of simulations, one for each input sample set generated by the sampling algorithm. This step can take days to weeks depending on the simulation time, the number of samples, and the computational resources available.
4. Select a regression method and use the method to generate a response surface that maps simulation inputs to the outputs from the ensemble runs.
5. Analyze the response surface and then quantify the accuracy of the ROM.
6. Use iterative approaches to improve the ROM accuracy by incorporating new samples.
7. Export the ROM for integration in larger scale system-wide simulations.

1.3. Supported Features

The iREVEAL toolkit provides all of the major set of features needed to generate a ROM from any CFD model or any non-CFD model that involves material and energy flow through a device. The features available are:

1. Generate samples from input parameter space: A user can select any input parameters and generate samples using any of the following sampling methods (for other available sampling methods consult the *FOQUS User Manual*):
 - *Latin Hypercube Sampling*: Generates samples uniformly distributed in the parameter space.
 - *Normal Distribution Sampling*: Generates samples from a Gaussian distribution, with samples centered around the mean value of the parameter and spread out based on the standard deviation.

$$\text{mean} = (\text{min} + \text{max})/2$$

$$\text{standard deviation} = (\text{max}-\text{min})/7$$
 - *Quasi Monte Carlo Sampling*: Uses either Halton, Sobol, or Torus sequence to generate samples.
2. Build the ROM: Once the results of all the simulations become available, ROM building algorithms inside iREVEAL can be invoked. The framework invokes the selected regression method and creates the mapping between input parameters and output parameters.
3. Export the ROM in ACM or CAPE-OPEN format: The user has the option to export the ROM to a file in ACM or CAPE-OPEN format, after the user has built the ROM. This generates an ACM file with an extension “acmf” or a CAPE-OPEN compatible output file which can then be used to integrate the ROM in a process simulation.

1.4. Restrictions

The iREVEAL framework is intended to be used for creating a steady-state ROM for a CFD model (e.g., MFIIX, Barracuda, Fluent) or other high-fidelity model that involves material flow in gas, liquid, and solid phases.

2. TUTORIAL

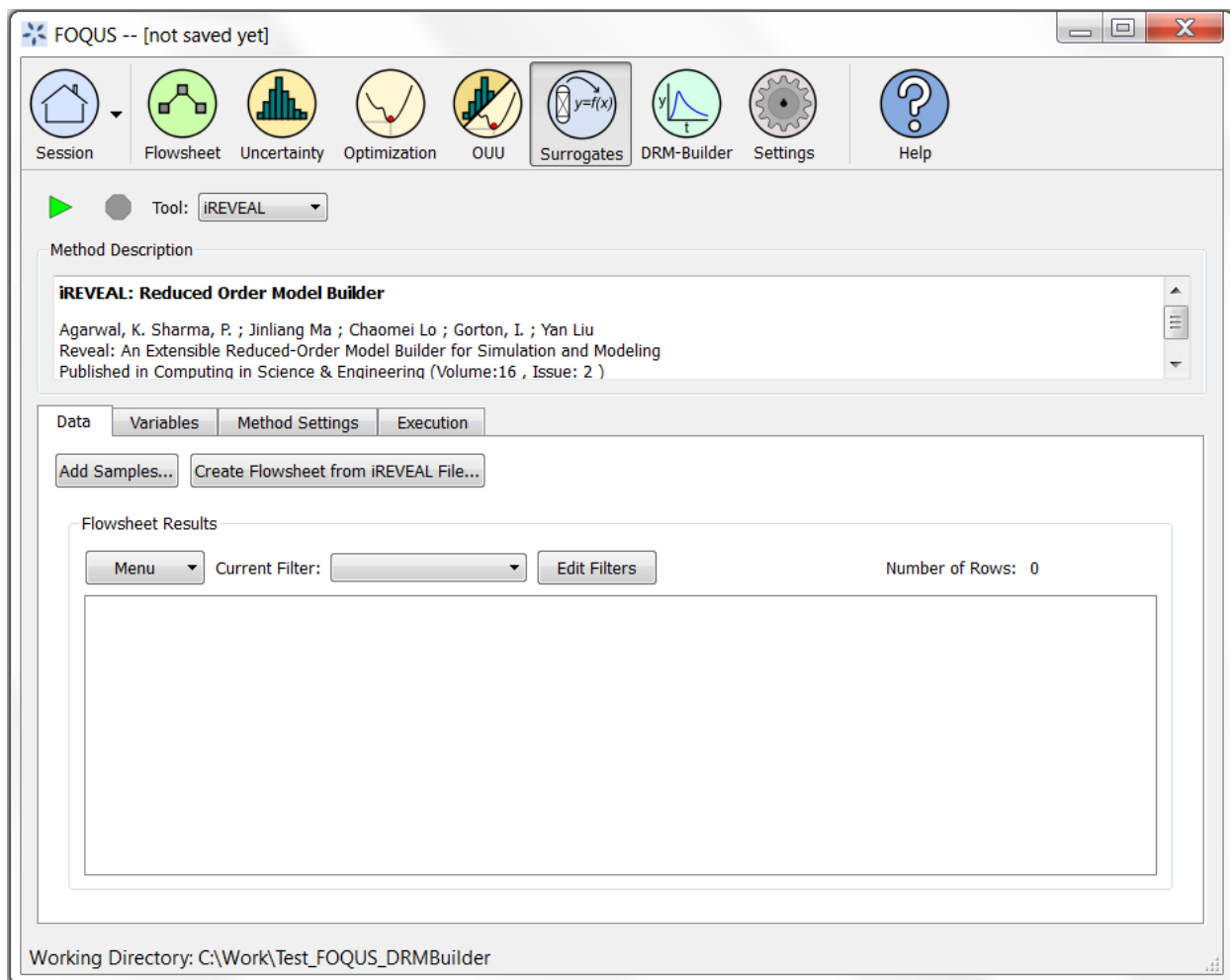
For the purpose of this tutorial, it is assumed that user has created a configuration input file in JSON format (as described in Appendix) containing species and phases involved in the high-fidelity model, data in feed and product streams including energy streams, and other input and output data to be included in the ROM. An example file has been provided in “examples/iREVEAL/boiler.json” for testing. The high-fidelity model for the example is a 1-D and 3-D hybrid boiler model (1-D for reacting flow and 3-D for radiative transfer) developed by CCSI (Ma et al., 2016). It models PacifiCorp’s Hunter Unit 3 boiler located in Castledale, Utah with a thermal input of 1325 MW. The wall-fired subcritical unit has 40 low-NO_x burners and 10 overfire air ports. For demonstration purposes, the ROM for the boiler model is generated assuming that the primary air and coal feed rates are fixed. The only two varied input parameters are the total flow rates of secondary air and overfire air. The output parameters are total heat loss to all boundaries and heat absorption by the platen superheater as well as the properties of flue gas and fly ash streams.

The directory *\$work_dir*, is set to the \$FOQUS_WORK_DIR/iREVEAL by default. All the files generated by iREVEAL for a particular session will be available in this folder. (To set/view \$FOQUS_WORK_DIR, user can go to the “Session” tab in FOQUS interface and click on “FOQUS Settings” (refer to the *FOQUS User Manual for details*).

Note: iREVEAL is now a part of FOQUS bundle and is installed with the bundle. The configuration of iREVEAL path under the “Session” tab is no longer needed. iREVEAL requires Java standard edition. JDK version 1.8 needs to be installed as a prerequisite.

In all of the steps below, it is assumed that the user has performed the following steps:

1. Open the FOQUS user interface and then select the “Surrogates” tab.
2. Select “iREVEAL” under the “Tools” drop-down box. The iREVEAL home window with the “Data” frame displays as shown in Figure 2.

**Figure 2: iREVEAL Home Window Under FOQUS**

2.1. Configuring ROM and Generating Input Samples

Description

The ROM generated by iREVEAL can eventually be integrated with process simulation software such as ACM, certain model configuration options and parameters need to be specified including the chemical species involved in the gas, liquid, and solid phases. The variables related to the material streams have to be specified such as pressure, temperature, flow rate, and species composition. Some of the variables might be “fixed” at certain operating conditions while others are “varied” within certain ranges. In the current version of iREVEAL those configuration options and parameters are prepared by the user in a configuration file in JSON format. The JSON format is widely used in the IT community especially for web applications. The JSON format is a simple text file with name and value pairs in a hierarchy structure. Visit json.org for details of the JSON format. The required configuration parameters for iREVEAL are described in Section 7.

Example

The user needs to prepare a JSON input file that contains the configuration data. A sample input file named “boiler.json” is included in the “examples/iREVEAL” folder under the FOQUS installation directory.

Steps in iREVEAL

1. Click “Create Flowsheet from iREVEAL File” in the “Data” frame. An “Open iREVEAL File” dialog window displays. Browse to the input JSON file. For testing with sample browse to “FOQUS_installation/examples/iREVEAL” directory, select the “boiler.json” file, and then click “Open” (as shown in Figure 3).

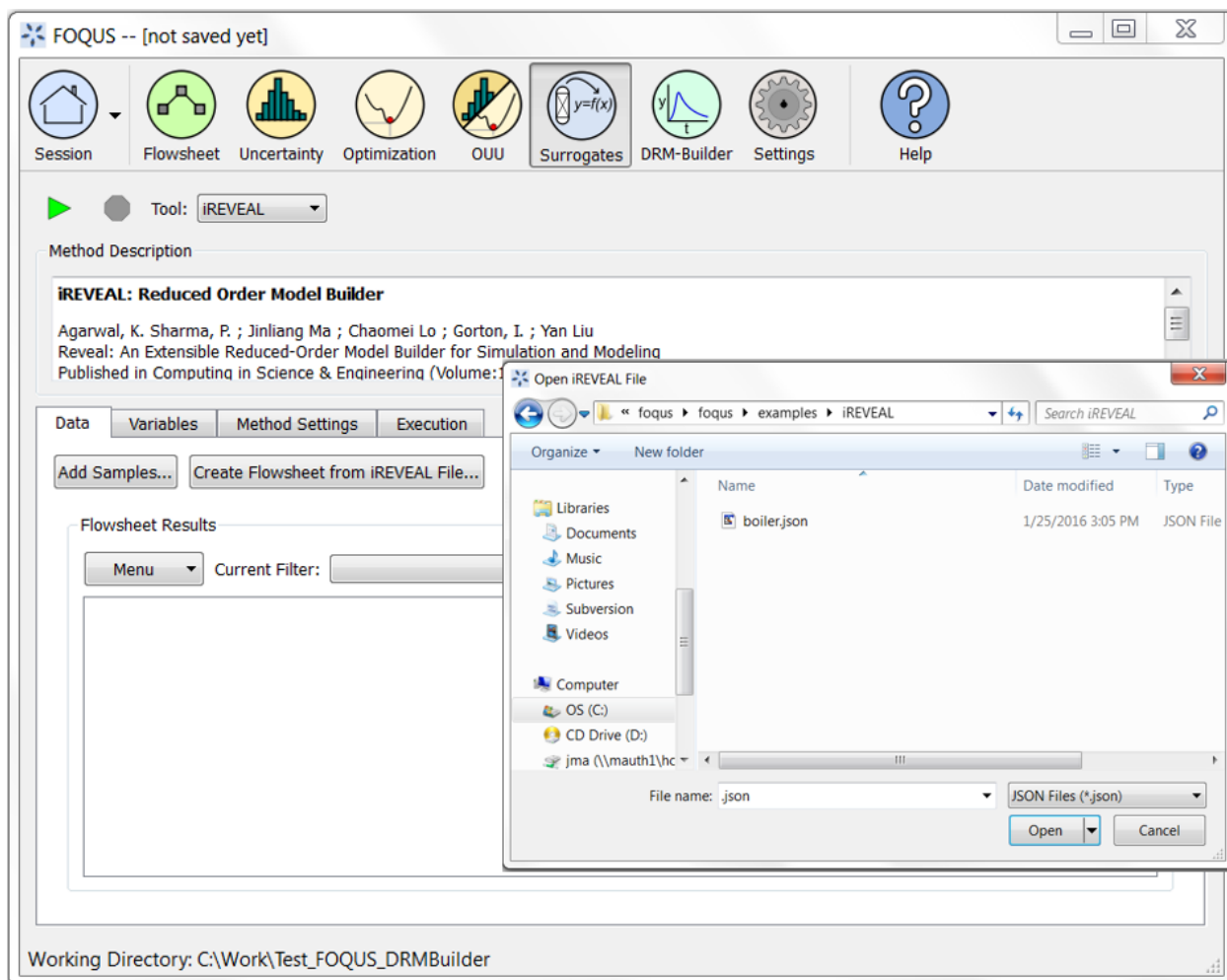


Figure 3: Create a Flowsheet From Configuration JSON File

- Click “Add Samples” in the “Data” frame of the iREVEAL window. An “Add New Ensemble – Model Selection” dialog window as shown in Figure 4 displays. Click “OK”. A “Simulation Ensemble Setup” window as shown in Figure 5 displays and contains variables marked in the JSON input file as “varied” under the “Distributions” tab. The parameters and the ranges shown here are included in ROM for sampling. For an input variable that is part of an input boundary, the name of the variable is usually the string that contains the name of the physical quantity in the JSON file appended by the corresponding boundary name, phase name, and phase index. For example, if the physical quantity is “Temperature”, the boundary name is “Feed”, and the phase is the first solid phase, the input variable name is “Temperature_FeedSP0”. Here “SP” stands for “Solid Phase” and 0 is the zero-based solid phase index. If the temperature is for the gas/liquid mixture phase, the input variable name is “Temperature_FeedGP” and no phase index is appended since there is always one gas/liquid mixture phase only. Notice that the two input parameters shown in Figure 5 are “TotalMassFlow_OFAGP” and “TotalMassFlow_SecGP”, representing the total overfire air mass flow rate and total secondary air mass flow rate, respectively. A prefix “iREVEAL” is added to the two input parameters. The values in the “Min” and “Max” columns are the data for lower and upper limits of the parameters. To make changes to the parameter list or the lower and upper limits of parameters, the user should make the corresponding changes in the JSON file and then repeat Steps 1 and 2.

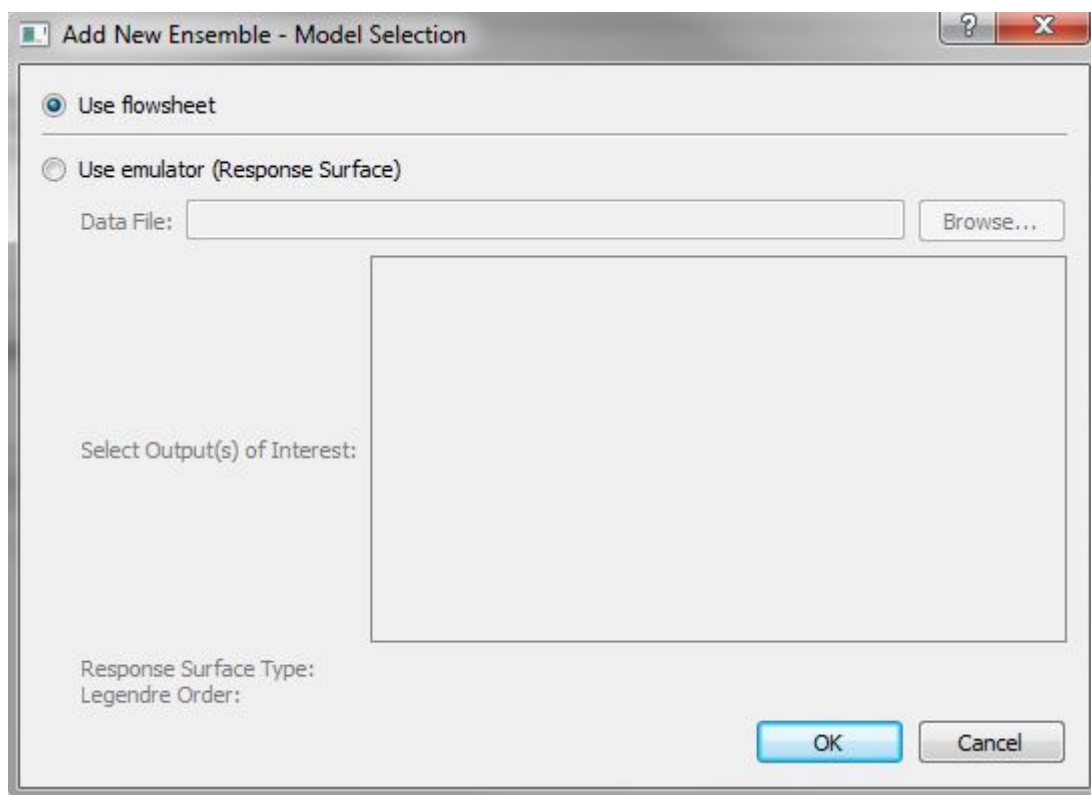


Figure 4: Add New Ensemble Window for Sampling Input Space

Simulation Ensemble Setup

Choose how to generate samples:

☒ Choose sampling scheme

☐ Load flowsheet samples

☐ Load all samples from a single file

Distributions | **Sampling scheme**

	Name	Type	Default	Min	Max	PDF	Param 1
1	iREVEAL.TotalMassFlow_OFAGP	Variable	nan	109.62	133.98	Uniform	
2	iREVEAL.TotalMassFlow_SecGP	Variable	nan	260.64	318.56	Uniform	

Figure 5: Input Parameters and Ranges

3. Select the “Sampling scheme” tab and then select the “Latin Hypercube” sampling scheme in the list on the right. Leave the radio button of “All” selected. In the “# of samples?” box, enter number of simulations. For the tutorial example, enter “20” as shown in Figure 6. This is the number of samples or high-fidelity model cases the user wants to run. Click “Generate Samples”. iREVEAL generates 20 sets of input data to be used in the ROM.

