

**SOFTWARE FRAMEWORK FOR ADVANCED
POWER PLANT SIMULATIONS**

COOPERATIVE AGREEMENT NO DE-FC26-05NT42443

FINAL TECHNICAL REPORT

Reporting Period Start Date: August 2, 2005

Reporting Period End Date: August 1, 2010

John Widmann, Sorin Munteanu, Aseem Jain, Pankaj Gupta,
Mark Moales, Erik Ferguson, Lewis Collins
ANSYS, Inc.

David Sloan and Woodrow Fiveland
ALSTOM Power

Yi-dong Lang and Larry Biegler
Carnegie Mellon University

Michael Locke, Simon Lingard and Jay Yun
Aspen Technology, Inc.

October 2010

DOE Cooperative Agreement No. DE-FC26-05NT42443

ANSYS Inc.

10 Cavendish Court, Lebanon, NH 03766

ALSTOM Power

US Power Plant Laboratories, 2000 Day Hill Rd, Windsor, CT 06095

Carnegie Mellon University

Dept. of Chem. Engr., Pittsburgh, PA 15213

Aspen Technology, Inc.

200 Wheeler Rd, Burlington, MA 01803

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ABSTRACT

This report summarizes the work accomplished during the Phase II development effort of the Advanced Process Engineering Co-Simulator (APECS). The objective of the project is to develop the tools to efficiently combine high-fidelity computational fluid dynamics (CFD) models with process modeling software. During the course of the project, a robust integration controller was developed that can be used in any CAPE-OPEN compliant process modeling environment. The controller mediates the exchange of information between the process modeling software and the CFD software. Several approaches to reducing the time disparity between CFD simulations and process modeling have been investigated and implemented. These include enabling the CFD models to be run on a remote cluster and enabling multiple CFD models to be run simultaneously. Furthermore, computationally fast reduced-order models (ROMs) have been developed that can be “trained” using the results from CFD simulations and then used directly within flowsheets. Unit operation models (both CFD and ROMs) can be uploaded to a model database and shared between multiple users.

TABLE OF CONTENTS

ABSTRACT.....	ii
EXECUTIVE SUMMARY	1
1. Introduction.....	3
2. Scope of Work	5
2.1 Task 1: Advanced Model Integration.....	5
2.2 Task 2: Parallel Execution of the Integrated Model.....	7
2.3 Task 3: Development of Reduced Order Models (ROMs)	9
2.4 Task 4: Virtual Plant Walkthrough	11
2.5 Task 5: Model Calibration and Industrial Testing	13
2.6 Task 6: Dynamics Simulations	15
2.7 Task 7: Project Management.....	16
3. Results and Discussion	18
4. References to Accompanying Documents	19
5. Bibliography	20
6. List of Acronyms and Abbreviations	22

EXECUTIVE SUMMARY

This project continued the development of the software infrastructure for advanced power plant simulation that was initiated under the DOE-Fluent Cooperative Agreement No. DC-FC26-00NT40954. The primary objective of the project was to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. The software toolkit, named the Advanced Process Engineering Co-Simulator (APECS), links a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. An important enhancement to APECS was the creation of computationally efficient reduced order models (ROMs) based on information from high-fidelity Computational Fluid Dynamics (CFD) models. Integration of the steady-state APECS tool with a virtual reality tool allows process designers to navigate through a 3D virtual representation of a power plant with visualization of simulation results. A related secondary objective was to develop a prototype dynamic simulator within APECS for advanced power plants and to demonstrate a dynamic simulation that integrates plant-level and equipment-level models.

APECS consists of a user interface, integration controller, advanced model database, reduced order models, and a ROM builder for the creation of reduced-order models. The user interface (UI) provides a simple and intuitive method for APECS users to browse/search the model database, upload new models to the database, download models from the database, and change databases. The interface can also be used to select CFD models from the database to be used to “train” ROMs. The UI contains design-of-experiments algorithms for the selection of input values when creating ROMs. The ROM builder can automatically execute CFD simulations either locally or remotely using as inputs the values generated by the design of experiments. At the completion of the simulations, the ROM builder can generate the desired ROM type and compare the ROM predictions to the CFD predictions. Some ROMs are also capable of providing the user with contour and vector plots of solution variables (e.g., temperature, pressure) within the unit operation.

The APECS integration controller mediates the exchange of information between the process modeling environment (PME) and the APECS model (e.g., CFD model or ROM). The controller uses the international standard CAPE-OPEN (CO) for information exchange, thereby remaining vendor-neutral. The APECS controller can be used to couple any CO-compliant PME with any CO-compliant model, including in-house proprietary codes. Upon installation, APECS installs two integration controllers and registers them with the computer operating system. One is intended for use with a single model, while the other is designed to run multiple CFD models simultaneously. The process engineer then simply chooses the appropriate controller for his or her needs when building the flowsheet.

During Phase II of this project, ANSYS Engineering Knowledge Manager (EKM) became an integral part of APECS. By leveraging an existing ANSYS product, an advanced model database was coupled to APECS with minimal cost and significant time savings. This database can be installed locally or remotely, can be used by a single user or an entire company, contains built-in security features and can serve multiple APECS users simultaneously.

The reduced-order modeling capability in APECS includes the creation of ROMs from CFD simulations and the use of these ROMs in PME flowsheets. APECS ROMs include second-order full-polynomial regression, non-linear regression, Kriging, and artificial neural networks (ANN). Several of these methods were realized by coupling APECS with ANSYS DesignXplorer (DX) and creating a CAPE-OPEN wrapper so that the response surfaces in that software could function as APECS ROMs. As with the model database, coupling APECS to an existing ANSYS product permitted significantly greater functionality to be implemented given the time and resource constraints.

In addition to the project deliverable reports, the Phase II results were disseminated in seventeen technical publications, workshop presentations, and dedicated web pages. The software capabilities were documented in a comprehensive APECS Users Manual, and ANSYS continued the ongoing process to evolve the APECS technologies into commercially supported products.

1. Introduction

To accelerate the development of advanced power plants, the DOE identified the need for an integrated suite of software tools that could be used to simulate and visualize new plant concepts. Existing process simulation software did not meet this objective of *virtual-plant simulation*. Sophisticated models of many individual equipment items are available; however, a seamless coupling capability that would integrate the advanced equipment models to the process simulation software remained to be developed. A team consisting of Fluent Inc. (now ANSYS, Inc.), ALSTOM Power, Inc., Aspen Technology, Inc., Intergraph Corporation and West Virginia University, in collaboration with the National Energy Technology Laboratory (NETL), addressed this challenge in a project (Cooperative Agreement No. DC-FC26-00NT40954) performed over the period from October 2000 through December 2004.

This project, referred to as Phase II, continued the development of the software infrastructure for advanced power plant simulation initiated under the original project. The primary objective of the Phase II project was to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of APECS, the simulator would link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. An important enhancement to the APECS tool would be the creation of computationally efficient reduced-order models (ROM) based on information from high-fidelity computational fluid dynamics (CFD) models.

This report details the results of the research conducted under Cooperative Agreement No. DE-FC26-05NT42443 during 2005-2010 with the primary goal of developing the software tools required to couple process modeling and computational fluid dynamics software for advanced power plant design. The project included the following major tasks:

- Task 1: Advanced Model Integration
- Task 2: Parallel Execution of the Integrated Model
- Task 3: Development of Reduced Order Models (ROMs)
- Task 4: Virtual Plant Walkthrough
- Task 5: Model Calibration and Industrial Testing
- Task 6: Dynamic Simulations
- Task 7: Project Management

For clarity, this report will discuss individually each of the tasks listed above. For each task, a brief overview will be presented, followed by a more detailed discussion of the work accomplished within the task. Due to the uncertain nature of technical research, the work scope

of specific tasks was revisited regularly and adjusted periodically. With this in mind, the original goals outlined in the project proposal will be described and compared with the work actually conducted, with detailed explanations of project modifications given. This report is intended as an integrative summary rather than a “how to” guide describing all aspects of the software developed. The reader is referred to the APECS Users Manual for detailed descriptions of all APECS features and instructions on the use of the software.

2. Scope of Work

2.1 Task 1: Advanced Model Integration

Task 1 involved extending the capabilities of the FLUENT CAPE-OPEN Configuration Wizard for greater functionality and usability. Also addressed in Task 1 was the incorporation of a robust model database for the storage and retrieval of CAPE-OPEN (CO) unit operation models. This included both the core components of the database as well as a user interface from which APECS users can access the database. Task 1 also extended FLUENT physical model support for APECS, including additional DPM injections, PDF-based chemical reaction models, and flexibility in Aspen/FLUENT species mapping to allow intermediate species in the FLUENT model. Finally, upgrading of the software documentation and development of APECS tutorials were part of Task 1. The APECS User's Manual (ANSYS, Inc. 2010), which comprehensively details the software's function, interface, features and installation, accompanies this report.

In the original proposal, Task 1 was broken down into four subtasks as follows:

Subtask 1.1: Replace the prototype (Phase I) file system for equipment models with an advanced data management service having the ability to check in or check out model files, track versions and history, browse and search, and control access.

Subtask 1.2: (a) Modify the Configuration Wizard (CW) to insert and modify CFD model files in the model database maintained using the advanced data management service. (b) Enhance the CW to work with model databases maintained on the local as well as remote (over the LAN) computers.

Subtask 1.3: (a) Add event handlers to the Integration Controller (IC) to handle error conditions that may arise in the course of establishing a connection with local and remote servers. (b) Develop the functionality to support integration of new ROMs. (c) Add multi-threading, interrupt, and start/stop capabilities to the FLUENT– Aspen Plus® interface for remote running so that the user has full control over the FLUENT code during Aspen Plus executions.

Subtask 1.4: Enhance the solution strategy capability to allow switching between models of an equipment item (e.g., Aspen Plus, ROM, proprietary, CFD) based on user-controlled switching criteria.

The objective of Subtask 1.1 of implementing an advanced data management system was accomplished by coupling APECS to ANSYS Engineering Knowledge Manager® (EKM). ANSYS EKM contains advanced database functions, can be shared by multiple users, and can be

accessed remotely through the http protocol. APECS can use an EKM server installed on the user's machine, remotely over a LAN, or in another location entirely using the web.

Subtask 1.2 specified that the Configuration Wizard (CW) should be modified to permit users to add/modify models in the database. In this phase of the APECS project, a single unified user interface (UI) was developed that provides the ability to configure and upload models to the EKM database, search the model database, and download models from the database. This UI also contains the APECS ROM Builder, discussed in Section 2.2, for building reduced order models from CFD simulation results. The APECS UI can interface with both local and remote databases, and can easily switch from one database to another.

Subtask 1.3 involves error handling and the ability to run FLUENT simulations remotely. Extensive error handling was added to the APECS Controller and other components to avoid error conditions that may arise in the course of co-simulation using local or remote servers. Clear error messages have been provided for instances when the software cannot recover from an error. It was decided not to add start/stop capabilities to the FLUENT-Aspen Plus interface when running FLUENT remotely because of limitations in the CAPE-OPEN standards. Once Aspen Plus requests that the APECS block "Calculate", no further control is given to the user until the calculations have completed. Any workaround provided by the project team would have involved violating the standard, which was an undesirable approach. Instead of providing start/stop capabilities to the user when running remotely, effort was placed on tasks deemed to be of greater importance (e.g., incorporation of COM-based models, incorporation of additional FLUENT models).

Solution strategy controls were enhanced as indicated in Subtask 1.4 to allow switching between models of an equipment item during the iterative PME solution based upon user-controlled settings. This can be done by calling one model for a set number of iterations before moving to the next. The switching can also be triggered by monitoring the inputs to the APECS block and transitioning when the inputs are constant to within a user-specified tolerance. It is also possible to have Aspen Plus explicitly set the model to use by setting the value of a certain CAPE-OPEN parameter. This allows users to change between different models either manually or programmatically.

Other enhancements to APECS that were added to Task 1 during the course of the project include the following:

- the ability to use COM-based CAPE-OPEN models in addition to CORBA-based models
- modification of the FLUENT CAPE-OPEN Wizard to support flexible species mapping
- support of additional DPM injection types
- support of PDF-based reaction models via automated recompilation of look-up tables

- support of FLUENT NO_x and SO_x pollutant models
- support for the FLUENT transient solver
- the ability to time-average output quantities when using the FLUENT transient solver
- support for the FLUENT Volume of Fluid (VOF) model
- support for the 3-stream PDF reaction model in FLUENT
- support for the Laminar Steady Flamelet combustion model in FLUENT
- support for the new Dense Discrete Phase model in FLUENT
- consolidation of the CAPE-OPEN Wrapper into the standard installation of FLUENT
- enhancements to APECS to permit geometry parameterization using only a GAMBIT journal file, without requiring user-supplied FLUENT journal files, UDFs, etc.
- enhancements to the APECS Controller to permit loading of Aspen Plus flowsheets containing references to obsolete databases or servers
- enhancements to the APECS Controller to permit modifying CFD execution attributes (e.g., local execution vs. remote execution, serial FLUENT vs. parallel FLUENT) after the APECS block has been instantiated and run
- enhancements to the APECS Controller so that different APECS blocks in the same flowsheet can reference different model databases
- enhancements to the APECS Controller so that chemical species can be automatically mapped if the FLUENT and Aspen Plus names are very similar (not necessarily identical)
- enhancements to the APECS Controller so that users can easily return to the original FLUENT data file if the solution diverges during co-simulation

2.2 Task 2: Parallel Execution of the Integrated Model

Task 2 explored hybrid simulation strategies that address the speed disparity between process simulation and CFD by taking advantage of parallel computing technology. The ultimate goal was to minimize the extra cost that integrated simulations add on top of simple process simulations. With FLUENT already well-parallelized so that an individual CFD model can make full use of multiprocessing hardware, this task was broken down into subtasks as follows:

Subtask 2.1: Research methods to use CFD more selectively within Aspen Plus, including concepts such as dual representation, more intelligent solver selection logic, subcycling, and customized convergence control. Implement successful new hybrid algorithms into the IC.

Subtask 2.2: Develop the capability to run multiple CFD equipment models attached to different sections of a flowsheet simultaneously (in parallel with each other).

The original concepts for reducing the computational costs associated with co-simulation contained some significant disadvantages. One of the greatest of these was the specificity to

Aspen Plus. The benefit of using the CAPE-OPEN standard interfaces is that it permits APECS to be used with a wide variety of commercial process modeling environments. Subtasks 2.1 and 2.2 as originally conceived could not be implemented within the CAPE-OPEN framework and would therefore have been unique to Aspen Plus. Furthermore, such an approach would have a high probability of “breaking” every time a new version of Aspen Plus was released. Due to these issues, it was decided to pursue alternative approaches that would allow APECS to remain CAPE-OPEN compliant.

Three approaches to reducing the computational time of co-simulation were explored and implemented into APECS. The first involves running CFD models off-line and creating reduced order models from the simulation predictions, thereby leveraging the results of Task 3 to be described below. This approach involved the development of the APECS ROM Builder. The ROM Builder contains design-of-experiments tools to calculate the optimal inputs, automatic execution of the CFD simulations, and visual and statistical tools for evaluating the accuracy of the ROMs generated. The ROM Builder was also enhanced to interface with CAPE-OPEN Property Packages for transferring material properties to the FLUENT model when creating a ROM.

The second approach to reducing simulation time was the development of the APECS Multiblock. The Multiblock permits parallel execution of multiple CFD models with the context of the CAPE-OPEN framework. Because the Multiblock is CAPE-OPEN compliant, it can be used within any CO-compliant simulator. The APECS Multiblock is essentially a CO-compliant “wrapper” that contains multiple APECS blocks. When instantiated in an Aspen Plus flowsheet, for example, the Aspen simulation engine perceives the Multiblock as a single unit operation. Aspen Plus treats the block as any other CO-compliant unit without “knowledge” that multiple CFD simulations are being simultaneously executed.

The final approach to reducing computational time is to permit FLUENT to exit during each PME iteration when the APECS block is finished calculating. In this way, all of the user’s FLUENT licenses can be applied to each running simulation instead of sharing them among all of the FLUENT models instantiated in the flowsheet. The simulation software (e.g., Aspen Plus) will still run the APECS models sequentially, but the models can be run using the parallel FLUENT solver, thereby increasing the speed.

Application testing confirmed the merit of all three approaches. The advantage of the first approach, which does not strictly require multiprocessing hardware, was observed to depend on the CFD mesh size, the complexity of the ROM required to capture the equipment performance trends, and the “stability” of the equipment design information over time. For example, suppose that a plant performance study would require N flowsheet iterations on a particular equipment item and the computational cost of one “average” CFD simulation is C . If the ROM execution is

1000 times faster than that CFD simulation, but the off-line ROM-building effort requires twenty CFD simulations, then the ratio of the cost using ROM to the cost without ROM for the study is

$$R = \frac{20C + 0.001NC}{NC}$$

Although this simple equation neglects the effects of CFD initial conditions on re-run costs – a reduction that APECS takes advantage of with or without the ROM – it shows that the technique is generally most cost-effective for relatively large N , valuable for real-world design studies that may require hundreds or even thousands of cumulative flowsheet iterations. A further advantage, less easily quantified, is the increased schedule flexibility possible when high-performance computing facilities can be used to prepare the off-line ROMs in advance. The second and third approaches reduce elapsed time only when parallel computers and corresponding multiple FLUENT licenses are available.

2.3 Task 3: Development of Reduced Order Models (ROMs)

Task 3 involved the investigation of various techniques for model-order reduction and the determination of which methods are most suitable for APECS. Computer code was developed based upon the identified ROM methods, and ROM results were compared with CFD predictions. ROMs capable of predicting field data (e.g., temperature contours) within the domain were also developed. ROMs were developed for two “FutureGen” equipment items, one based upon Artificial Neural Networks (ANN) and the other based upon non-linear regression coupled with Principal Component Analysis (PCA).

Subtasks for this task were originally defined as follows:

Subtask 3.1: Develop methods to systematically derive ROMs for equipment items based on high fidelity simulation results. The models will use two alternative techniques: multiple non-linear regression and proper orthogonal decomposition (POD).

Subtask 3.2: Assess the performance and accuracy of the ROM approximations derived in Subtask 3.1. Identify suitable methods for integration of the ROMs within cycle simulations and confirm viability of the integrations via testing.

Subtask 3.3: Develop computer code based on the ROM methods identified in Subtask 3.2.

Subtask 3.4: In cooperation with DOE, select two FutureGen equipment items for ROM demonstration. Candidate modules include a gasifier, heat recovery steam generator (HRSG),

steam turbine, air preheater, and syngas cooler. Develop ROMs for the selected items using either CFD results or proprietary design code results.

At the beginning of the Phase II research effort, it was envisioned that non-linear regression and proper orthogonal decomposition (POD), also called Principal Component Analysis (PCA), would be the best approaches for model-order reduction in APECS. Through the course of the project it was determined that non-linear regression was a very useful approach for predicting scalar outputs, while the PCA approach was more appropriate for predicting field data. Both of these methods were implemented into APECS and resulted in ROMs capable of providing the user with post-processing functionality. ROMs using PCA provide the ability for users to view contours of field data (temperature, pressure, etc.) within the domain. Comparisons between these ROMs and CFD results showed very good agreement under many circumstances, particularly non-reacting cases. Details of this study and investigation are included in the accompanying report "Reduced Order Model Based on Principal Component Analysis for Process Simulation and Optimization" (Lang 2008).

The feasibility of using a "trust region" approach to programmatically choose between a ROM and a CFD model was investigated. This approach was abandoned after consultation with Aspen Technology, which revealed experience that such repeated "switching" between solvers could prevent flowsheet convergence. It was suggested instead that the flowsheet would be more likely to converge if the ROM returned results even if the inputs were outside of the trust region. APECS now displays warning messages to the user if the ROM is outside of the trust region. Only if the *converged* flowsheet results in inputs to the ROM that are outside of the trust region should the user take any action. In this case, the user can manually change the CFD model with a few clicks of the mouse and re-converge the flowsheet.

Initially implemented for displaying contour plots from ROMs generated with 2D FLUENT models, PCA ROMs were extended to also display vector plots. In addition, APECS was enhanced so that PCA ROMs could be created from 3D FLUENT simulations and contour plots on specified 2D sub-domains displayed to the user. To ensure computational tractability, these "display" surfaces must be chosen in the 3D FLUENT model before creating the PCA ROM. In this way, only the data on the specified surfaces need to be exported and processed. This approach also reduced the effort associated with enhancing the CFD Viewer for displaying contours in 3D, enabling it to be completed within this project.

Subtask 3.4 involves the creation of two demonstration cases using ROMs for selected FutureGen equipment items. The items chosen were a two-stage gasifier and a radiant syngas cooler. One ROM was created using Artificial Neural Network (ANN) methodology, and the other used non-linear regression with PCA. The details of these investigations are summarized in accompanying reports (Sloan, Demonstration of a Regression ROM for a Heat Recovery

Steam Generator in an Integrated Co-Simulation 2009) and (Sloan, Demonstration of a Calibrated ROM for a Radiant Syngas Cooler in an Integrated Co-Simulation 2009).

During the course of this Phase II effort, APECS was coupled to ANSYS DesignXplorer (DX) to take advantage of the design-of-experiments and response-surface functionality in DX. In addition to the Latin Hypercube sampling algorithm in APECS, DX offers several other methods that are now available to APECS users. Response surfaces in DX, like ROMs in APECS, are essentially computationally fast methods of mapping inputs to outputs. By “wrapping” the DX response surfaces to make them CAPE-OPEN compliant, they can be used in Aspen Plus and other CO-compliant flowsheets. Coupling APECS to DX provided additional ROM methods, including Kriging and a robust, well-tested implementation of Artificial Neural Network.

2.4 Task 4: Virtual Plant Walkthrough

In Task 4, APECS was connected to a virtual reality environment with the intent of enabling a virtual plant walkthrough. Under separate DOE funding coordinated with this project, the Ames Laboratory developed a data integration link with APECS and Aspen Plus called Virtual Engineering Process Simulator Interface (VE-PSI). These real-time data integration links provide a toolkit to seamlessly interoperate between VE-Suite, Aspen Plus, and APECS. At various times throughout the project, demonstrations of the current feature set provided by VE-PSI and the new features in VE-Suite to support the virtual plant walk-through were provided. These demonstrations also served as a way to illustrate the new workflows that are possible with VE-Suite and VE-PSI.

Task 4 was divided into the following subtasks:

Subtask 4.1: Extend the existing link between FLUENT and VE-Suite to include the integrated model, enabling real-time update and interaction through the virtual engineering environment. Allow user to interact directly with Aspen Plus, FLUENT, and other proprietary software packages from VE-Suite.

Subtask 4.2: Develop the three-dimensional geometries and rendered components needed to portray a selected advanced power plant simulation.

Subtask 4.3: Couple the software framework to interactive optimization tools and demonstrate these tools in a power plant design application.

VE-Suite software installers and training were provided to ANSYS and NETL for the duration of the APECS project. The VE-Suite software installers were provided on an as needed basis to ANSYS and NETL personnel to review the current status of the project. In addition, providing

VE-Suite software to the development team enabled ANSYS, NETL, and ALSTOM Power engineers to participate in the design and development of the VE-PSI toolkit.

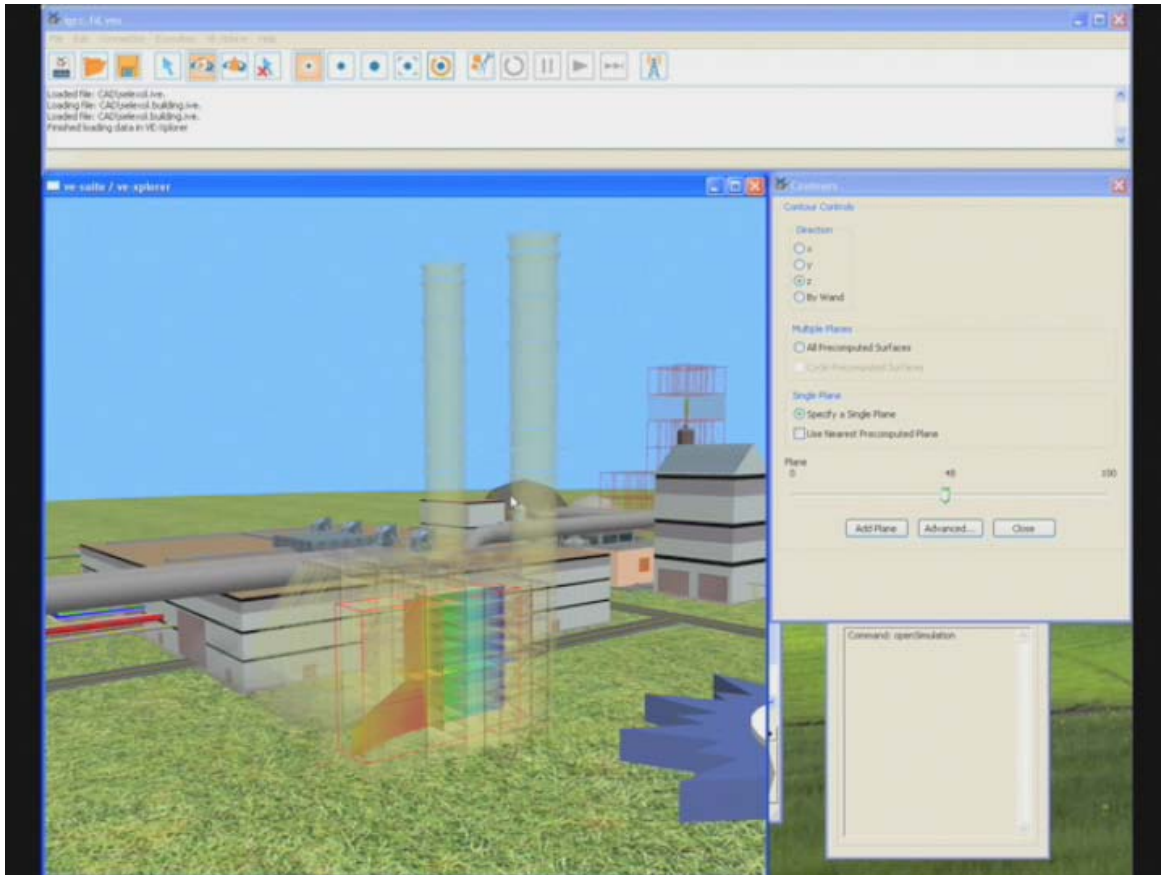
To facilitate the visualization and usage of the VE-PSI toolkit with Aspen Plus and APECS, some basic CAD representations of structures in power plants were created by Ames Lab and NETL. These representations can be overlaid on the unit operations that are loaded into VE-Suite through VE-PSI. In addition to the 3-D CAD models, 3-D icons were also created to show the 2-D flowsheet in the 3-D view of VE-Suite. These icons enable a consistent look and feel when engineers and scientists work with Aspen software in connection with VE-Suite. For any CAD data that is loaded in VE-Suite there are standard operations (e.g., scale, translation, rotation, transparency) that can be applied to any CAD data.

To integrate VE-Suite, Aspen Plus and APECS, the Aspen Plus type libraries were used to create a link between the unmanaged code (i.e. VE-PSI) and the managed code (i.e. Aspen Plus). This integration enabled VE-Suite to have complete access to the underlying Aspen Plus simulation data and provided some level of control over the Aspen Plus simulator. For example, the user can start, stop, and pause an Aspen Plus simulation. The user also has the ability to set any variable within the Aspen Plus simulation. This includes any variable that can be seen in the Aspen Plus variable explorer. This capability is also accessible for any APECS block. As a part of the Aspen Plus integration with VE-Suite, 2D icons were created for Aspen Plus unit operations that enabled the 2D Aspen Plus flowsheet to be replicated in VE-Suite. These icons include the APECS and FLUENT icons that appear in Aspen Plus flowsheets.

As mentioned earlier, a demonstration was created to highlight the various features of the VE-Suite integration with Aspen Plus and APECS enabling plant walkthroughs. This demonstration highlighted the ability to load and interact with FLUENT data, render high fidelity CAD data, render a complete power plant, plant walkthrough capability, and real-time interaction with components of the power plant. One of the key features that was created to polish the content of the demonstration was the dynamic display of textual data in the virtual world. These displays provide the ability to present the user with information that would typically be displayed in the GUI (i.e., VE-Conductor). In addition to illustrating basic tools available within VE-Suite the demonstration also utilizes the scripting capability developed for demonstration purposes. The scripting tools enable a user to set up the commands in advance for an operation such as the creation of contour planes for FLUENT data, changing CAD properties of power plant components, and walkthroughs of the plant. The script can then be loaded and executed and the demonstration will “play” without any user input. This capability would enable a series of demos to be created to illustrate the connections between VE-Suite, Aspen Plus, and APECS.

The figure below is a screen-capture from the demonstration of VE-Suite performed by Ames Laboratory referenced above. The demonstration shows a 3-D representation in VE-Suite of an example integrated gasification combined cycle (IGCC) plant, which is based on an Aspen Plus flowsheet that, itself, contains APECS unit operation blocks. The APECS unit operation block

shown in the figure represents a FLUENT CFD model of an HRSG, with contours of pressure displayed on planar surfaces.



2.5 Task 5: Model Calibration and Industrial Testing

The objectives of Task 5 were to calibrate a reduced-order model based upon a CFD model, develop a parameterized FLUENT CFD equipment module, and systematically test the previously constructed ROMs and component models. The parameterized FLUENT model required the implementation of geometry scaling and re-meshing into APECS. This was accomplished using the preprocessing tool GAMBIT. Task 5 also included the management of the Industrial Advisory Board consisting of industry experts for the purpose of obtaining feedback and recommendations.

The original proposal divided Task 5 into the following subtasks:

Subtask 5.1: Calibrate a ROM developed in Subtask 3.4 and ensure that the predictive accuracy is adequate over the parameter space required for convergence of the FutureGen cycle at the full load condition.

Subtask 5.2: Develop a FLUENT CFD equipment module (e.g., cyclone, air preheater) and demonstrate the ability to do geometric scaling within an integrated simulation. Automate the module to control the geometry with a few parameters, generate a new grid, interpolate the old solution onto the new grid, and solve the model. Use the geometric parameters for manipulation from Aspen Plus for design specification, sensitivity analyses, or optimization of components within a cycle flowsheet framework.

Subtask 5.3: Systematically test the previously constructed ROMs and component models with simplified versions of a FutureGen cycle (see the accompanying reports referenced in Subtask 3.4 above).

Subtask 5.4: Form an Advisory Board within two months after the start of the project. Conduct a total of five semiannual meetings over the Internet to demonstrate the software and review simulation results. Document the board feedback and recommendations.

Reduced order models were created and used in several industrial test cases. These included ROMs of a radiant syngas cooler and a heat recovery steam generator. In addition, a study was performed to investigate pollutant emissions from an oxy-fired boiler for various flue-gas cleanup scenarios. The results of this study are summarized in an accompanying report (Sloan, Demonstration of Emissions Calculations for and Oxy-Fired Boiler in and Integrated Simulation 2009). As described in Subtask 5.2, a demonstration case was created to show the ability to perform automated geometric scaling in APECS. This study is also documented in an accompanying report (Sloan, Demonstration of Geometric Scaling in an Integrated Simulation 2009).

An Industrial Advisory Board was created early in Phase II of the project. The board was organized by ALSTOM Power and consisted of experts from industry and academia with an interest in co-simulation and reduced order modeling. Five advisory board meetings were held during the project period, where the APECS team provided updates on the status of the project and demonstrated new features. The comments and questions of the Advisory Board were summarized in the meeting minutes. A summary of the comments and responses is presented in the table below.

	Advisory Board Question/Comment	Response
1	ROM reliability: "How many CFD simulations are needed to get a reliable ROM for a gasifier?"	For simple modules and regression methods, such as our CSTR example, the number may be less than e.g., 20; however, we do not yet know how that number will scale as the number of input parameters to the CFD block increases.
2	Mass balance: "Has the APECS team given any consideration to the closing of mass and energy balances on ROMs? The team should begin to look at forcing mass conservation constraints around ROMs."	Implemented capability to compare relative difference between mass flow rate at outlet and inlet boundaries and declare balance when value reaches user-specified value.

3	License management: "How many FLUENT licenses are required to instantiate two FLUENT blocks simultaneously on the flowsheet?"	Having two FLUENT blocks active simultaneously in a flowsheet requires two FLUENT licenses. It is possible, however, to release the license when FLUENT is not iterating, thereby requiring only one license for a flowsheet with multiple FLUENT blocks.
4	Model library: "Does a library of FLUENT equipment items exist (that can either exist as templates or that are "instantiation-ready")?"	Although the PMEs do have a library of modules, it would be difficult to provide FLUENT models in the form of a generalized template, since CFD can be applied to the wide variety of vendor-specific versions for any given equipment item. Site-specific customization or templating for most common equipment items by in-house CFD experts or ANSYS Services might help approach the library concept.
5	ROM graphical representations: "Is there a way to know whether the particular PCA (contour plot) view desired by the user constitutes an interpolation or an extrapolation of the set of FLUENT runs?"	In the event of an extrapolation, the contour plot is still provided to the user, but a warning message is issued that the solicited condition is "outside the range."
6	Database management: "Is there a mechanism for downloading selected files from EKM in a form that can be sent to someone else?"	APECS allows the CFD analyst and/or database administrator to download model files to a user-specified directory on a local machine, after which the files can easily be transmitted to another user.
7	Multiple APECS blocks in a flowsheet: "The concept behind a number of different FLUENT blocks grouped into an APECS Multiblock and run in parallel, may mean that a lot of the flow sheet may be embedded within that Multiblock. It is apparent that there is a lot of room for research on the convergence issues for various Multiblock configurations."	Each FLUENT block may have very distinct run times relative to the other FLUENT blocks in the Multiblock. Indeed, the overall convergence will be both block and flow sheet specific. Consequently, the Multiblock will be positioned as a option that the user may select in cases where it provides an advantage
8	Model parameters: "Are there any specific guidelines for conducting CFD simulation with reacting flow using the APECS system? Such information will be helpful for users."	The CFD simulation can use any of the combustion or chemical kinetic sub-models available within FLUENT. The option or approach to deal with reacting flow within a particular FLUENT run is pre-specified by the CFD analyst and stored within the FLUENT case and data files prior to coupling the FLUENT model with the PME. Consequently, the reacting flow models are independent of the APECS tool kit, and a wealth of guidance is available in the FLUENT product documentation, training materials, and publications.

2.6 Task 6: Dynamics Simulations

Task 6 involved developing a prototype coupling between FLUENT and Aspen Dynamics. Because Aspen Dynamics is not CAPE-OPEN compliant, this effort involved using the proprietary application programming interfaces in Aspen Dynamics that were recommended and supported by Aspen Technology Inc. The prototype dynamic simulation capability in APECS used FLUENT to model an equipment item, as is done for steady-state simulations. An event-based quasi-steady state approach was adopted for coupled dynamic simulations, in which FLUENT was run in steady-state mode in response to an event or time-varying input or

throughput conditions. A ROM developed from FLUENT predictions was also coupled with Aspen Dynamics to explore the time savings that can be realized when using ROMs instead of CFD models in dynamic co-simulations.

This task was originally divided into the following subtasks:

Subtask 6.1: Survey and document the requirements for an integrated dynamic simulator.

Subtask 6.2: (deleted)

Subtask 6.3: Upgrade the IC to link CFD/custom models into AD via an existing interface (SAX).

Subtask 6.4: Test the prototype with an integrated dynamic simulation. Use a quasi-steady state approximation for CFD and use results stored in a CFD database.

The requirements for a prototype dynamic co-simulator were investigated and summarized in a Software Requirements Document. A prototype coupling between Aspen Dynamics and FLUENT was then implemented using these requirements. Both FLUENT CFD and ROM blocks were used in an Aspen Dynamics flowsheet, and the computation time required for co-simulation investigated. The details of this study were summarized in an accompanying report (Widmann 2010).

2.7 Task 7: Project Management

ANSYS Inc. was responsible for managing the project and for planning tasks with input from the other members of the project team. Project team members reviewed the contract work scope and provided input to the project manager as ANSYS Inc. formally planned the project. The project management plan was established and periodically updated in coordination with DOE.

The original proposal outlined the following subtasks for Task 7:

Subtask 7.1: Plan the overall work scope and coordinate with all team members and NETL. Review the contract work scope, establish key milestones and dates for deliverables, negotiate sub-award agreements, and submit final plan to DOE within 60 days after the start of the project. In consultation with DOE, periodically adjust project plans as necessary based on the Advisory Board recommendations from Subtask 5.4.

Subtask 7.2: Submit the mandated monthly/quarterly/annual reports, and execute the requisite managerial responsibilities and coordinating efforts.

Subtask 7.3: Prepare and submit the final technical report.

Managerial tasks were all performed as specified in the original proposal.

3. Results and Discussion

APECS Build 2.0.4, incorporating all of the previously described work, was delivered to DOE/NETL on July 22, 2010 as the final deliverable version of the software. The table below summarizes the main features of the APECS software and how those features help the end user. A status comparison is presented of the Phase I integration toolkit and the present enhanced capabilities.

Feature	Benefit	Integration Toolkit (Phase I)	APECS 2.0.4 (Phase II)
Configuration Wizard	Make FLUENT models compatible with CO-compliant PME software	Separate application to be installed. XML-based solver data.	Enhanced GUI is available in standard release of FLUENT 13.
Integration Controller	Select equipment model from database for convenient integration as a unit operation block in a PME flowsheet.	Separate model selection and edit GUIs	Enhanced APECS unit operation interface combines model selection and configuration for use by PME.
Flexible model database	Convenient storage configuration, management and integration of equipment models.	Manual file-based setup and management. XML-file based model data.	ANSYS EKM-based solution with automatic uploading and downloading, history tracking, access control, etc.
Parallel/remote execution	Run multiple CFD cases in a single flowsheet on either local or remote machines or clusters.	Custom installation of Controller components required. Manually specify ports, server names.	No custom installation needed. Database admin registers server names in advance; user selects from a list.
Reduced Order Models	Use previous CFD solutions to generate faster estimates	ROMs must be created through Custom Model Configuration Wizard.	Enhanced GUI to facilitate ROM creation; more ROM methodologies available through linkage with ANSYS DX
Virtual Engineering linkage	Explore results in context-based, user-centered interface	Not available	Demonstration case of power plant walkthrough to view CFD results overlaid on CAD model of unit operation.
CFD Viewer	Display CFD-based results from inside PME interface	View only pre-defined CFD results	View 2-D/3-D ROM results and compare side-by-side with CFD results
Geometric scaling	Choose geometric parameters for automatic remeshing and solution of scaled model in PME	Not available	Define geometry model parameters; create GAMBIT journal file in advance to enable remeshing
Dynamic simulation	Perform co-simulation for transient flowsheets	Not available	Prototype simulator demonstrated integrating Aspen Dynamics with a FLUENT equipment model

4. References to Accompanying Documents

ANSYS, Inc. "APECS User's Manual." Lebanon, NH, Version 2.0, 2010.

Lang, Y. and Biegler, L.T. *Reduced Order Model Based on Principal Component Analysis for Process Simulation and Optimization*. Topical Report, Pittsburgh, PA: Carnegie Mellon University, 2008.

Sloan, D. and Fiveland, W. *Demonstration of a Calibrated ROM for a Radiant Syngas Cooler in an Integrated Co-Simulation*. Topical Report, Windsor, CT: Alstom Power Inc., 2009.

Sloan, D. and Fiveland, W. *Demonstration of a Regression ROM for a Heat Recovery Steam Generator in an Integrated Co-Simulation*. Topical Report, Windsor, CT: Alstom Power Inc., 2009.

Sloan, D. and Fiveland, W. *Demonstration of Emissions Calculations for and Oxy-Fired Boiler in an Integrated Simulation*. Topical Report, Windsor, CT: Alstom Power Inc., 2009.

Sloan, D. and Fiveland, W. *Demonstration of Geometric Scaling in an Integrated Simulation*. Topical Report, Windsor, CT: Alstom Power Inc., 2009.

Widmann, J. *Dynamics Research and Prototype*. Topical Report, Lebanon, NH: ANSYS, Inc., 2010.

5. Bibliography

This section lists the published and unpublished papers, presentations and web pages resulting from this project.

- B1. APECS Open Energy Info Wiki, en.openei.org/wiki/APECS
- B2. Bryden, K.M., and McCorkle, D.S., "Virtual Engineering", *Mechanical Engineering*, November 2005.
- B3. Diwekar U., Shastri, Y., Subramanyan, K., Zitney, S.E., "CAPE-OPEN Compliant Stochastic Modeling and Reduced-order Model Computation Capability for APECS System", In Proceedings of the AIChE 2007 Annual Meeting, Paper 288b, Salt Lake City, UT, November 2007.
- B4. Ferguson, E., Madsen, J., "Integration of APECS and VE-Suite", Presented at the Virtual Engineering Workshop, Ames, IA, May 2007.
- B5. Lang, Y., Biegler, L.T., Munteanu, S., Madsen, J., Zitney, S.E., "Advanced Process Engineering Co-Simulation Using CFD-based Reduced Order Models", In Proceedings of the AIChE 2007 Annual Meeting, Paper 398c, Salt Lake City, UT, November 2007.
- B6. Lang, Y., Malacina, A., Biegler, L.T., Munteanu, S., Madsen, J., Zitney, S.E., "Reduced Order Model Based on Principal Component Analysis For Process Simulation and Optimization", *Energy & Fuels*, 23(3), 1695–1706, 2009.
- B7. McCorkle, D.S., Yang, C., Jordan, T., Swensen, D.A., Zitney, S.E., Bryden, K. M., "Towards the Integration of APECS with VE-Suite to Create a Comprehensive Virtual Engineering Environment," In Proceedings of the 32nd International Technical Conference on Coal Utilization & Fuel Systems, Clearwater, FL, June 2007.
- B8. Osawe, M, Sloan, D., Fiveland, W., Madsen, J., "Fast Co-Simulation of Advanced Power Plants Using Neural Network Component Models", In Proceedings of the AIChE 2006 Annual Meeting, Paper 626b, San Francisco, CA, November 2006.
- B9. Proceedings of the APECS Workshop & Project Review Meeting, Pittsburgh, PA, October 2009, www.netl.doe.gov/publications/proceedings/09/APECS/
- B10. Sloan, D., Fiveland, W., Zitney, S.E., Osawe, M., "Focus on O&M: Plant Design", *POWER Magazine*, August 2007.
- B11. Widmann, J., "Coupling Computational Fluid Dynamics with Process Modeling for Improved Plant Design", Presented at AspenTech User Conference, Houston, TX, April 2008.
- B12. Zitney, S.E., "Power Plant Virtual Engineering with Advanced Process Co-Simulation", Presented at the Virtual Engineering Workshop, Ames, IA, May 2006.
- B13. Zitney, S. E., Osawe, M., Collins, L., Ferguson, E., Sloan, D., Fiveland, W., Madsen, J., "Advanced Process Co-Simulation of the FutureGen Power Plant," In Proceedings of the 31st International Technical Conference on Coal Utilization & Fuel Systems, Clearwater, FL, May 2006.

- B14. Zitney, S.E., "CAPE-OPEN Integration for Advanced Process Engineering Co-Simulation", In Proceedings of the AIChE 2006 Annual Meeting, Paper 535a, San Francisco, CA, November 2006.
- B15. Zitney, S.E., McCorkle, D., Yang, C., Jordan, T., Swensen, D., Bryden, M., "Towards the Integration of APECS and VE-Suite for Virtual Power Plant Simulation", Presented at the Virtual Engineering Workshop, Ames, IA, May 2007.
- B16. Zitney, S.E., "Computational Research Challenges and Opportunities for the Optimization of Fossil Energy Power Generation Systems", In Proceedings of the 32nd International Technical Conference on Coal Utilization & Fuel Systems, Clearwater, FL, June 2007.
- B17. Zitney, S.E., "Process/equipment co-simulation for design and analysis of advanced energy systems", *Computers & Chemical Engineering*, 34(9), 1523–1542, 2010.

6. List of Acronyms and Abbreviations

Name	Description
AD	Aspen Dynamics software
ANN	Artificial Neural Network
APECS	Advanced Process Engineering Co-Simulation
CAPE-OPEN, CO	Computer Aided Process Engineering -- Open Simulation Environment Interface definitions for exchanging information with process simulation software (www.colan.org)
CFD	Computational Fluid Dynamics
COM	Component Object Model -- refers both to a specification and implementation developed by Microsoft Corporation that provides a framework for integrating software components
CORBA	Common Object Request Broker Architecture -- a specification of a standard architecture for object request brokers (ORBs)
CW	Configuration Wizard component of APECS
DOE	U.S. Department of Energy
DPM	Discrete Phase Model in FLUENT
DX	DesignXplorer -- ANSYS parametric analysis software
EKM	Engineering Knowledge Manager -- ANSYS database software that serves as the model repository for APECS
HRSG	Heat Recovery Steam Generator
IC	Integration Controller component of APECS
IGCC	Integrated Gasification Combined Cycle
LAN	Local Area Network
NETL	National Energy Technology Laboratory
PCA	Principal Component Analysis
PME	Process Modeling Environment
POD	Proper Orthogonal Decomposition
ROM	Reduced-Order Model
UDF	User-Defined Function in FLUENT
UI	User Interface
VE-PSI	Virtual Engineering Process Simulator Interface
VOF	Volume of Fluid multiphase model in FLUENT

Technical Report

Reduced Order Model Based on Principal Component Analysis For Process Simulation and Optimization

Yi-dong Lang and Lorenz T. Biegler

**Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213**

Abstract

By replacing conventional simulation models of the important equipment with Computational Fluid Dynamic (CFD) models, it is clear that we can obtain more reliable and accurate results. Moreover, solving optimization problems based on these extended models will also be closer to the reality. Since solving the CFD model is time-consuming compared with the process simulator itself, it is essential to develop reduced order models (ROMs) based on CFD results in order to approximate the CFD models in the flowsheet. For instance, Proper Orthogonal Decomposition (POD) is a popular method for ROM in the area of fluid dynamic research. On the other hand, it is difficult to apply POD to process engineering, because of its limitations with complicated geometries, sophisticated models for thermo-dynamic properties, chemical reactions, flow patterns, and other related aspects. In this report, we propose a strategy to develop the ROM based on Principal Component Analysis (PCA), instead of POD. With the proposal methodology, and by taking advantage of the commercial software of simulators and CFD (for example, Aspen Plus and Fluent), all these limitations are tackled successfully.

In particular, this approach has no restrictions due to complicated equipment geometries and there is no additional burden to manage the PDEs in the model explicitly. Importantly, the validity of the ROM within well-specified ranges of the inputs is more robust and the CPU time is significantly reduced. Typically it is only one CPU second for ROM, compared with more than 2,000 seconds for the CFD model.

Two case studies are implemented that apply and demonstrate this methodology. They show strong support our proposed approach.

Keywords: ROM, POD, PCA, CFD, Simulation

Table of Contents

1.	INTRODUCTION.....	3
2.	LITERATURE SURVEY.....	5
2.1	LITERATURE SURVEY FOR POD.....	6
2.1.1	<i>Fundamentals of POD.....</i>	<i>6</i>
2.2	LITERATURE SURVEY FOR PCA.....	11
2.2.1	<i>Fundamentals of PCA.....</i>	<i>12</i>
2.2.2	<i>Applications of PCA in Literature.....</i>	<i>14</i>
2.3	APPROACHES OF DECOMPOSITION.....	15
2.3.1	<i>Singular Value Decomposition(SVD).....</i>	<i>15</i>
2.3.2	<i>Lanczos method.....</i>	<i>15</i>
3.	EXPERIENCES WITH POD.....	19
3.1	ALGORITHM FOR OBTAINING POD MODES.....	19
3.2	IMPLEMENTATION OF POD ON BURGER’S EQUATION.....	22
3.3	IMPLEMENTATION OF POD ON NAVIER-STOKES EQUATIONS.....	25
3.4	SUMMARY.....	32
3.4.1	<i>Conceptual comparison of POD and PCA.....</i>	<i>32</i>
3.4.2	<i>Restrictions of POD.....</i>	<i>34</i>
3.4.3	<i>Why we proposed PCA based ROM.....</i>	<i>35</i>
4.	METHODOLOGY OF PCA BASED ROM.....	37
4.1	VECTORS, SPACES AND STATE SPACE MODEL.....	37
4.2	THEORETICAL CONCEPTS OF PCA AND SVD.....	38
4.3	METHODOLOGY OF PCA BASED ROM.....	39
5.	CASE STUDIES.....	42
5.1	CASE STUDY I: FLOW IN MANIFOLD.....	42
5.2	CASE STUDY II: GAS TURBINE COMBUSTOR.....	48
5.2.1	<i>Overview of Gas Turbine combustor.....</i>	<i>49</i>
5.2.2	<i>Experimental Design.....</i>	<i>51</i>
5.2.3	<i>Building the Database.....</i>	<i>53</i>
5.2.4	<i>PCA Applied to the Database.....</i>	<i>53</i>
5.2.5	<i>Regression using Neural Network.....</i>	<i>54</i>
5.2.6	<i>Evaluation of ROM.....</i>	<i>56</i>
5.2.7	<i>Remarks.....</i>	<i>61</i>
5.2.8	<i>Parametric ROM.....</i>	<i>68</i>
5.2.9	<i>Summary of case study II.....</i>	<i>71</i>
6.	CONCLUSIONS.....	72
	APPENDIX: CATEGORIES OF THE FLAMES.....	74
	REFERENCES.....	77

1. Introduction

Steady-state process simulators (PS) solve equations consisting of mass-energy balances, phase and reaction equilibria, either sequentially or simultaneously. Generally, they do not consider fluid dynamic behavior within the equipment or the equipment geometry, which also affect the process. Additionally, they cannot account for operation parameters, such as shaft speed in a stirred tank reactor. In contrast, Computational Fluid Dynamics (CFD), on the other hand, focuses on all of these aspects. In other words, PS is suited for macroscopic processes, while CFD is adept at microscopic behavior inside the equipment. In process design, analysis and optimization, it is not uncommon that engineers are interested not only in process design but also in equipment behavior, as this has an important effect on the process. For example, shaft speed affects the reactions and the output in a stirred tank reactor. Similarly, nozzle design is critical for gasifiers, spray drying units and many others. .

For advanced process design, analysis and optimization, we intend to use both PS and CFD models for “*Co-Simulation*” (S. Zitney, 2004). Since 2004, Co-simulation has been implemented through the efforts by NETL and its contractors (D. G. Sloan et al, 2005), in particular, through development of the software package APECS (*Advanced Process Engineering Co-Simulation*). APECS invokes Aspen Plus as a process simulator and Fluent as the equipment simulator using the CAPE-OPEN (CO) industrial standard. With APECS, one can replace the relevant equipment in flowsheet of Aspen Plus with Fluent CFD models. With co-simulation, the results of the process design or solution of the process optimization are closer to industrial practice. In particular, M. Osawe (2002) presented a benefit of optimal shaft speed of a CSTR using APECS which could not be obtained by using Aspen Plus only.

It is evident that CFD models, especially for large, complicated equipment are both computationally- and memory-intensive. One of the solutions for this prohibitive drawback is the implementation of *Reduced Order Models* (ROMs) to approximate the rigorous CFD model.

Over the past 18 months we have been focusing on ROM methodologies through a literature investigation as well as working on Fluent case studies. As a result of this search we propose an improved algorithm that targets specific needs of CFD-based ROMs. The resulting approach will achieve the task goals more effectively and efficiently with much less effort and time. The proposed algorithm is based on Principal Components Analysis (PCA) instead of Proper Orthogonal Decomposition (POD). The algorithm is rooted in the following factors:

- The ROM objects are process equipment with complex geometries that also describe behavior of complex chemical species. In order to simulate these processes, numerous models must be introduced that calculate necessary information such as thermodynamic properties of the chemicals or materials, phase equilibrium, chemical reactions, etc. All of these features make the task

more difficult compared to fluid dynamics models where the POD method has become popular.

- Because our models are steady state, there are no demands on time-varying behavior of state variables. Differential equations in time can be avoided.
- Facilities in process simulators and CFD tools are well developed to introduce those models and handle sophisticated computations. In particular, the CFD code has already solved the complicated Partial Differential-Algebraic Equations (PDAEs) model with its own proprietary or in-house algorithms. Thus, it seems impractical to redevelop and solve these PDAEs again through POD during the development of the ROM.

The proposed algorithm takes full advantage of the above characteristics, and directly uses accurate solutions obtained by the CFD dataset. For our reduced order model, we imply that, with the independent variables (inputs) specified, the model will determine values of all of the dependent (i.e., state and output) variables. We note that the form of our ROMs is not equation-based, as in POD, but relies on a tabular construction. To illustrate, the simple model, $y = \sin(x)$ implemented as an intrinsic function is actually calculated as an interpolated lookup table. With a converged CFD model, the solution is obtained similarly, i.e, given coordinates (e.g. x, y, z), any state at any point in the field bounded by the equipment can be extracted. Thus, to build our ROM, if we design a set of inputs, and let the CFD code calculate a set of corresponding solutions, then we can also develop more powerful models in tabular form. The model is then able to give us the state values according to specified inputs and coordinates. Here we no longer consider the restrictions of geometric shape and form of the PDAEs, as these are already handled within powerful CFD tools.

Instead, our task is to utilize all the information stored in the CFD solutions to develop the ROM. With the developed ROM of the equipment model, we are able to obtain the state values in the field and outputs from the equipment corresponding to any specified input in our domain.

ROM approaches have been applied in areas such as boundary control (M. Fahl 2001) and aerodynamics (Willcox, 2000), and Proper Orthogonal Decomposition (POD) is the most popular method used in ROM development. Usually, the ROM approximation for a set of PDAEs is developed using a set of accurate solutions obtained by solving the rigorous PDAE model. With POD, one first decomposes the PDAE solution, $u(t, \mathbf{x})$ into the form

of $\sum_{j=1}^r \alpha_j(t) \psi_j(\mathbf{x})$ where t is temporal and \mathbf{x} is spatial. Here $\psi_j(\mathbf{x})$, $j=1, r$ are the so-

called POD basis functions, which can be derived from a set of solutions (snapshots) of the PDAE model. Using the decomposed form of the profiles, we replace $u(t, \mathbf{x})$ in the Galerkin projection of the PDAE in order to obtain a set of Ordinary Differential Equations (ODE). Solving the ODE, one can obtain the coefficients $\alpha_j(t)$, $j=1, r$.

Finally, POD based ROM is given by: $u(t, \mathbf{x}) = \sum_{j=1}^r \alpha_j(t) \psi_j(\mathbf{x})$.

Most POD literature case studies do not address issues such as geometry (arising from the shape of the equipment), complicated mass mixtures that require thermodynamic property calculations, complicated energy calculation, phase equilibrium and chemical reactions. Having applied POD to some simple examples, we found that it is impractical for POD to be used in developing ROMs for APECS.

We have considered two examples for development of ROMs with POD. The first is Burgers equation having only one PDE; the second consists of 2-D Navier-Stokes equations. Our results show that even with these simple examples, the implementations are still tedious, error-prone. The resulting ODEs are complicated to be reformulated and solved, and this motivates us to search for an alternate approach. Moreover, we recognize that APECS deals only with steady state systems, which is different from the cases in POD literature. This motivated us to switch our attention to the treatment of datasets and images, where Principal Component Analysis (PCA) is popular. In fact, PCA and POD are closely related and rooted in the same method called *Karhunen-Loeve Decomposition* (KLD). However, they are used in different areas. Here, we propose a new methodology for APECS in terms of *PCA based ROM*. With the proposed methodology, all the restrictions in POD are not obstacles anymore. Instead, it is shown that the methodology of PCA based ROM is simple and easy to implement, and the resulting ROM is efficient and effective.

In order to examine the effectiveness and efficiency of our proposed methodology, ROM based on PCA, we spent more than twelve months to conduct two case studies a) Flow in a manifold and b) Gas turbine combustor. The former is a simple example with only one input, the inlet mass flow rate. The latter case study has ten inputs and is a component of the FutureGen flowsheet. A comprehensive set of techniques is invoked in this case study: experimental design (e.g. Latin Hypercube Sampling, LHS), Singular value decomposition (SVD), neural networks (NNet), visualization of data scatter as well as Principal component analysis (PCA). 128 cases of the FLUENT model are implemented to formulate the database and an additional 45 cases of Fluent are implemented for ROM evaluation. Having tested and evaluated the ROM with given ranges of ten inputs, we recommend to introduce and develop the concept of “*Parametric ROM*,” similar to referring to “Parametric Programming” applied in the control field (M. Morari, 2007).

In next section, we provide a literature survey related not only to POD but also to other topics of ROM. As examples to implement POD based ROM, in section 3, we give the results of derivations of POD-based ROM for Burgers equation and 2-D Navier-Stokes Equation; these show the motivation of our proposed methodology. In section 4, we will discuss the methodology of PCA based ROM. To illustrate the advantages of the proposed methodology, we present two case studies in section 5. Finally we conclude this report.

2. Literature survey

At the beginning of the survey, we worked on the scope of POD-based ROM. With a comprehensive understanding of POD, we felt there were obstacles for POD used in APECS. Then we expanded the scope of literature research into some other topics such as balanced truncation, neural network, finite elements and PCA as well.