Note: The “Number of Samples” set in the user interface should be the same as what the user specified in the input JSON file (i.e., boiler.json file).

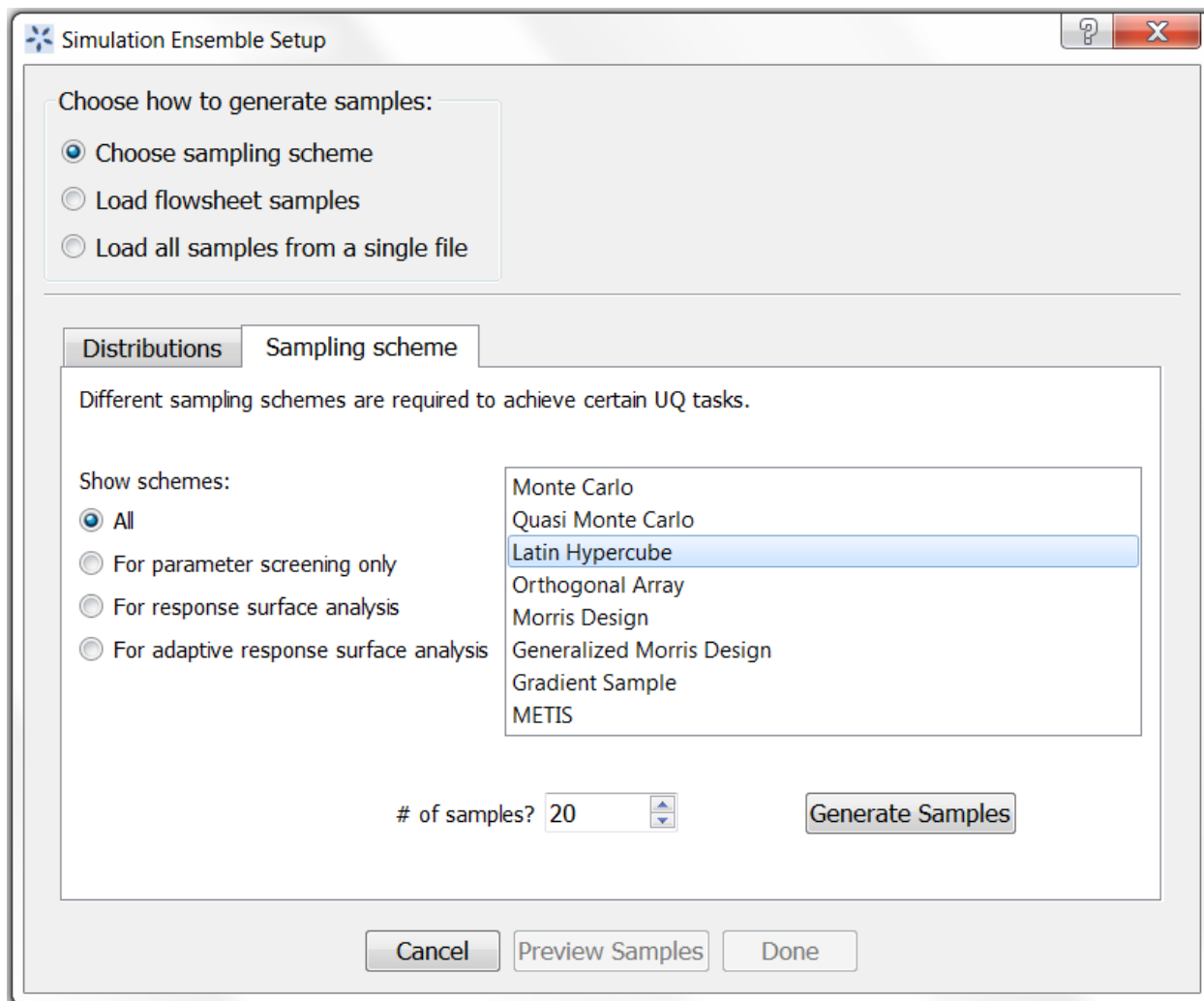


Figure 6: Window for Sampling Scheme and Number of Samples

4. (Optional) Click “Preview Samples”. A “CCSI UQ/Opt Framework – Preview Inputs” dialog window as shown in Figure 7 displays. The samples of input data are tabulated on the left and the user can use the data to setup the input files for the CFD cases. The input data can be visualized in 1-D and 2-D plots using the widgets shown on the right side of the window. Figure 8 shows the 2-D plot of the 20 sample points. Note: Since the Latin Hypercube sampling algorithm is random in nature, the samples obtained by the user may be different from those shown in Figure 8. Click “OK” to close the plot window.
5. Click “Done” in the “Simulation Ensemble Setup” window to accept the generated samples. A message window displays confirming the successful generation of the samples.

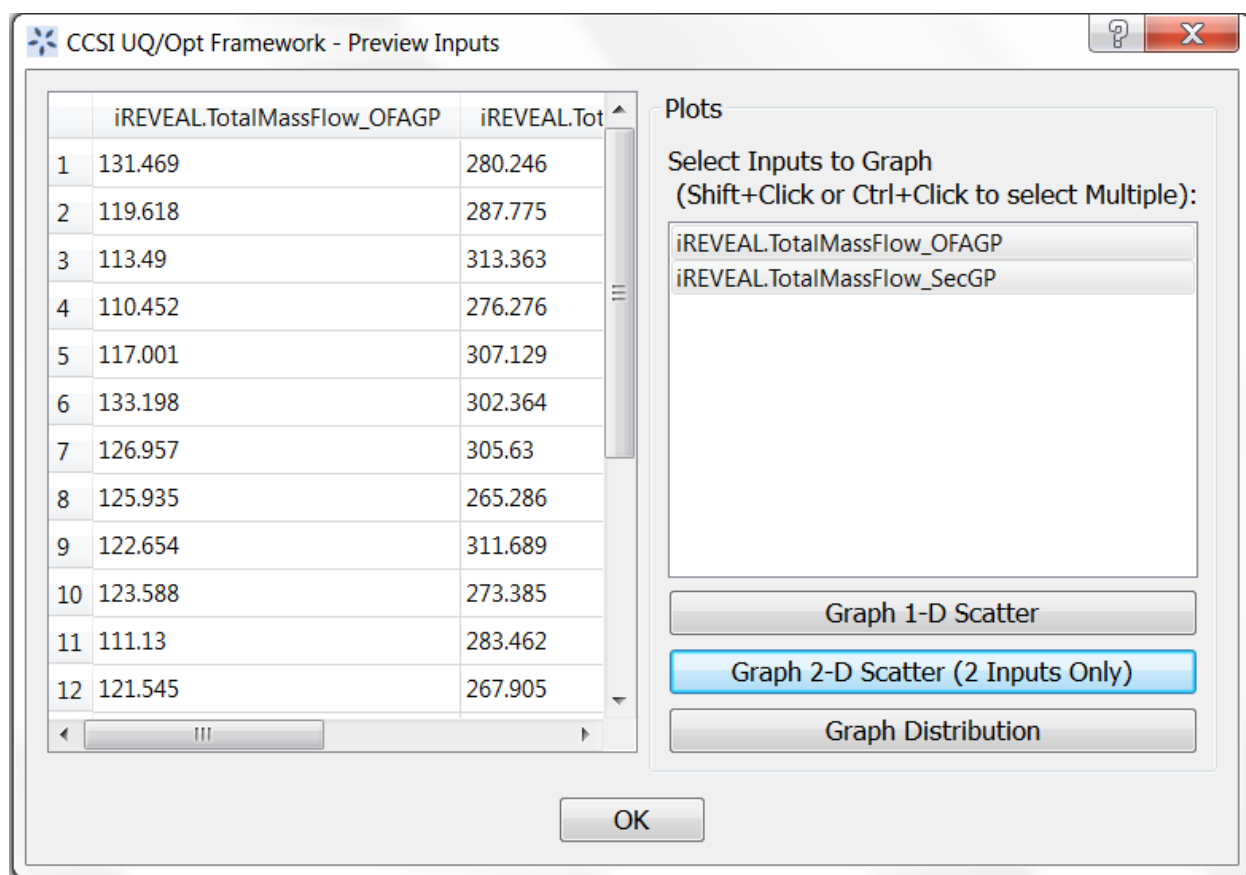


Figure 7: Dialog Window for Previewing Input Samples

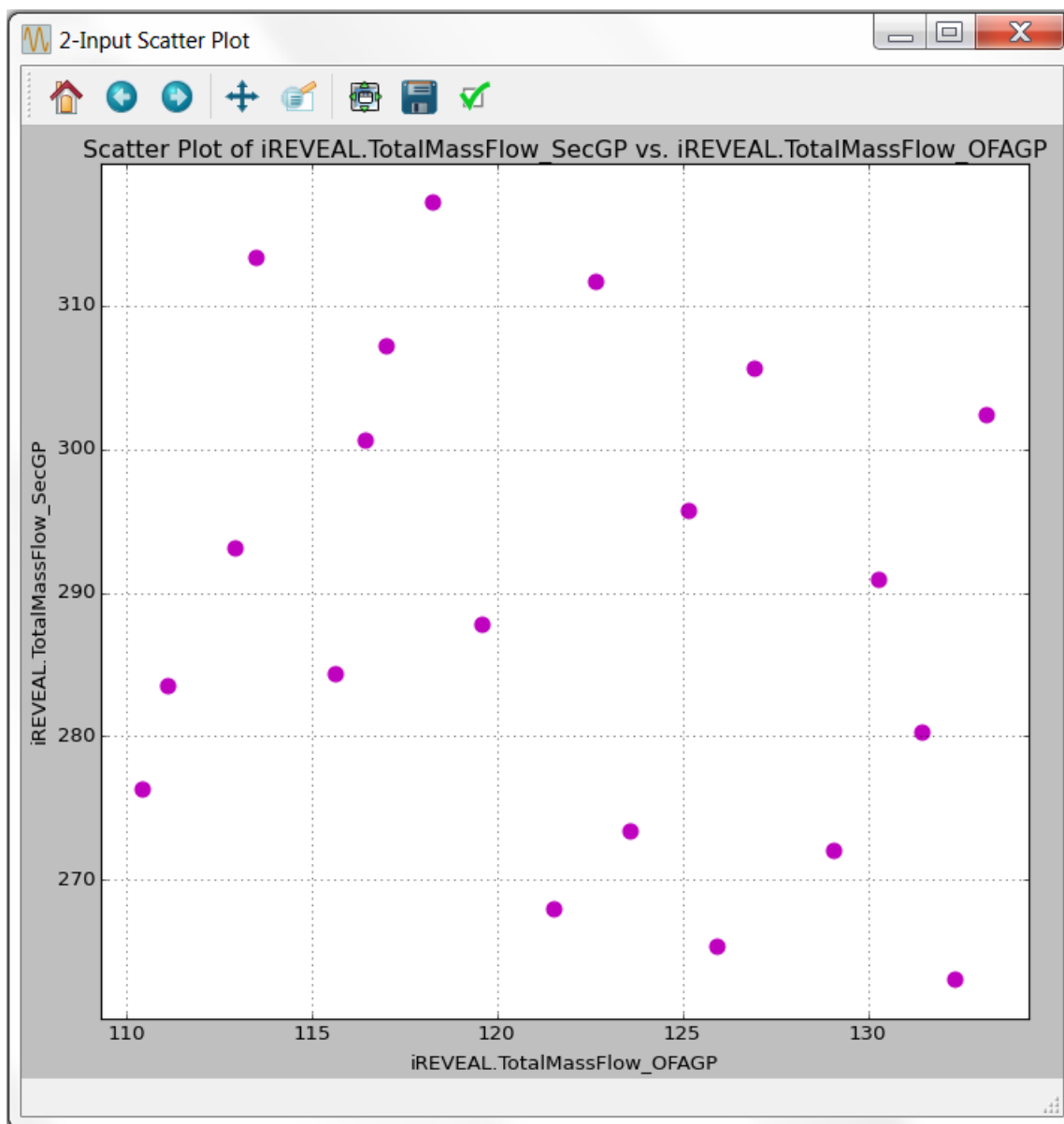


Figure 8: 2-D Plot of 20 Sampled Points

Output of Configuration and Sampling Steps

On successful execution, iREVEAL generates two files in the *\$work_dir* directory, “psuade.in” and “psuadeData” and several other temporary files. The “psuadeData” file contains values of the sampled input parameters for the individual samples. The user can use the values from this file to prepare the inputs for 20 boiler model cases. The user can then run those cases and process the results. Notice that there are 20 sets of data in “psuadeData” file. Each set contains 2 input variables and 19 output variables in this example. The two input variables are total overfire air flow rate and total secondary air flow rate (notice the order specified in “psuade.in” file). The output variables in the “psuadeData” file are irrelevant at the moment and each has a value of “9.999999999999997e+034”. The two files that are generated in the steps described

above are used in subsequent steps to build the ROM and export the ROM in CAPE-OPEN and ACMF compliant formats.

2.2. Building and Exporting ROM

Description

This section of the tutorial describes the steps to create the functional mapping from the input space to the output space and to export the ROM to files.

Example

Once the results of all high-fidelity model runs are available, the user needs to create a file named “results” in the *\$work_dir* directory, containing the values of all the output parameters listed between “Output_Parameter_Starts” and “Output_Parameter_Ends” lines in the “param.in” file. The format of the “results” file is in tab separated format. The first row in the “results” file contains the name of the output variables shown in the “param.in” file. The simulation results are then listed and separated by tabs, one line for each simulation. The order of the output parameter values in each row should be the same and match the order in the “param.in” file, generated in *\$work_dir* in previous step. For this tutorial, 20 hybrid boiler models need to be run and 19 output variables need to be entered in the “results” file for each run.

Notes: For the purpose of tutorial we provide a pre-generated input (“psuadeData”) and output (“results”) files in “FOQUS_installation/examples/iREVEAL” directory. User can copy these pre-generated files to *\$work_dir* for the tutorial.

1. The 20 samples generated by iREVEAL in Section 2.1 are random in nature, meaning that they are different each time the user issues the sampling command. For the tutorial purpose, FOQUS installation provides a “psuadeData” example file that was created when the tutorial was prepared. The corresponding “results” file is also provided in the same example directory.
2. Copy the “psuadeData” and “results” files in the “examples/iREVEAL” folder under the FOQUS installation’s folder to the working directory, *\$work_dir*. Allow the “psuadeData” file created in Section 2.1 to be replaced.
3. The name of an output variable corresponding to an output boundary is similar to the name of an input variable corresponding to an input boundary with the boundary name, phase name, and phase index appended to the physical quantity’s name.

Steps:

1. Navigate to the “Surrogates” tab in FOQUS, select “iREVEAL” under “Tools”, and then click the “Method Settings” tab in the stacked frame. Make sure the “Value” column of the first row is “results” and the “Value” column of the second row is “Kriging”. Note: Only the Kriging regression method is implemented in the current version of iREVEAL.
2. Click the “green triangle”. This generates and exports the ROM in the *\$work_dir* directory. The user can view the progress of the execution under the “Execution” tab as shown in Figure 9.