2.1 Literature survey for POD

2.1.1 Fundamentals of POD

Generally, this approach is classified as continuous or infinite-dimension and finite dimension (or snapshot) POD (A. Chatterjee, 2000, W. Cazemier, 1968). In this survey, we review finite dimension POD only. Assume that a PDAE model in form of

$$F(\dot{z}_x(x,t), \dot{z}_t(x,t), z(x,t), x,t) = 0 \quad (2-1)$$

has the solution $z(x,t)$, a spatiotemporal function. We split the solution into a summation of products with both temporal function and spatial function terms:

$$z(x,t) \approx \sum_{k=1}^r a_k(t)\phi_k(x) \quad (2-2)$$

where $\phi_k(x)$, $k=1,r$ are the basis functions and r is the rank of POD. There is no fundamental difference between t and x . In steady state, if there is no temporal term, one can use a spatial dimension instead, as we did with Burgers equation given in a later section. Also the basis functions $\phi_k(x)$, $k=1,r$ can be derived through some method including Fourier series, Legendre polynomials, Chebyshev polynomials, as well as POD. It has been proved that the optimal approximation of (2-1) in the sense of least squares is Proper Orthogonal Decomposition (POD), i.e. it can capture more energy than other method with the same order. POD is also known as *Karhunen-Loeve decomposition*, the *method of empirical eigenfunctions* or *Principal Component Analysis* (M. Fahl, 2000). The special, ordered, orthonormal functions $\phi(x)$ are called the *POD basis functions*, which have the property:

$$\phi_i(x)\phi_j(x) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases},$$

Taking advantage of orthogonality, we have

$$z(x,t)\phi_j(x) = \sum_{k=1}^r (\alpha_k(t)\phi_k(x))\phi_j(x) \quad (2-3)$$

$$\alpha_k(t) = z(x,t)\phi_k(x) \quad (2-4)$$

Equation (2-4) shows that the determination of the coefficient function $\alpha_k(t)$, $k=1,r$ depends only on its corresponding basis function $\phi_k(t)$, $k=1,r$ and not other basis functions. To develop the ROM, we need a set of n snapshots stored in a

matrix $A \in \mathfrak{R}^{m \times n}$, $A_i = z(x, t_i)$, $i = 1, n$. A simply optimization problem is formed to prove the optimality of POD basis functions and to show how to obtain them.

$$\max \frac{1}{n} \|(\mathbf{A}^T \boldsymbol{\psi})\|^2 \quad \text{s.t.} \quad \boldsymbol{\psi}^T \boldsymbol{\psi} = 1 \quad [\text{P-1}]$$

Taking the Lagrangian of [P-1] gives

$$L = \frac{1}{n} \|(\mathbf{A}^T \boldsymbol{\psi})\|^2 - \bar{\lambda} (\boldsymbol{\psi}^T \boldsymbol{\psi} - 1)$$

with optimality conditions: $\frac{\partial}{\partial \boldsymbol{\psi}} L = 2 \frac{1}{n} \mathbf{A}(\mathbf{A}^T \boldsymbol{\psi}) - 2 \bar{\lambda} \boldsymbol{\psi} = 0$

$$\text{and this leads to: } \mathbf{A}\mathbf{A}^T \boldsymbol{\psi} = \lambda \boldsymbol{\psi} \quad \text{where } \lambda = n \bar{\lambda} \quad (2-5)$$

Eq. (2-5) indicates that the POD modes are the eigenvectors of the correlation matrix of the snapshot, $\mathbf{A}\mathbf{A}^T$. If we denote the first solution of [P-1] as Ψ_1 , then we can solve [P-1] again with the additional constraint $\boldsymbol{\psi}^T \Psi_1 = 0$ to get a solution $\boldsymbol{\psi}_2$. Implementing the same procedure we obtain all n solutions Ψ_j and λ_j , $j = 1, n$, and we can store the eigenvectors in descending order of their corresponding eigenvalues for the eigenvector matrix, $\boldsymbol{\Psi}$. Note that the columns Ψ_j , $j = 1, n$ in $\boldsymbol{\Psi}$ are also the POD modes which have the orthogonality property

$$\Psi_i^T \Psi_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Since the snapshot columns are linearly independent, the matrix $\mathbf{A}\mathbf{A}^T$ has n nonzero eigenvalues. If we chose the first r eigenvectors as POD basis functions in Eq. (2-2), then we can calculate how much energy in the snapshot is captured, which also is the cut-off criterion

$$R_r = \frac{\sum_{k=1}^r \lambda_k}{\sum_{k=1}^n \lambda_k} \leq 1 \quad (2-6)$$

Thus, we showed that the POD modes are the eigenvectors of correlation matrix of the snapshots, $\mathbf{A}\mathbf{A}^T$. This also means that POD modes can be derived from solving the eigenproblem of the correlation matrix $\mathbf{A}\mathbf{A}^T$.

It is worth noting that the POD modes are just a set of data, in other words, they are not spatial functions as expected in expression (2-2). To make them spatially dependent, some functions are introduced. Typical spatial function used in POD is *linear finite element basis function* (M. Fahl, 2000).

Let a function vector $\Phi(x) = [\varphi_1(x) \ \varphi_2(x) \ \cdots \ \varphi_m(x)]$ denote a collection of finite element linear basis functions defined as follows (refer to Figure 2-1),

1. piecewise linear and continuous between the nodes
2. $\varphi_i(x_j) = \begin{cases} 0 & \text{for } i \neq j \\ 1 & \text{for } i = j \end{cases}$
3. $\varphi_i(x) > 0$ only when $x_i < x < x_{i+1}$ for $i = 0, \dots, m+1$

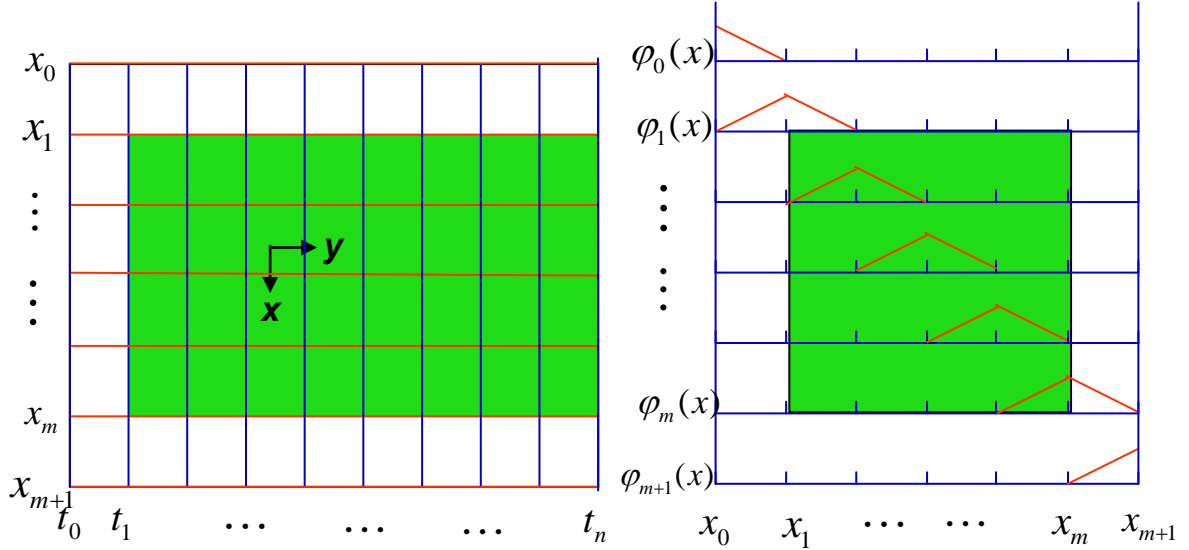


Figure 2-1. Finite element basis function

It is noted that the snapshot according to the lattice does not contain initial and boundary conditions as shown in green area in Figure 2-1. In that case, a very important property of the test function, consisting of basis function $H_0^1(x_0, x_{n+1})$, holds. That means the function has first derivatives and vanishes at $x = x_0$ and $x = x_{n+1}$, which is crucial for Galerkin projection used later.

With these functions, we can formulate the spatial POD basis functions.

$$\phi_j(x) = \Phi(x)\Psi_j, \quad j = 1, r \quad (2-7)$$

It is important that these POD spatial basis functions inherit the property of $H_0^1(x_0, x_{n+1})$ from finite element basis functions.

Now we are ready to obtain coefficients as temporal functions $\alpha_k(t)$ in Eq. (2-2). But we have to go back to the PDAE model again. With decomposed solution (2-2) and (2-7) obtained using Galerkin projection, we convert the original PDAE to DAEs which can be used to solve for $\alpha_k(t)$, $k = 1, r$. Remember, for each state, there are r unknown $\alpha_k(t)$,

which means for each state in original PDAEs, r DAEs will be produced. If there are p states in the PDAE model, the resulted DAEs will have dimension of $(p \times r)$.

2.1.2 Applications of POD in literature

POD is used as a ROM technique mostly in processes relating to fluid dynamics. Among literature of POD there are two categories of using POD for reduced-order models.

- 1) The system is modeled by PDEs such as Navier-Stokes equations.
- 2) The system is modeled by state space models (Willcox, 2000; Rowley, 2005)

The materials involving in the system are mainly water or air. The properties of the materials in the PDAE models are density and viscosity that are easy to obtain or calculate. The most popular case studies in the first category are driven cavity flow in 2-D (M. Fahl, 2000; B. Kragel, 2005; Cazemier, W, 1968) and plane Couette flow in 3-D (J. Moehils, et al, 2003; T.R. Smith et al, 2005) as shown in Figure 2-2.

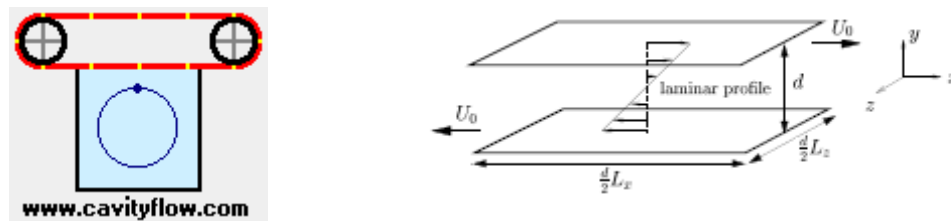


Figure 2-2. Driven cavity flow and plane Couette flow

From Figure 2-2 we can characterize these two cases as follows:

- The number of model equations is small. Only Navier-Stokes equations are considered.
- Geometric shapes of the objects are simple which makes boundary conditions easy to formulate.
- There are no equations for calculation of material properties.

K. Kunisch and S. Volkwein (1999) derived ROM results for Burgers equation with POD. This system is a single spatiotemporal PDE with one spatial dimension. Here we provide the detailed derivation to show the complexity of POD application, even with one PDE. P.G. Cizmas et al.(2003) used POD to simulate a laboratory scale fluidized bed. The model consists of four PDEs in the rectangular field (2-D).

Typical cases of the second category can be found in K. Willcox's work dealing with model reduction of CFD system using POD for real-time simulation and control. Willcox states that the system of PDE governing a general fluid flow can be discretized in the spatial domain using a CFD formulation to yield a set of nonlinear ODEs. For unsteady flows, these ODEs can be linearized about the steady state solution to obtain a linear CFD

model that is valid for small perturbation of the flow from the steady-state condition. A general linearized CFD model for unsteady-state can be written as

$$G : \frac{d}{dt} x = Ax + Bu, \quad y = Cx + Du \quad (2-8)$$

where $x \in \mathfrak{R}^n$ contains the n unknown perturbation flow quantities at each point in the computational grid. The vectors u and y contain the system inputs and outputs, respectively. The definition of the inputs and outputs will depend upon the problem at hand. For active control applications, the outputs might monitor a flow condition at a particular location, which varies in response to a disturbance in the incoming flow, while inputs describe both actuators and disturbances.

The order of the system is still prohibitively high for many applications. Therefore Willcox develops a method to find a low-order, stable, linear time invariant (LTI), state-space model

$$\hat{G} : \frac{d}{dt} \hat{x}(t) = \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t), \quad \hat{y}(t) = \hat{C}\hat{x}(t) + \hat{D}u(t) \quad (2-9)$$

where $\hat{x}(t) \in \mathfrak{R}^k$. Model (2-9) approximates the given model (2-8) well. Willcox estimates that typically, A in model (2-8) is a sparse, square matrix of very large dimension, $n > 10^4$, and the desired order k in model (2-9) is less than 50.

The errors of the approximation are given by:

$$\|\hat{G} - G\|_{\infty} = \sup_{\omega \in \mathfrak{R}} \|\hat{G}(j\omega) - G(j\omega)\| \quad (2-10a)$$

$$\|\hat{y} - y\| = \int_t |\hat{y}(t) - y(t)|^2 dt \quad (2-10b)$$

Willcox then uses snapshot POD to reduce the order of x by an appropriate choice of input to the simulation:

$$x = V\hat{x} \quad (2-11)$$

where $V \in \mathfrak{R}^{n \times k}$ contains k columns as the POD basis modes. Using state-space models as the discretized PDE in spatial domain with CFD formulation, Willcox avoids introducing spatial variables into the system model, which eliminates restrictions due to geometric shapes. There is no need to define boundary conditions for the geometric bounds of the flow field. Also there is no need to work with the PDEs again as the authors in the first category do. Consequently, this methodology has two features to handle large scale, complicated systems.

The main problem in POD-based ROM is that the POD modes are sensitive to the input used to obtain the snapshot. When the input changes during simulation or control calculations, there is no guarantee that the POD modes are still valid. Some authors tried to overcome the limited validity problem of POD. For example, M. Fahl (2000) proposed trust-region POD for flow optimal control. Later, B. Kragel (2005) developed streamline

diffusion POD. Both Fahl and Kragel developed POD methods that belong to the first category.

In addition, M. Schelegel et al (2002) use POD in context of dynamic optimization of a process flowsheet. They tried to reduce order of the dynamic system expressed in DAEs of state and independent variables. First they transfer the original space into a space better revealing the important contribution to the process dynamics,

$$x - x^* = Uz$$

where $x \in \mathfrak{R}^n$ is original state vector, $x^* \in \mathfrak{R}^n$ is non-zero nominal operation condition. U is a homomorphic transformation and $z \in \mathfrak{R}^n$ is transformed vector. Then the state z is decomposed into dominant and non-dominant subspaces, z_1 and z_2 , respectively.

$$Uz = U_1z_1 + U_2z_2, \quad U = [U_1 \quad U_2]$$

By setting $z_2 = 0$, the only space involving in DAEs is z_1 and the order of the original DAEs is reduced. Here POD is used to decompose U to get U_1 and U_2 .

Finally, with the state-space model and concepts of controllability and observability Grammians, balanced POD is proposed to take account of both inputs and outputs of the system in order to determine which states to retain in the reduced-state representation (B.C. Moore, 1981; K. Willcox, 200X; C.W. Rowley, 2005).

2.2 Literature Survey for PCA

Fluent, a powerful CFD software package, generally displays the solutions of the discretized CFD model in visualized color contours of the flow field, or inside the geometry of the underlying equipment. These also include contour images of the temperature, pressure, velocity components, and so on. Figure 2-3 shows the contour image of temperature of a heat exchanger displayed by Fluent.

Let “*set*” denote the images of all states corresponding to one input. Let “*group*” denote the images of the same state corresponding to all inputs. Each set of images consists of all state variables at the nodes of the mesh, which corresponds to a specified input of the equipment. Let p and q denote the numbers of variables in the input space and the state space, respectively. There will be q groups at hand and p images in each group. It is obvious that in solving the PDAE model with CFD, the finer the mesh is, the more accurate the solution will be. However, this makes the dimension of the images prohibitively high. So to represent the obtained solution, a coarse mesh appears to be sufficient for ROM construction and is strongly recommended.

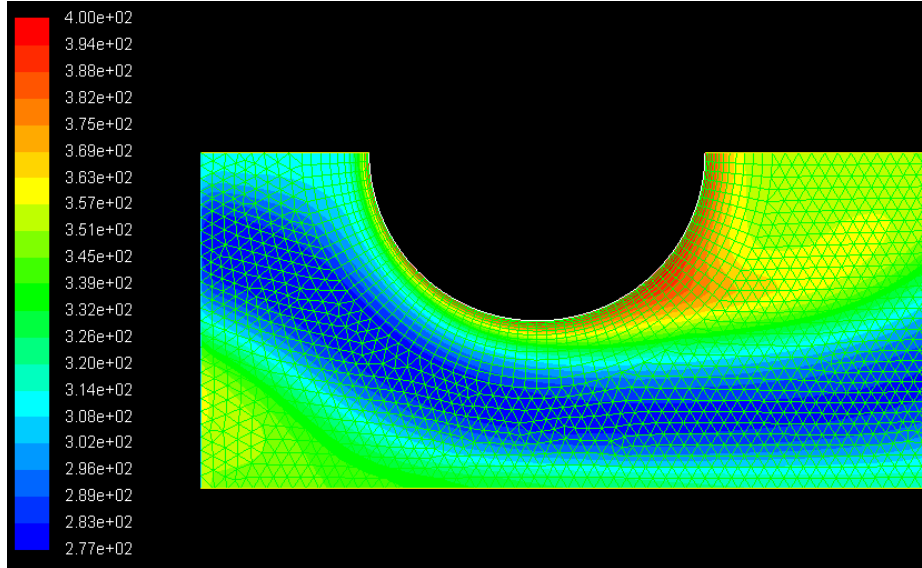


Figure 2-3. A color filled contour image of temperature displayed by Fluent

2.2.1 Fundamentals of PCA

PCA is probably the oldest and best known multivariable analysis technique. It was first introduced by Pearson in 1901, and developed thereafter by Hotelling in 1933 (Jolliffe 2002). In statistics, PCA is a powerful tool to simplify a multidimensional dataset to a lower dimensional analysis. Mathematically, PCA is a linear transformation that transforms the data to a new orthogonal coordinate system such that the greatest variance by any projection of the data lies in the first coordinate, i.e., the first Principal Component (PC), the second greatest variance on the second PC, and so on.

PCA is also a useful statistical technique in face recognition and image compression. Also it is a common technique for reducing the number of variables and detecting structure in the relationships between variables in multi-dimensional dataset. The PCA provides a transformation of the original data space spanned by the vectors called *Principal Components* (PCs). Assume a dataset or a snapshot is given in a matrix, $\mathbf{X} \in \mathfrak{R}^{m \times n}$, then any column in matrix \mathbf{X} can be decomposed as a linear combination of its PCs,

$$\mathbf{X}_i = \sum_{j=1}^n \boldsymbol{\Psi}_j \boldsymbol{\Theta}_i \quad \text{and} \quad \mathbf{X} = \boldsymbol{\Psi} \boldsymbol{\Theta} \quad (2-12)$$

where the columns of the matrix $\boldsymbol{\Psi} \in \mathfrak{R}^{m \times n}$ are PCs, and entries in vector $\boldsymbol{\Theta}$ are scores that are the coordinates of the data elements in transformed vector space. The PCs are the eigenvectors of correlation matrix $\mathbf{X}\mathbf{X}^T$ which arranged according to their corresponding eigenvalues in descent order of $\lambda_1 > \lambda_2 > \dots > \lambda_n$. There are a few methods to calculate the PCs such as *Singular Value Decomposition* (SVD), Lanczos method and the Arnoldi method. Among them, SVD is the simplest. For instance, matrix \mathbf{X} can be decomposed by SVD after eliminating zeros as

$$\mathbf{X} = \mathbf{W}\mathbf{\Sigma}V^T \quad (2-13)$$

where $\mathbf{\Sigma} = \text{diag}(s_1, s_2, \dots, s_n)$ and the square matrix $V \in \mathfrak{R}^{n \times n}$ are the eigenvectors of $\mathbf{X}^T\mathbf{X}$ with the eigenvalues of $\lambda_j, j = 1, n$, too. It can be proved that eigenvalues of correlation matrix $\mathbf{X}\mathbf{X}^T$ and matrix $\mathbf{X}^T\mathbf{X}$ are connected to SVD as $\lambda_j = s_j^2$. These stem from the variance in the dataset captured by the corresponding PC.

Using Eq. (2-13) with rank r that results from the cut-off criterion in (2-17), we can express reduced order dataset as

$$X \approx \hat{\mathbf{X}} = W^{(r)}\mathbf{\Sigma}^{(r)}(V^{(r)})^T \text{ and } \hat{\mathbf{X}} = \mathbf{\Psi}\mathbf{\Theta} \quad (2-14)$$

where the superscript ^(r) indicates that the first r columns taken from the original matrix to formulate a new matrix, and the reduced score matrix is:

$$\mathbf{\Theta} = \begin{bmatrix} s_1 V_{11} & s_1 V_{21} & \cdots & s_1 V_{n1} \\ s_2 V_{12} & s_2 V_{22} & \cdots & s_2 V_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ s_r V_{1r} & s_r V_{2r} & \cdots & s_r V_{nr} \end{bmatrix} \in \mathfrak{R}^{r \times n}. \quad (2-15)$$

Then any column X_i in the dataset \mathbf{X} can be expressed in reduced order space as

$$\mathbf{X}_i \in \mathfrak{R}^m = \mathbf{\Psi}\mathbf{\Theta}_i \quad (2-16)$$

The majority of its variances in PC space captured by the reduce order dataset can be calculated as

$$R_r = \frac{\sum_{j=1}^r \lambda_j}{\sum_{j=1}^n \lambda_j} = \frac{\sum_{j=1}^r s_j^2}{\sum_{j=1}^n s_j^2} \leq 1 \quad (2-17)$$

Equation (2-17) is usually used as the cut-off criterion and the truncation error is

$$e_r = (1 - R_r) = \frac{\sum_{j=r+1}^n \lambda_j}{\sum_{j=1}^n \lambda_j} = \frac{\sum_{j=r+1}^n s_j^2}{\sum_{j=1}^n s_j^2} \quad (2-18)$$

2.2.2 Applications of PCA in Literature

PCA applications are found largely in image recognition, data analysis and compression. Jolliffe (2002) gives a number of systematic examples in his book. Q. Jiang is one of the authors who worked on face image recognition with PCA. As shown in Figure 2-4; there is a group of images of 593 human faces. First, all images are expressed as a matrix in which each column corresponds to one image detected using software from MIT's Artificial Intelligence Lab.

With PCA, the eigenfaces are calculated and the original images can be approximated well with the first 100 eigenfaces. This is good for later finding image patterns for recognition. The purpose of quoting these photos here is that if we consider the filled color contours produced by Fluent with different boundary conditions in the same way; we certainly can work on them similarly to reconstruct and represent these contours in reduced order.

It is worth noticing that there are no rules on how the faces were created, even though there are rules on how Fluent created the contour images. Nevertheless, it is our interest to reconstruct and express the contour images in low-dimensional space, and then find the relationships between the images and the inputs. In this sense, PCA allows us to establish this relationship with Fluent models.



Figure 2-4. Images of human faces used by PCA method; 593 pictures were processed to construct 100 eigenfaces.

2.3 Approaches of Decomposition

Both POD and PCA need to calculate new basis functions; there are three methods to do so in literature: SVD, Lanczos method and Arnoldi method. According to Willcox, the Arnoldi method is recommended for aerodynamics where the snapshot is in the frequency domain. However, as it is similar in concept to the Lanczos method, we will not consider it in this survey.

2.3.1 Singular Value Decomposition(SVD)

Singular value decomposition is the simplest method to calculate the principal component and the scores given a data matrix. As we mentioned in previous section that a given real matrix $\mathbf{X} \in \mathfrak{R}^{m \times n}$ with $m > n$ can be written using a so-called singular value decomposition (SVD) of the form (2-13)

$$\mathbf{X} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^T$$

Note that there are several conflicting notational conventions in use in the literature. Press (*et al.* 1992) define matrix $\mathbf{W} \in \mathfrak{R}^{m \times n}$, $\mathbf{\Sigma} \in \mathfrak{R}^{n \times n}$ and $\mathbf{V} \in \mathfrak{R}^{n \times n}$. However, *Mathematica* defines $\mathbf{W} \in \mathfrak{R}^{m \times m}$, $\mathbf{\Sigma} \in \mathfrak{R}^{m \times n}$ and $\mathbf{V} \in \mathfrak{R}^{n \times n}$. We prefer the former to the later. With SVD and ranked principal components and scores the original dataset can be approximated as (2-16), i.e. $\mathbf{X} = \mathbf{\Psi}\mathbf{\Theta}$.

Essentially $6mn^2 + 11n^3$ floating point operations (flops) (G.H. Golub and C.F.Van Loan, 1989) are necessary to compute all singular values and the related left singular vectors of $\mathbf{X} \in \mathfrak{R}^{m \times n}$. Therefore SVD is not suitable for very large problems.

2.3.2 Lanczos method

The relationship between SVD and eigenvalue problems can be described by the following result (M. Fahl, 2000)

Lemma: Let $\mathbf{X} \in \mathfrak{R}^{m \times n}$, $m \geq n$, with SVD $\mathbf{X} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^T$, where $\{\mathbf{W}_i\}_{i=1}^m$ form the columns $\mathbf{W} \in \mathfrak{R}^{m \times m}$, $\mathbf{\Sigma} = \text{diag}(\{s_i\}_{i=1}^n) \in \mathfrak{R}^{m \times n}$ and $\{\mathbf{V}_i\}_{i=1}^n$ form the columns of $\mathbf{V} \in \mathfrak{R}^{n \times n}$. Then, the following results hold:

1. The eigenvalues of the symmetric matrix $\mathbf{B}_1 = \mathbf{X}^T \mathbf{X} \in \mathfrak{R}^{n \times n}$ are s_i^2 . The right singular vectors \mathbf{V}_i are corresponding orthogonal eigenvectors.
2. The eigenvalues of the symmetric matrix $\mathbf{B}_2 = \mathbf{X}\mathbf{X}^T \in \mathfrak{R}^{m \times m}$ are s_i^2 . The left singular vectors \mathbf{W}_i are corresponding orthogonal eigenvectors.

3. The eigenvalues of the symmetric matrix

$\mathbf{B}_3 = \begin{bmatrix} 0 & \mathbf{X} \\ \mathbf{X}^T & 0 \end{bmatrix} \in \mathfrak{R}^{(m+n) \times (m+n)}$ are $s_i, -s_i$ and $m - n$ zeros. The vectors

$\frac{\sqrt{2}}{2}(W_i, V_i)^T$ and $\frac{\sqrt{2}}{2}(W_i, -V_i)^T$ are corresponding orthogonal eigenvectors for the eigenvalues s_i and $-s_i$, respectively.

The Lemma shows that the computation of SVD for a rectangular matrix is equivalent to solving the eigenvalue and eigenvector problems of several specific symmetric matrices.

$$Bz = \lambda z$$

This is the main idea of Lanczos method. Let $B \in \mathfrak{R}^{n \times n}$ be a symmetric matrix. For any nonzero vector $z \in \mathfrak{R}^n$ we define the *Rayleigh quotient* by

$$r(z) = \frac{z^T Bz}{z^T z}$$

Then let its gradient be

$$\nabla r(z) = \frac{2}{z^T z} (Bz - r(z)z) = 0, \quad (2-19)$$

leading to $Bz = r(z)z$, which shows that z is the eigenvector and $r(z)$ is the corresponding eigenvalue. Since we are interested in the largest eigenvalue and corresponding eigenvector, we need to solve a maximization problem

$$\lambda_{\max} = r(z^*) = \max_{z \neq 0} r(z) \quad (2-20)$$

To satisfy orthogonality of the eigenvectors, we have to restrict z to a subspace with orthogonal basis using the columns in matrix $\mathbf{Q}^{(j)} = [\mathbf{Q}_1 \ \mathbf{Q}_2 \ \cdots \ \mathbf{Q}_j] \in \mathfrak{R}^{n \times j}$.

Let $z = \mathbf{Q}^{(j)} \chi$, where $\chi \in \mathfrak{R}^j$, then Eq. (2-20) becomes

$$\max_{\chi \neq 0} r(\mathbf{Q}^{(j)} \chi) \quad (2-21)$$

When $j = n$, equations (2-20) and (2-21) are identical and give the exact eigenvalue-eigenvector pair, (λ_{\max}, z^*) . But two cases are possible:

- If there is a stationary point $(\chi_*^{(j)})$ of $r(\mathbf{Q}^{(j)} \chi)$ such that $(\mathbf{Q}^{(j)} \chi_*^{(j)})$ is a good approximation of z^* , then a good approximation of λ_{\max} can be expected;
- If above is not satisfied with given $\mathbf{Q}^{(j)} = [\mathbf{Q}_1 \ \mathbf{Q}_2 \ \cdots \ \mathbf{Q}_j]$, then we try to construct a vector \mathbf{Q}_{j+1} such that the corresponding Rayleigh quotient yields $r(\mathbf{Q}^{(j)} \chi_*^{(j)}) \leq r(\mathbf{Q}^{(j+1)} \chi_*^{(j+1)}) \leq \lambda_{\max}$, where $\chi_*^{(j+1)}$ is the stationary point for Eq. (2-21) with enlarged basis matrix $\mathbf{Q}^{(j+1)}$

Requiring $\nabla r(\mathbf{Q}^{(j)} \chi_*^{(j)}) \in \text{span}\{\mathbf{Q}_1, \dots, \mathbf{Q}_j, \mathbf{Q}_{j+1}\}$ and Eq. (2-19) implies $\nabla r(z) \in \text{span}\{z, Bz\}$. Recursively this leads to the problem: Find \mathbf{Q}_{j+1} such that

$$\text{span}\{\mathbf{Q}_1, \dots, \mathbf{Q}_j, \mathbf{Q}_{j+1}\} = \text{span}\{\mathbf{Q}_1, B\mathbf{Q}_1, \dots, B^j \mathbf{Q}_1\} \quad (2-22)$$

The right hand of Eq. (2-22) is known as Krylov subspace defined as

$$K^{(j+1)}(q, B) \in \text{span}\{q, Bq, \dots, B^j q\}$$

Thus, we can move to discuss the Basic Lanczos Iteration. The orthonormal basis of the Krylov subspace in Eq. (2-22) can be derived by exploiting a tridiagonalization of B . For illustration of the idea, we assume that $\mathbf{Q} = \mathbf{Q}^{(n)}$ is chosen such that

$$\mathbf{Q}^T \mathbf{B} \mathbf{Q} = \mathbf{T} \text{ with } \mathbf{Q} \mathbf{e}_1 = \mathbf{Q}_1; \mathbf{e}_1 = (1; 0; \dots; 0)^T; \quad (2-23)$$

where

$$\mathbf{T} = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \dots & 0 \\ \beta_1 & \alpha_2 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \alpha_{n-1} & \beta_{n-1} \\ 0 & \dots & 0 & \beta_{n-1} & \alpha_n \end{bmatrix}$$

:

Hence, the orthonormal matrix \mathbf{Q} tridiagonalizes \mathbf{B} where its first column consists of the starting vector \mathbf{Q}_1 . Using $\mathbf{B}\mathbf{Q} = \mathbf{Q}\mathbf{T}$, Eq. (2-23) guarantees

$$[\mathbf{Q}_1, B\mathbf{Q}_1, \dots, B^{n-1}\mathbf{Q}_1] = \mathbf{Q}[\mathbf{e}_1; \mathbf{T}\mathbf{e}_1; \dots; \mathbf{T}^{n-1}\mathbf{e}_1], \quad (2-24)$$

Thus, Eq. (2-24) shows that the columns of \mathbf{Q} form an orthonormal basis for the Krylov subspace $K^{(n)}(\mathbf{Q}_1, B)$.