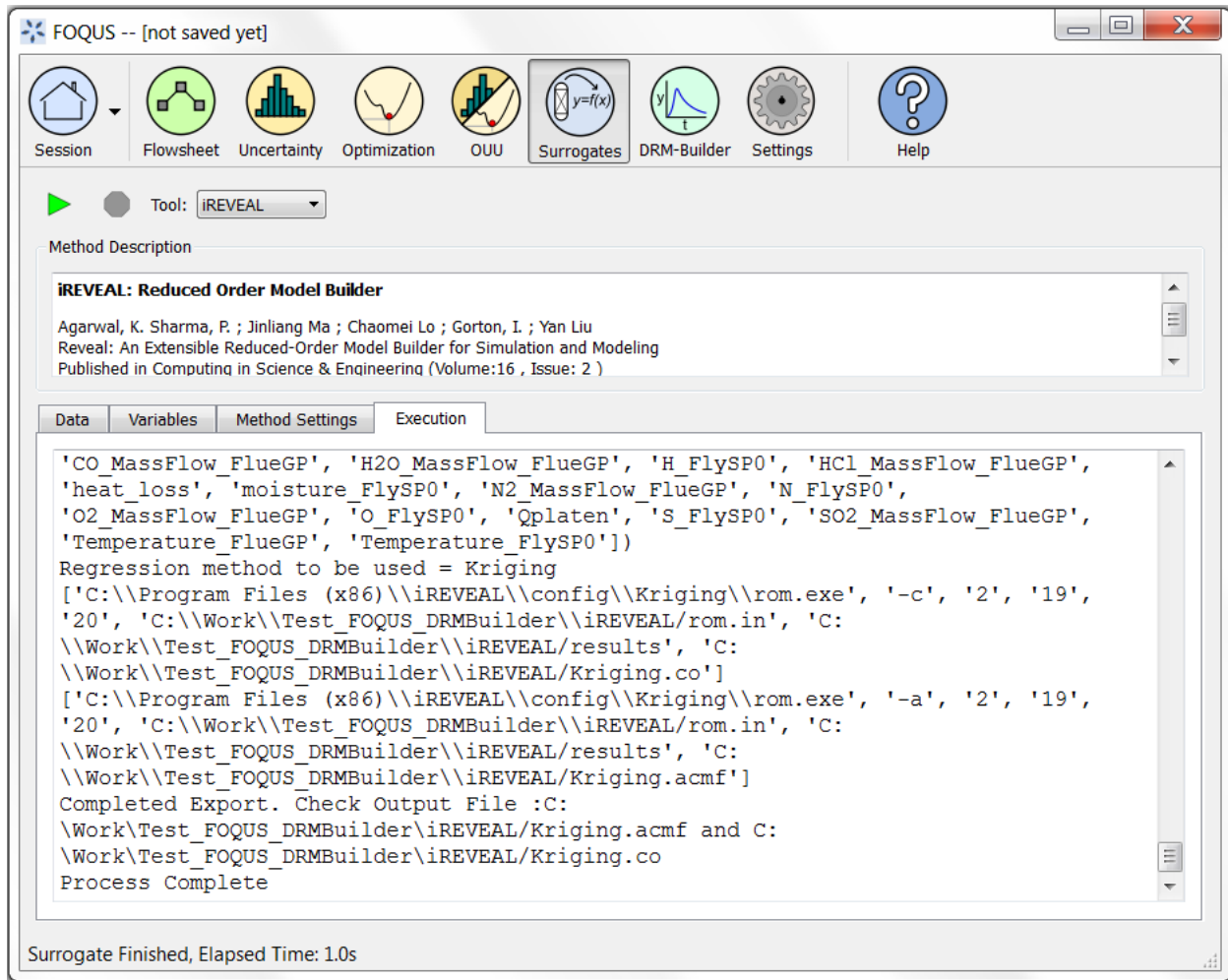


Figure 9: Monitoring Execution Process during ROM Generation

Output Files and Directory

1. On successful execution the system generates the “*\$work_dir\Kriging*” directory. The “errors” file under this directory contains the meanings of the error values after the N-fold cross validation test. The “predicted_results” file in this directory (in the same format as the “*\$work_dir\results*” file) contains the predicted values of the output parameters (one simulation per line) using the ROM.
2. On successful execution the system generates the “*\$work_dir\model.acmf*” and “*\$work_dir\model.rom*” files. These files are ROM files generated from the high-fidelity model, which are ACMF and CAPE-OPEN compliant, respectively.

2.3. Visualize Response Surface (Optional)

Description

To visualize the response surface in FOQUS interface, the user can follow the steps listed below.

Note: This visualization step is optional and not needed to use the ROM created in section 2.2 inside a Aspen process simulation

Steps

1. Open the “Data” frame under the “Surrogates” tab. Under the “Menu” option select “Export to csv file” to export the flowsheet to a csv file as shown in Figure 10. Let’s call this file “results1.csv”. Note that this file does not contain output parameter data.
2. Copy the output values from the “\$work_dir/results” file to the exported “results1.csv” file column by column, matching column names.

Note: User can open “results” file in Excel (enable space and tab as delimiters) and copy the matching data from “results” to results1.csv.

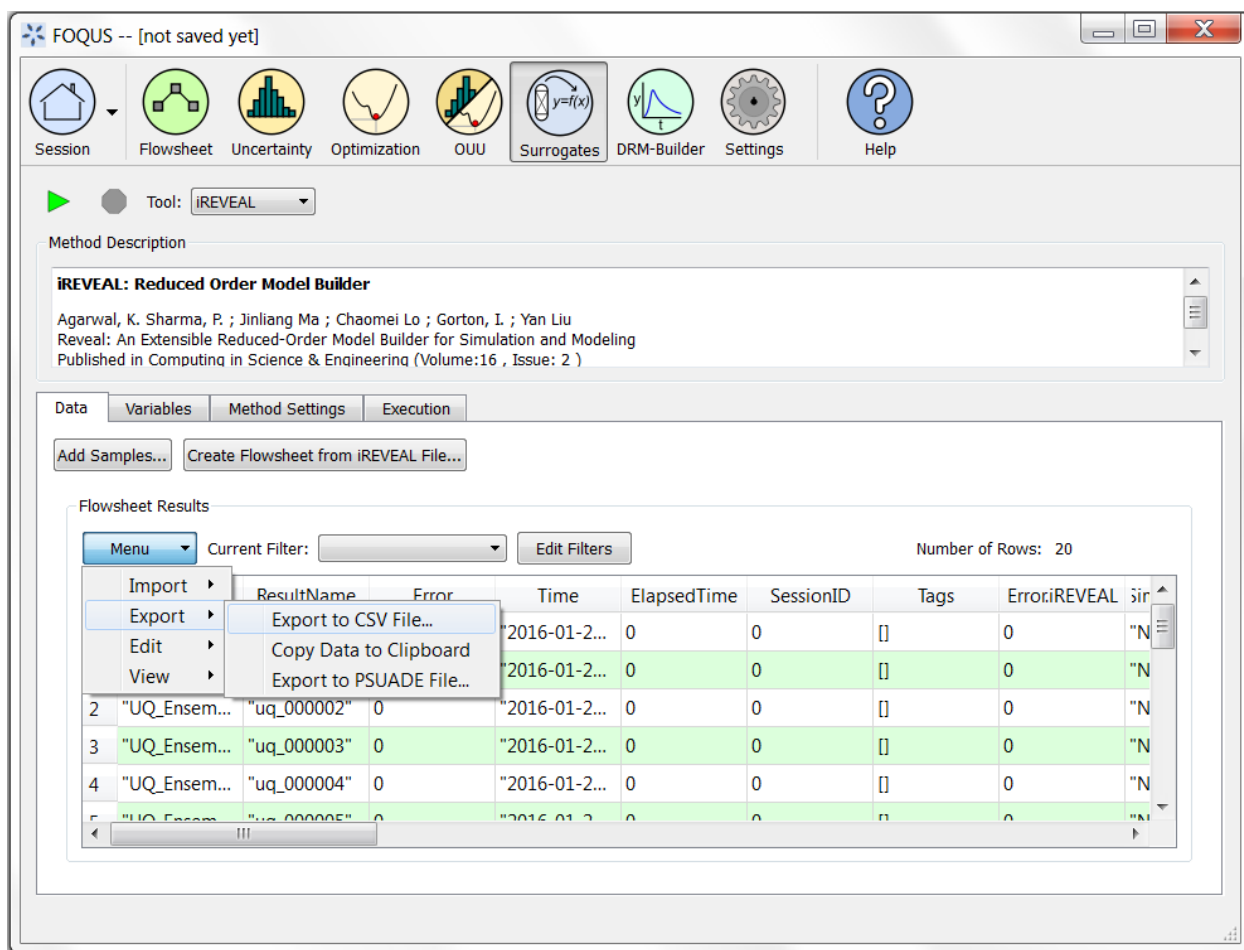


Figure 10: Visualize Response Surface: Export and Import Flowsheet Results

3. **(For the tutorial only)** Copy the input values contained in the “*\$work_dir /rom.in*” file to the results1.csv file column by column. (Please complete step 2 before executing step 3)

Note:

- 1) Step 2 is always needed (copying CFD output parameter data to csv file is not optional)
- 2) Step 3 is optional (copying input parameter data is optional), to be done for this tutorial only since the “psuadeData” file has been replaced by the pre-generated one and the input data displayed in the table has not been updated.
- 3) To make the task of copying a little easier, user can open “rom.in” file in Excel (enable space and tab as delimiters) and copy the data columns easily into matching input columns in results1.csv
- 4) Also, note that the exported csv file, prefixes “Input.iREVEAL” and “Output.iREVEAL” to input parameter names and output parameter names as compared to names in “rom.in” or “results” file. Do not replace the headers in csv file, just copy the data.

- Open the “Data” frame under the “Surrogates” tab. Under the “Menu” option select “Import from csv file” to import the csv file (results1.csv) containing the values from the “results” file.

Navigate to the “Uncertainty” tab and then select “Add New...”. Under the opened dialog select “Use flowsheet” and then click “OK” as shown in Figure 11.

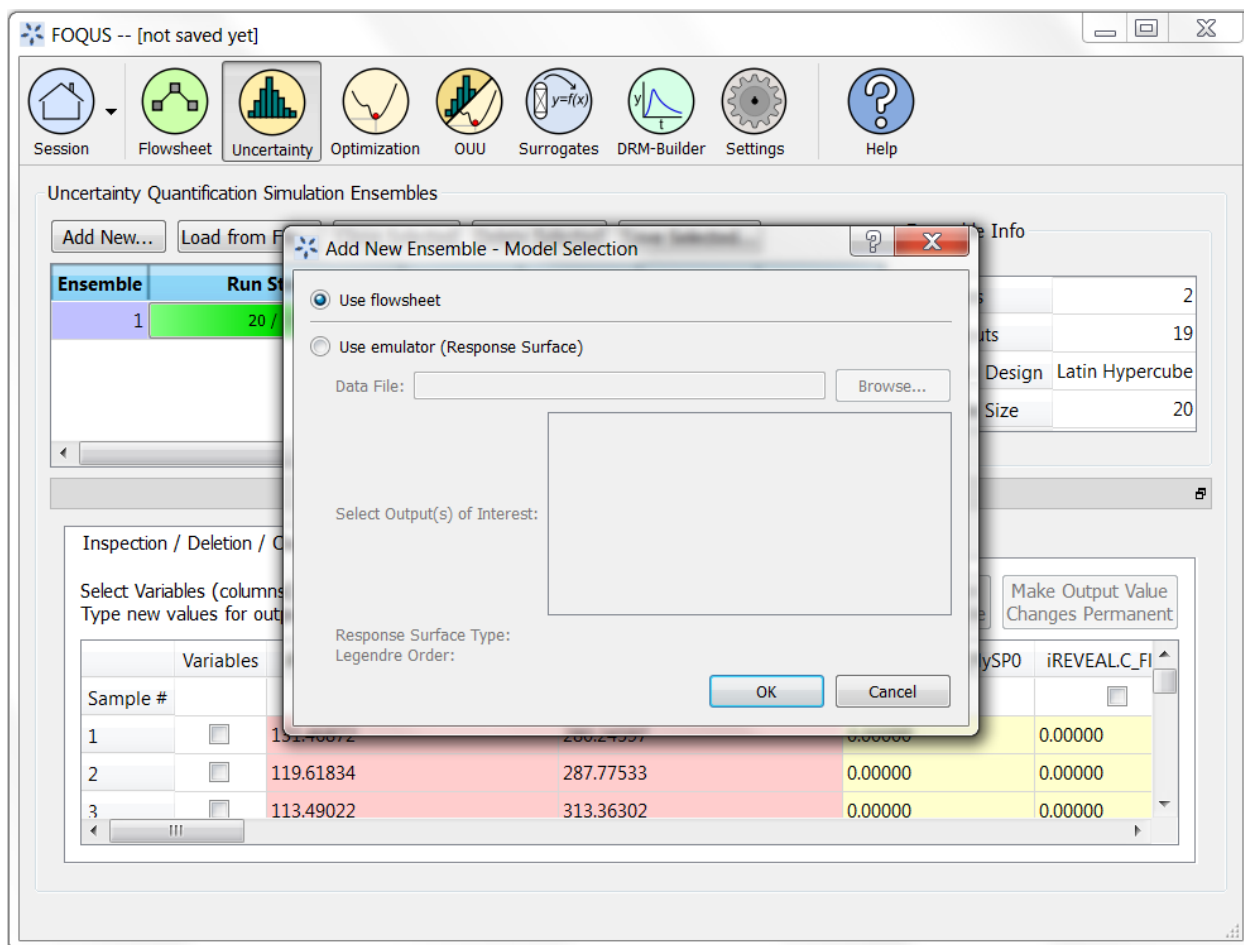


Figure 11: Visualize Response Surface: Create New Ensemble 1

5. In the new opened dialog select the “Load Flowsheet Samples” option and then click “Done”. This creates a new Ensemble in the “Uncertainty” tab as shown in Figure 12.

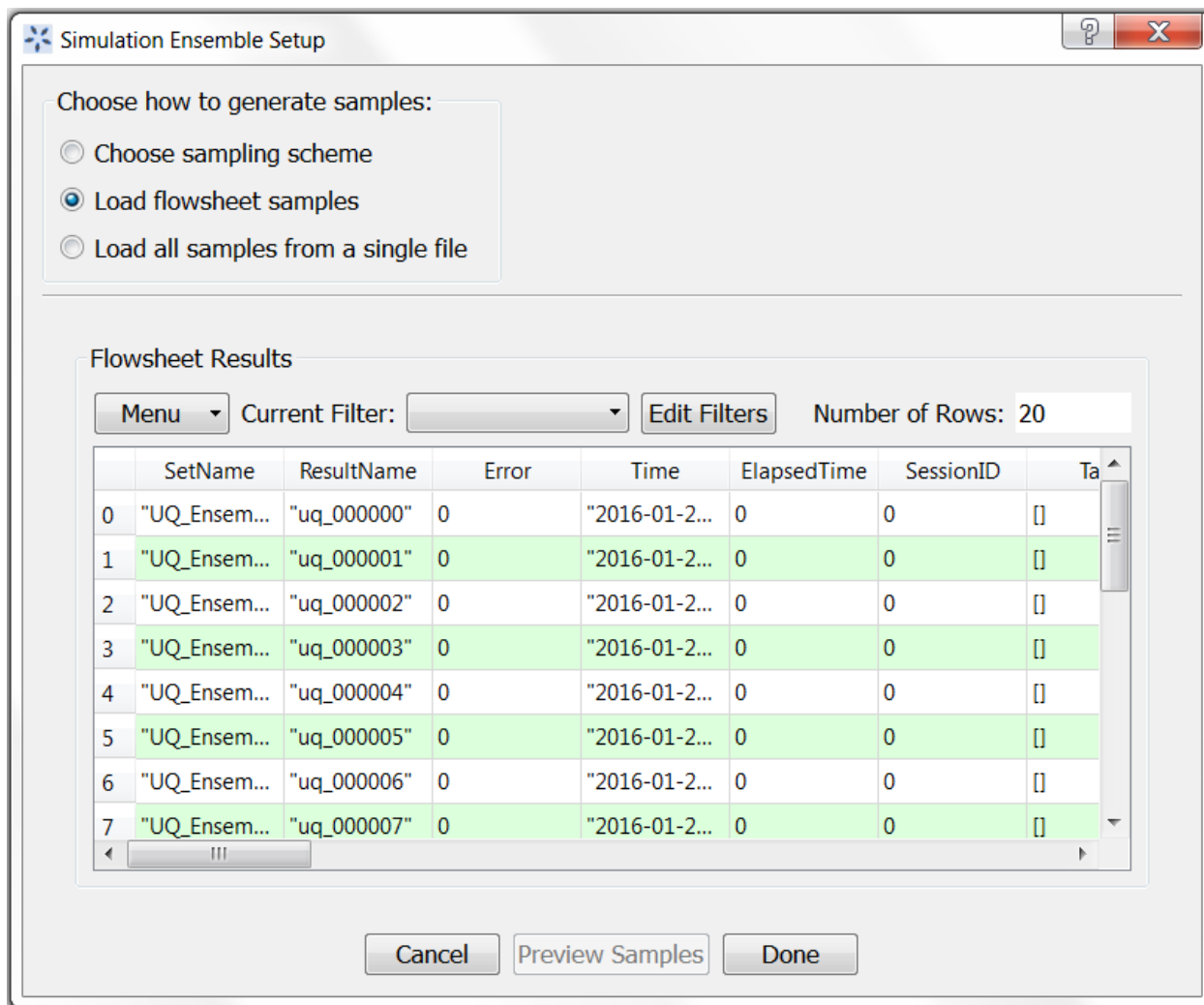


Figure 12: Visualize Response Surface: Create New Ensemble

6. Select the newly created ensemble and then click “Analyze” on the new ensemble (ensemble 2, with descriptor = psuadeData2.tmp) as shown in Figure 13. The “Analysis of Ensemble” dialog window displays.

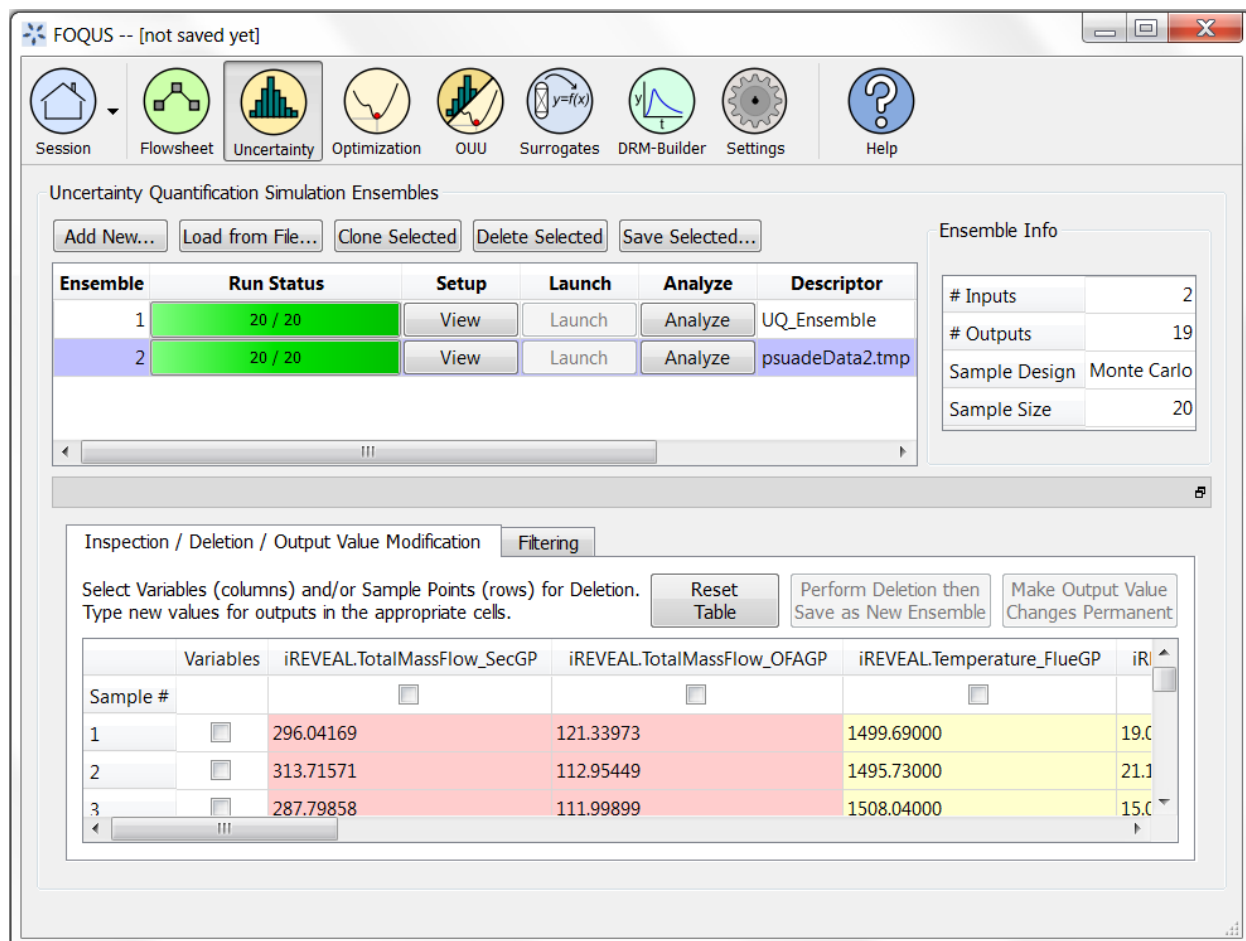


Figure 13: Visualize Response Surface: Select New Ensemble

7. In the “Analysis” section under the “Analysis” frame, select the appropriate output parameters (Item 2) and response surface method (Item 3) as shown in Figure 14. For this tutorial, choose flue gas temperature “iREVEAL.Temperature_FlueGP” as output variable and choose “Kriging” as response surface method. Note: The implementation of “User Regression” to perform user selected response surface method is a work in progress.

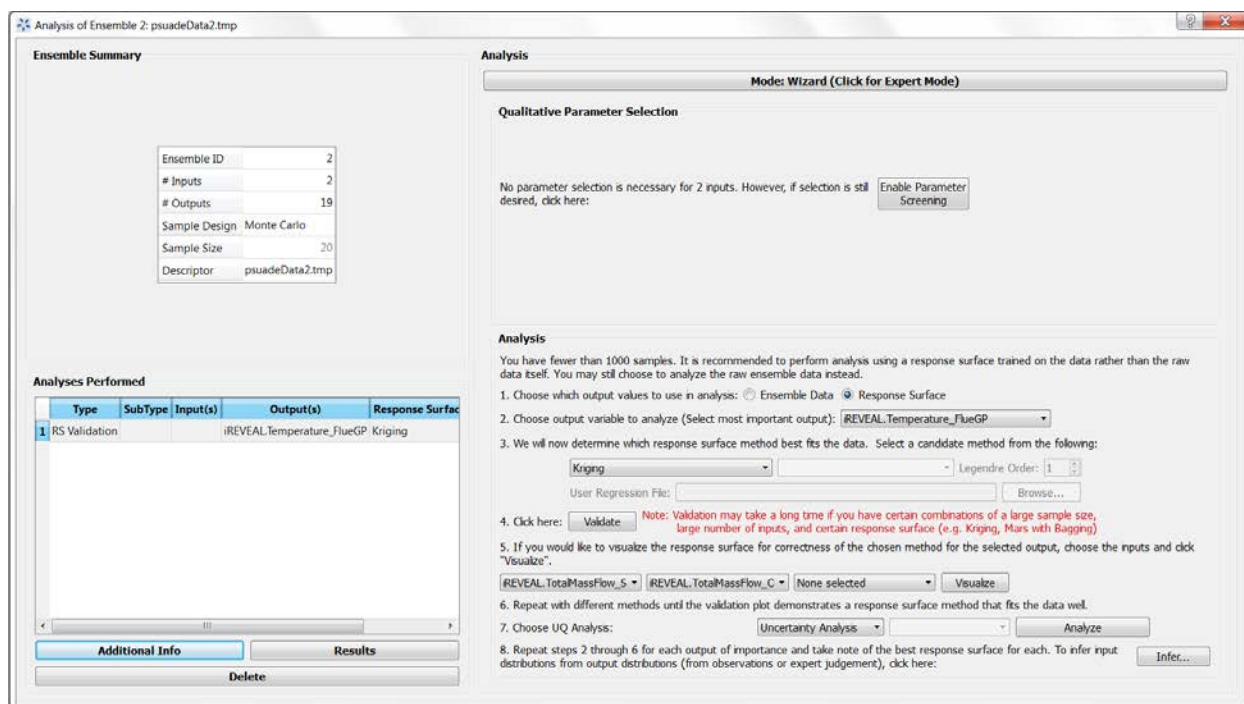


Figure 14: Visualize Response Surface: Analyze Ensemble

8. Click “Validate” to view the “Model Error Histogram” and “Actual Vs Predicted” plots.
 9. To visualize response surface. Select the two input parameters in Item “5”. Click “Visualize” to view the response surface.

Steps Output

On successful execution the system generates the following plots:

1. The results after clicking “Validate” as shown in Figure 15.

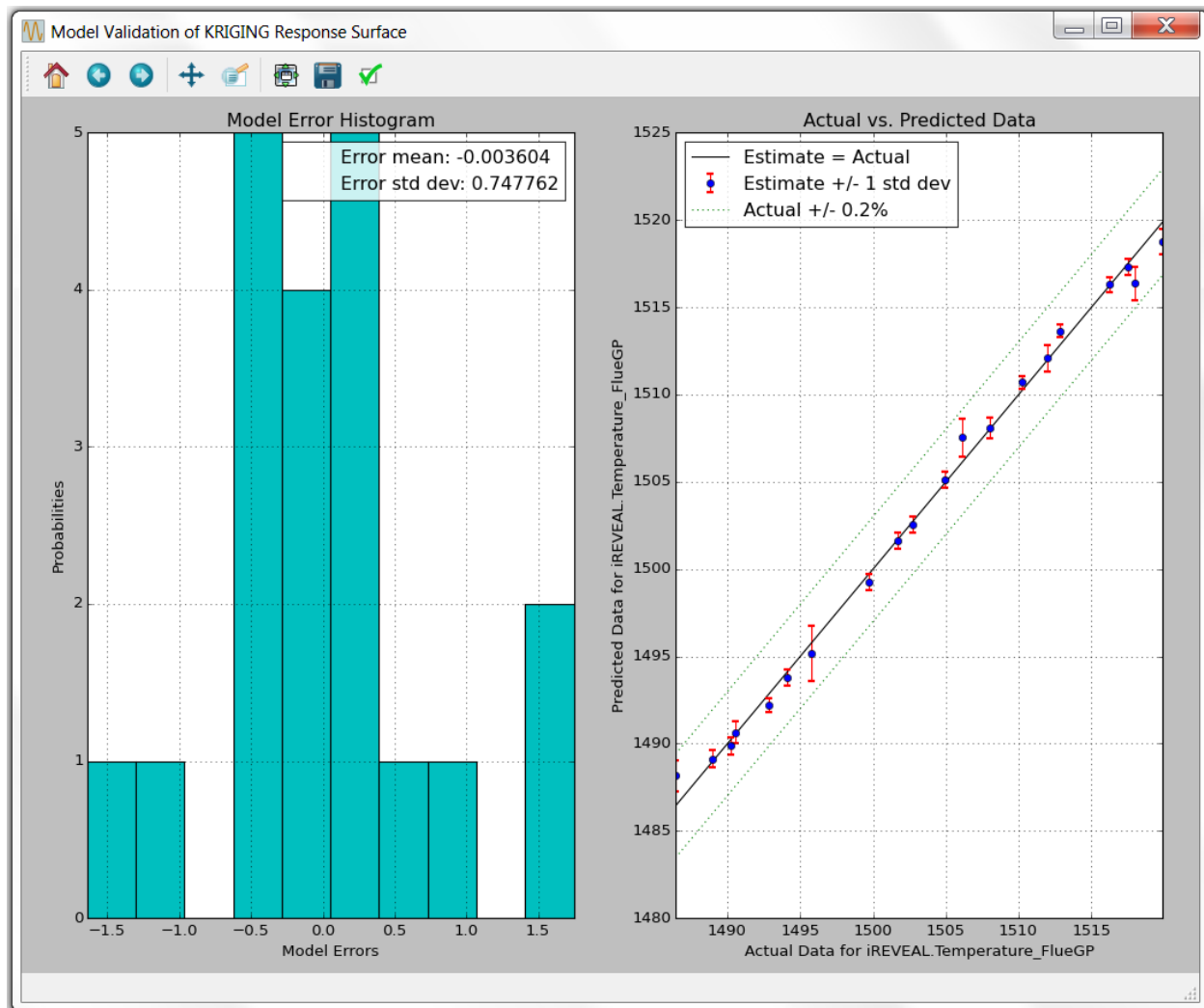


Figure 15: Visualize Response Surface: Validation Plot

- The results after clicking “Visualize” as shown in Figure 16.

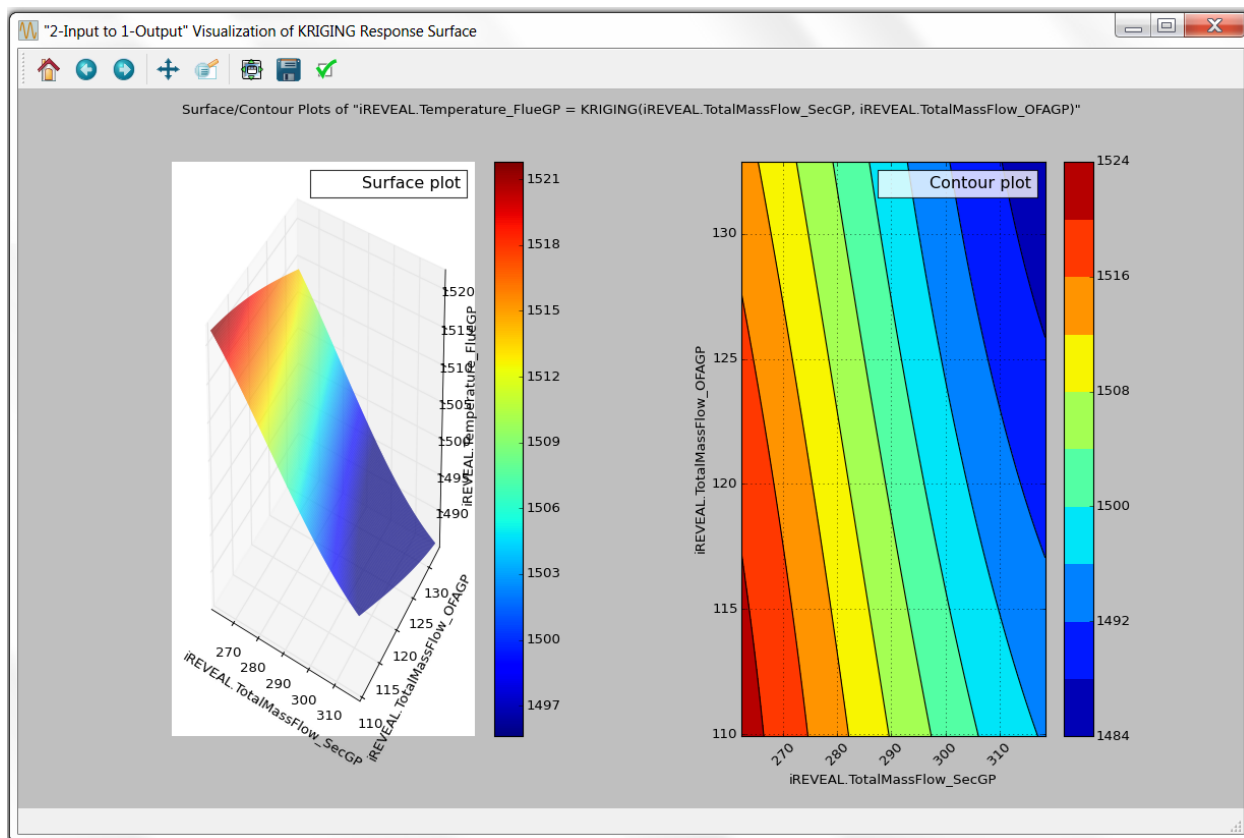


Figure 16: Visualize Response Surface: Response Surface

2.4. Integrating ROM with ACM and Aspen Plus

Description

This section of the tutorial describes the steps to integrating the ROM generated in the previous section with the process simulation packages such as ACM and Aspen Plus. Although the CAPE-OPEN format of the ROM is also generated in the previous section, it is recommended that the ACM format of the ROM be used and embedded to the flowsheet of ACM and Aspen Plus since the ACM format supports solid phase species and electrolyte solutions. Note: Integration of the ROM with Aspen Plus requires the compilation of the ACM file to the dynamic linked library (DLL) and, therefore, requires the installation of a compatible version of Microsoft® Visual Studio® on the computer where ACM is installed. Refer to the document of ACM for the compatible versions of Visual Studio. Microsoft® Visual Studio® 2013 Professional version has been tested to be compatible with Aspen Plus and ACM Versions 8.8. The example used in this tutorial requires Aspen Plus version 8.8 to handle non-conventional solid such as coal and fly ash in the boiler model.