Moreover, \mathbf{Q}_j can be calculated iteratively. Using Eq. (2-23), and comparing the columns of $\mathbf{B}\mathbf{Q} = \mathbf{Q}\mathbf{T}$ leads to the three-term recurrence

$$\mathbf{B}\mathbf{Q}_j = \beta_{j-1}\mathbf{Q}_{j-1} + \alpha_j\mathbf{Q}_j + \beta_j\mathbf{Q}_{j+1} \quad (2-25)$$

with $\beta_0\mathbf{Q}_0 \equiv 0$. The \mathbf{Q}_j can be computed iteratively. Obviously, Eq. (2-23) shows that each α_j given by $\alpha_j = \mathbf{Q}_j^T \mathbf{B} \mathbf{Q}_j$ for known \mathbf{Q}_j (Steps 1, 2 in algorithm below). For given α_j , β_{j-1} and \mathbf{Q}_j , Eq. (2-25) yields the relationship for computation of \mathbf{Q}_{j+1}

$$\beta_j\mathbf{Q}_{j+1} = \mathbf{B}\mathbf{Q}_j - \beta_{j-1}\mathbf{Q}_{j-1} - \alpha_j\mathbf{Q}_j \quad (2-26)$$

Thus, the basic Lanczos iteration algorithm can be formulated as shown below

Initialization: choose $\beta_0 = 0$, $Q_0 = 0$ and $b \neq 0$. Set $Q_1 = \frac{b}{\|b\|_2}$

For $j = 1, 2, 3, \dots$ Repeat

- 1) Compute $v = \mathbf{B}Q_j$
- 2) Compute $\alpha_j = Q_j^T v$
- 3) Set $v = v - \beta_{j-1}Q_{j-1} - \alpha_j Q_j$ and $\beta_j = \|v\|_2$
- 4) Set $Q_{j+1} = \frac{v}{\beta_j}$

After k Lanczos iterations, a partial tri-diagonalization, $T^{(k)} \in \mathfrak{R}^{k \times k}$, of symmetric matrix \mathbf{B} is produced. It is efficient to obtain the eigenvalues $\lambda_i^{(k)}$ and the eigenvectors $v_i^{(k)}$ of $T^{(k)}$ because it is tri-diagonal, such that:

$$T^{(k)} v_i^{(k)} = \lambda_i^{(k)} v_i^{(k)}, \quad i = 1, k$$

Then we can derive the approximation of pair (λ_i, z_i)

$$(Q^{(k)} v_i^{(k)})^T \mathbf{B} Q^{(k)} v_i^{(k)} = \lambda_i^{(k)} \text{ and } z_i^{(k)} = Q^{(k)} v_i^{(k)}, \quad i = 1, k$$

Consequently, increasing k enlarges the subspace where we are looking for approximation of the desired eigenvectors with the largest eigenvalues of \mathbf{B} .

In order to solve the eigenproblems without potential ill-conditioning, usually the symmetric matrix \mathbf{B} is rewritten as the extended form, \mathbf{B}_3 , as described in the Lemma above.

3. Experiences with POD

At the first stage of the project, we followed the idea of the investigators in the first category. We derive reduce order model for Burgers equation using POD basis functions and Galerkin projection. The derivation is presented below. It is worth noting that there is only one partial differential equation with respect to a single spatial variable and time. Of course, we can replace time with a spatial dimension to solve 2-D problems in our project. The deviation also illustrates that the methodology is complicated and error-prone.

3.1 Algorithm for obtaining POD modes

Since we have been considering steady fluid flows, the temporal variable will never appear and only spatial coordinates will be in concern. Therefore the model of 2-D in steady case is equivalent to the 1-D in spatiotemporal model. We will use y-coordinate to represent the flow direction which is similar to time in the dynamic case.

We first guess the solution of the model for every state variable, such as temperature, pressure, velocity components along two directions, etc. This can be expressed in form of

$$\zeta(x, y) = \sum_{j=1}^l \alpha_j(y) \psi_j(x) \quad (3-1)$$

where

$$\begin{aligned} \psi_j(x) &= \sum_{i=1}^n \varphi_i(x) \psi_{i,j} \\ &= [\varphi_1(x) \quad \cdots \quad \varphi_n(x)] \begin{bmatrix} \psi_{1,j} \\ \vdots \\ \psi_{n,j} \end{bmatrix} \\ &= \vec{\varphi}(x)^T \vec{\psi}_j \end{aligned}$$

Note that $\psi_j(x = x_0) = \psi_j(x = x_{n+1}) = 0$ for $j = 1, \dots, n$ which satisfies $H_0^1(x_0, x_{n+1})$. Let matrix $A \in \mathfrak{R}^{n \times m}$ of rank k be the snapshot based on designated lattice corresponding one input of the equipment in the flowsheet. The columns $\vec{A}_j = \{A(:, y_j)\}_{j=1}^m$ of A is the x-coordinate vector at $y = y_j$; the rows of $\{A(x_i, :)\}_{i=1}^n$ of A is the y-coordinate vector evaluated at $x = x_i$.

We assume that $A \in \mathfrak{R}^{n \times m}$ is the discretization of a state variable in the 2D system using a piece-wise linear finite element basis function as shown in Figure 2-1.

$$\zeta(x, y_j) = \sum_{i=1}^n A_{i,j} \varphi_i(x) \quad (3-2)$$

Where $\varphi_i, i=1, n$ denoting the finite-element basis functions. Note the range of $i = [1, n]$.

To find the POD modes of the ensemble we formulate an optimal problem as

$$\max \sum_{j=1}^m |(\zeta(x, y_j), \psi)_L|^2 \quad [\text{P-2}]$$

s.t.

$$\|\psi\|_{L^2} = 1$$

$$\begin{aligned} |(\zeta(x, y_j), \psi_l)|^2 &= \left(\int_{\Omega} \zeta(x, y_j) \cdot \psi dx \right)^2 = \left(\int_{\Omega} \sum_{i=1}^n [A_{i,j} \varphi_i(x)] \left[\sum_{k=1}^n \varphi_k(x) \psi_{k,l} \right] dx \right)^2 \\ &= \left(\sum_{i=1}^n \sum_{k=1}^n A_{i,j} \psi_{k,l} \int \varphi_i(x) \varphi_k(x) dx \right)^2 = \left(\sum_{i=1}^n \sum_{k=1}^n A_{i,j} \psi_{k,l} S_{i,k} \right)^2 \\ &= (\bar{A}_j^T S \bar{\psi}_l)^2 = (\bar{\psi}_l^T S^T \bar{A}_j \bar{A}_j^T S \bar{\psi}_l) \end{aligned}$$

where matrix $S \in \mathfrak{R}^{n \times n}$ is the so-called *mass matrix*, S is symmetric and its entry is

$$S_{i,j} = (\varphi_i(x), \varphi_j(x))_{L^2} = \int_{\Omega} \varphi_i(x) \varphi_j(x) dx$$

Furthermore, we can expressed the objective and normalized vector in original optimization problem as

$$\begin{aligned} \sum_{j=1}^m |(\zeta(x, y_j), \psi_l)|^2 &= \sum_{j=1}^m \bar{\psi}_l^T S^T \bar{A}_j \bar{A}_j^T S \bar{\psi}_l = \bar{\psi}_l^T S A A^T S \bar{\psi}_l \\ \|\psi_l\|^2 &= \int \bar{\psi}_l^T \bar{\varphi}(x) \bar{\varphi}(x)^T \bar{\psi}_l dx = \bar{\psi}_l^T S \bar{\psi}_l \end{aligned}$$

Let us solve an equivalent maximization problem

$$\max \bar{\psi}_l^T S A A^T S \bar{\psi}_l \quad [\text{P-3}]$$

s.t.

$$\bar{\psi}_l^T S \bar{\psi}_l = 1$$

To solve this problem let us first formulate the Lagrange function

$$L = \bar{\psi}_l^T S A A^T S \bar{\psi}_l + \lambda_l (1 - \bar{\psi}_l^T S \bar{\psi}_l)$$

By $\frac{\partial L}{\partial \bar{\psi}_l} = 0$, we get the KKT system as

$$A A^T S \bar{\psi}_l = \lambda_l \bar{\psi}_l$$

$$\bar{\psi}_l^T S \bar{\psi}_l = 1$$

Let $\tilde{A} = S^{1/2} A$ and $\tilde{u}_l = S^{1/2} \bar{\psi}_l$, then $A = S^{-1/2} \tilde{A}$ and the KKT becomes

$$S^{-1/2} \tilde{A} \tilde{A}^T S^{-1/2} S \bar{\psi}_l = \lambda \bar{\psi}_l$$

$$\tilde{A} \tilde{A}^T \bar{u}_l = \lambda_l \bar{u}_l$$

$$\bar{u}_l^T \bar{u}_l = 1$$

With Singular Value Decomposition (SVD) the matrix can be decomposed as

$$\tilde{A} = U \Sigma V^T \quad \text{with } \tilde{A} \in \mathfrak{R}^{n \times m}, U \in \mathfrak{R}^{n \times n} = [u_1 \ u_2 \ \cdots \ u_n],$$

$$\Sigma \in \mathfrak{R}^{n \times m} = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}, \quad D = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k), \quad \text{and } V \in \mathfrak{R}^{m \times m} \text{ with orthogonality of}$$

$U^T U = I$ and $V^T V = I$ then we obtain

$$\begin{aligned} \tilde{A} \tilde{A}^T \bar{u}_l &= U \Sigma V^T V \Sigma^T U^T \bar{u}_l = U \Sigma \Sigma^T U^T \bar{u}_l \\ &= U^K (D^K)^2 (U^K)^T \bar{u}_l = \left(\sum_{i=1}^K \sigma_i^2 u_i u_i^T \right) \bar{u}_l = \lambda_l \bar{u}_l \end{aligned}$$

Finally the maximization problem becomes leads us to solve two equations for two unknowns: \bar{u}_l and λ_l

$$\begin{aligned} \left(\sum_{i=1}^K \sigma_i^2 u_i u_i^T \right) \cdot \bar{u}_l &= \lambda_l \bar{u}_l \\ \bar{u}_l^T \bar{u}_l &= 1 \end{aligned}$$

The solution is that \bar{u}_l must be one of the columns of U that are orthogonal. i.e.

$$\begin{aligned} \bar{u}_l &= u_l \\ \lambda_l &= \sigma_l^2 \end{aligned}$$

for any $1 \leq l \leq K$ with descending order of $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_K$.

$$\bar{\psi}_l = S^{-1/2} u_l, \quad \psi_l(x) = \bar{\varphi}(x) S^{-1/2} u_l$$

The ratio of modeled energy to the total energy contained in the system \mathbf{A} is calculated as

$$\varepsilon(L) = \frac{\sum_{i=1}^L \lambda_i}{\sum_{i=1}^m \lambda_i} = \frac{\sum_{i=1}^L \sigma_i^2}{\sum_{i=1}^m \sigma_i^2} \quad (3-3)$$

Thus, we proved how to get POD basis function with SVD to get the POD modes using snapshot. The one part of the POD mode is the columns of U. It is important to notice that U depends on snapshots only, and the snapshot is response of a specific input (initial and boundary condition). Therefore U is a response of the input.

3.2 Implementation of POD on Burger's Equation

The original Burger's equation is for dynamic system in one dimension.

$$\frac{dy}{dt} - \mu \frac{d^2 y}{dx^2} + y \frac{dy}{dx} = f \quad (3-4)$$

Replacing temporal variable with the second spatial variable, y-coordinate, we can write it as 2D model

$$\frac{d\zeta(x, y)}{dy} - \mu \frac{d^2 \zeta(x, y)}{dx^2} + \zeta(x, y) \frac{d\zeta(x, y)}{dx} = f \quad (3-5)$$

Boundary condition: $\zeta(0, y) = \zeta(1, y) = 0$; Initial condition: $\zeta(x, 0) = \Phi(x)$.

The solution can be expressed as:

$$\zeta_L(x, y) = \sum_{j=1}^L \alpha_j(y) \psi_j(x)$$

$$\psi_j(x) = \sum_{i=1}^n \varphi_i(x) \psi_{i,j}$$

$$\zeta_L(x, y) = \sum_{j=1}^L \alpha_j(y) \sum_{i=1}^n \varphi_i(x) \psi_{i,j} = \bar{\alpha}^T(y) \Psi \bar{\varphi}(x)$$

$$\text{with } \sum_{i=1}^n \varphi_i(0) \psi_{i,j} = \sum_{i=1}^n \varphi_i(1) \psi_{i,j} = 0$$

The solution satisfies $H_0^1(0,1)$ with the boundary conditions

Let $v(x) = \sum_{j=1}^L v_j \left(\sum_{i=1}^n \varphi_i(x) \psi_{i,j} \right) = \sum_{j=1}^L v_j \psi_j(x)$ be the test function for weak solution of the equation. Note that $v(x)$ satisfies $H_0^1(0,1)$ i.e. $v(0) = v(1) = 0$, in the case when $v_j = 1$, for $j=1, L$; this is known as Galerkin projection.

$$\underbrace{\int_{\Omega} v(x) \frac{d\zeta(x, y)}{dy} dx}_{[1]} - \underbrace{\mu \int_{\Omega} v(x) \frac{d^2 \zeta(x, y)}{dx^2} dx}_{[2]} + \underbrace{\int_{\Omega} v(x) \zeta(x, y) \frac{d\zeta(x, y)}{dx} dx}_{[3]} = \underbrace{\int_{\Omega} v(x) f dx}_{[4]} \quad (3-6)$$

The term[1] in equation (3-6) can expanded as

$$\begin{aligned} \int_{\Omega} v(x) \frac{d\zeta(x, y)}{dy} dx &= \int_{\Omega} \sum_{k=1}^L v_k \psi_k(x) \sum_{m=1}^L \frac{d\alpha_m(y)}{dy} \psi_m(x) dx = \sum_{k=1}^L v_k \sum_{m=1}^L \left(\int \psi_k(x) \psi_m(x) dx \right) \frac{d\alpha_m(y)}{dy} \\ &= \bar{v}^T \mathbf{M} \frac{d\bar{\alpha}(y)}{dy} = \bar{v}^T \frac{d\bar{\alpha}(y)}{dy} \end{aligned}$$

where

$$\begin{aligned} \bar{v}^T &= [v_1 \quad v_2 \quad \cdots \quad v_L] \\ \bar{\alpha}^T(y) &= [\alpha_1(y) \quad \alpha_2(y) \quad \cdots \quad \alpha_L(y)] \\ \mathbf{M} &= \begin{bmatrix} (\psi_1(x), \psi_1(x))_{L^2} & (\psi_1(x), \psi_2(x))_{L^2} & \cdots & (\psi_1(x), \psi_L(x))_{L^2} \\ (\psi_2(x), \psi_1(x))_{L^2} & (\psi_2(x), \psi_2(x))_{L^2} & \cdots & (\psi_2(x), \psi_L(x))_{L^2} \\ \vdots & \vdots & \ddots & \vdots \\ (\psi_L(x), \psi_1(x))_{L^2} & (\psi_L(x), \psi_2(x))_{L^2} & \cdots & (\psi_L(x), \psi_L(x))_{L^2} \end{bmatrix} \end{aligned}$$

Because of orthogonality of $\psi(x)$ \mathbf{M} is an identity matrix with rank L .

The term [2] in Eq. (3-6) becomes

$$\begin{aligned} \mu \int_{\Omega} v(x) \frac{d^2 \zeta(x, y)}{dx^2} dx &= \mu \int_{\Omega} \sum_{k=1}^L v_k \psi_k(x) \sum_{m=1}^L \alpha_m \frac{d^2 \psi_m(x)}{dx^2} dx \\ &= \mu \left(\sum_{k=1}^L v_k \sum_{m=1}^L \left(\int_{\Omega} \psi_k(x) \frac{d^2 \psi_m(x)}{dx^2} dx \right) \alpha_m(y) \right) \\ &= -\mu \sum_{k=1}^L v_k \sum_{m=1}^L \int_{\Omega} \frac{d\psi_k(x)}{dx} \frac{d\psi_m(x)}{dx} dx \alpha_m(y) \\ &= -\mu \bar{v}^T \mathbf{K} \bar{\alpha}(y) \end{aligned}$$

$$\mathbf{K} = \begin{bmatrix} \left(\frac{d\psi_1(x)}{dx}, \frac{d\psi_1(x)}{dx} \right)_{L^2} & \left(\frac{d\psi_1(x)}{dx}, \frac{d\psi_2(x)}{dx} \right)_{L^2} & \cdots & \left(\frac{d\psi_1(x)}{dx}, \frac{d\psi_L(x)}{dx} \right)_{L^2} \\ \left(\frac{d\psi_2(x)}{dx}, \frac{d\psi_1(x)}{dx} \right)_{L^2} & \left(\frac{d\psi_2(x)}{dx}, \frac{d\psi_2(x)}{dx} \right)_{L^2} & \cdots & \left(\frac{d\psi_2(x)}{dx}, \frac{d\psi_L(x)}{dx} \right)_{L^2} \\ \vdots & \vdots & \ddots & \vdots \\ \left(\frac{d\psi_L(x)}{dx}, \frac{d\psi_1(x)}{dx} \right)_{L^2} & \left(\frac{d\psi_L(x)}{dx}, \frac{d\psi_2(x)}{dx} \right)_{L^2} & \cdots & \left(\frac{d\psi_L(x)}{dx}, \frac{d\psi_L(x)}{dx} \right)_{L^2} \end{bmatrix}$$

with property of $K_{i,j} = (\psi_i(x), \psi_j(x))_{H_0^1}$

The term [3] in Eq. (3-6) leads to

$$\begin{aligned}
\int_{\Omega} v(x) \zeta(x, y) \frac{d\zeta(x, y)}{dx} dx &= \int_{\Omega} \sum_{k=1}^L v_k \psi_k(x) \sum_{m=1}^L \alpha_m(y) \psi_m(x) \sum_{j=1}^L \alpha_j(y) \frac{d\psi_j(x)}{dx} dx \\
&= \sum_{k=1}^L v_k \int_{\Omega} \psi_k(x) \sum_{m=1}^L \sum_{j=1}^L \psi_m(x) \frac{d\psi_j(x)}{dx} \alpha_m(y) \alpha_j(y) dx \\
&= \sum_{k=1}^L v_k \int_{\Omega} \psi_k(x) \vec{\alpha}^T(y) \mathbf{G} \vec{\alpha}(y) dx
\end{aligned}$$

where $\mathbf{G}(x) = \hat{\psi}(x)(\hat{\psi}'(x))^T$

$$\begin{aligned}
\hat{\psi}^T(x) &= [\psi_1(x) \quad \psi_2(x) \quad \cdots \quad \psi_L(x)], \\
(\hat{\psi}'(x))^T &= \left[\frac{d\psi_1(x)}{dx} \quad \frac{d\psi_2(x)}{dx} \quad \cdots \quad \frac{d\psi_L(x)}{dx} \right]
\end{aligned}$$

Let

$$\mathbf{T}_k = \int_{\Omega} \psi_k(x) \mathbf{G}(x) dx$$

Then

$$\sum_{k=1}^L v_k \int_{\Omega} \psi_k(x) \vec{\alpha}^T(y) \mathbf{G} \vec{\alpha}(y) dx = \sum_{k=1}^L v_k \vec{\alpha}^T(y) \mathbf{T}_k \vec{\alpha}(y) = \vec{v}^T \vec{\tau}(y)$$

where

$$\vec{\tau}(y) = [\vec{\alpha}^T(y) \mathbf{T}_1 \vec{\alpha}(y) \quad \vec{\alpha}^T(y) \mathbf{T}_2 \vec{\alpha}(y) \quad \cdots \quad \vec{\alpha}^T(y) \mathbf{T}_L \vec{\alpha}(y)]$$

Finally, the term [4] in Eq. (3-6) is

$$\begin{aligned}
\int_{\Omega} v(x) f(x, y) dx &= \int_{\Omega} f(x, y) \sum_{k=1}^L v_k \psi_k(x) dx = \sum_{k=1}^L v_k \int_{\Omega} f(x, y) \psi_k(x) dx \\
&= \vec{v}^T \vec{f}(y)
\end{aligned}$$

where

$$\vec{f}^T(y) = [(f(x, y), \psi_1(x))_{L^2} \quad (f(x, y), \psi_2(x))_{L^2} \quad \cdots \quad (f(x, y), \psi_L(x))_{L^2}]$$

Assembling all terms, we obtain $\vec{v}^T \left[\frac{d\vec{\alpha}(y)}{dy} + \mu \mathbf{K} \vec{\alpha}(y) + \vec{\tau}(y) \right] = \vec{v}^T \vec{f}(y)$ and finally

ODE of $\vec{\alpha}(y)$ with respect to y .

$$\frac{d\vec{\alpha}(y)}{dy} + \mu \mathbf{K} \vec{\alpha}(y) + \vec{\tau}(y) = \vec{f}(y) \quad (3-7)$$

Next we will derive initial condition for the ODE according to the PDE, $\zeta(x,0) = \Phi(x)$

$$\zeta_L(x,0) = \sum_{m=1}^L \alpha_m(0) \psi_m(x) = \Phi(x) \quad (3-8)$$

$$\int \sum_{m=1}^L \alpha_m(0) \psi_m(x) \psi_j(x) dx = \int_{\Omega} \Phi(x) \psi_j(x), \quad j = 1, L \quad (3-9)$$

Considering the orthogonality of $\psi(x)$, we get $\vec{\alpha}(0) = \vec{\Phi}$, where $\vec{\alpha}^T(0) = [\alpha_1(0) \quad \alpha_2(0) \quad \cdots \quad \alpha_L(0)]$

$$\vec{\Phi} = \begin{bmatrix} (\Phi(x), \psi_1(x))_{L^2} \\ (\Phi(x), \psi_2(x))_{L^2} \\ \vdots \\ (\Phi(x), \psi_L(x))_{L^2} \end{bmatrix} \quad (3-10)$$

Thus, the original PDE is converted into a set of ODEs depending on y-coordinate only and with a new set of initial conditions. The number of ordinary differential equations in the derived ODE is set equal to the rank of POD, which is L in this case.

3.3 Implementation of POD on Navier-Stokes Equations

Let us consider to convert the Navier Stokes equation and continuity equation in 2D to a set of ODEs with three dependent variables pressure (p), x-component (u) and y-component (v) of velocity.

$$\begin{aligned} 0 &= -\frac{\partial p}{\partial x} + \rho g_x + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\ 0 &= -\frac{\partial p}{\partial y} + \rho g_y + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \\ 0 &= \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \end{aligned} \quad (3-11)$$

We renamw the variables as $\vec{\zeta} = [\zeta_1 \quad \zeta_2 \quad \cdots \quad \zeta_{nz}]$ where $\zeta_1 = u(x, y)$, $\zeta_2 = v(x, y)$ and $\zeta_3 = p(x, y)$ with $nz = 3$, and rewrite the PDE as

$$\begin{aligned}
\mu\left(\frac{\partial^2 \zeta_1}{\partial x^2} + \frac{\partial^2 \zeta_1}{\partial y^2}\right) - \frac{\partial \zeta_3}{\partial x} &= 0 \\
\mu\left(\frac{\partial^2 \zeta_2}{\partial x^2} + \frac{\partial^2 \zeta_2}{\partial y^2}\right) - \frac{\partial \zeta_3}{\partial y} &= -\rho g \\
\frac{\partial \zeta_1}{\partial x} + \frac{\partial \zeta_2}{\partial y} &= 0
\end{aligned} \tag{3-12}$$

We introduce the tensor $\Psi = ((\Psi_{i,j,k})) \in \mathfrak{R}^{n \times L \times nZ}$ with $\Psi_{i,j,k} = (\psi_{i,j,k})$ and define sets $I = \{1, \dots, n\}$, $J = \{1, \dots, L\}$ and $NZ = \{1, \dots, nZ\}$. Then in POD context, the variables can be decomposed as

$$\zeta_k(x, y) = \sum_{j=1}^l \alpha_{j,k}(y) \left(\sum_{i=1}^n \psi_{i,j,k} \varphi_i(x) \right) \quad \text{for } i \in I, j \in J, k \in NZ \tag{3-13}$$

Formulating the weak form of the equations with the test function leads to:

$$v(x) = \sum_{m=1}^L v_m \sum_{i=1}^n \mathcal{G}_{i,m} \varphi_i(x) = \vec{v}^T \Theta^T \vec{\varphi}(x) \tag{3-14}$$

$$\begin{aligned}
\mu \left\{ \underbrace{\int_{\Omega} v(x) \frac{\partial^2 \zeta_1}{\partial x^2} dx}_{[1-1]} + \underbrace{\int_{\Omega} v(x) \frac{\partial^2 \zeta_1}{\partial y^2} dx}_{[1-2]} \right\} - \underbrace{\int_{\Omega} v(x) \frac{\partial \zeta_3}{\partial x} dx}_{[1-3]} &= 0 \\
\mu \left\{ \underbrace{\int_{\Omega} v(x) \frac{\partial^2 \zeta_2}{\partial x^2} dx}_{[2-1]} + \underbrace{\int_{\Omega} v(x) \frac{\partial^2 \zeta_2}{\partial y^2} dx}_{[2-2]} \right\} - \underbrace{\int_{\Omega} v(x) \frac{\partial \zeta_3}{\partial y} dx}_{[2-3]} &= -\rho g \underbrace{\int_{\Omega} v(x) dx}_{[2-4]} \\
\underbrace{\int_{\Omega} v(x) \frac{\partial \zeta_1}{\partial x} dx}_{[3-1]} + \underbrace{\int_{\Omega} v(x) \frac{\partial \zeta_2}{\partial y} dx}_{[3-2]} &= 0
\end{aligned} \tag{3-15}$$

And with this, we can now derive each term in the three equations in Eqs. (3-15).

Before we derive further, we note that with the convention of the finite element basis functions, we can decompose the integral range $[x_0, x_{n+1}]$ into $n+1$ elements. Each element has two nodes indexed as $k1$ and $k2$. The “generic” integrals that define the Galerkin approximation can be written as

$$\int_{\Omega} (\dots) dx = \sum_{k=1}^{n+1} \int_{x_{k1}}^{x_{k2}} (\dots) dx = \sum_{k=1}^{n+1} K_{k1,k2}^k$$

There are four possible forms of integrals define the Galerkin approximation

$$K(1)_{k1,k2} = \int_{\Omega} \frac{d\varphi_{k1}(x)}{dx} \frac{d\varphi_{k2}(x)}{dx} dx = \sum_{k=1}^{n+1} \int_{x_{k1}}^{x_{k2}} \frac{d\varphi_{k1}(x)}{dx} \frac{d\varphi_{k1}(x)}{dx} dx = \sum_{k=1}^{n+1} K(1)_{k1,k2}^k$$

$$K(2)_{k1,k2} = \int_{\Omega} \frac{d\varphi_{k1}(x)}{dx} \varphi_{k2}(x) dx = \sum_{k=1}^{n+1} \int_{x_{k1}}^{x_{k2}} \frac{d\varphi_{k1}(x)}{dx} \varphi_{k2}(x) dx = \sum_{k=1}^{n+1} K(2)_{k1,k2}^k$$

$$K(3)_{k1,k2} = \int_{\Omega} \varphi_{k1}(x) \varphi_{k2}(x) dx = \sum_{k=1}^{n+1} \int_{x_{k1}}^{x_{k2}} \varphi_{k1}(x) \varphi_{k2}(x) dx = \sum_{k=1}^{n+1} K(3)_{k1,k2}^k$$

$$Q_{k1} = \int_{\Omega} \varphi_{k1}(x) dx = \sum_{k=1}^{n+1} \int_{x_{k1}}^{x_{k2}} \varphi_{k1}(x) dx = \sum_{k=1}^{n+1} Q_{k1}^k$$

where $K(J)_{k1,k2}^k$ and Q_{k1} are defined as the contributions from element k to $K(J)_{k1,k2}$ and Q_{k1} , respectively.