Example

The user can use the “model.acmf” file created in the previous section to perform the integration of the ROM with ACM and Aspen Plus. The user can perform this section of the tutorial in the same or different working directory. The user needs to ensure that the working directory is not write-protected. A simple Aspen Plus example file “boiler_rom.bkp” is also provided in the “examples/iREVEAL” directory of the FOQUS installation. This example file contains the input and output streams that are compatible to the ROM generated in the previous section. The species involved and property models have been specified. The feed streams specified in the Aspen Plus model match the input ranges defined in the JSON configuration file. Note: The Aspen Plus model was created using Aspen Plus v8.8.

Steps

1. Select a working directory for this section of the tutorial and copy the “model.acmf” file from the iREVEAL working directory of the previous section and paste it to the current working directory. Copy the “boiler_rom.bkp” file from the FOQUS installation’s “examples/iREVEAL” folder and paste it to the current working directory.
2. Open the “model.acmf” file by double-clicking the file. The ACM window displays. Expand the “Custom Modeling” branch in the “Exploring – Simu...” window inside the main ACM window. Then expand the “Models” branch. The “Boiler_ROM” item is listed as shown in Figure 17. View the ACM source code by right-clicking the “Boiler_ROM” item and then selecting the “Edit” pop-up menu.

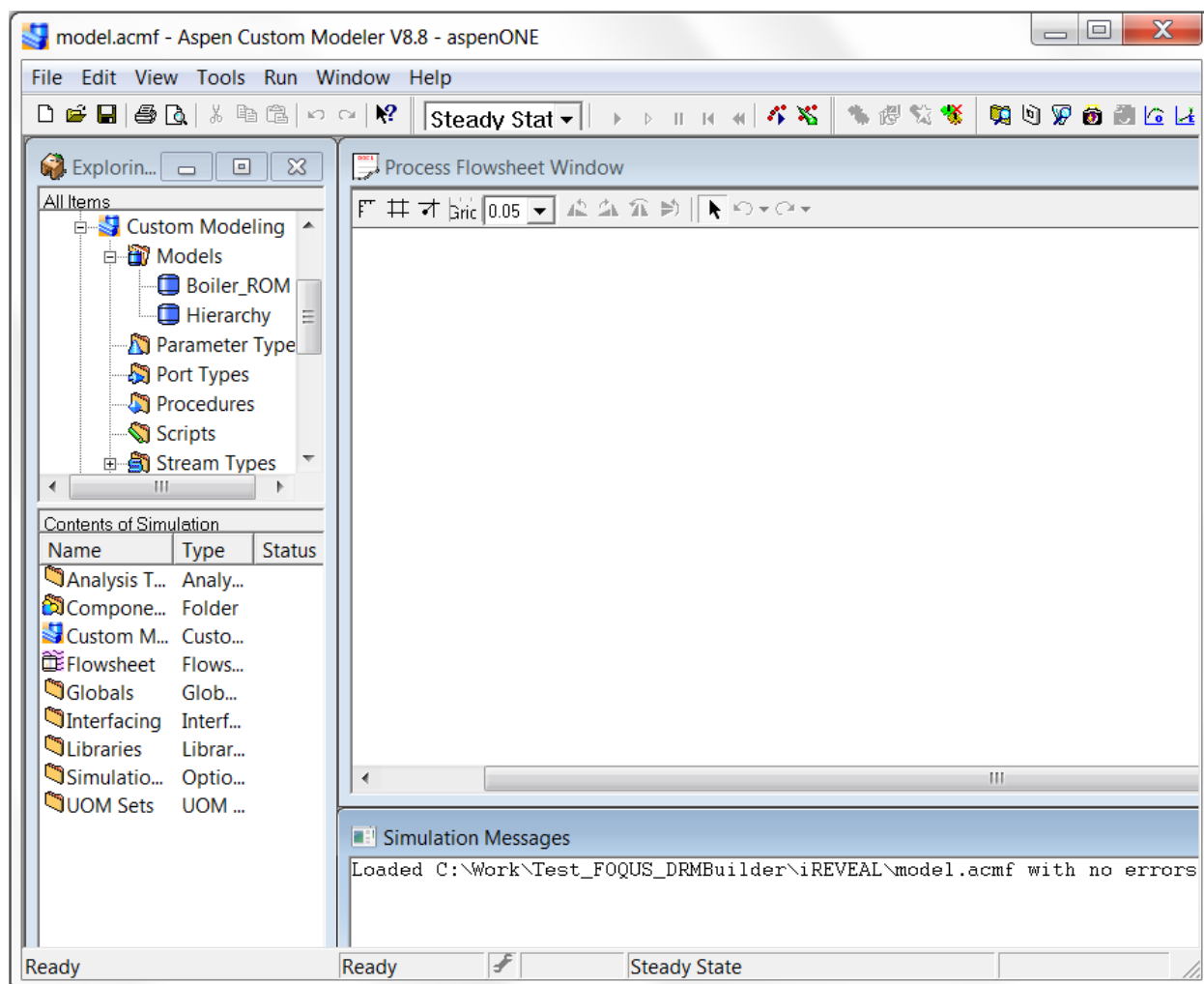


Figure 17: ROM Model Displayed in ACM Window

3. If the Visual Studio C++[®] compiler is installed, the ACM source code exported by iREVEAL can be compiled. Right-click the “Boiler_ROM” item and then select the “Export to DLL...” pop-up menu as shown in Figure 18. An “Export to DLL” window displays, enabling the user to specify a folder to which the exported DLL resides. The user can accept the default folder and then click “OK” to export the DLL. Note: “C:\Windows\System32” has to be included in your computer’s PATH in order to compile the C++ code to generate the DLL.

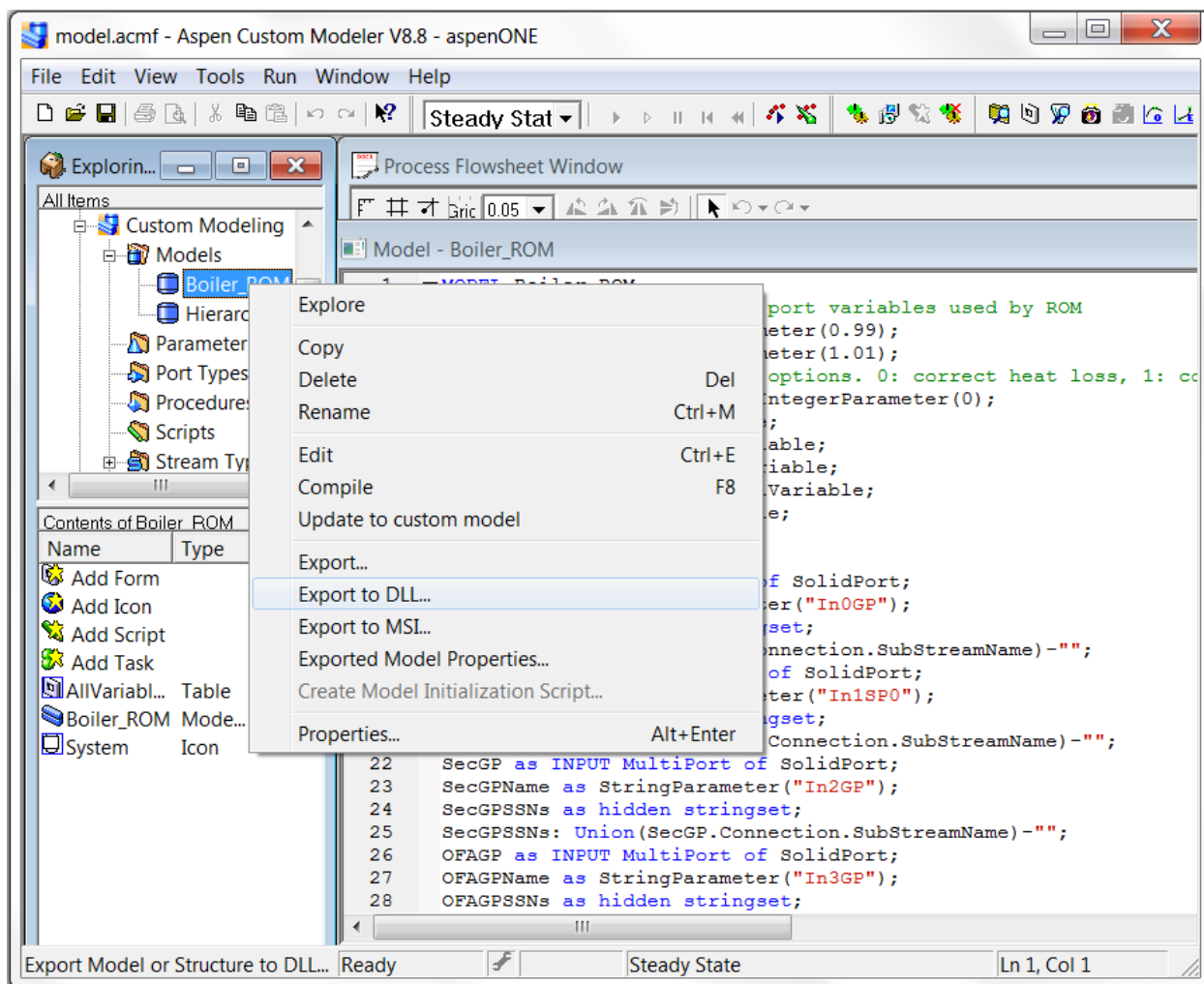


Figure 18: Compiling ACM Source Code and Exporting DLL

4. (Optional) If the user wants to export the DLL and install it to a different computer, the user can select the “Export to MSI...” pop-up menu. The exported file with the “msi” extension is a Microsoft installer file that can be run to install the DLL to a Windows operating system.
5. Open Aspen Plus window by selecting the “boiler_rom.bkp” file. The flowsheet contains four feed streams (FUEL, PA, SA and OFA) and two product streams (EXITGAS and EXITNC). In addition, there is an output heat stream QWALL. The user can browse the species and property setup for the flowsheet. Stream FUEL contains non-conventional

solid named COAL with the properties matching the coal fired at Hunter Unit 3. Streams PA, SA and OFA represent the primary air, secondary air, and overfire air streams, respectively. The flow rates of SA and OFA streams are allowed to be changed since they are set as “Varied” in the ROM generated in this tutorial. Stream EXITGAS represent the flue gas stream and Stream EXITNC represent the non-conventional solid fly ash stream. Heat stream QWALL represents the total heat absorbed the water and steam in the boiler.

- To activate the “Boiler_ROM” model, the user needs to issue the “Manage Libraries” command under the “Customize” menu as shown in Figure 19. The “Manage Libraries” dialog window displays as shown in Figure 20. Select the “ACM Models” check box under the “Available Libraries” list. An “ACM Models” tab is added to the model palette. Close the “Manage Libraries” dialog window. Select the “ACM Models” tab at the right of the palette. An icon named “BoilerROM” is displayed.

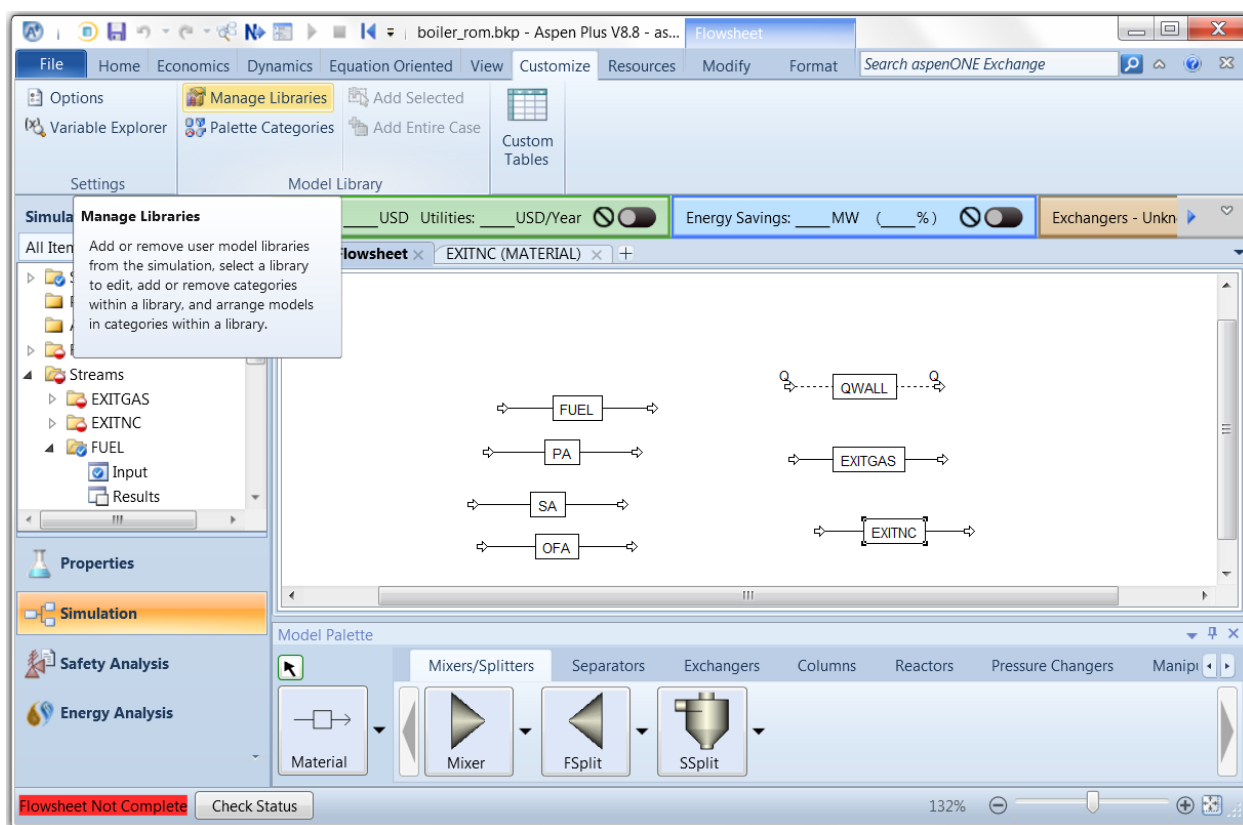


Figure 19: Manage Model Libraries Under Customize Menu

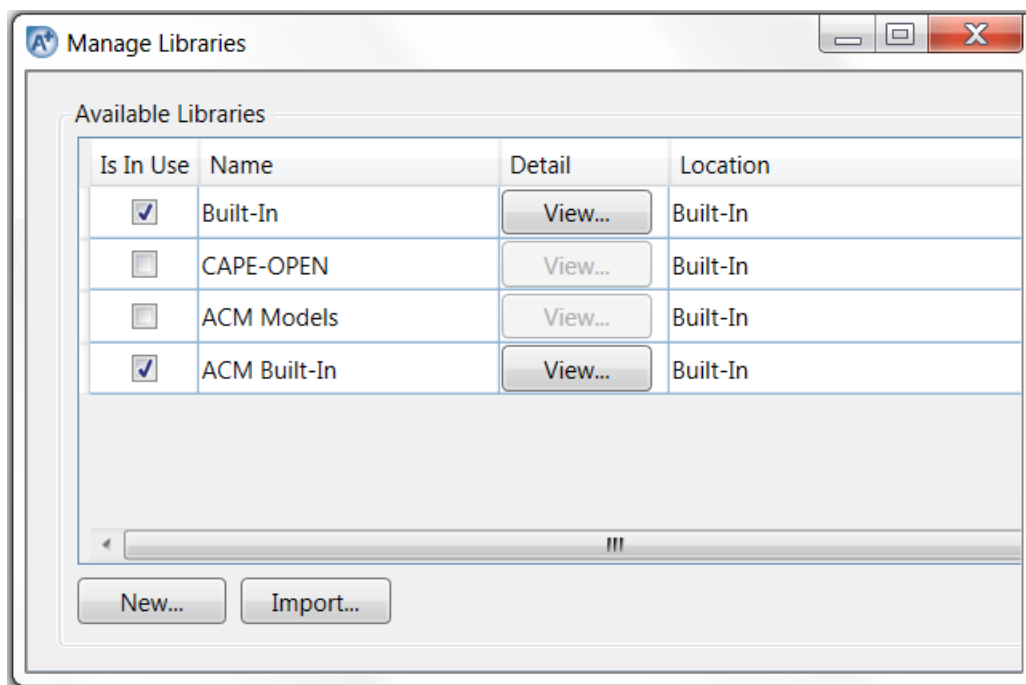


Figure 20: Activating ACM Models in Manage Libraries Window

7. Click the “BoilerROM” icon and drag and drop it to the white space between the four feed streams and the two product streams on the flowsheet. Rename the block to “ROM”.
8. Connect Stream FUEL to the “ROM” block. In the “Universal Port” dialog window that displays, select the “FuelSP0(IN)” port. Connect Stream PA to the “ROM” block. In the “Universal Port” dialog window that displays, select the “PrimGP(IN)” port. Connect Stream SA to the “ROM” block. In the “Universal Port” dialog window that displays, select the “SecGP(IN)” port. Then connect Stream OFA to the “ROM” block. This time no “Universal Port” dialog window displays and Stream OFA is connected to the “OFAGP(IN)” port.
9. Connect Stream EXITGAS to the “ROM” block. In the “Universal Port” dialog window that displays, select the “FlueGP(OUT)” port. Then connect Stream EXITNC to the “ROM” block. No “Universal Port” dialog window displays and Stream EXITNC is connected to the “FlySP0(OUT)” port. Finally, connect heat stream QWALL to the “ROM” block. Figure 21 shows the flow sheet with the streams connected to the “ROM” block.

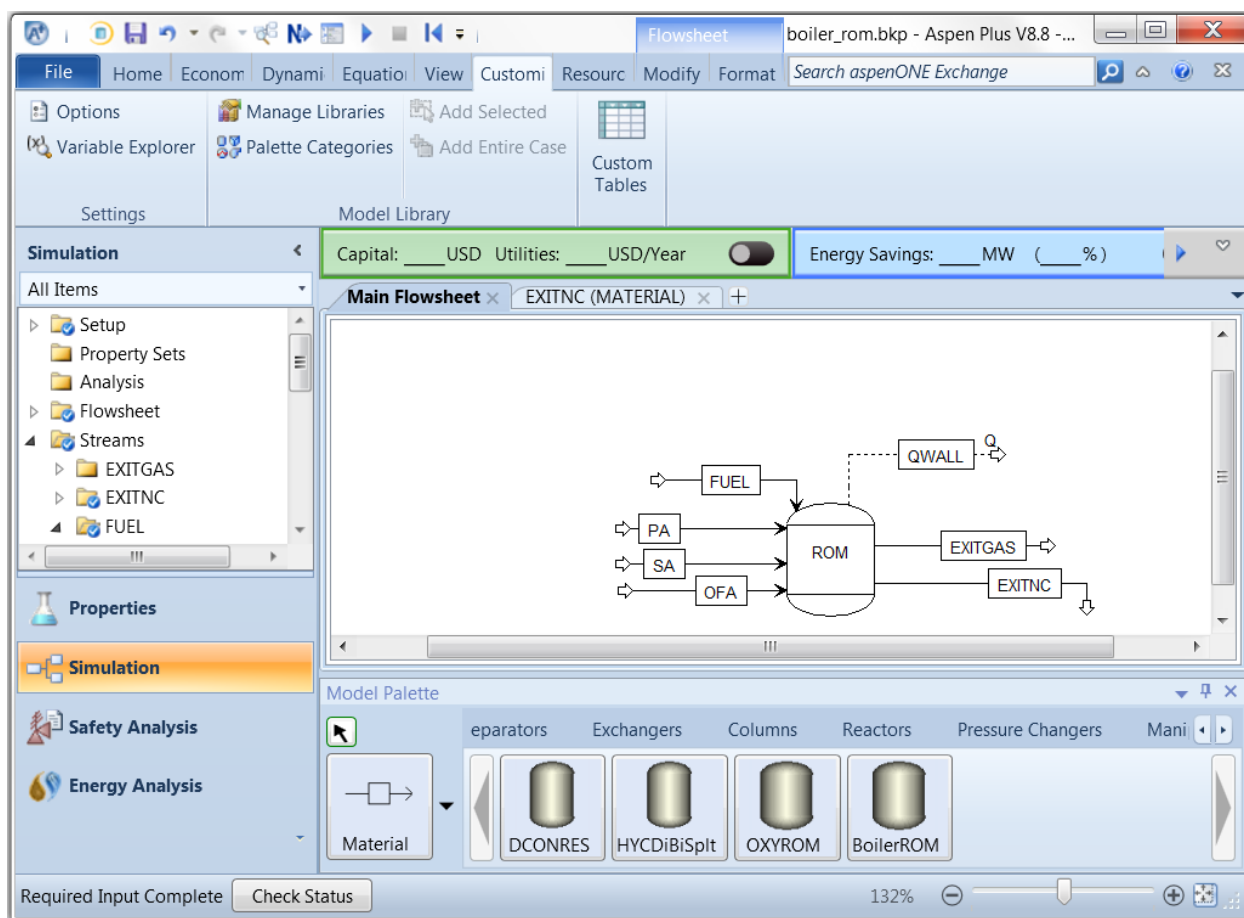


Figure 21: Aspen Plus Flowsheet With Feed and Product Streams Connected to ROM Block

10. After the four feed streams, the two product streams, and the heat stream are connected to the ROM, the “Run” command (the triangle under the “Home” menu) is enabled. Click the “Run” triangle to solve the flowsheet. No warning message is displayed.
11. Due to a minor Aspen Plus bug, the first “Run” command does not calculate the non-conventional solid substream in EXITNC correctly. The user can use a workaround to avoid this bug by modifying an input stream, say OFA, slightly and issuing the “Run” command again. For example, the user can set the mass flow rate of OFA stream to 121.81 and click the “Run” triangle icon.
12. The user can browse the block and product streams including the QWALL stream to view the simulation results. The user can now save the flowsheet. Meanwhile, the user can try to modify the flowrate of SA and OFA streams within the lower and upper limits set for the ROM to see the changes in the product streams and heat output stream. Detailed simulation results can be viewed by clicking on “Blocks/ROM/Variables” item in the “Simulation” tree on the left of the Aspen Plus window.

3. USAGE INFORMATION

3.1. Support

To obtain support for this tool, send an email to ccsi-support@acceleratecarboncapture.org.

3.2. Restrictions

This software is intended to be used for the CFD and other high-fidelity model.

4. ADVANCED FEATURES

The advanced features of the tool have been described in the appropriate subsections of Section 2 Tutorial.

5. DEBUGGING

5.1. How to Debug

The user may experience issues with installation and should perform installation checks mentioned in the *Installation Manual*. If the user experiences a problem the user should:

1. Email the “Python.log” file from the directory that the command line is running from to ccsi-support@acceleratecarboncapture.org.
2. Email the message on the command line window to ccsi-support@acceleratecarboncapture.org.

5.2. Known Issues

A. Code doesn’t run, relative path

If the user does not provide the complete path to \$work_dir and uses the relative path instead it will create problems running the code.

B. Reduced Order Models/Surrogates do not run, text window appear

When FOQUS runs Reduced Order Models (Surrogates) generated by iREVEAL or ALAMO, it runs them as Python code. However, this mechanism relies on Windows file associations. If the .py extension is associated with some program other than Python, that other program will run instead, and the ROM will not run. For example, some people have .py associated with notepad, so they can edit the python code. If this is the case on your machine, when FOQUS attempts to run the ROM, notepad will be opened instead.

To fix this, go to Start->Control Panel. Click “Programs” as in Figure 22.

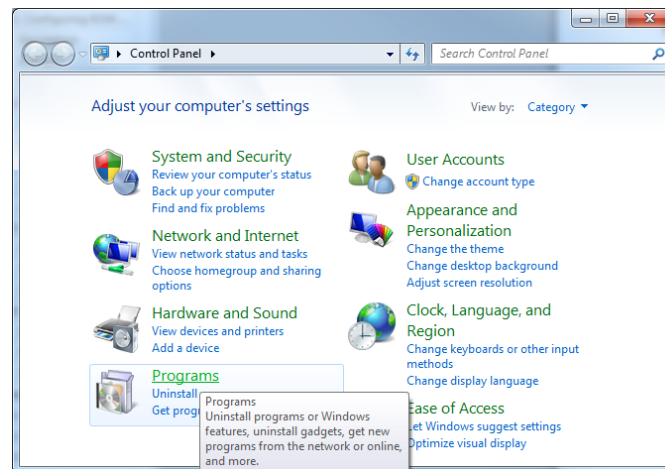


Figure 22: Select Programs in Control Panel

Next select “Make a file type always open in a specific program” as in Figure 23.

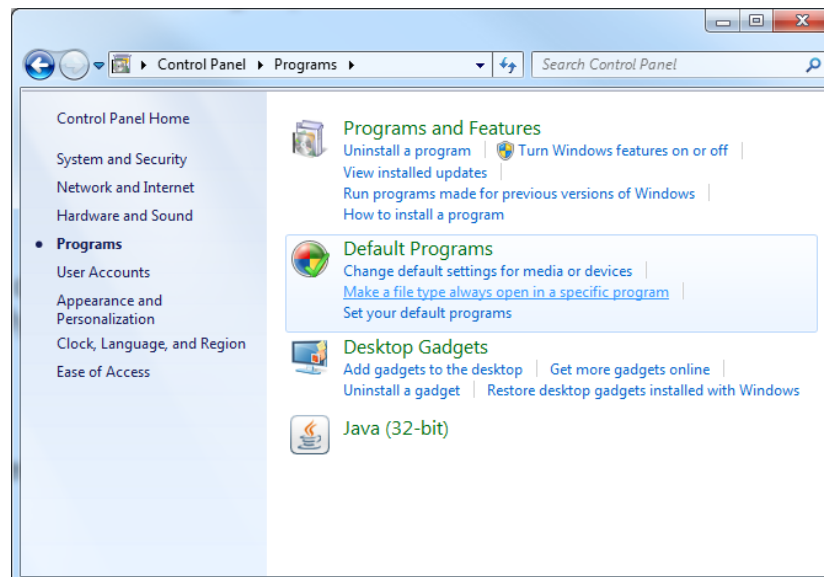


Figure 23: Select “Make a file type always open in a specific program”

Scroll down to “.py” and make sure it is associated with python.exe. In Figure 24 it is not and must be fixed.

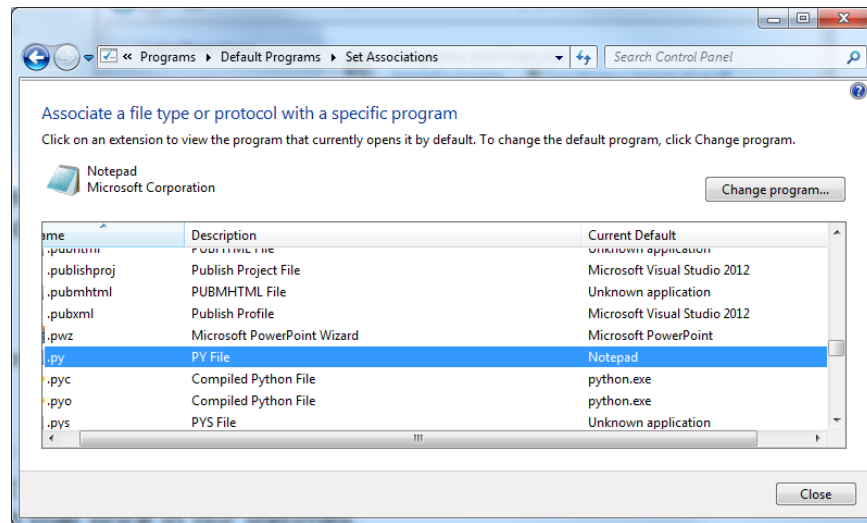


Figure 24: .py is associated with Notepad, that's bad.

Double click on the .py extension. You should be able to select “python.exe” as in Figure 25. If you cannot, click browse and navigate to your python executable, usually at: c:\Python27\python.exe. If python.exe is not in Python27, it was either installed in a different location, or is not installed at all. Please request assistance from your tech support in this case.

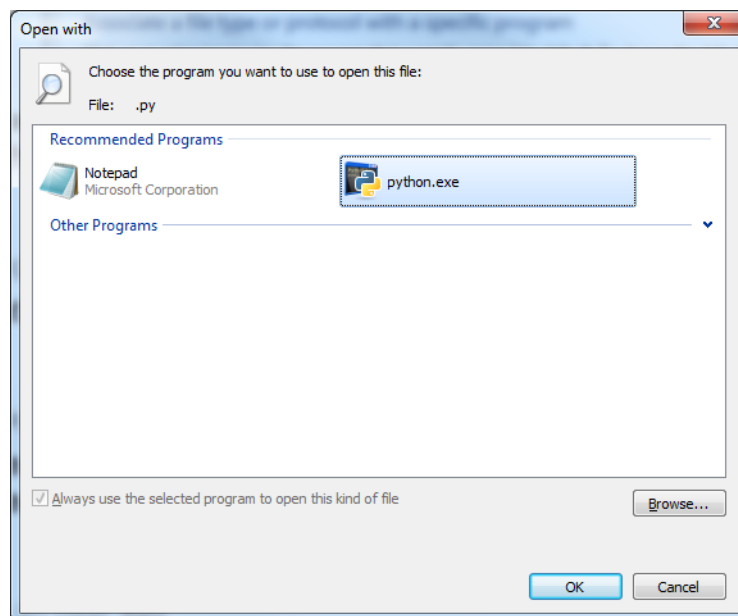


Figure 25: Select Python.exe

.py should now be associated with Python.exe as in Figure 26. Close the Window and try rerunning FOQUS and iREVEAL.

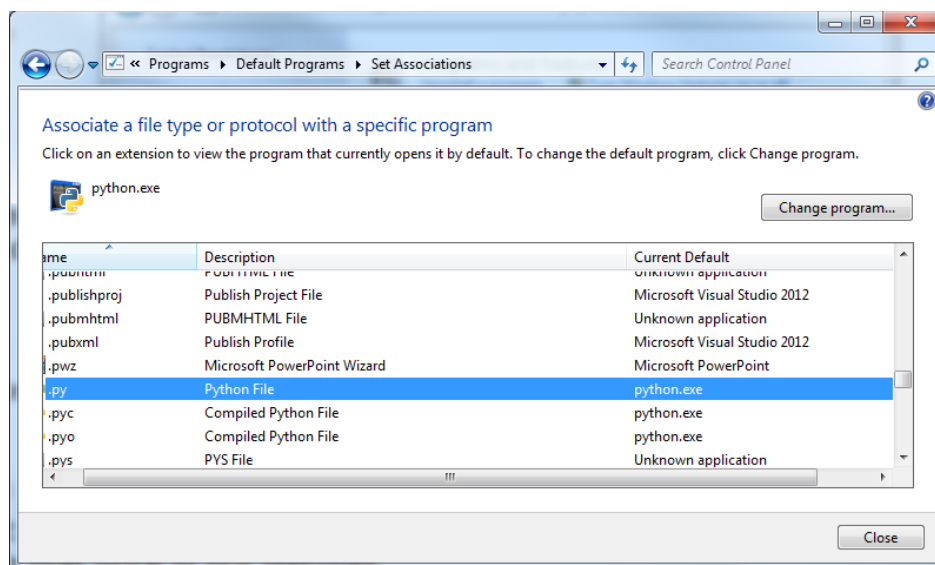


Figure 26: Python now associated with .py

C. C++ compiler not supported or not configured correctly by ACM

Exporting the ACM model to DLL such that it can be used in Aspen Plus as described in Section 2.4 requires a compatible C++ compiler installed. Sometimes a compiler may not be supported or not configured correctly by ACM. Please contact AspenTech support for choosing and configuring the C++ compiler.