These terminologies lead to following important conditions:

$$K(J)_{k1,k2}^k = 0 \quad J=1,3 \text{ unless both nodes } k1 \text{ and } k2 \text{ are associated with element } k.$$

$$Q_{k1}^k = 0 \quad \text{unless both nodes } k1 \text{ are associated with element } k.$$

Each 1D linear element contributes six nonzero terms to the system, which can be identified as contributions from the nodes associated with the particular element (referring to finite element basis functions as shown in Figure 2-1). We can condense these six nonzero terms into two element matrices:

$$K(J)^k = \begin{bmatrix} K(J)_{k1,k1}^k & K(J)_{k1,k2}^k \\ K(J)_{k2,k1}^k & K(J)_{k2,k2}^k \end{bmatrix} \quad \text{and} \quad Q^k = \begin{bmatrix} Q_{k1}^k \\ Q_{k2}^k \end{bmatrix}$$

Assembling all matrices from each element we can construct a $(n+1) \times (n+1)$ matrix

$$K(J) = \begin{bmatrix} K(J)_{1,1}^1 & K(J)_{1,2}^1 & 0 & \cdots & \cdots & 0 \\ K(J)_{2,1}^1 & K(J)_{2,2}^1 + K(J)_{2,2}^2 & K(J)_{2,3}^1 & \ddots & \ddots & \vdots \\ 0 & K(J)_{3,2}^1 & K(J)_{3,3}^2 + K(J)_{3,3}^3 & K(J)_{3,4}^1 & \ddots & \vdots \\ \vdots & \ddots & K(J)_{4,3}^1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & K(J)_{kn,kn}^n + K(J)_{kn,kn}^{n+1} & K(J)_{kn,kn+1}^{n+1} \\ 0 & \cdots & \cdots & \cdots & K(J)_{kn,kn+1}^{n+1} & K(J)_{kn+1,kn+1}^{n+1} \end{bmatrix}$$

There are four possible forms of integrals that define the Galerkin approximation

$$K(1)^k \Rightarrow \begin{cases} \int_{x_{k1}}^{x_{k2}} \frac{d\varphi_{k1}(x)}{dx} \frac{d\varphi_{k1}(x)}{dx} dx = \int_{x_{k1}}^{x_{k2}} \frac{d\varphi_{k2}(x)}{dx} \frac{d\varphi_{k2}(x)}{dx} dx = \frac{1}{h_x} \\ \int_{x_{k1}}^{x_{k2}} \frac{d\varphi_{k1}(x)}{dx} \frac{d\varphi_{k2}(x)}{dx} dx = -\frac{1}{h_x} \end{cases}$$

$$K(2)^k \Rightarrow \begin{cases} \int_{x_{k1}}^{x_{k2}} \frac{d\varphi_{k1}(x)}{dx} \varphi_{k1}(x) dx = \int_{x_{k1}}^{x_{k2}} \frac{d\varphi_{k2}(x)}{dx} \varphi_{k2}(x) dx = 0.5 \\ \int_{x_{k1}}^{x_{k2}} \frac{d\varphi_{k1}(x)}{dx} \varphi_{k2}(x) dx = \int_{x_{k1}}^{x_{k2}} \frac{d\varphi_{k2}(x)}{dx} \varphi_{k1}(x) dx = -0.5 \end{cases}$$

$$K(3)^k \Rightarrow \begin{cases} \int_{x_{k1}}^{x_{k2}} (\varphi_{k1}(x))^2 dx = \int_{x_{k1}}^{x_{k2}} (\varphi_{k2}(x))^2 dx = \frac{h_x}{3} \\ \int_{x_{k1}}^{x_{k2}} \varphi_{k1}(x) \varphi_{k2}(x) dx = \frac{h_x}{6} \end{cases}$$

$$Q^k \Rightarrow \int_{x_{k1}}^{x_{k2}} \varphi_{k1}(x) dx = \int_{x_{k1}}^{x_{k2}} \varphi_{k2}(x) dx = 0.5h_x$$

where $h_x = \frac{x_{n+1} - x_0}{n+1}$, is the length of the element.

Now we can derive terms one-by-one in (3-15). First the terms [1-1] and [2-1] in (3-15) are similar and term [1-1] can be derived further to get

$$\begin{aligned} \int_{\Omega} v(x) \frac{\partial^2 \zeta_1}{\partial x^2} dx &= - \int_{\Omega} \frac{dv(x)}{dx} \frac{\partial \zeta_1}{\partial x} dx = \int_{\Omega} \sum_{m=1}^L v_m \sum_{k=1}^n \mathcal{G}_{k,m} \frac{d\varphi_k(x)}{dx} \sum_{j=1}^l \alpha_{j,1}(y) \left(\sum_{i=1}^n \psi_{i,j,1} \frac{d\varphi_i(x)}{dx} \right) dx \\ &= \sum_{m=1}^L v_m \sum_{i=1}^n \left(\int_{\Omega} \frac{d\varphi_m(x)}{dx} \frac{d\varphi_i(x)}{dx} dx \right) \sum_{j=1}^l \psi_{i,j,1} \alpha_{j,1}(y) = \sum_{m=1}^n v_m \sum_{i=1}^n \left(\int_{\Omega} \frac{d\varphi_m(x)}{dx} \frac{d\varphi_i(x)}{dx} dx \right) \bar{\psi}_{i,1}^T \bar{\alpha}_1(y) \\ &= \sum_{m=1}^n v_m \bar{\varphi}_m^T \Psi_1 \bar{\alpha}_1(y) = v^T K(1) \Psi_1 \bar{\alpha}_1(y) \end{aligned}$$

Or

$$\begin{aligned} \int_{\Omega} v(x) \frac{\partial^2 \zeta_1}{\partial x^2} dx &= - \int \bar{v}^T \Theta_1^T \frac{d}{dx} \bar{\varphi}(x) \frac{\partial}{\partial x} (\bar{\varphi}^T(x) \Psi_1 \bar{\alpha}_1(y)) dx = - \bar{v}^T \Theta_1^T \int \frac{d\bar{\varphi}(x)}{dx} \frac{d\bar{\varphi}^T(x)}{dx} dx \Psi_1 \bar{\alpha}_1(y) \\ &= - \bar{v}^T \Theta_1^T K(1) \Psi_1 \bar{\alpha}_1(y) \end{aligned}$$

where

matrix $\Psi_1 \in \mathfrak{R}^{n \times l}$ with $\Psi_{i,j,1} = \psi_{i,j,1}$

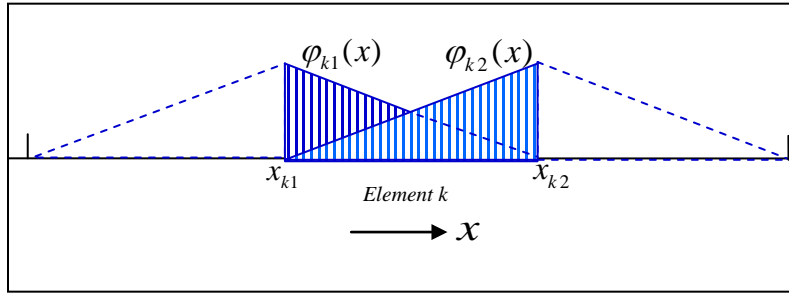
vector $\vec{v}^T = [v_1 \ v_2 \ \dots \ v_n] \in \mathfrak{R}^n$

vector $\vec{\alpha}_1^T(y) = [\alpha_1(y) \ \alpha_2(y) \ \dots \ \alpha_L(y)] \in \mathfrak{R}^L$

vector $\vec{\varphi}_m^T = \left[\int_{\Omega} \frac{d\varphi_m}{dx} \frac{d\bar{\varphi}_1}{dx} dx \quad \int_{\Omega} \frac{d\varphi_m}{dx} \frac{d\bar{\varphi}_2}{dx} dx \quad \dots \quad \int_{\Omega} \frac{d\varphi_m}{dx} \frac{d\bar{\varphi}_n}{dx} dx \right] \in \mathfrak{R}^n$

$$\text{matrix } K(1) = \frac{1}{h_x} \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & \ddots & \vdots & \vdots \\ 0 & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & \dots & -1 & 2 \end{bmatrix} \in \mathfrak{R}^{n \times n}$$

The figure below illustrates the meaning of the terms of finite element basis functions involved in element k .



Scheme of finite element basis function in element k

The terms [1-2] and [2-2] in (3-15) are similar and term [1-2] leads to

$$\begin{aligned} \int_{\Omega} v(x) \frac{\partial^2 \zeta_1}{\partial y^2} dx &= \int \vec{v}^T \Theta^T \vec{\varphi}(x) \frac{\partial^2}{\partial y^2} (\vec{\varphi}^T(x) \Psi_1 \vec{\alpha}_1(y)) dx = \vec{v}^T \Theta^T \int \vec{\varphi}(x) \vec{\varphi}^T(x) dx \Psi_1 \frac{d^2}{dy^2} \vec{\alpha}_1(y) \\ &= \vec{v}^T \Theta^T K(3) \Psi_1 \frac{d^2}{dy^2} \vec{\alpha}_1(y) \end{aligned}$$

where

$$K(3) = \frac{h_x}{6} \begin{bmatrix} 4 & 1 & 0 & \dots & 0 \\ 1 & 4 & \ddots & \vdots & \vdots \\ 0 & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 4 & 1 \\ 0 & 0 & \dots & 1 & 4 \end{bmatrix} \in \mathfrak{R}^{n \times n}$$

The term [1-3] in (3-15) leads to

$$\begin{aligned} \int_{\Omega} v(x) \frac{\partial \zeta_3}{\partial x} dx &= \int \bar{v}^T \Theta^T \bar{\varphi}(x) \frac{\partial}{\partial x} (\bar{\varphi}^T(x) \Psi_3 \bar{\alpha}_3(y)) dx = \bar{v}^T \Theta^T \int \bar{\varphi}(x) \frac{\partial}{\partial x} \bar{\varphi}^T(x) dx \Psi_3 \bar{\alpha}_3(y) \\ &= \bar{v}^T \Theta^T K(2) \Psi_3 \bar{\alpha}_3(y) \end{aligned}$$

where

$$K(2) = \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ -1 & 0 & \ddots & \vdots & \vdots \\ 0 & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & -1 & 0 \end{bmatrix} \in \mathfrak{R}^{n \times n}$$

The term [2-3] in (3-15) leads to

$$\begin{aligned} \int_{\Omega} v(x) \frac{\partial \zeta_3}{\partial y} dx &= \int \bar{v}^T \Theta^T \bar{\varphi}(x) \frac{\partial}{\partial y} (\bar{\varphi}^T(x) \Psi_3 \bar{\alpha}_3(y)) dx = \bar{v}^T \int \bar{\varphi}(x) \bar{\varphi}^T(x) dx \Theta^T \Psi_3 \frac{d}{dy} \bar{\alpha}_3(y) \\ &= \bar{v}^T K(3) \Theta^T \Psi_3 \frac{d}{dy} \bar{\alpha}_3(y) \end{aligned}$$

The term [2-4] in (3-15) becomes

$$\int_{\Omega} v(x) dx = \int v^T \Theta^T \bar{\varphi}(x) dx = v^T \Theta^T Q = v^T \Theta^T 0.5 h_x \bar{e}$$

The term [3-1] in (3-15) is derived as

$$\begin{aligned} \int_{\Omega} v(x) \frac{\partial \zeta_1}{\partial x} dx &= \int \bar{v}^T \Theta_3^T \bar{\varphi}(x) \frac{\partial}{\partial(x)} (\bar{\varphi}^T(x) \Psi_1 \bar{\alpha}_1(y)) dx = \bar{v}^T \Theta_3^T \int \bar{\varphi}(x) \left(\frac{d\bar{\varphi}(x)}{dx} \right)^T dx \Psi_1 \bar{\alpha}_1(y) \\ &= \bar{v}^T \Theta_3^T K(2) \Psi_1 \bar{\alpha}_1(y) \end{aligned}$$

The term [3-2] in (3-15) is derived as

$$\begin{aligned} \int_{\Omega} v(x) \frac{\partial \zeta_2}{\partial y} dx &= \int \bar{v}^T \Theta_3^T \bar{\varphi}(x) \frac{\partial}{\partial y} (\bar{\varphi}^T(x) \Psi_2 \bar{\alpha}_2(y)) dx = \bar{v}^T \int \bar{\varphi}(x) \bar{\varphi}^T(x) dx \Theta_3^T \Psi_2 \frac{d}{dy} \bar{\alpha}_2(y) \\ &= \bar{v}^T K(3) \Theta_3^T \Psi_2 \frac{d}{dy} \bar{\alpha}_2(y) \end{aligned}$$

where $\bar{e}^T = [1 \ 1 \ \cdots \ 1] \in \mathfrak{R}^{1 \times n}$

Eventually, we can convert the NSE into ODE in form as below,

$$-\mu \bar{w}^T \Theta^T K(1) \Psi_1 \bar{\alpha}_1(y) + \mu \bar{w}^T \Theta^T K(3) \Psi_1 \frac{d^2}{dy^2} \bar{\alpha}_1(y) + \bar{v}^T \Theta^T K(2) \Psi_3 \bar{\alpha}_3(y) = 0 \quad (3-16)$$

Let each equation have its own matrix Θ^T in the test function matrix. Then we have the final converted ODE's:

$$\begin{aligned} \mu \Theta_1^T K(3) \Psi_1 \frac{d^2}{dy^2} \bar{\alpha}_1(y) - \mu \Theta_1^T K(1) \Psi_1 \bar{\alpha}_1(y) - \Theta_1^T K(2) \Psi_3 \bar{\alpha}_3(y) &= 0 \\ \mu \Theta_2^T K(3) \Psi_2 \frac{d^2}{dy^2} \bar{\alpha}_2(y) - \mu \Theta_2^T K(1) \Psi_2 \bar{\alpha}_2(y) - \Theta_2^T K(2) \Psi_3 \frac{d}{dy} \bar{\alpha}_3(y) \\ &= 0.5 \rho g h_x \Theta_2^T \bar{e} \\ \Theta_3^T K(3) \Psi_2 \frac{d \bar{\alpha}_2(y)}{dy} + \Theta_3^T K(2) \Psi_1 \bar{\alpha}_1(y) &= 0 \end{aligned}$$

Let $\Theta_1 = K(3)^{-1} \Psi_1$, $\Theta_2 = K(3)^{-1} \Psi_2$ and $\Theta_3 = K(3)^{-1} \Psi_2$. This leads to:

$$\begin{aligned} \frac{d^2}{dy^2} \bar{\alpha}_1(y) - \Psi_1^T K(3)^{-1} K(1) \Psi_1 \bar{\alpha}_1(y) - \frac{1}{\mu} \Psi_1^T K(3)^{-1} K(2) \Psi_3 \bar{\alpha}_3(y) &= 0 \\ \frac{d^2}{dy^2} \bar{\alpha}_2(y) - \Psi_2^T K(3)^{-1} K(1) \Psi_2 \bar{\alpha}_2(y) - \frac{1}{\mu} \Psi_2^T K(3)^{-1} K(2) \Psi_3 \frac{d}{dy} \bar{\alpha}_3(y) \\ &= \frac{\rho g h_x}{2\mu} \Psi_2^T K(3)^{-1} \bar{e} \\ \frac{d \bar{\alpha}_2(y)}{dy} + \Psi_2^T K(3)^{-1} K(2) \Psi_1 \bar{\alpha}_1(y) &= 0 \end{aligned}$$

$$K(1)_{i,j} = \left(\frac{d\varphi_i}{dx}, \frac{d\varphi_j}{dx} \right)_{L^2} \quad K(2)_{i,j} = \left(\frac{d\varphi_i}{dx}, \varphi_j \right)_{L^2} \quad K(3)_{i,j} = (\varphi_i, \varphi_j)_{L^2}$$

For simplicity, we use a compact nomination to express a matrix defined by a product of four matrices

$$M(ijkl) = \Psi_i^T K(j)^{-1} K(k) \Psi_l, \text{ for example, } M(1323) = \Psi_1^T K(3)^{-1} K(1) \Psi_3, \dots$$

$$\begin{aligned}
\frac{d^2}{dy^2} \bar{\alpha}_1(y) - M(1311) \bar{\alpha}_1(y) - \frac{1}{\mu} M(1323) \bar{\alpha}_3(y) &= 0 \\
\frac{d^2}{dy^2} \bar{\alpha}_2(y) - M(2312) \bar{\alpha}_2(y) - \frac{1}{\mu} M(2323) \frac{d}{dy} \bar{\alpha}_3(y) &= \frac{\rho g h_x}{2\mu} \Psi_2^T K(3)^{-1} \vec{e} \\
\frac{d \bar{\alpha}_2(y)}{dy} + M(2321) \bar{\alpha}_1(y) &= 0
\end{aligned}$$

3.4 Summary

3.4.1 Conceptual comparison of POD and PCA

In fact, both POD and PCA are the different forms of *Karhunen-Loeve Decomposition*. Therefore, there is little conceptual difference between POD and PCA, but there are some differences in the implementations (refer to Figure 3-1). First, POD is generally used for dynamic models, the solutions or variables of dynamic model are spatial-temporal, while PCA is used in time-independent or steady state models; the solutions of the models are input-spatial. This difference determines the different representation of the dataset matrices. The columns in the snapshot for POD are the values of the states in the spatial coordinates measured at corresponding time points, while the columns in the dataset for PCA are state responses to corresponding inputs.

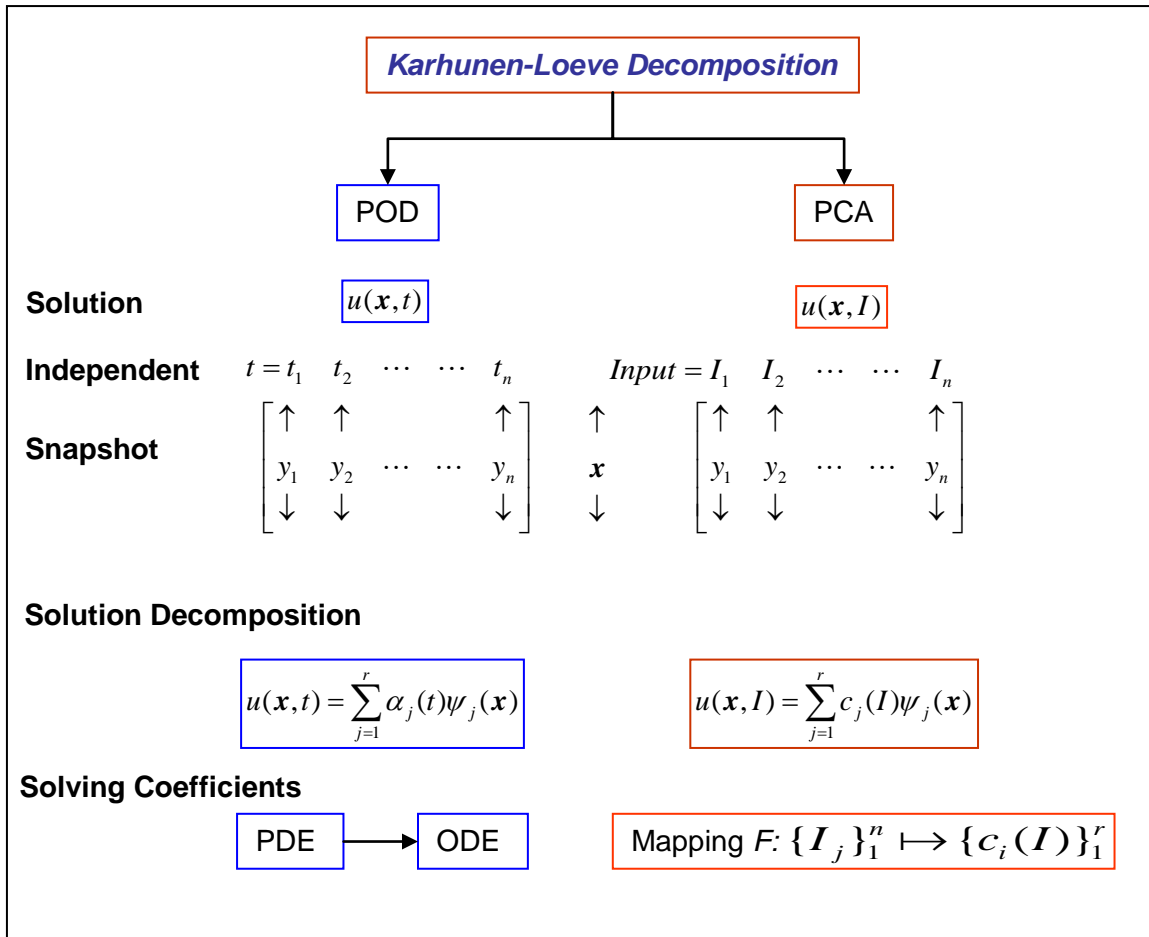


Figure 3-1. Comparison of POD and PCA

After constructing the matrices, the procedures of POD and PCA are the same and the same methods can be used to solve eigenproblems to obtain POD modes or principal components.

Second, the purposes and procedures of the post-decomposition to calculate the coefficients or scores are different. In POD case, to calculate time-dependent coefficients, $\alpha_j(t)$, we have to go back to the original PDAE and use the decomposed form of the solution and Galerkin projection to convert the PDAE model into a set of ODEs. The number of resulting ODEs is $(n_y \times r)$, where n_y is the number of the variables or states in the PDE and r is the rank of the POD modes. The compute cost is $O(r^3)$. In PCA case, the coefficients PCs or coordinates of states in transformed space are called scores that are determined only by the inputs used for constructing the dataset. It is often straightforward to find the mapping from the input into the scores.

Finally, validity of the POD modes is not guaranteed when the input changes, while the PCs are still valid when the input changes as long as the dataset is well-designed. It is

obvious that this difference makes PCA superior to POD for the iterative process simulation and optimization of steady state.

3.4.2 Restrictions of POD

Having experiences with POD, we found that there are many restrictions due to some elementary factors of the PDAE model itself, which are listed as follows:

(1) Geometry

When we tried to replace temporal with one spatial variable, we have to maintain the rectangular geometry. Otherwise, the boundary condition and Dirichlet conditions are very difficult to formulate; consequently, Galerkin projection can not be used in converting PDE to ODE which is crucial for obtain the coefficients of POD modes.

Usually the POD modes are not functions of coordinates and linear finite basis functions are used

$$\psi_j(x) = \psi_j \cdot \varphi(x)$$

where ψ_j are the POD modes and $\varphi(x)$ is the finite element basis function as described in Figure 2-1. To make Galerkin projection valid, Dirichlet boundary condition must be guaranteed, i.e.

$$\psi_j(x = 0) = \psi_j(x = x_{m+1}) = 0$$

as shown in Figure 2-1. If we intend to replace the time with y , as we did in Burgers equation, then the geometry of domain must be regular. Obviously this is impractical for the ROM methodology for process equipment.

(2) Complexity of PDAE model

Most POD investigations worked on fluid dynamics applications, such as driven cavity flow in a 2-D square, or plane Couette flow in a rectangular channel. The Navier-Stokes equations are the main governing equations. Even the POD used in fluidized bed considers four partial differential equations (PDEs) in a 2-D rectangular domain (Cizmas, 2003). Only one or two properties, density and viscosity, were considered in the model and these are kept constant.

We could not expect the PDAE models of the equipment in any chemical process flowsheet to be as simple as those in the POD literature. There might be many additional PDEs and algebraic equations included in the model to take account of energy balances, mass balances of many components, thermo-properties, and chemical reactions as well as flow patterns (laminar or turbulent). The complexity of the PDAE model makes the procedure of Galerkin projection even more impractical. Otherwise, there is no generic method to implement Galerkin automatically. This has

been shown by our experiences with POD on Burgers equation, where there is only one PDE.

(3) Dimension of the model, 2-D or 3-D

Since the POD based ROM heavily depends upon the spatial coordinates, the dimension of original model from 2-D to 3-D significantly affects the complexity of the resulted ROM, which makes the procedure of Galerkin more complicated.

(4) Validation of POD modes

Since the POD modes are derived according to a specific set of boundary conditions, they are valid only when the boundary conditions are close enough to the original one. Otherwise, there is no guarantee of validity of the POD modes. That is why Fahl develop a trust-region POD (TRPOD).

3.4.3 Why we proposed PCA based ROM

Recognizing the restrictions and difficulties of applying POD based ROM to the equipment in process flowsheeting in APECS, we pay attention to principles and applications of PCA in literature. Combining the concepts of *state space models* used widely in control area with the characteristics of steady state, we propose a methodology of PCA based ROM for the APECS project.

(1) Space state model

Usually, state space model is in form

$$\dot{x} = Ax + Bu; \quad y = Cx + Du$$

where x is state variable, u is control, y is output and A , B , C and D are matrices. It is worth noting that there are no coordinates involved in the model and all the members of the model are independent on the spatial coordinates. The locations of the state variable, x_i are determined by the index i , that is determined by the protocol to construct the state space or the mesh in CFD. By introducing the concepts of the state space model and avoiding the spatial coordinates in the model, we can eliminate the restrictions due to complicated geometry and the dimension of the model as discussed in section 3.1.2.

(2) Taking advantage of steady state of our objects

The equipment in process flowsheet, is always in steady state and has nothing to do with dynamic behaviors. In the prospective ROM there is no temporal variable and the ROM will remain as algebraic equations.

(3) Taking advantage of results solved by ASPEN and Fluent

It is well known that Aspen Plus is a convenient tool to simulate the processes and Fluent is a powerful CFD tool to simulate the flow field inside the equipment. Both contain specialized, sophisticated algorithms to handle complicated calculations listed in section 3.1.2 (2). In particular, the CFD code solves complicated PDAE models with its own proprietary or in-house algorithms. Thus, it seems impractical to redevelop and solve these PDAEs again through POD during the development of the ROM. As shown in Figure 3-2, our task is to establish systematic methodology to develop the ROM to express the relationship between the output from the equipment (for example, the CSTR in the flowsheet) and the inputs entering it, according to the results or dataset obtained by Aspen Plus and Fluent. Optionally, the ROM can also give the field information if required. The PCA based ROM will be a methodology that is efficient, easy to implement, generic for all types of equipment and therefore superior to POD based ROM in the present task.

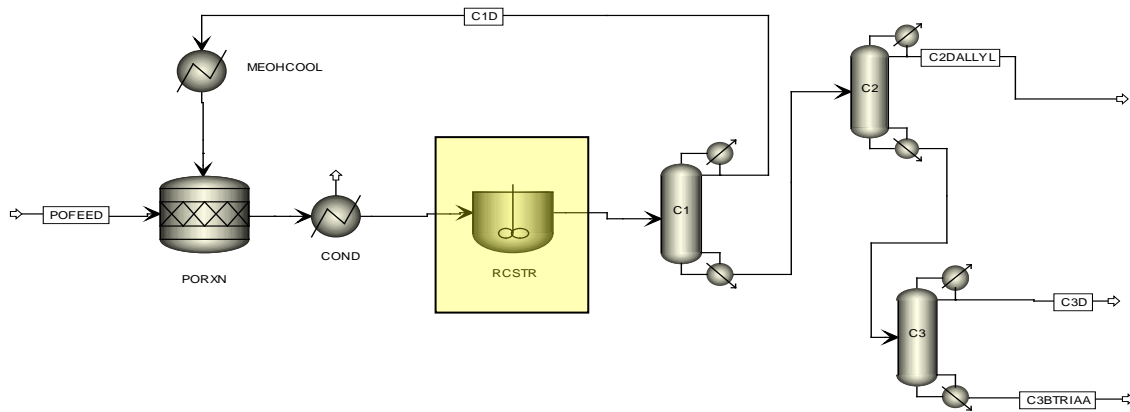


Figure 3-2. An example of process flowsheet in APECS

(4) Validity of principal components in PCA

As long as the inputs used to obtain the dataset cover the range of the prospective inputs during iterative simulation and/or optimization, the developed PCA-based ROM, including *PCs* and the input *mapping* (defined later), is still valid. This property is crucial to iterative optimization of process flowsheet.

4. Methodology of PCA based ROM

This section provides a self-contained and detailed description of the PCA-based approach, including basic concepts and implementation.

4.1 Vectors, spaces and state space model

First, we define input space and output space for the particular equipment. The input space consists of all input variables as its coordinates, such as temperature T_{in} , pressure P_{in} , feed rate of each species f_i , $i = 1, nci$, (here nci denotes number of species feeding into the equipment), as well as operation parameters, for instance shaft speed in CSTR. An input vector u in the input space contains the values of all input variables, i.e. $u = [T_{in} \ P_{in} \ f_1 \ \cdots \ f_{nci} \ \cdots]^T$, where T_{in} and P_{in} are the temperature and pressure of the streams entering the equipment, respectively. f_i , $i = 1, nci$, are flow rates of the species in the input stream. Similarly the output space is spanned by all variables in output streams leaving the equipment. An output vector y in the output space contains the magnitudes of all output variables, i.e. $y = [T_{out} \ P_{out} \ d_1 \ \cdots \ d_{nco}]^T$, where T_{out} and P_{out} are the temperature and pressure of the streams from the equipment, respectively. d_i , $i = 1, nco$, (nco is the number of species in the stream) are flow rates of the species in the output stream.

Then we define a state vector x in the state space. Each entry in column vector x represents a value of a state variable in one node of the mesh.

$$x = \begin{bmatrix} u_x \\ T \\ P \\ \vdots \end{bmatrix} \quad \text{where } u_x, T \text{ and } P \text{ themselves are also column vectors representing } x-$$

components of the velocity, temperature and pressure at all nodes of the mesh in the field, respectively. For simplifying and not losing the generality, from now on, we denote x as one specific state variable, e.g. temperature at all the mesh.

Conventionally, we also define state space model in form of

$$\begin{aligned} \dot{x} &= \Lambda x + \Pi u \\ y &= \Gamma u \end{aligned} \tag{4-1}$$

to consider the responses of state x and output y to input u . However, since our system is steady state, then the model can be simplified further as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} B \\ D \end{bmatrix} \cdot u \tag{4-2}$$

where we treat the matrix B as a mapping that transforms u in the input space to x in the state space and D as a mapping that transfer u to y in the output space. Thus, we focus on identifying the two matrices B and D first, then we can calculate the state x and the output y corresponding to a given input u in form (4-2), the prospective ROM.