5.3. Reporting Issues

To report an issue send an email to ccsi-support@acceleratecarboncapture.org.

6. TERMS

- ACMF – Aspen Custom Modeler File Format
- CFD – Computational Fluid Dynamics
- CO – CAPE-OPEN
- MFIX – Multiphase Fluid Interphase Exchange (<https://mfix.netl.doe.gov/>)
- Model – A new computational fluid dynamics model built for the MFIX simulator
- ROM – Reduced Order Model

7. JSON CONFIGURATION FILE FORMAT

A JSON file is a text file that contains the name and value pairs that are enclosed by a pair of curly brackets “{ }”. The names are simply strings and the values can be primitive literals such as strings and numbers (integer, double), and collections such as array, map, and object. A name and a value in a pair are separated by a column “:”. Multiple name and value pairs are separated by commas “,” with the last one ending without a comma. An array contains a list of values of the same type and those values are separated by commas “,” and enclosed by square brackets “[]”. A map is a list of key/value pairs, separated with a comma “,” with each pair consisting of a unique key and a value separated by a column “:”. The list of entries in the map is enclosed by a pair of curly brackets “{ }”. An object is enclosed by a pair of curly bracket “{ }”, which could contain multiple name and value pairs. A value within an object could be another object, forming a hierarchy of data structure. The entire JSON data can be considered as an object enclosed by a pair of curly brackets “{ }”. The order of the name and value pairs inside the JSON file or inside an object can be arbitrary. Note: An empty array value is allowed by using a pair of square brackets “[]” with nothing enclosed inside.

The iREVEAL configuration file contains some simple properties of the ROM to be generated such as the name of the ROM, the regression method, and the number of samples used to create the ROM. The configuration file also contains the data related to the species involved in the gas/liquid mixture phase and Aspen’s CISOLID-type and NC-type solid phases, the data related to the feed and product material streams, and the special model input and output data. The required names and data types are listed in Table 1. Some data types are object types and are listed in Table 2. If the value corresponding to a name is an array, “Yes” is listed in the “Array” column.

Note: Some name and data pairs for solid phase specifications are required for some specific CFD models. In case they are irrelevant to a specific CFD model, they still need to be included in the configuration file but the values could be arbitrary since they are not used by iREVEAL. For example, granular temperature is only applicable to CFD models such as MFIX that use the Eulerian framework to model dense particle phases. The user still needs to provide some arbitrary value for the “glannularTemperature” parameter even if a CFD model does not use the Eulerian framework to solve the particle phase fluid dynamics. Likewise, the volatile mater content is only valid for Aspen’s NC (non-conventional) solid material such as coal, the parameter “vm” still needs to be specified even if only Aspen’s CISOLID solid phase is involved.

iREVEAL treats all floating point (double) inputs as a “Parameter” object type that contains a flag to indicate if the parameter is “varied”, a default value, a minimum value, a maximum value, and a parameter name as a string (see the “Parameter” object description at the end of Table 2). If the flag “isVaried” is “False”, the parameter is “fixed” and it is not built in the mapping function from inputs to outputs. However, the parameter still needs to be specified in the iREVEAL configuration file. The “defaultValue” of the parameter is used for the integration of the ROM with process modeling software such as ACM and Aspen Plus. If the fixed parameter is a feed stream property such as temperature, it is used to check if the feed stream temperature matches the temperature parameter of the ROM. If the fixed parameter is a product stream

property such as product temperature, the mapping function inside the ROM does not include that product stream temperature and the “defaultValue” of the parameter is used and assigned to the product stream property. Note: The default, minimum, and maximum values in the “Parameter” object are in SI unit.

Table 1: Names and Types of the Data in the iREVEAL Configuration File

Name	Data Type	Array	Description
name	string	No	The name of the ROM to be generated.
regMethod	string	No	Regression method. Either “KRIGING” or “ANN”.
nSample	integer	No	The number of samples used to generate the ROM.
gasSpeciesList	Species object	Yes	A list of species in the gas/liquid mixture phase. Note: The Gas phase in iREVEAL actually corresponds to the gas/liquid “MIXED” phase in Aspen Plus and ACM.
solidSpeciesList	Species object	Yes	A list of species in a solid phase (CISOLID or NC).
solidPhaseList	string	Yes	A list of solid phase names.
solidPhaseTypeList	integer	Yes	“0” if CISOLID or “1” if NC solid.
inletBoundaryList	Boundary object	Yes	A list of inlet boundary condition data, corresponding to feed streams.
outletBoundaryList	Boundary object	Yes	A list of outlet boundary condition data, corresponding to product streams.
inputParameterList	Parameter object	Yes	A list of extra model input parameters that are not related to the inlet boundary data. The list could be empty.
outputParameterList	Parameter object	Yes	A list of extra model output parameters that are not related to the outlet boundary data. The list could be empty.

Table 2: Descriptions of Object Types Used in the iREVEAL Configuration File

Object Type	Name	Data Type	Array	Description
Species	name	string	No	The name of a species.
	formula	string	No	The molecular formula of a species.
Boundary	hasGasPhase	boolean	No	“True” if the material stream through the boundary contains a gas/liquid mixture phase.
	hasSolidPhase	boolean	No	“True” if the material stream through the boundary contains at least a solid phase.
	boundaryIndex	integer	No	The boundary index used for certain CFD solvers such as MFIX. This input is for the user’s reference only and does not affect the generated ROM.
	boundaryName	string	No	The name of the boundary or stream.
	gasMixture	GasMixture object	Yes	The object represents a gas/liquid mixture phase. There is always only one element in the array.
	solidMixtures	SolidMixture object	Yes	The object represents a solid mixture phase. A boundary could have one or more solid mixtures.
GasMixture	pressure	Parameter object	No	The pressure of a feed or product stream.
	temperature	Parameter object	No	The temperature of a feed or product stream.
	volumeFraction	Parameter object	No	The volume fraction of the gas/liquid mixture phase. It should be “1.0” if there is no solid phase.
	hasConstMass Fractions	boolean	No	“True” if the mass fractions of the species in the mixture is “fixed”.
	totalMassFlow	Parameter object	No	The total mass flow rate through the boundary.
	speciesFlow Map	FlowMap object	Yes	If “hasConstMass” is “True”, this contains the mass fraction map. Otherwise it contains the map of the mass flow rates of the individual species. The key for the map is 0-based species index in the gas/liquid mixture phase.

Object Type	Name	Data Type	Array	Description
SolidMixture	phaseIndex	integer	No	The 0-based index of solid phases listed in “solidPhaseList”.
	solidType	integer	No	The 0-based index of solid phase types in “solidPhaseTypeList”.
	isGranular EnergySolved	boolean	No	A flag to indicate if the granular energy equation is solved, applicable to some CFD such as MFIX.
	pressure	Parameter object	No	The pressure of a feed or product stream.
	temperature	Parameter object	No	The temperature of a feed or product stream.
	volumeFraction	Parameter object	No	The volume fraction of the gas/liquid mixture phase. It should be “1.0” if there is one solid phase and there is no gas/liquid mixture phase.
	diameter	Parameter object	No	The diameter of particles in the solid phase.
	density	Parameter object	No	The density of particles in the solid phase.
	granular Temperature	Parameter object	No	The granular temperature parameter, used by some CFD such as MFIX, to describe the dense solid phase.
	vm	Parameter object	No	The volatile mater mass fraction on an “as received” basis, used for NC solid only.
	hasConstMass Fractions	boolean	No	“True” if the mass fractions of species in the mixture is “fixed”.
	speciesFlow Map	FlowMap object	Yes	If “hasConstMass” is “True”, this contains the mass fraction map. Otherwise it contains the map of the mass flow rates of the individual species. The key for the map is 0-based species index in the corresponding solid phase.
FlowMap	speciesIndex	integer	No	The species index (0-based integer) in the species list corresponding to the gas/liquid or solid phase.

Object Type	Name	Data Type	Array	Description
	speciesFlowRate	Parameter object	No	If the “hasConstMassFractions” is “True”, the parameter is mass fraction. Otherwise, the parameter is the mass flow rate of the species.
Parameter	isVaried	boolean	No	“True” if the parameter is “varied” and “False” if “fixed”.
	defaultValue	double	No	The default value of the parameter if “fixed”.
	minValue	double	No	The minimum value of the parameter, used when “isVaried” is “True”.
	maxValue	double	No	The maximum value of the parameter, used when “isVaried” is “True”.
	name	double	No	The name of the parameter.

The example iREVEAL configuration file for the boiler model in the tutorial containing gas and non-conventional solid phases is listed below.

```
{
  "regMethod": "KRIGING",
  "nSample": 20,
  "gasSpeciesList": [
    {
      "name": "O2",
      "formula": "O2"
    },
    {
      "name": "N2",
      "formula": "N2"
    },
    {
      "name": "H2O",
      "formula": "H2O"
    },
    {
      "name": "CO2",
      "formula": "CO2"
    },
    {
      "name": "CO",
      "formula": "CO"
    },
    {
      "name": "SO2",
      "formula": "SO2"
    },
    {
      "name": "HCl",
      "formula": "HCl"
    }
  ]
}
```

```

    }
  ],
  "solidSpeciesList": [
  [
    {
      "name": "C",
      "formula": "C"
    },
    {
      "name": "H",
      "formula": "H"
    },
    {
      "name": "N",
      "formula": "N"
    },
    {
      "name": "Cl",
      "formula": "Cl"
    },
    {
      "name": "S",
      "formula": "S"
    },
    {
      "name": "O",
      "formula": "O"
    },
    {
      "name": "Ah",
      "formula": "Ah"
    },
    {
      "name": "H2O",
      "formula": "H2O"
    }
  ]
  ],
  "solidPhaseList": [
    "COAL"
  ],
  "solidPhaseTypeList": [
    1
  ],
  "inletBoundaryList": [
    {
      "hasGasPhase": true,
      "hasSolidPhase": false,
      "boundaryIndex": 1,
      "boundaryName": "Prim",
      "gasMixture": [
        {
          "pressure": {
            "isVaried": false,
            "defaultValue": 86126.3,
            "minValue": 86126.3,
            "maxValue": 86126.3,

```

```

    "name": "Pressure"
  },
  "temperature": {
    "isVaried": false,
    "defaultValue": 338.7,
    "minValue": 338.7,
    "maxValue": 338.7,
    "name": "Temperature"
  },
  "volumeFraction": {
    "isVaried": false,
    "defaultValue": 1.0,
    "minValue": 0.0,
    "maxValue": 0.0,
    "name": "VolumeFraction"
  },
  "hasConstMassFractions": true,
  "totalMassFlow": {
    "isVaried": false,
    "defaultValue": 99.2,
    "minValue": 99.2,
    "maxValue": 99.2,
    "name": "TotalMassFlow"
  },
  "speciesFlowMap": {
    "0": {
      "isVaried": false,
      "defaultValue": 0.23003,
      "minValue": 0.23003,
      "maxValue": 0.23003,
      "name": "O2_mass_fraction"
    },
    "1": {
      "isVaried": false,
      "defaultValue": 0.759625,
      "minValue": 0.759625,
      "maxValue": 0.759625,
      "name": "N2_mass_fraction"
    },
    "2": {
      "isVaried": false,
      "defaultValue": 0.0098313,
      "minValue": 0.0098313,
      "maxValue": 0.0098313,
      "name": "H2O_mass_fraction"
    },
    "3": {
      "isVaried": false,
      "defaultValue": 0.000513497,
      "minValue": 0.000513497,
      "maxValue": 0.000513497,
      "name": "CO2_mass_fraction"
    }
  }
},
],
"solidMixtures": []

```

```

    },
    {
      "hasGasPhase": false,
      "hasSolidPhase": true,
      "boundaryIndex": 2,
      "boundaryName": "Fuel",
      "gasMixture": [],
      "solidMixtures": [
        {
          "phaseIndex": 0,
          "solidType": 1,
          "isGranularEnergySolved": false,
          "pressure": {
            "isVaried": false,
            "defaultValue": 86126.3,
            "minValue": 101325.0,
            "maxValue": 101325.0,
            "name": "Pressure"
          },
          "temperature": {
            "isVaried": false,
            "defaultValue": 338.7,
            "minValue": 338.7,
            "maxValue": 338.7,
            "name": "Temperature"
          },
          "volumeFraction": {
            "isVaried": false,
            "defaultValue": 1.0,
            "minValue": 0.0,
            "maxValue": 1.0,
            "name": "VolumeFraction"
          },
          "diameter": {
            "isVaried": false,
            "defaultValue": 0.0001,
            "minValue": 0.0,
            "maxValue": 0.1,
            "name": "Diameter"
          },
          "density": {
            "isVaried": false,
            "defaultValue": 1350.0,
            "minValue": 0.0,
            "maxValue": 1400.0,
            "name": "Density"
          },
          "granularTemperature": {
            "isVaried": false,
            "defaultValue": 1.0,
            "minValue": 0.0,
            "maxValue": 1.0,
            "name": "granularTemperature"
          },
          "vm": {
            "isVaried": false,
            "defaultValue": 40,

```

```

    "minValue": 0.0,
    "maxValue": 50.0,
    "name": "VolatileMatter"
  },
  "hasConstMassFractions": true,
  "totalMassFlow": {
    "isVaried": false,
    "defaultValue": 50.4,
    "minValue": 45.36,
    "maxValue": 55.44,
    "name": "TotalMassFlow"
  },
  "speciesFlowMap": {
    "0": {
      "isVaried": false,
      "defaultValue": 0.6449,
      "minValue": 0.1,
      "maxValue": 1.0,
      "name": "C"
    },
    "1": {
      "isVaried": false,
      "defaultValue": 0.0444,
      "minValue": 0.1,
      "maxValue": 1.0,
      "name": "H"
    },
    "2": {
      "isVaried": false,
      "defaultValue": 0.0118,
      "minValue": 0.1,
      "maxValue": 1.0,
      "name": "N"
    },
    "3": {
      "isVaried": false,
      "defaultValue": 0.0001,
      "minValue": 0.0,
      "maxValue": 1.0,
      "name": "Cl"
    },
    "4": {
      "isVaried": false,
      "defaultValue": 0.0063,
      "minValue": 0.1,
      "maxValue": 1.0,
      "name": "S"
    },
    "5": {
      "isVaried": false,
      "defaultValue": 0.0925,
      "minValue": 0.1,
      "maxValue": 1.0,
      "name": "O"
    },
    "6": {
      "isVaried": false,

```

```

        "defaultValue": 0.1114,
        "minValue": 0.1,
        "maxValue": 1.0,
        "name": "Ah"
    },
    "7": {
        "isVaried": false,
        "defaultValue": 0.0886,
        "minValue": 0.1,
        "maxValue": 1.0,
        "name": "H2O"
    }
}
]
},
{
    "hasGasPhase": true,
    "hasSolidPhase": false,
    "boundaryIndex": 3,
    "boundaryName": "Sec",
    "gasMixture": [
        {
            "pressure": {
                "isVaried": false,
                "defaultValue": 86126.3,
                "minValue": 86126.3,
                "maxValue": 86126.3,
                "name": "Pressure"
            },
            "temperature": {
                "isVaried": false,
                "defaultValue": 548.7,
                "minValue": 350.0,
                "maxValue": 600.0,
                "name": "Temperature"
            },
            "volumeFraction": {
                "isVaried": false,
                "defaultValue": 1.0,
                "minValue": 0.0,
                "maxValue": 0.0,
                "name": "VolumeFraction"
            },
            "hasConstMassFractions": true,
            "totalMassFlow": {
                "isVaried": true,
                "defaultValue": 289.6,
                "minValue": 260.64,
                "maxValue": 318.56,
                "name": "TotalMassFlow"
            },
            "speciesFlowMap": {
                "0": {
                    "isVaried": false,
                    "defaultValue": 0.23003,
                    "minValue": 0.23003,