4.2 Theoretical concepts of PCA and SVD

Given a matrix $\Psi \in \mathfrak{R}^{m \times n}$ with columns $\Psi_j \in \mathfrak{R}^m$ as eigenvectors of correlation matrix XX^T , where $X \in \mathfrak{R}^{m \times n}$ is the dataset matrix, then the columns $X_j, j = 1, n$ in X can be expressed as a linear combination of $\Psi_j, j = 1, n$, called the principal components (PCs). For example, the j^{th} column of X can be expressed as

$$X_j = \sum_{i=1}^r \Psi_i \alpha_{i,j} = \Psi \alpha_j \quad (4-3)$$

where $\alpha_j \in \mathfrak{R}^r, j = 1, n$ are vectors with entries $\alpha_{i,j}, i = 1, r; j = 1, n$ which stores the “score” coefficients and $r \leq n$ is the rank of the new coordinate system. Here the $\Psi_j, j = 1, r$ for all the columns of X are the same and each column $X_j, j = 1, n$ has its own *score vector* $\alpha_j \in \mathfrak{R}^r$. Equivalently, this procedure transforms dataset X into a new coordinate system spanned by PCs with r coordinate variables that account for the majority of the variance in the dataset X .

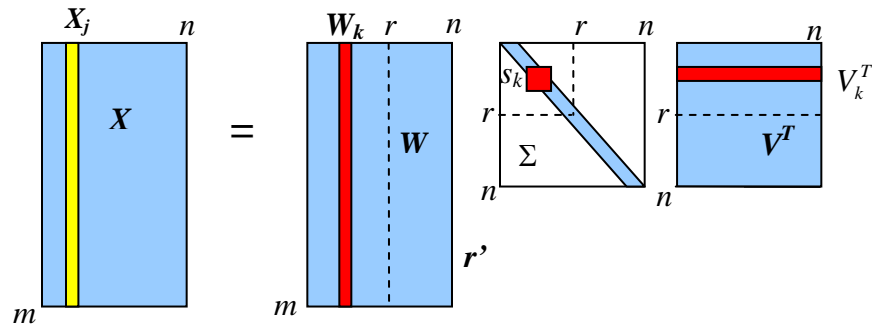


Figure 4-1. Illustration of SVD

From another perspective, we write the Singular Value Decomposition (SVD) of matrix X as:

$$X = W \Sigma V^T \quad (4-4)$$

where $W \in \mathfrak{R}^{m \times n}$, called the left-singular matrix, as in PCA, is the eigenvector matrix of correlation matrix XX^T , $V \in \mathfrak{R}^{n \times n}$, called the right-singular matrix, is the eigenvector

matrix of matrix $X^T X$, $\Sigma \in \mathfrak{R}^{n \times n}$ and $\Sigma = \text{diag}(s_1, s_2, \dots, s_n)$, has components s_k called the singular values. If rank of X is L , then $s_k > 0$ and s_k^2 equal to k^{th} largest eigenvalue of $X^T X$ when $1 \leq k \leq L$, otherwise $s_k = 0$. Figure 4-1 illustrates the structure of a matrix and its SVD.

The ratio of variance captured by the first r principal components to the total variance in the dataset is given by:

$$R_r = \frac{\sum_{k=1}^r s_k^2}{\sum_{k=1}^n s_k^2} \leq 1 \quad (4-5)$$

Based on the cut-off value of R_r , we select the first r columns of W as PCs, and the dataset can be reduced to r -dimensional coordinates spanned by $W_i, i = 1, r$.

$$X = W^{(r)} \Sigma^{(r)} (V^{(r)})^T \quad (4-6)$$

Let $\Psi = W^{(r)}, \alpha = \Sigma^{(r)} (V^{(r)})^T$, where $\alpha \in \mathfrak{R}^{r \times n} = \Sigma^{(r)} (V^{(r)})^T$ is the coefficient matrix. Its column vector $\alpha_j \in \mathfrak{R}^r$ stores the coefficients for X_j , which can now be expressed as a linear combination of r PCs.

$$X_j = \Psi \alpha_j = \sum_{i=1}^r \Psi_i \alpha_{i,j}, \quad j = 1, n \quad (4-7)$$

Comparing equation (4-7) with Eq. (4-3), we find that they are identical. Therefore we can use SVD in Matlab as a convenient tool to implement PCA for construction of the ROM.

4.3 Methodology of PCA based ROM

We begin with a set of data (SOD) obtained from running Fluent for a series of p input cases. Each input case is characterized by a vector defined in the input space, which contains all given information such as boundary conditions, equipment parameters, operation conditions and input information streams, e.g., defined from the (ASPEN) process simulator. We define an input matrix $U \in \mathfrak{R}^{p \times n}$ to store the set of input vectors. The dimensions of the matrix U indicate that p elements for each input vector and n cases under investigation. Values of U are bounded by domain Γ (also in term of ‘‘operation window’’), which defines the range of each variable in the input. Column $U_j, j=1, p$ stores the values of all input variables for a single CFD case.

Extracting one state snapshot from the SOD, we can formulate a dataset matrix, $X \in \mathfrak{R}^{m \times n}$ for the single state. We assume that there are q states in the equipment field, m

nodes in the mesh and n responses for each state to the n inputs. Therefore matrix $X \in \mathfrak{R}^{m \times n}$ has elements $x_{i,j}$, $j = 1, m$ of the columns $X_i \in \mathfrak{R}^m, i = 1, n$ that store the particular state value at the j^{th} node corresponding to the i^{th} input. We denote X_i as the “snapshot” for each input case.

We now apply SVD to X . Based on the criterion (2-17), we obtain ranks r , and then represent reduced versions of X as

$$X \approx X^{(r)} = W^{(r)} \Sigma^{(r)} (V^{(r)})^T = \Psi \alpha \quad (4-8)$$

where $\Psi = W^{(r)} \in \mathfrak{R}^{m \times r}$ and $\alpha = \Sigma^{(r)} (V^{(r)})^T \in \mathfrak{R}^{r \times n}$

Referring to Eq. (2-14), we decompose X into its PC matrix $\Psi \in \mathfrak{R}^{m \times r}$ and score matrix $\alpha \in \mathfrak{R}^{r \times n}$. We already indicated that PCs of the state are the same for all responses as long as the corresponding inputs are kept in the domain Γ . In other words, for any input $u_j \in \mathfrak{R}^p$ restricted in the domain Γ , we need only find its score vector $\alpha_j \in \mathfrak{R}^r$ to calculate its response $X_j \in \mathfrak{R}^m$. The only varying components in Eq. (4-7) are the score α_j obtained from PCA of the dataset and there are potential relationships between α_j and input u_j . Additional functions between them can be built with techniques such as regression, neural networks, etc., according to the complexity of the inputs. Equivalently, this is the procedure that builds the mapping $F(u) : U \mapsto \alpha(u)$ that allows us to calculate the state responding to any input u_s specified in the domain Γ ,

$$\alpha_s = \begin{bmatrix} \uparrow \\ \alpha_s(u_s) \\ \downarrow \end{bmatrix} = F(u) \cdot u = \begin{bmatrix} \leftarrow & F_1(u) & \rightarrow \\ \leftarrow & F_2(u) & \rightarrow \\ & \vdots & \\ \leftarrow & F_r(u) & \rightarrow \end{bmatrix} \cdot \begin{bmatrix} \uparrow \\ u_s \\ \downarrow \end{bmatrix} \quad (4-9)$$

where α_s is the score vector for the state responding to the specified input vector u_s . F_j , $j=1, r$ are the row component of matrix F containing all of the mapping coefficients.

By substituting α_s into Eq. (4-8), we can obtain the state x_s at all nodes of the mesh responding to input u_s as

$$X_s = \Psi \alpha_s = \Psi F(u) \cdot u_s \quad (4-10)$$

Indeed, Eq. (4-10) is the expected ROM of the state where PCs, $\Psi(x) \in \mathfrak{R}^{m \times r}$ and mapping $F \in \mathfrak{R}^{r \times k}$ are given by previous derivations, with which for any given input, u_s , the corresponding state, x_s can be computed directly. Thus the development of ROM for

the equipment is based on creating the mapping $F : U \mapsto \alpha$ from the CFD solutions, which leads to the mapping: $xROM : U \mapsto X$ for every state x . Comparing Eq. (4-10) with Eq. (4-2), it is obvious that the matrix B in Eq. (4-2) is decomposed into two matrices, i.e.

$$B = \Psi F(u) . \quad (4-11)$$

Also, it is clear how to identify them. Consequently, there are no restrictions due to geometric shape, field dimension, type of flow, material properties, chemical reactions, etc.

To recap the development of ROM from Fluent case studies, we summarize the main methodology below:

1. Obtain an accurate solution on a fine mesh. If desired, re-mesh this solution more coarsely using the equipment geometry or location of interest. This leads to more efficient ROM development and can be done with any available meshing tools.
2. Design a set of n inputs within predefined ranges for all p input variables. Formulate the inputs as a matrix $U \in \mathfrak{R}^{p \times n}$ as described above.
3. Solve the CFD cases one-by-one with the designed inputs under the original finer mesh and formulate the set of data (SOD).
4. Do a loop for each state, for $j = 1, q$ (q is number of the states to be considered)
 - 1) Extract the j^{th} state from the SOD. Formulate the dataset $X \in \mathfrak{R}^{m \times n}$ for the j^{th} state variable in the flow field e.g. temperature, pressure, velocity, etc.
 - 2) Implement SVD on $X = W \Sigma V^T$
 - 3) According to required accuracy and the cut-off criterion select the reduced rank, r .
 - 4) obtain the ranked principal components $\Psi = W^{(r)}$ and scores $\alpha = \Sigma^{(r)} (V^{(r)})^T$
 - 5) Build mapping $xROM : U \mapsto X$.
 - 6) Next j .

The collection of xROMs obtained from the loop is indeed the result PCA based ROM of the equipment. Additionally, we implement mapping from input to outlet vector of the equipment, $yROM : U \mapsto Y$. Eventually, we can formulate the ROM in state space model as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} B \\ D \end{bmatrix} \cdot u = \begin{bmatrix} \Psi & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} F(u) \\ G(u) \end{bmatrix} \cdot u \quad (4-12)$$

where $F(u)$ is defined in Eq. (4-9) and $G(u)$ is resulting from the mapping of $yROM$.

5. Case Studies

5.1 Case study I: flow in manifold

We next consider a small CFD example to further illustrate the ROM methodology. This initial study is the simplest one; it is in 2D and has only pressure and two components of velocity as its state variables. Neither energy nor chemical reaction equations are involved.

Figure 5-1 (a) shows an intersection view of a multiple-branch unit with a width of 1000 mm. Water at 20°C, 1 atm flows in at the top with given velocity (input) and outlets are open at the ends of the two branches at pressure of 1 atm. While the ROM methodology can, of course, be applied to an arbitrary geometry, we focus only on a part of the manifold, as shown in Figure 5-1 (b) to simplify the illustration. Using *pdetool* in Matlab, we mesh this area more coarsely, as depicted in Figure 5-2, than the mesh used in CFD computation, in order to reduce the computation cost and maintain sufficient accuracy for the ROM.

In this example, we design the range of the input variable as $v = 0.2-2.0$ [m/s] and 10 cases. Each case i corresponds to input of $v = i*0.2$, $i = 1,10$.

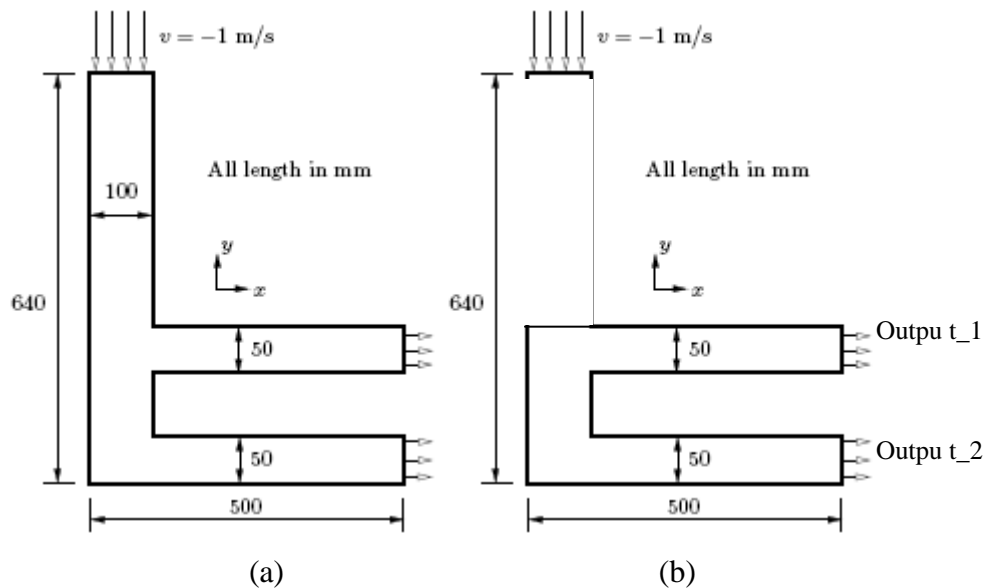


Figure 5-1. Geometry of manifold
(a) the whole geometry; (b) the part of area for ROM derivation.

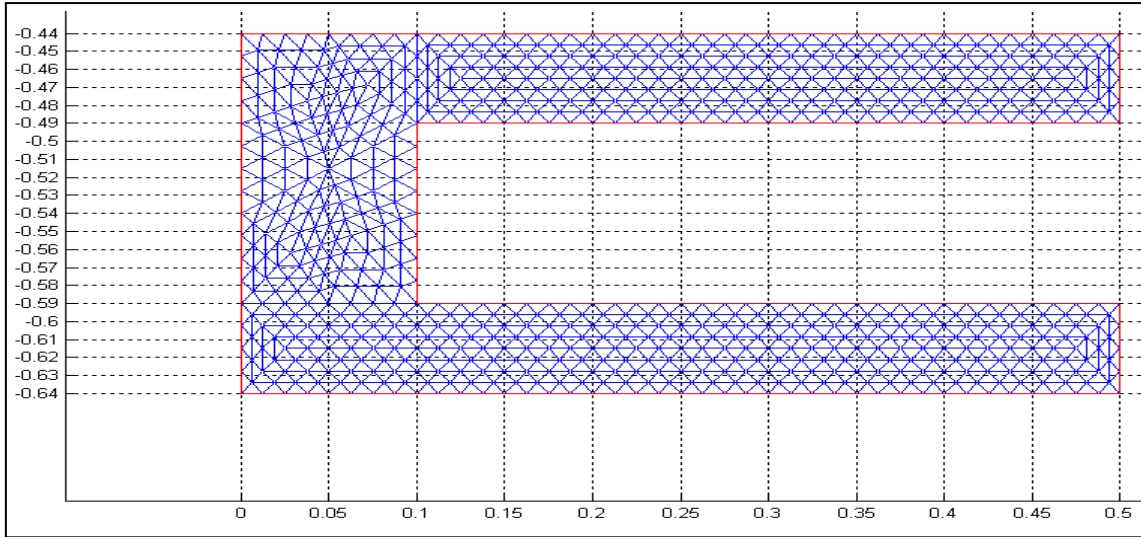


Figure 5-2. Coarser re-mesh for interested area.

The results of PCA for the case are shown in Figure 5-3. In the left column of Figure 5-3, we see that the eigenvalues decline rapidly and only two PCs are enough to capture more than 99% of variances in the dataset. The other two columns show the relationships between the scores of $F(u)$ for calculate components of velocity in x,y-coordinates and the pressure; these are linear for u_x and u_y and quadratic for P .

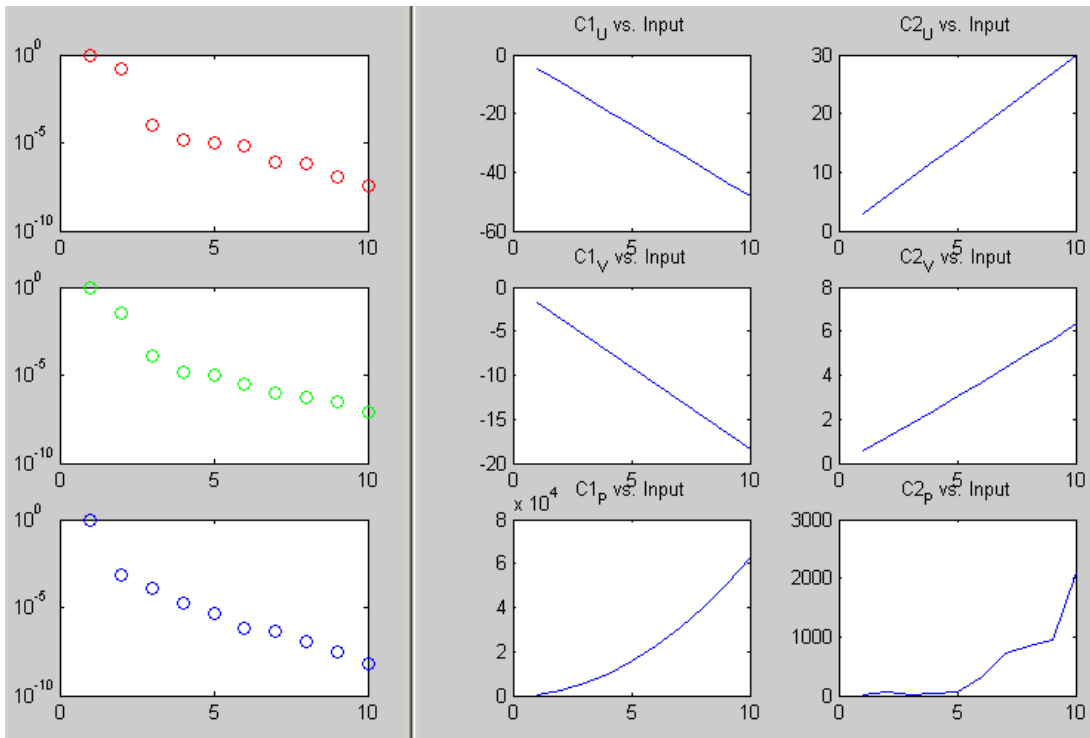


Figure 5-3. eigenvalues and relationship of scores

The linearity of the scores for u_x and u_y to the inputs can be verified by the linearity of the two outputs vs. input obtained from Fluent results for 10 cases, as shown in Figure 5-4.

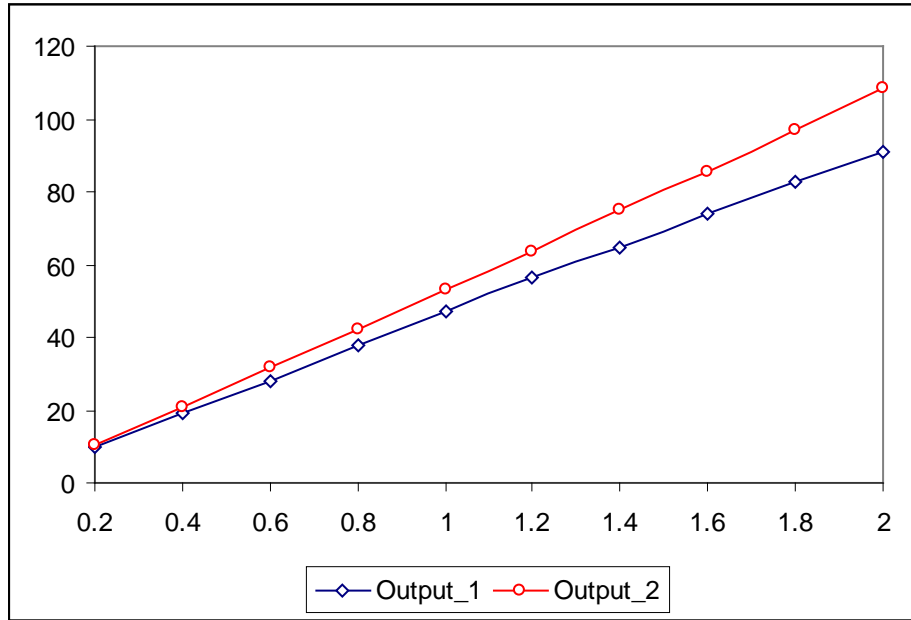


Figure 5-4. Two outputs vs. input

The dimensions of these cases are

- Number of variables in the input vector $p = 1;$
- Number of input vectors in the input space $n = 10;$
- Number of points in the state space or nodes in the mesh $m = 809;$
- Number of kinds of states involved $q = 3.$

The mapping $F : \beta_k(u) \mapsto \alpha_r(x)$ is a regression polynomial of the input variable v , in the form of

$$\alpha_{l,k}(x) = F_{k,0}^{(l)} + F_{k,1}^{(l)}v + F_{k,2}^{(l)}v^2 + F_{k,3}^{(l)}v^3 \quad (5-1)$$

where $F_{k,j}^{(l)}$, $l=1,3$, $k=1,2$ $j = 0,3$ are the regression coefficients, superscript (l) corresponds to the index of states, and subscript k is the index of the PC. The regression results, obtained by MS Excel, are listed in Table 5-1.

Table 5-1. Values of scores in ROM

	$F_{k,0}^{(l)}$	$F_{k,1}^{(l)}$	$F_{k,2}^{(l)}$	$F_{k,3}^{(l)}$
$\alpha_{1,1}(x)$	7.532	-24.088	-7.5E-3	-3.83E-5
$\alpha_{1,2}(x)$	-22.577	14.795	2.367E-2	9.279E-3
$\alpha_{2,1}(x)$	4.713E-1	-9.259	1.0845E-1	-4.236E-2
$\alpha_{2,2}(x)$	-4.442	3.010	3.085E-2	3.064E-2
$\alpha_{3,1}(x)$	-16.309	-1.189E+2	1.588E+4	-76.998
$\alpha_{3,2}(x)$	-1.077E+3	4.887E+2	-8.261E+2	5.423E+2

With all results from PCA and the mapping, the ROM can be expressed as following in the form (4-11) with $u = [1 \quad v \quad v^2 \quad v^3]$ where v is inlet flow rate,

$$X_l = \sum_{k=1}^2 \Psi_{l,k}(x) \alpha_{l,k}(x) = \begin{bmatrix} \uparrow & \uparrow \\ \Psi_{l,1} & \Psi_{l,2} \\ \downarrow & \downarrow \end{bmatrix} \begin{bmatrix} F_{1,0}^{(l)} & F_{1,1}^{(l)} & F_{1,1}^{(l)} & F_{1,2}^{(l)} \\ F_{2,0}^{(l)} & F_{2,1}^{(l)} & F_{2,2}^{(l)} & F_{2,3}^{(l)} \end{bmatrix} \begin{bmatrix} 1 \\ v \\ v^2 \\ v^3 \end{bmatrix} \quad (5-2)$$

where the subscript is for l^{th} state, i.e. $l = 1,2$ for u_x and u_y , respectively, $l = 3$ for the pressure P . and $\Psi_{l,k}(x)$, $k = 1,2$ are the PCs for l^{th} state.

Using the developed ROM, we calculate the state responses for three inputs $v = 0.4, 1.0$ and 2.0 , respectively. The contour plots generated by Matlab (right columns) are shown in Figure 5-5, 5-6 and 5-7. To compare the performance of the ROM, the contours plotted by Matlab with the data extracted from the CFD solution according to the coarser re-mesh, are shown in the left column.

It is evident that the ROM performs well. The wall clock time for Fluent to converge one manifold case is around 100 seconds while the ROM in Matlab for the same case runs in less than a second.

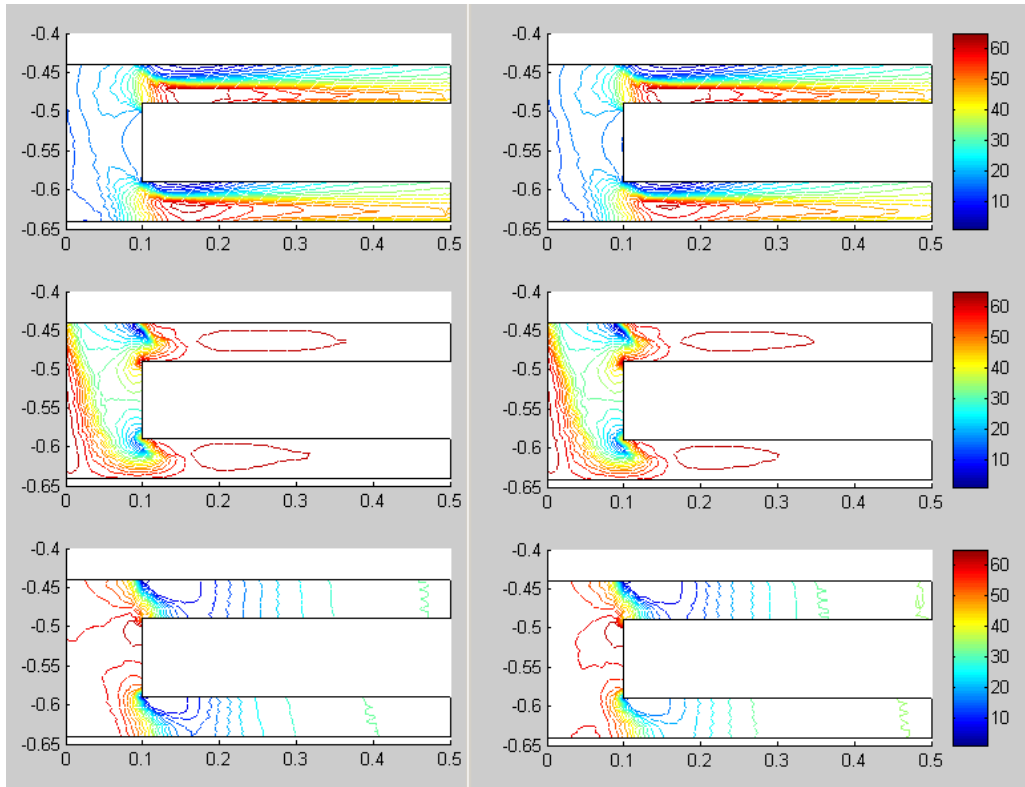


Figure 5-5. Comparison of ROM results with input $v_y = -0.4$

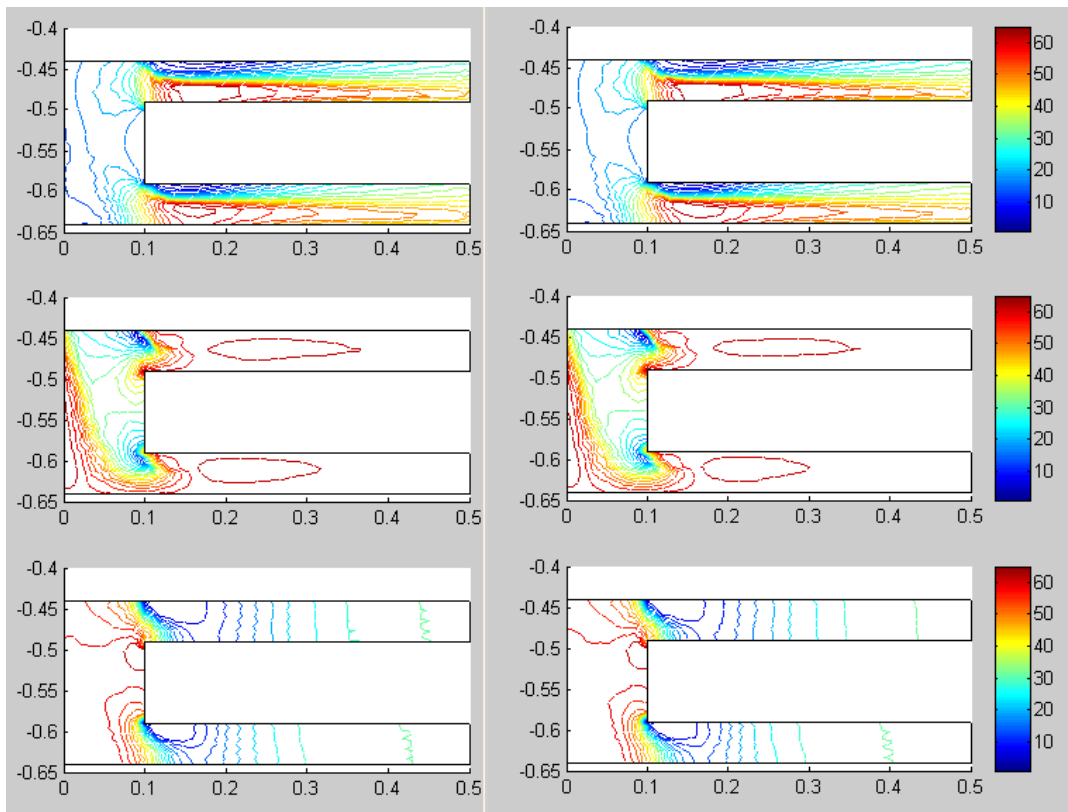


Figure 5-6. Comparison of ROM results with input $v_y = -1.0$

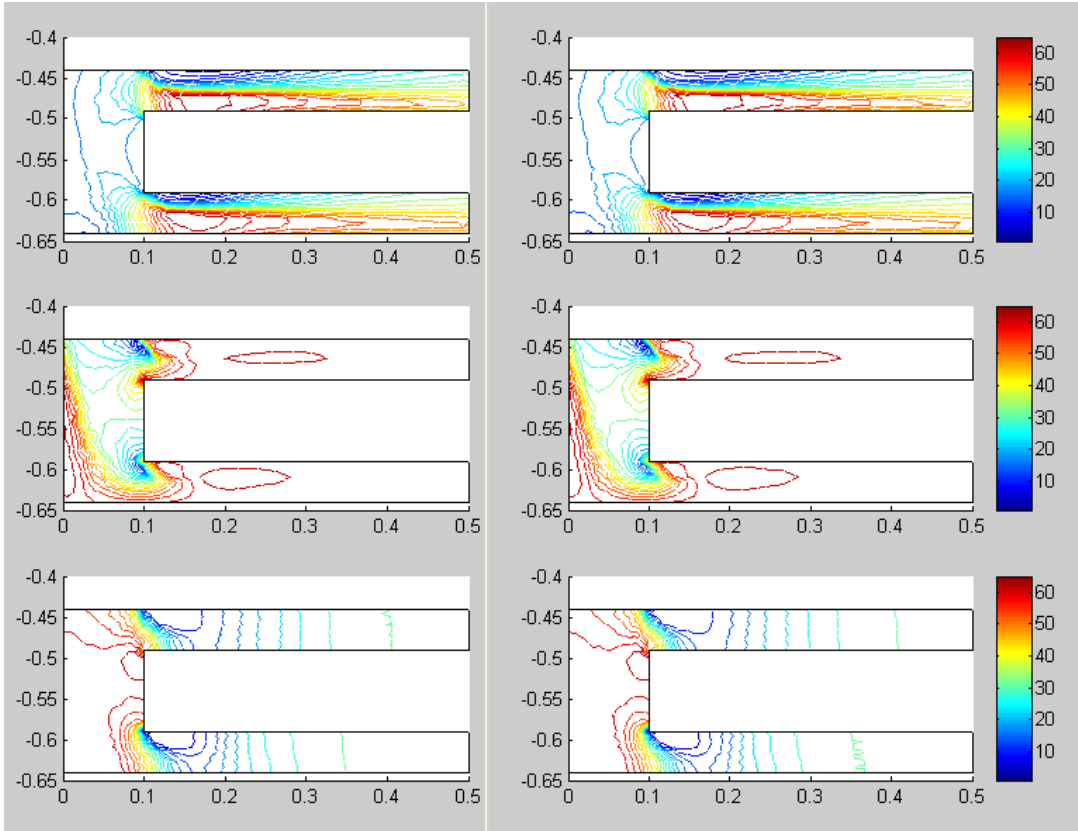


Figure 5-7. Comparison of ROM results with input $v_y = -2.0$

5.2 Case study II: Gas Turbine Combustor

There are two important equipment units in the FuetureGen process model flowsheet, the gas-turbine combustor and the gasifier, which are simulated with CFD models in FLUENT/APECS. In order to test the methodology of Reduced Order Models (ROM), comprehensively, the combustor was selected as the subject of the case study because of its complexity. First of all, the model of the combustor has 10 input variables defined as operating conditions and boundary conditions, in FLUENT interface. This multiple-input model introduces the necessity of invoking techniques in the area of experimental design. Here the general method of Latin Hypercube Sampling (LHS) is used to generate an experimental set of the inputs to produce a set of databases corresponding to the designed range of the inputs.

After investigation of the properties of the inputs, we found that some inputs are correlated and could not be changed independently. To guarantee feasible operation of the combustor in IGCC flowsheet, we had to remap inputs to make them independent. For example, in order to keep combustion feasible, we created a new variable, mole ratio of oxidant (O₂) to fuels (CO, H₂ and CH₄). According to the outlet of the gasifier, we constrain the molar ratio of CO to H₂ as an independent input. With these assumptions, we reduced the number of inputs from ten to nine independent ones. Then based on the range of the inputs which cover the possible range of input variables (e.g., during iterative convergence of a flowsheet simulation), we determined the range of inputs within which the developed ROM is valid. In this study, 128 experimental samples are designed and 128 corresponding FLUENT cases are solved with the same mesh as shown.

It is obvious that the interesting portion of the temperature field is located at the entrance of the combustor, while the temperatures remain uniform elsewhere. Therefore, we consider only this rectangular entrance portion, shown in Figure 5-10(b), as the object of our study. In that case, we can use the graphical facilities of Matlab to plot the results and compare the differences of the temperature fields from the original solution and the ROM. With the database obtained from FLUENT corresponding to the area of interest, we implement Principal Component Analysis (PCA) using Singular Value Decomposition (SVD) in Matlab.

We showed that with rank 35, the ranked PCA can capture 99.999% of the energy of the 128 cases, and we use 35 principal components to develop the ROM. The PCA decomposes the database into two parts; the principal components are new coordinates that are fixed. The scores are the projection of the database into the coordinates, determined by the inputs sets. In other words, if we establish the relationship between the scores and inputs, we can relate the temperature field to the inputs combined with the coordinates. This is our goal with the derived ROM. In addition, we use a neural network to regress the functional relationship of the scores from the PCA to the inputs of the FLUENT cases. Mathematically speaking, this implements the mapping from the inputs to the scores.

Once the ROM is derived, we test and analyze its accuracy and validity. Here we make two observations.

- The ROM can reproduce the temperature field and the outlet temperature from the combustor accurately at the 128 experimental points. The error of the outlet temperature is less than 10^{-7} .
- We observed that in the neighborhood of the existing data points, if the radius of the input neighborhood is small, the derived ROM provides a good approximation in comparing with the solution with FLUENT at the same input point. This phenomenon predicts that if the density of the existing points in the input space is high enough, then the derived ROM based on that database will be good for that input space.
- Also, from a random sampling of the input space, we found that the output temperatures from the ROM were within 3% of those predicted by FLUENT. Nevertheless, from an analysis of the sampled data in the space of the principal components, further improvements would be useful for ROM development. In particular, we believe that the database obtained with 128 points is sufficient to approximate the model response with acceptable engineering accuracy. The accuracy for state field approximation can be improved by further work.

5.2.1 Overview of Gas Turbine combustor

The flowsheet for the FutureGen process is shown in Figure 5-8. In this process, the combustor is an important reactor that combusts synthesis gas to produce a high temperature exhaust stream, which drives the turbine. The fuel is originally produced by a gasifier followed by a series of pre-treatment procedures such as water gas shift reaction and CO₂ quench. The syngas is premixed with compressed atmospheric air to form the combustor input stream. The variables included in this input stream are mass flow rate, temperature and mass fractions of eight components (seven out of eight are independent). The components are CO₂, CO, O₂, H₂, H₂O, CH₄, Ar and N₂. Among these, only CO, H₂, CH₄ and O₂ are involved in the combustion reactions.

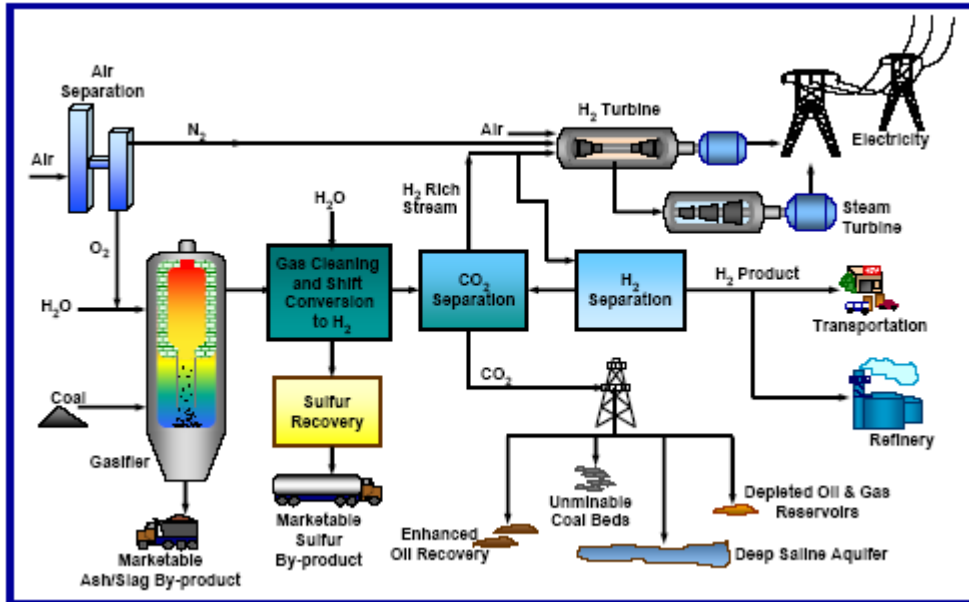


Figure 5-8. Scheme of flowsheet of IGCC in FutureGen power plant

In this study, we use a FLUENT CFD model of the combustor. This is a turbulent, lean-premixed, swirl-stabilized combustor based on the NETL research combustor described by Sidwell et al. (2005). The CFD model is scaled up to represent a single combustor can; a 250MW, 16-can gas turbine is used in the FutureGen power plant. The 2D axisymmetric combustor geometry with nearly 6000 computational cells is used to solve the CFD model, as shown in Figure 5-9 (Zitney et al. 2005).

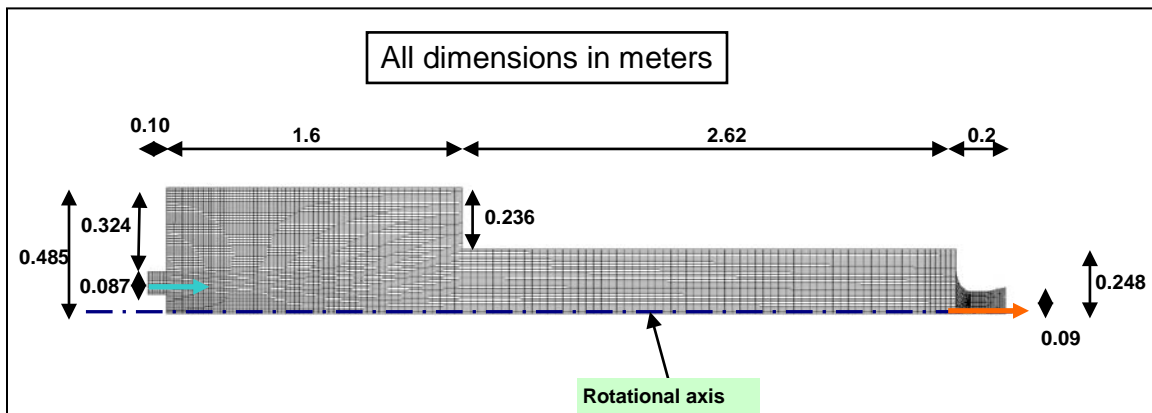


Figure 5-9. 2-D Geometry of a single axisymmetric combustor can and its mesh

The flame is anchored by careful scaling and design of the combustor, so that the nozzle velocity of the NETL research combustor is matched. In the FLUENT model of the combustor, the finite rate/eddy dissipation model is selected. The reaction rate is defined

by taking the minimum of the chemical reaction rate and the turbulent mixing rate. The global reactions in the combustor are:



With the CFD model, we can change the input variables (in terms of inlet stream information in Aspen Plus/APECS) that represent operating conditions and boundary conditions in FLUENT, to obtain corresponding state fields and output streams. In this study, we consider one state, temperature, only. Nevertheless, the methodology of deriving the ROM for temperature can be applied directly to any other state without any restriction. Figure 5-10 shows a typical temperature field displayed as filled contours.

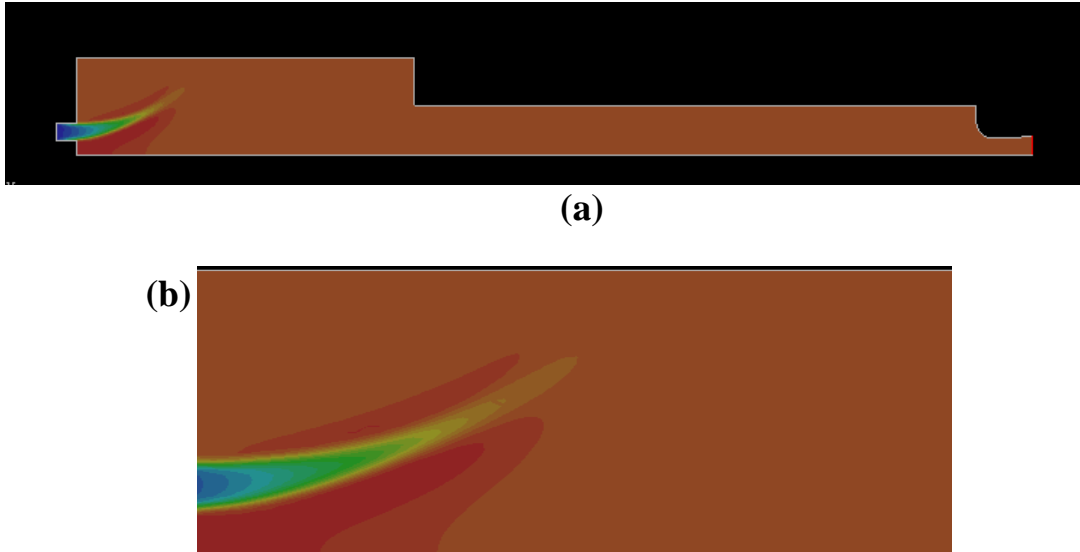


Figure 5-10. A typical temperature field (a) entire geometry; (b) Enlargement of the interesting portion of the field

From the entire temperature field, Figure 5-10 (a) we can immediately find that the temperature changes significantly only in the left portion of the geometry; while in the rest of the geometry, the temperature stays uniform. In order to use the graphic facility in MATLAB to display the results easily without loss of generality, we focus only on the left rectangle shown in Figure 5-10 (b), excluding the nozzle part.

5.2.2 Experimental Design

Because the combustor model has multiple inputs, we need to minimize the number and size of the database used for development of the ROM and maximize the information covered for the database. Here it is necessary to invoke the techniques of experimental design to obtain a designed set of inputs that cover the ranges of the inputs. There are a few methods of experimental design. In this study; we chose the Latin Hypercube

Sampling (LHS) method with uniform probability distribution for the nine independent input variables.

Before we implement LHS, we reformulate the input space first to make all components or coordinates spanning the input space non-correlating, i. e. independent of each other.

By inspecting the chemical behaviors of the combustor and the streams, we found the molar fractions of CO and H₂ are correlated and dominated by the ratio of CO/H₂ controlled by the gasifier. In addition, to maintain proper combustion the molar ratio of oxidant to the fuel must be constrained. According to the flowsheet, we assume that there are two streams, Fuel and Air, mixing together and forming one stream entering the combustor. With some assumptions for simplicity, we finally end up with the following nine independent input variables: operation pressure, inlet temperature, mass flowrate, inlet mole fractions of CO, CO₂, H₂O, CH₄, the molar ratio of CO/H₂ in fuel stream, and the molar ratio of (CO+H₂+CH₄)/O₂.

The ranges of the input samples are based on data offered by FLUENT shown in Table 5-2 and modified according to chemical behavior of the role of the combustor in the flowsheet of FutureGen as in Table 5-3.

Table 5-2. Original inputs given by FLUENT

		Min	Max
<i>u1</i>	Operation. Pressure [MPa]	1.4	2.2
<i>u2</i>	Inlet Mass fraction of CO ₂ [-]	0.0	0.04
<i>u3</i>	Inlet Mass fraction of CO [-]	0.0	0.01
<i>u4</i>	Inlet Mass fraction of O ₂ [-]	0.10	0.25
<i>u5</i>	Inlet Mass fraction of H ₂ [-]	0.0	0.04
<i>u6</i>	Inlet Mass fraction of H ₂ O [-]	0.0	0.06
<i>u7</i>	Inlet Mass fraction of CH ₄ [-]	0.0	0.04
<i>u8</i>	Inlet Mass fraction of Ar [-]	0.0	0.02
<i>u9</i>	Inlet Mass Flow Rate [Kg/s].	10.0	25.0
<i>u10</i>	Inlet Temperature [K].	300	1000

The experimental design is implemented using Latin Hypercube Sampling to produce a set of experimental independents with parameters:

- No. of independents without correlation = 9
- No. of experimental samples = 128
- Probability distribution = Uniform

Table 5-3. Modified ranges of independent design variables

			Min	Zitney	Max
Fuel	<i>_u2</i>	Molar fraction of CO2 [-]	0.01	0.026	0.05
	<i>_u3</i>	Molar fraction of CO [-]	0.02	0.035	0.04
	<i>_u6</i>	Molar fraction of H2O [-]	0.15	0.211	0.25
	<i>_u7</i>	Molar fraction of CH4 [-]	0.02	0.037	0.05
	<i>_u5</i>	Molar fraction of CO+H2 [-]	0.5	0.565	0.65
Air	<i>_u6</i>	$R2=O2/(0.5(CO+H2)+2CH4[-])$	0.98	---	5.50
Phys	<i>_u1</i>	Operation. Pressure [MPa]	1.4	1.85	2.2
	<i>_u8</i>	Inlet mass flow rate [Kg/s]	15.0	20.5	25.0
	<i>_u9</i>	Inlet Temperature [K].	300	659	1000

LHS gives a designed set of independent variables with 128 cases. In each case, there are sampling values for the nine variables (*_u1* -- *_u9*) within the ranges. Since some of these independent inputs cannot be prescribed as boundary conditions directly in FLUENT, we map them into FLUENT acceptable parameters for each case. Also, we map them into corresponding FLUENT inputs for each case.

5.2.3 Building the Database

It is straightforward to perform a batch of FLUENT simulation for the designed sampling set. For each group of the input in the designed sampling set, a case is established by solving the FLUENT CFD model first, then extracting the temperature field and integrated (mass average weighted) outlet temperature. When all 128 cases are solved, we obtained a database matrix with 5996 (number of mesh points) rows and 128 columns, each column of the matrix corresponds to the temperature field of one case. Also, we obtain a vector where each entry represents the outlet temperature of corresponding case.

As we mentioned above, since most of the temperature field is uniform, we consider only the area of interest in Figure 5-10(b) for our investigation and development of the ROM.

5.2.4 PCA Applied to the Database

The established database is expressed as the snapshot matrix X_T . Implementing PCA on the matrix, X_T we obtain

$$X_T = W\Sigma V^T \quad (5-4)$$

as described previously. To develop a more accurate ROM, we prefer to not lose the accuracy in PCA approximation by taking higher rank. In this case, we choose rank of 35 and the ratio of captured energy is more than 99.999%. Then, we use the first 35 columns of W and first 35 rows of V^T to formulate the PCA approximation of the database matrix, i.e.

$$X_T \approx W_r \Sigma_r V_r^T \quad (5-5)$$

Defining the scores through the following matrix according to Eq. (4-8):

$$\Psi = W_r \in \mathfrak{R}^{2601 \times 35}; \quad \alpha \approx \Sigma_r V_r^T \in \mathfrak{R}^{35 \times 128}$$

leads to:

$$X_T \approx \Psi \alpha \quad (5-6)$$

It is known that the scores only correspond to the input that determines the database. In the next section we will discuss mapping the input set to the score with the neural network method.

Using obtained parameters of the network we build a new network for network simulation, i.e. for a given input vector, u , and calculate the corresponding output of the network, the score,

$$\begin{bmatrix} F(u) \\ G(u) \end{bmatrix} = F_2(LW \cdot (F_1(IW \cdot u + b\{1\}) + b\{2\})) \in \mathfrak{R}^{36} \quad (5-7)$$

as well as outlet temperature. Then, getting back the principal component, we formulate the Reduced Order Model for the temperature field in the combustor and outlet temperature as,

$$\begin{bmatrix} X_T \\ T_{ROM}^{out} \end{bmatrix} = \begin{bmatrix} \Psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F(u) \\ G(u) \end{bmatrix} \cdot u \quad (5-8)$$

5.2.5 Regression using Neural Network

1. Principal Component Analysis on the temperature snapshot matrix $X_T \in \mathfrak{R}^{2601 \times 128}$, with >99.999% energy captured, produced 35 corresponding ranked principal components $\Psi \in \mathfrak{R}^{2601 \times 35}$ and scores $F \in \mathfrak{R}^{35 \times 126}$.

Using one neural network for the ROM to accommodate the scores and the outlet temperature, we add outlet temperatures in 128 cases into the last row of the score matrix we get

$$\Omega = \begin{bmatrix} \alpha \\ T^{out} \end{bmatrix} \in \begin{matrix} \mathfrak{R}^{35 \times 128} \\ \mathfrak{R}^{1 \times 128} \end{matrix} \quad (5-9)$$

- In order to relate the independent experimental design $U \in \mathfrak{R}^{9 \times 128}$ to the ranked scores and outlet temperature in $\Omega \in \mathfrak{R}^{36 \times 128}$, a back-propagation neural network is created with 800 nodes in the hidden layer and 36 nodes in output layer, respectively as shown in Figure 5-11.

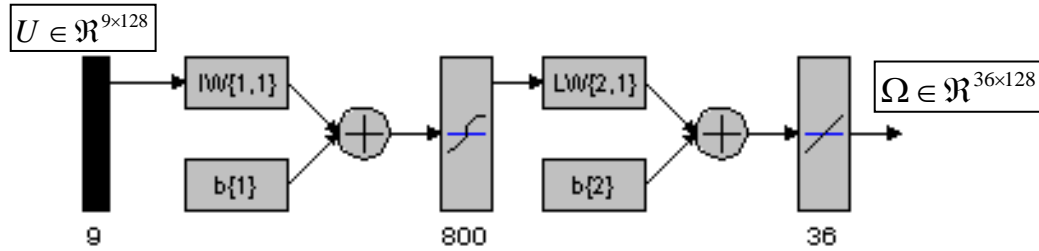


Figure 5-11. Scheme of neural network

To justify this scheme, it is worth noting the following property of neural networks:

“Multiple-layer networks are quite powerful. For instance, a network of two layers, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well.”

--- H. Demuth et al, “Neural Network Toolbox” (2005)

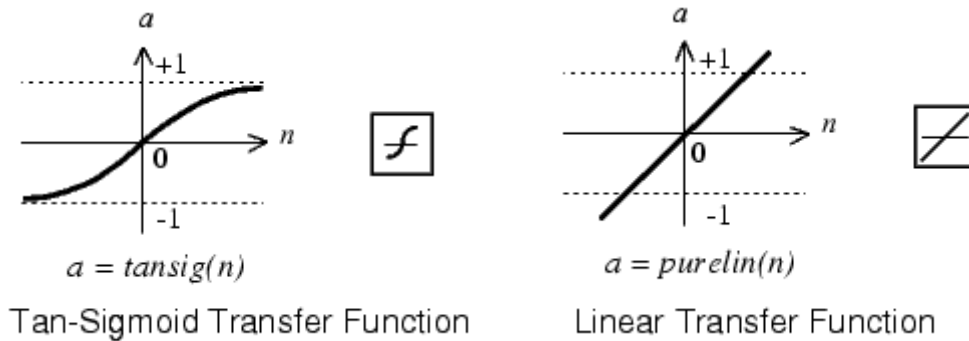


Figure 5-12. Transfer functions used in NNET

- The mapping can be expressed as

$$\begin{bmatrix} F(u) \\ G(u) \end{bmatrix} \cdot u = L(LW \cdot S(IW \cdot u + b\{1\}) + b\{2\}) \quad (5-10)$$

where S is the Tan-Sigmoid transfer function with parameters IW and $b\{1\}$ for layer 1 and L is the linear transfer functions, with parameters LW , and $b\{2\}$ in layer 2.

4. With $\Omega \in \mathfrak{R}^{36 \times 128}$ as the target and $U \in \mathfrak{R}^{9 \times 128}$ as the input, we train the neural network with $MSE(\text{Mean Square Error}) \leq 10^{-16}$ to get the parameters

$$\begin{aligned} IW\{1,1\} \in \mathfrak{R}^{800 \times 9}, \quad b\{1\} \in \mathfrak{R}^{800 \times 1}, \\ LW\{2,1\} \in \mathfrak{R}^{36 \times 800}, \quad b\{2\} \in \mathfrak{R}^{36 \times 1}, \end{aligned} \quad (5-11)$$

simultaneously

5. Using obtained parameters of the network we use the same network for simulation. That is, for a given input vector, u , we calculate the corresponding output of the network, the score and outlet temperature,

$$\begin{bmatrix} F(u) \\ G(u) \end{bmatrix} \cdot u = L(LW \cdot S(IW \cdot u + b\{1\}) + b\{2\}) \quad (5-10)$$

as well as outlet temperature.

Then, using the principal component set, we formulate the Reduced Order Model for the temperature field in the combustor and outlet temperature as,

$$\begin{bmatrix} X_T \\ T_{ROM}^{out} \end{bmatrix} = \begin{bmatrix} \Psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} F(u) \\ G(u) \end{bmatrix} \cdot u \quad (5-12)$$

5.2.6 Evaluation of ROM

In order to evaluate the ROM thoroughly, two different comparisons are made. First, we test the predictive accuracy of the ROM at the LHS design points. Second, we evaluate the approximation quality of the ROM at the random points within the designed range in the nine-dimensional input space.

5.2.6.1 Evaluation at the designed data points

We now have three data groups obtained by FLUENT simulation, PCA approximation and the derived ROM. Each set contains 128 cases with different operating and boundary conditions. For each FLUENT simulation, we have one temperature field of interest

which consists of 2601 nodes, as well as the value of the outlet temperature averaged over the mass flow. All of the temperature files of 128 cases construct a 2601 x 128 snapshot matrix. With PCA on the matrix, we obtained ranked principal components and corresponding scores with rank of 35 with 99% energy captured. Then we developed a ROM for PCA approximation along with outlet temperatures included in the ROM. We denote the resulting FLUENT temperature fields as true values of the combustor model, written as TF_{true} . Similarly TF_{PCA} represents the fields approximated by the ranked PCA and TF_{ROM} represents the field calculated with the derived ROM. Let T_{true}^{out} and T_{ROM}^{out} denote the outlet temperatures integrated by FLUENT and calculated by the ROM, respectively.

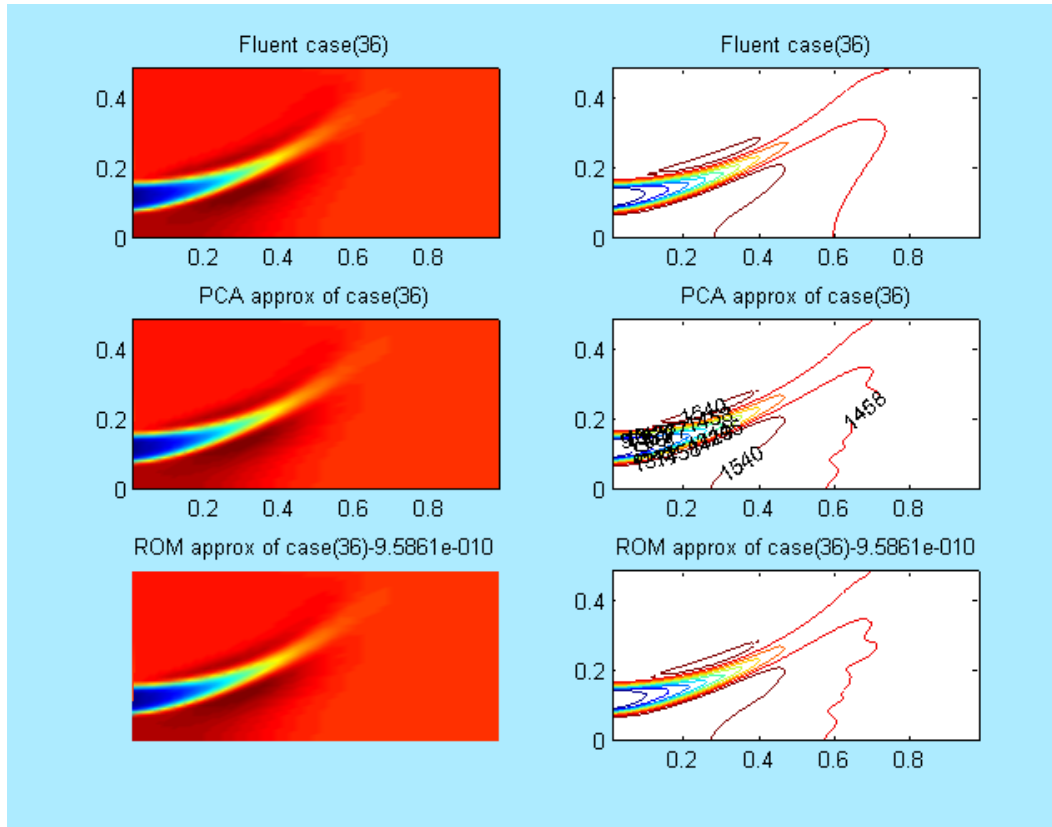


Figure 5-13. An example case comparisons

We plot the three temperature fields in filled and unfilled contours separately using data in $TF_{true}(i)$, $TF_{PCA}(i)$ and $TF_{ROM}(i)$, respectively, corresponding to case i . An example is shown in Figure 5-13. Then we compare these plots one-by-one and conclude that in some of them there is a little observable distortion between TF_{true} and TF_{PCA} , even though most of them have no distinguishable differences. It is possible to compare the outlet temperatures obtained by FLUENT and the ROM quantitatively. The magnitude of the differences is 10^{-8} to 10^{-9} , since we used the MSE (Mean Square Error) of 10^{-16} when we trained the neural network for developing the ROM. Therefore, we have:

$$TF_{ROM}(i) = TF_{PCA}(i) \approx TF_{true}(i), \quad i = 1, 128$$

And because the error is so small, we can state that quantitatively,

$$T_{ROM}^{out}(j) = T_{true}^{out}(j), \quad j = 1, 128$$

Summarizing the above, we are confident that the ROM is an accurate approximation of the FLUENT model *at all of the 128 designed data points*.