```



```

        "maxValue": 0.23003,
        "name": "O2_MassFraction"
    },
    "1": {
        "isVaried": false,
        "defaultValue": 0.759625,
        "minValue": 0.759625,
        "maxValue": 0.759625,
        "name": "N2_MassFraction"
    },
    "2": {
        "isVaried": false,
        "defaultValue": 0.0098313,
        "minValue": 0.0098313,
        "maxValue": 0.0098313,
        "name": "H2O_MassFraction"
    },
    "3": {
        "isVaried": false,
        "defaultValue": 0.000513497,
        "minValue": 0.000513497,
        "maxValue": 0.000513497,
        "name": "CO2_MassFraction"
    }
}
],
"solidMixtures": []
},
{
    "hasGasPhase": true,
    "hasSolidPhase": false,
    "boundaryIndex": 4,
    "boundaryName": "OFA",
    "gasMixture": [
        {
            "pressure": {
                "isVaried": false,
                "defaultValue": 86126.3,
                "minValue": 86126.3,
                "maxValue": 86126.3,
                "name": "Pressure"
            },
            "temperature": {
                "isVaried": false,
                "defaultValue": 548.7,
                "minValue": 350.0,
                "maxValue": 600.0,
                "name": "Temperature"
            },
            "volumeFraction": {
                "isVaried": false,
                "defaultValue": 1.0,
                "minValue": 0.0,
                "maxValue": 0.0,
                "name": "VolumeFraction"
            }
        }
    ]
}

```

```

    "hasConstMassFractions": true,
    "totalMassFlow": {
      "isVaried": true,
      "defaultValue": 121.8,
      "minValue": 109.62,
      "maxValue": 133.98,
      "name": "TotalMassFlow"
    },
    "speciesFlowMap": {
      "0": {
        "isVaried": false,
        "defaultValue": 0.23003,
        "minValue": 0.23003,
        "maxValue": 0.23003,
        "name": "O2_MassFraction"
      },
      "1": {
        "isVaried": false,
        "defaultValue": 0.759625,
        "minValue": 0.759625,
        "maxValue": 0.759625,
        "name": "N2_MassFraction"
      },
      "2": {
        "isVaried": false,
        "defaultValue": 0.0098313,
        "minValue": 0.0098313,
        "maxValue": 0.0098313,
        "name": "H2O_MassFraction"
      },
      "3": {
        "isVaried": false,
        "defaultValue": 0.000513497,
        "minValue": 0.000513497,
        "maxValue": 0.000513497,
        "name": "CO2_MassFraction"
      }
    }
  ],
  "solidMixtures": []
},
"outletBoundaryList": [
  {
    "hasGasPhase": true,
    "hasSolidPhase": false,
    "boundaryIndex": 5,
    "boundaryName": "Flue",
    "gasMixture": [
      {
        "pressure": {
          "isVaried": false,
          "defaultValue": 86126.3,
          "minValue": 86126.3,
          "maxValue": 86126.3,
          "name": "Pressure"
        }
      }
    ]
  }
]

```

```

},
"temperature": {
  "isVaried": true,
  "defaultValue": 298.15,
  "minValue": 298.15,
  "maxValue": 2500.0,
  "name": "Temperature"
},
"volumeFraction": {
  "isVaried": false,
  "defaultValue": 1.0,
  "minValue": 0.0,
  "maxValue": 1.0,
  "name": "VolumeFraction"
},
"hasConstMassFractions": false,
"totalMassFlow": {
  "isVaried": false,
  "defaultValue": 0.0,
  "minValue": 0.0,
  "maxValue": 0.0,
  "name": "TotalMassFlow"
},
"speciesFlowMap": {
  "0": {
    "isVaried": true,
    "defaultValue": 0.0,
    "minValue": 0.0,
    "maxValue": 0.1,
    "name": "O2_MassFlow"
  },
  "1": {
    "isVaried": true,
    "defaultValue": 0.0,
    "minValue": 0.0,
    "maxValue": 2.0,
    "name": "N2_MassFlow"
  },
  "2": {
    "isVaried": true,
    "defaultValue": 0.0,
    "minValue": 0.0,
    "maxValue": 0.3,
    "name": "H2O_MassFlow"
  },
  "3": {
    "isVaried": true,
    "defaultValue": 0.0,
    "minValue": 0.0,
    "maxValue": 0.3,
    "name": "CO2_MassFlow"
  },
  "4": {
    "isVaried": true,
    "defaultValue": 0.0,
    "minValue": 0.0,
    "maxValue": 0.3,

```

```

        "name": "CO_MassFlow"
      },
      "5": {
        "isVaried": true,
        "defaultValue": 0.0,
        "minValue": 0.0,
        "maxValue": 0.3,
        "name": "SO2_MassFlow"
      },
      "6": {
        "isVaried": true,
        "defaultValue": 0.0,
        "minValue": 0.0,
        "maxValue": 0.3,
        "name": "HCl_MassFlow"
      }
    }
  ],
  "solidMixtures": []
},
{
  "hasGasPhase": false,
  "hasSolidPhase": true,
  "boundaryIndex": 6,
  "boundaryName": "Fly",
  "gasMixture": [],
  "solidMixtures": [
    {
      "phaseIndex": 0,
      "solidType": 1,
      "isGranularEnergySolved": false,
      "pressure": {
        "isVaried": false,
        "defaultValue": 86126.3,
        "minValue": 86126.3,
        "maxValue": 86126.3,
        "name": "Pressure"
      },
      "temperature": {
        "isVaried": true,
        "defaultValue": 288.15,
        "minValue": 288.15,
        "maxValue": 2500.0,
        "name": "Temperature"
      },
      "volumeFraction": {
        "isVaried": false,
        "defaultValue": 1.0,
        "minValue": 0.0,
        "maxValue": 1.0,
        "name": "VolumeFraction"
      },
      "diameter": {
        "isVaried": false,
        "defaultValue": 0.0001,
        "minValue": 0.0,

```

```

    "maxValue": 0.1,
    "name": "Diameter"
  },
  "density": {
    "isVaried": false,
    "defaultValue": 1350.0,
    "minValue": 1000.0,
    "maxValue": 1400.0,
    "name": "Density"
  },
  "granularTemperature": {
    "isVaried": false,
    "defaultValue": 1.0,
    "minValue": 0.0,
    "maxValue": 1.0,
    "name": "granularTemperature"
  },
  "vm": {
    "isVaried": false,
    "defaultValue": 0.0,
    "minValue": 0.0,
    "maxValue": 10.0,
    "name": "VolatileMatter"
  },
  "hasConstMassFractions": false,
  "totalMassFlow": {
    "isVaried": false,
    "defaultValue": 0.0045,
    "minValue": 0.0045,
    "maxValue": 0.0047,
    "name": "TotalMassFlow"
  },
  "speciesFlowMap": {
    "0": {
      "isVaried": true,
      "defaultValue": 0.1,
      "minValue": 0.1,
      "maxValue": 1.0,
      "name": "C"
    },
    "1": {
      "isVaried": true,
      "defaultValue": 0.05,
      "minValue": 0.1,
      "maxValue": 1.0,
      "name": "H"
    },
    "2": {
      "isVaried": true,
      "defaultValue": 0.01,
      "minValue": 0.01,
      "maxValue": 1.0,
      "name": "N"
    },
    "3": {
      "isVaried": true,
      "defaultValue": 0.0,

```

```

        "minValue": 0.0,
        "maxValue": 1.0,
        "name": "Cl"
    },
    "4": {
        "isVaried": true,
        "defaultValue": 0.005,
        "minValue": 0.001,
        "maxValue": 1.0,
        "name": "S"
    },
    "5": {
        "isVaried": true,
        "defaultValue": 0.1,
        "minValue": 0.1,
        "maxValue": 1.0,
        "name": "O"
    },
    "6": {
        "isVaried": true,
        "defaultValue": 0.1,
        "minValue": 0.1,
        "maxValue": 1.0,
        "name": "Ash"
    },
    "7": {
        "isVaried": true,
        "defaultValue": 0.1,
        "minValue": 0.1,
        "maxValue": 1.0,
        "name": "moisture"
    }
  },
  ],
  "inputParameterList": [],
  "outputParameterList": [
    {
      "isVaried": true,
      "defaultValue": 0,
      "minValue": 0,
      "maxValue": 1e10,
      "name": "heat_loss"
    },
    {
      "isVaried": true,
      "defaultValue": 0,
      "minValue": 0,
      "maxValue": 1e10,
      "name": "Qplaten"
    }
  ],
  "name": "Boiler_ROM"
}

```

In this case, the name of the ROM to be generated is “Boiler_ROM”. The regression method to be used is the Kriging method. 20 samples are to be generated by iREVEAL and the user needs to run 20 cases and provide 20 sets of results. The “gasSpeciesList” has only one entry, which is the gas/liquid mixture consisting of 7 species: O₂, N₂, H₂O, CO₂, CO, SO₂, and HCl. The “Species” object contains a “name” string and a “formula” string. Note: iREVEAL allows only one entry for the “gasSpeciesList”. The “solidSpeciesList” could contain multiple solid phases. This example has only one solid phase, which consists of NC components for the fuel of the boiler named COAL: C, H, N, Cl, S, O, Ash, and H₂O, representing the elements of C, H, N, Cl, S, O, ash, and moisture, respectively. This species list is fixed and the species **must** be in this order. The name of the solid phase is specified in the “solidPhaseList” array, which in this example contains only one name “COAL”. The “solidPhaseTypeList” contains an array of one integer entry, which is “1”, indicating the “NC” type. For CISOLID type, the integer is 0.

The “inletBoundaryList” array contains four entries of the “Boundary” object. A “Boundary” object contains the “hasCasePhase” and the “hasSolidPhase” boolean variables, the “boundaryIndex” integer variable, the “boundaryName” string variable, the “gasMixture” array of type “GasMixture”, and the “solidMixtures” array of type “SolidMixture”. The first inlet boundary has a gas/liquid mixture phase (“hasGasPhase” is “True”) and does not have any solid phase (“hasSolidPhase” is “False”). The “boundaryIndex” is “1” and the “boundaryName” is “Prim”, which is the feed stream name for primary air. Note: The “boundaryIndex” is for the user’s reference only and is actually not used in the exported ROM. Some CFD models such as MFX assign each boundary a boundary index while others do not. Since the “hasGasPhase” is “True” for the first inlet boundary named “Prim”, the “gasMixture” array contains exactly one “GasMixture” object entry. Since the “hasSolidPhase” is “False”, the “solidMixtures” array is empty. The “GasMixture” object contains several “Parameter” objects including “pressure”, “temperature”, “volumeFraction”, and “totalMassFlow”. The “GasMixture” object also contains a boolean variable “hasConstMassFractions” and a map “speciesFlowMap”. If the “hasConstMassFraction” is “True”, the mass fractions of individual species in the gas/liquid mixture must be “fixed” and the “speciesFlowMap” contains a map of species index to the mass fraction. The “totalMassFlow” is the total mixture mass flow rate, which could be “fixed” or “varied”. If the “hasConstMassFraction” is “False”, the “totalMassFlow” is ignored and the “speciesFlowMap” contains a map of the species index to the mass flow rate of the individual species. Note: The species index in the “speciesFlowMap” is a 0-based index in the species list of the corresponding phase. In this example, the gas/liquid mixture of the first inlet boundary “Prim” has a “fixed” pressure, temperature, volume fraction, and mass fractions. The total mass flow rate is “fixed” at 99.2 kg/s (“isVaried” is false). The minimum and the maximum values of the total mass flow rate are 99.2 and 99.2 kg/s, respectively. Note: If a parameter is “fixed”, the minimum and maximum values are irrelevant and arbitrary numbers can be used.

The second entry in the “inletBoundaryList” in this example is an inlet boundary named “Fuel” which contains a single solid phase (“hasGasPhase” is “False” and “hasSolidPhase” is “True”). The “boundaryIndex” is “2” and the “gasMixture” list is “empty”. The “solidMixtures” list has one entry of “SolidMixture” object only. The “SolidMixture” object contains a “phaseIndex” which is the 0-based solid phase index in the “solidPhaseList”. Since there is only one solid phase named “Fuel” in this example, “0” is specified for the solid mixture. The “SolidMixture” object also contains a “solidType” integer, which is “1” in this example, indicating the NC type.

The “isGranularEnergySolved” is set to “False”, indicating no granular energy equation is specified in the high-fidelity model and the “granularTemperature” parameter is not used (the “isVaried” property should be “False”). Other parameters in the “SolidMixture” object include pressure, temperature, volume fraction, particle diameter, particle density, and volatile matters (“vm”) of the solid particles. In this example, all of the parameters are “fixed” (“isVaried” is “False”). Note: The “vm” is used for NC solid only and is irrelevant in this case. As in the “GasMixture” object, the “hasConstMassFractions” indicates if the mass fractions of the individual species in the solid mixture are constant. If it is “True”, the “speciesFlowMap” represents a map of the species index to the species mass fractions with the “isVaried” set to “False” for any species. Otherwise the “speciesFlowMap” represents a map of the species index to the mass flow rate of the individual species and the “totalMassFlow” is ignored. In this example, the “hasConstMassFraction” is “True” and the “totalMassFlow” is fixed at 50.4 kg/s.

The third entry in the “inletBoundaryList” in this example is an inlet boundary named “Sec” which contains gas phase only (“hasGasPhase” is “True” and “hasSolidPhase” is “False”) and represents the secondary air stream. The “boundaryIndex” is “3” and the “solidMixture” list is “empty”. The pressure, temperature, and volume fraction are all fixed. However, the “totalMassFlow” is varied from 260.64 to 318.56 kg/s.

The fourth entry in the “inletBoundaryList” in this example is an inlet boundary named “OFA” which contains gas phase only (“hasGasPhase” is “True” and “hasSolidPhase” is “False”) and represents the overfire air stream. The “boundaryIndex” is “4” and the “solidMixture” list is “empty”. The pressure, temperature, and volume fraction are all fixed. However, the “totalMassFlow” is varied from 109.62 to 133.98 kg/s.

The “outputBoundaryList” array contains one entry of “Boundary” object in this example. The outlet boundary has both gas/liquid mixture phase and solid phase (both the “hasGasPhase” and “hasSolidPhase” variables are “True”). The first outlet boundary has a “boundaryIndex” is “5” and the “boundaryName” is “Flue”, representing the flue gas stream. Since it contains the gas/liquid phase, the boundary has one entry in the “gasMixture” array. The pressure of the outlet boundary is “fixed” at the “pressure” parameter’s default value of 86126.3 Pa. The temperature is “varied”, indicating that it is calculated by the ROM. The volume fraction is “fixed” at 0.5. The “hasConstMassFractions” is always “False” for the outlet boundaries. The flow rates of the individual species are always “varied” and predicted by the ROM for the outlet boundaries. The “totalMassFlow” is always not specified and set as “not varied” since it can be calculated as the sum of the flow rates of the individual species. Any entry in the “speciesFlowMap” does not require the default, minimum, and maximum values in the configuration input file since the result is calculated by the ROM. However, the user needs to specify what species exist in the gas/liquid mixture by providing the 0-based species index in the species list of the gas/liquid mixture phase.

The second outlet boundary contains the solid phase named “Fly” to represent fly ash leaving the boiler. It has one “SolidMixture” type object entry in the “solidMixtures” array. The “phaseIndex” is “0” since there is only one solid phase in the model. The “solidType” is “1”, indicating the solid type is NC. The “isGranularEnergySolved” is “False”, indicating the “granularTemperature” parameter entry is arbitrary. The “pressure” parameter is “not varied”

and its default value is set to 86126.3 Pa. The “temperature” parameter is “varied” and is calculated by the ROM. The “volumeFraction”, “diameter”, and “density” are “fixed” at “0.5”, “0.001 m”, and “1350 kg/m³”, respectively. The “vm” parameter is set to 0. As for the gas/liquid phase, the “hasConstMassFrations” variable is always set to “False”, indicating that the mass flow rates of the individual species are to be calculated by the ROM and are “varied” in the “speciesFlowMap”. As in the gas/liquid mixture phase, the species indices need to be listed in the “speciesFlowMap”. Here eight fly ash components are listed.

The “inputParameterList” array contains any additional input parameters that are not related to the material feed streams through the “inletBoundaryList”. In this example, there is no entry.

The “outputParameterList” array contains any additional output parameters that are not related to the material product streams through the “outletBoundaryList”. In this example, there is one entry name “heat_loss” indicating the heat loss to water wall and superheater. Usually “heat_loss” is always included as a result from high-fidelity model and predicted in the ROM. There is an additional output parameter “Qplaten”, representing the heat absorbed by platen super heater.

8. REFERENCES

[1] Ma, Jinliang, John P. Eason, Alexander W. Dowling, Lorenz T. Biegler, David C. Miller, “Development of a first-principles hybrid boiler model for oxy-combustion power generation system,” *International Journal of Greenhouse Gas Control*, 46, 136-157 (2016).