5.2.6.2 Evaluation at random points

We test the ROM at a set of 45 random points in the nine dimensional input space. First we pick up a pair of random integer number between 1 and 128, e.g. ip_1 and ip_2 , as the indices of cases in the database, called “parents”. The inputs corresponding to these parent cases are denoted as $V(ip_1)$ and $V(ip_2)$, or simply V_1 and V_2 . Second we obtain a random number between 0 and 1, called w , as the weight to produce a “child” test point expressed as

$$V_{child} = w \cdot V_1 + (1 - w)V_2 \tag{5-13}$$

Having expressed the points as vector in the input space, we can measure the “closeness” of the parents and child to its closet parent in term of 2-norm of the differences of two vectors. These are illustrated in Figure 5-14, where D_C is the distance between the child vector and its closet parent, and D_P is the distance between the pair of the parent vectors.

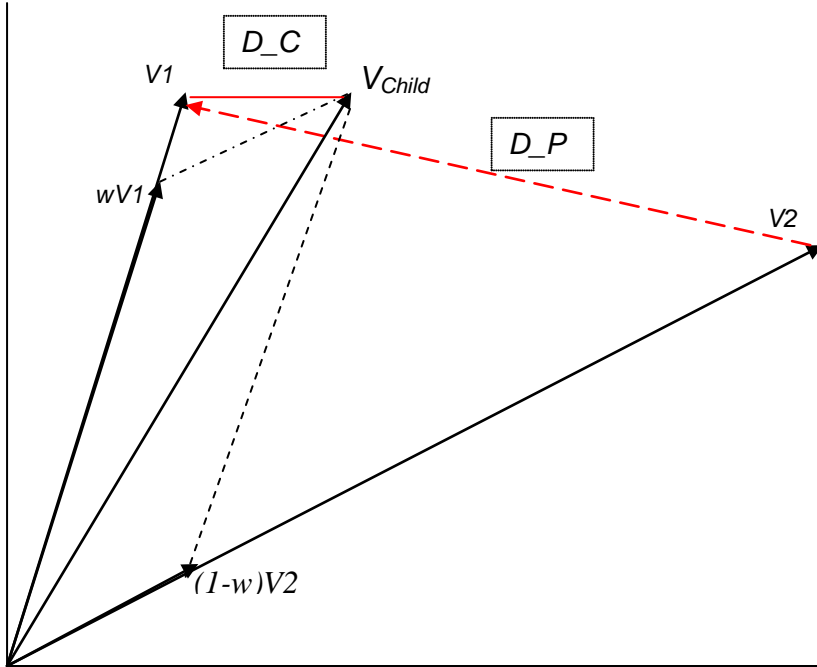
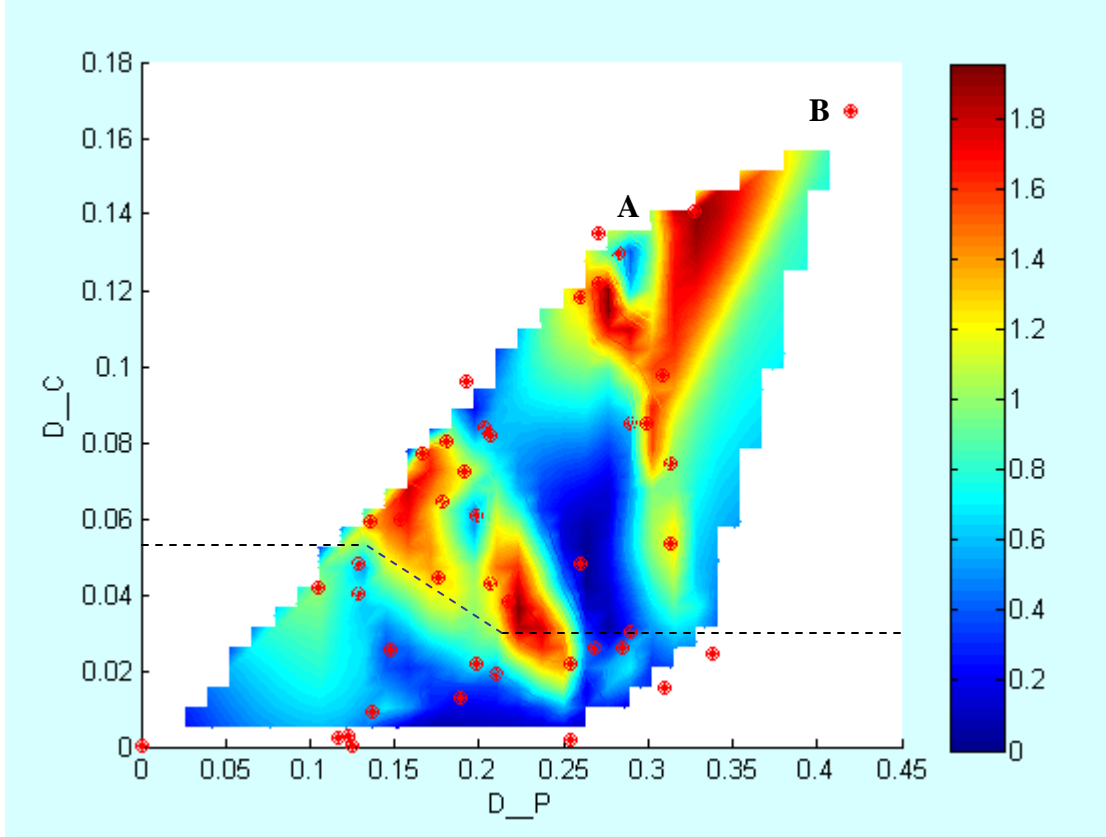


Figure 5-14. Scheme of new data point produced by random selected pair of existing points with random weight

The vector V_{child} poses 9 components corresponding 9 independent variables as described in the experimental design.

Using the produced child vector, V_{child} we obtain calculated response of the temperature field and the outlet temperature using the derived ROM. On the other hand, for comparing the results of the ROM, we convert V_{child} into the operation and boundary conditions of the FLUENT model of the combustor. Running FLUENT, we obtain a temperature field and the outlet temperature accordingly.

45 random test points are implemented and investigated. Comparing the temperature field obtained by ROM and FLUENT qualitatively by plotting them with filled and unfilled contours, Not too surprisingly, we find that the temperature fields are approximated with less error when the random point is in the proximity of a LHS design point.



A quantitative analysis was necessary to understand the performance of the ROM at non-designed data points. As the fields are too complicated to do this, we focus on the values of the outlet temperatures, in other words, we analyze the differences between T_{true}^{out} and T_{ROM}^{out} , where T_{true}^{out} and T_{ROM}^{out} denote the outlet temperatures integrated by FLUENT and calculated by the ROM, respectively. The results are encouraging. At all 45 test points, the errors of the outlet temperatures calculated by the ROM do not exceed 3%, i.e.:

$$err(k) = \frac{|T_{ROM}^{out}(k) - T_{true}^{out}(k)|}{T_{ROM}^{out}(k)} \leq 3\%, \quad k = 1, 45 \quad (5-14)$$

In order to discover the relationship of the errors to the characteristics of V_{child} and its parent vectors V_1 and V_2 , the errors of outlet temperature are shown as a contour plot in the coordinate system of D_P (X-coordinate) vs. D_C (Y-coordinate) in Figure 5-15, where the red dots represent the data at the 45 test points, D_C and D_P are defined as:

$$D_C = \min(\|V_{child} - V_1\|, \|V_{child} - V_2\|),$$

$$D_P = \|V_1 - V_2\|$$

As depicted in the Figure 5-15, it is evident that in the cases where either D_C or D_P are small, the errors are also smaller (the area below the dashed line). Otherwise the errors are larger. But there are some exceptions, for example cases A and B, as shown in Figure 5-15. Even though neither D_P nor D_C is small, the errors are smaller (located in the valley of the contour). One possible explanation is that there must have been a designed point in its neighborhood. This was indeed verified by exploring this area and finding a design point that is closer to the test point than its parent point.

5.2.7 Remarks

Having evaluated the developed ROM, we state the following two conclusions:

- At a designed point, the ROM can reproduce either snapshot without significant difference or outlet temperature with an error of less than 10^{-6} K;
- At random points in the input space, the approximation accuracy heavily depends on the distance between the point and the closest existing designed point, i.e. the closer the test point is to the neighborhood of the designed point, the less the error there is.

Further investigation leads us to discover some characteristics of given design data and the database obtained from FLUENT solutions, which provide further insight into the ROM evaluation.

5.2.7.1 Flame shapes categorization

From an inspection of the contours of the 128 cases, we find the shapes of the flames range widely, and we have categorized them into three groups according to the flame shapes in terms of

- Width
- Length
- Angle
- Temperature

The typical flame shapes are shown in Figure 5-16, all of the 128 flame portraits are given in Appendix.

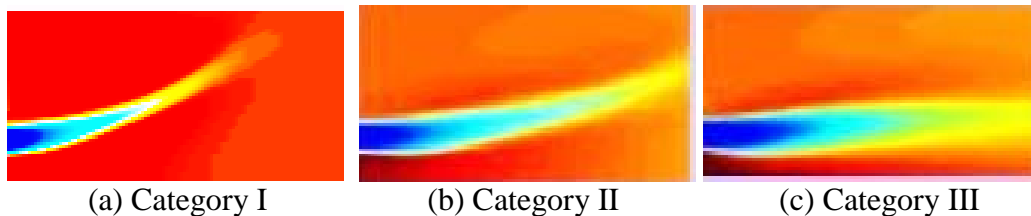


Figure 5-16. Typical flames in the three categories

Generally, the flame in the category I is narrow, shorter, and about 45 degrees with respect to the horizon. Also, temperatures of the field are more uniform than in Category III, which have wider, longer and flatter temperature gradients in the field. Category II is in between Categories I and III.

We do not investigate these categories in detail. Roughly speaking, according to the work at NETL, the feasible operating window for the combustor in the flowsheet of IGCC falls in Category I which was proved to be “*stable*” combustion. The number of cases in category I among the 128 cases is 42, a third of the 128. We observed that the shapes of the flames, i.e. the combustion statuses are determined mainly by ratio of the oxidant to the fuel and the temperature of input. This phenomenon implies that the ranges of the ratio and temperature in our experimental design are overly large. This reduces the density of the designed points of feasible operating window in the input space. Consequently, the accuracy of the developed ROM based on the experimental design for the operating window is sacrificed.

5.2.7.2 Visualization of data scatter

In order to investigate how the data in the database scatter in the PC space is spanned by the principal components (PC's), we invoke the technique of visualization in data analysis. This should indicate that, even though we use LHS to design the inputs for the CFD simulation, we cannot guarantee that the resulting database of the state (e.g., temperature) field, would distribute evenly in the PC space, because of higher nonlinearity of mapping from input to the state field through the CFD model in FLUENT.

The most significant variances of the database are in the first four principal components revealed by PCA, as depicted in Figure 5-17. Consequently, we are interested in the scatter of data along the first four coordinates in the PC space.

It is interesting to visualize the spectrum of the PC's, which reveals the recognizable patterns given in Figure 5-18. All three PC's show the wave structures with variant but structural envelope lines. The undergone state field responding to any input within the input range can be expressed as linear combination of such waves, which is similar to the Fourier approximation.

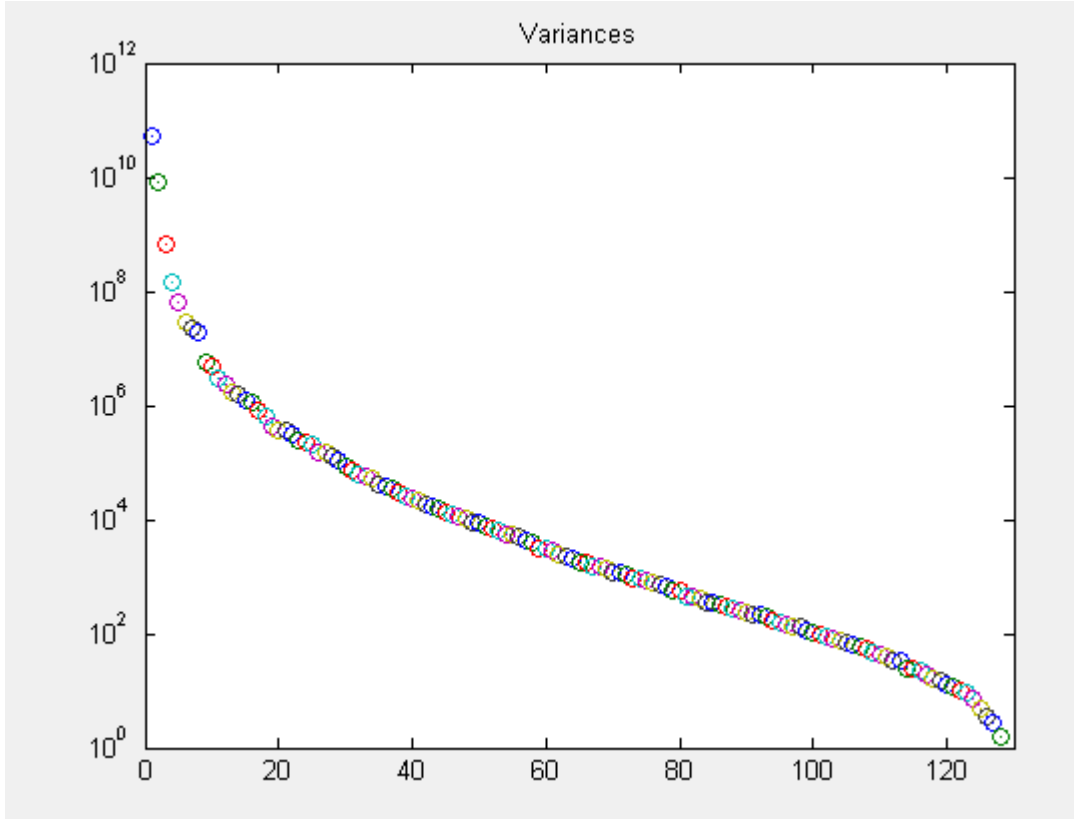


Figure 5-17. Variance profile in PC's

We use the concept of projection to plot the data scatter on the planes formed by two of the first four principal components respectively, i.e. $X_T^T \Psi_1$ vs. $X_T^T \Psi_2$, $X_T^T \Psi_1$ vs. $X_T^T \Psi_3$, $X_T^T \Psi_2$ vs. $X_T^T \Psi_3$ and $X_T^T \Psi_4$ vs. $X_T^T \Psi_4$. Here X_T is the database matrix, whose columns constitute a snapshot of the temperature field, obtained by running FLUENT corresponding to one point in the input space. Also, Ψ_i , $i=1,2,3,4$ are the first four columns of the matrix Ψ ; these are the principal components obtained by SVD and PCA which span the Principal Component (PC) Space. It is shown that the projections are easily extracted with orthogonality of the PC's, i.e.

$$X_T^T \Psi_i = (W_r \Sigma_r V_r^T)^T \Psi_i = \alpha^T \Psi^T \Psi_i = \alpha^T e_i \quad (5-15)$$

where $e_i = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ \dots \ 0]^T$ is the unit vector with the same dimension as the rank (35). The resulting plots are given in Figure 5-19.

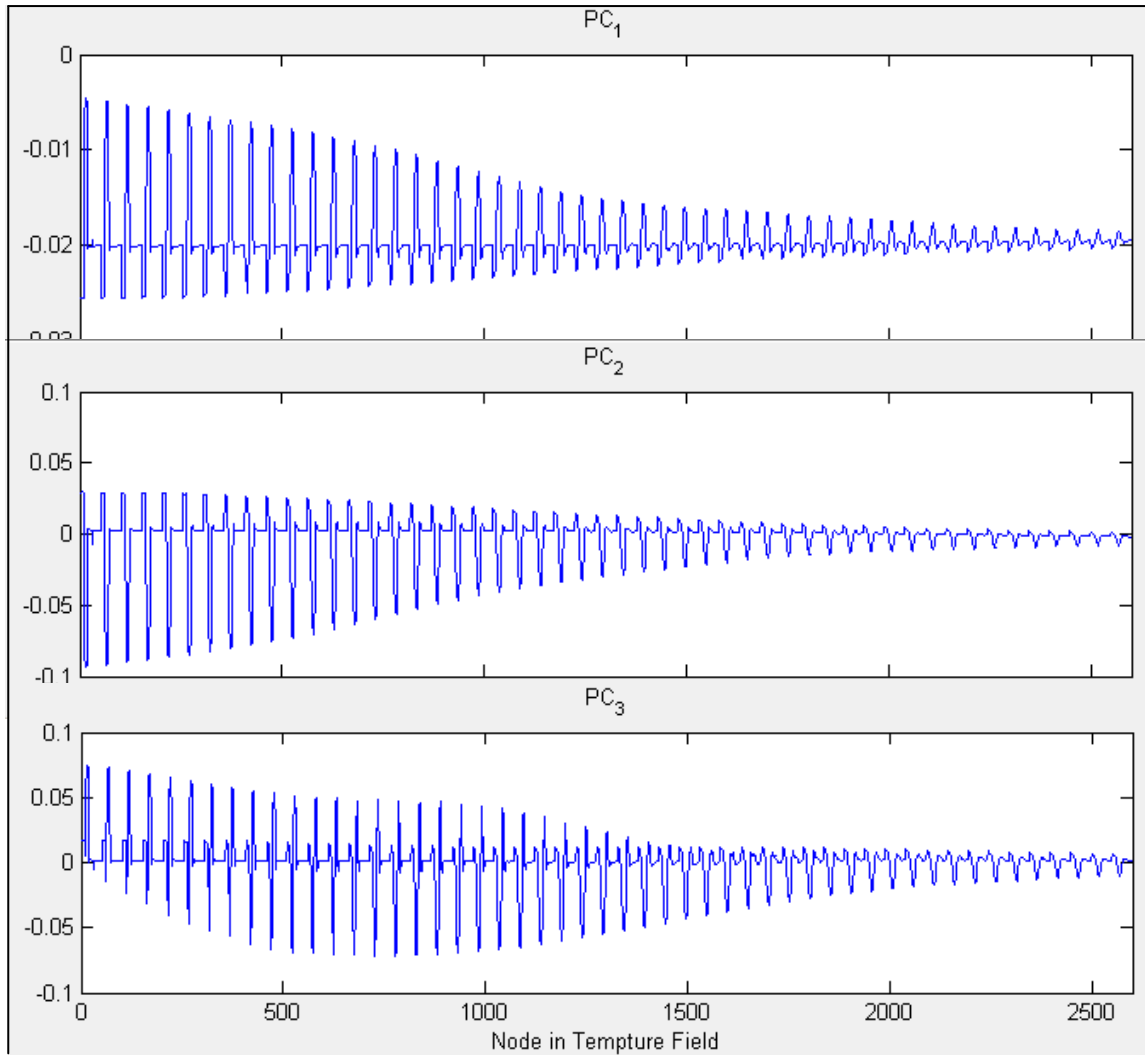


Figure 5-18. *Spectrum of the first three PC's*

In Figure 5-19, we observe that the sampling points used in our experimental design lead to output data that scatter widely in the PC space. This phenomenon implies that the mapping from input space into temperature field of the combustor through the FLUENT model is highly nonlinear. In other words, well designed inputs do not produce an even distribution in the PC space. It is worth noting that this is essential for PCA-based ROM development. In general, experimental design is used to develop model between sampling inputs and resulted data points directly, and a set of well-designed sampling points should lead to good coverage of the respective model. On the other hand, in this investigation, the resulting database requires a more even distribution than the one used here. Good distributions of the database in the PC space are vital within the valid range of the ROM.

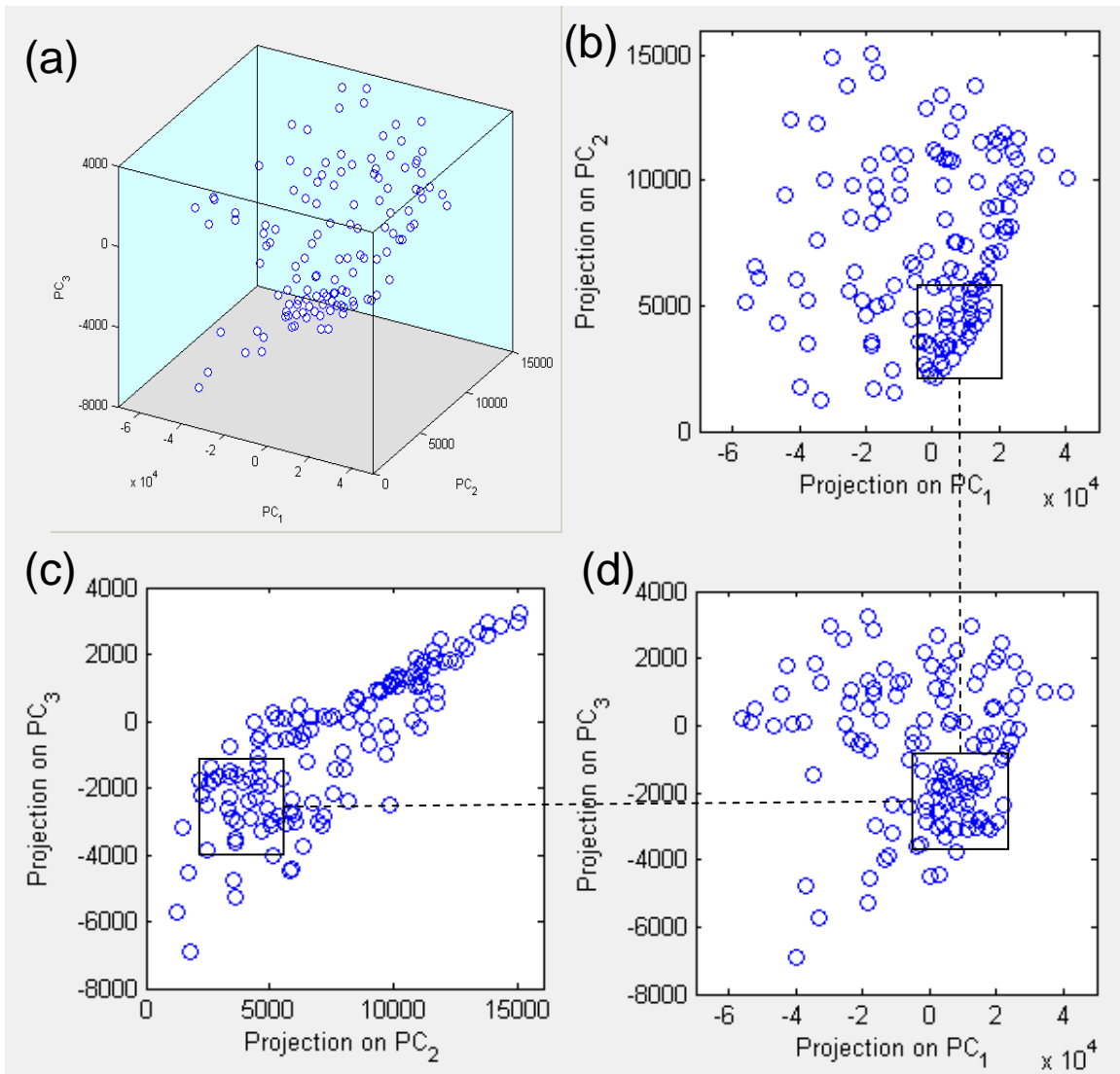


Figure 5-19. Visualized data scatter plots. Temperature field produced by FLUENT using inputs designed with LHS are illustrated in (a) 3-D space spanned by the first three PC's; (b) a projection scatter plot on the plane scanned by PC1 and PC2. (c) a projection scatter plot on the plane scanned by PC1 and PC3. (d) a projection scatter plot on the plane scanned by PC2 and PC3

5.2.7.3 Flame shape categories vs. data scatter

It is interesting to note that when we colored the individual data points to identify their categories illustrated in Figure 5-20, we found that almost all the data points belonging to Category I (red) fall into the box, where they concentrate to form a higher density cluster or 'data cloud'. The data points belonging to Category III (dark) spread sparsely in other areas of the PC space, while the data points of Category II (green) lie somewhere in-between. This is consistent with the criterion of the case categorization based on the roughly observed characteristics of the flames in the 128 cases. This is an unexpected, but quite favorable result.

Researchers at NETL have indicated that the combustor should be operated under stable conditions that can produce flames as in Category I. By invoking a concept of “operating window”, we can emphasize that the ranges of design input variables must fall in the operating window to ensure feasible operation of the combustor; otherwise the operation of the combustor is infeasible from an engineering point of view.

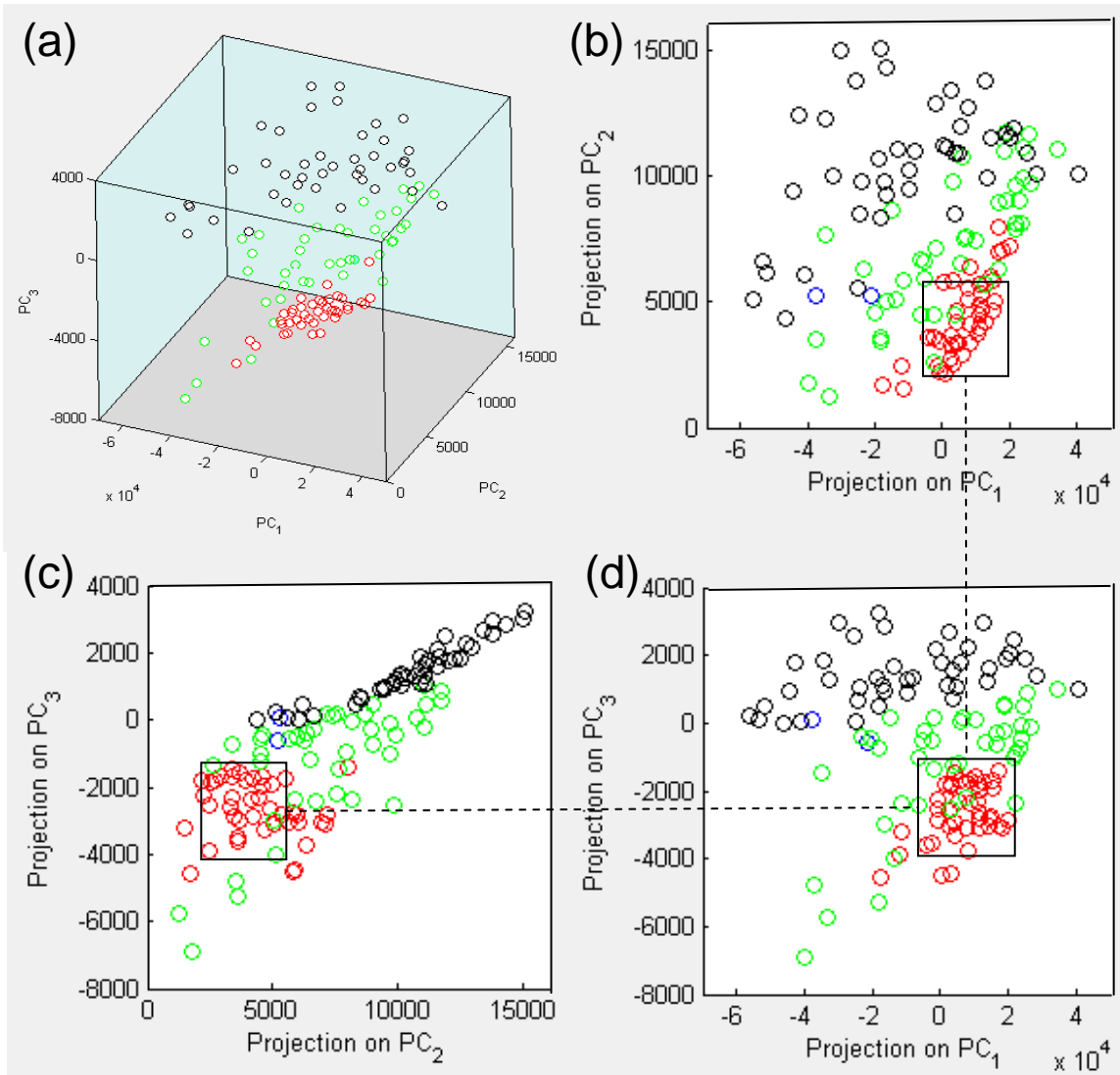


Figure 5-20. Projection scatter of the three categories of data in the planes as defined in Figure 5-19

It is obvious that the ranges of the independent design variables we used (according to that offered by FLUENT) for experimental design is not restricted inside the operation window, therefore we could not ensure that the cases produced by LHS are all corresponding to the feasible operations of the combustor and the produced flames are all stable like in category I. For example, the molar ratio of oxidant to the fuel is from 0.98

to 5.5, the inlet temperature is from 300K to 1000 K; this is evidently too wide and splits the resulting flame into three categories.

To make it clearer, we delete the data points of Category II in the scatter as in Figure 5-21; now it is obvious that the two extremes, category I and category III are distinguished significantly. This implies that the flame shapes or temperature profiles in the flow field can be detected easily by their projections to the first few principal components. This visualization of the data scatter shows that the data points colored in dark and green are sparse. This also explains why there are some points, mentioned in section 5.2.6, at which the developed ROM could not approximate the FLUENT model well, as there are no other existing points in their near neighborhoods.

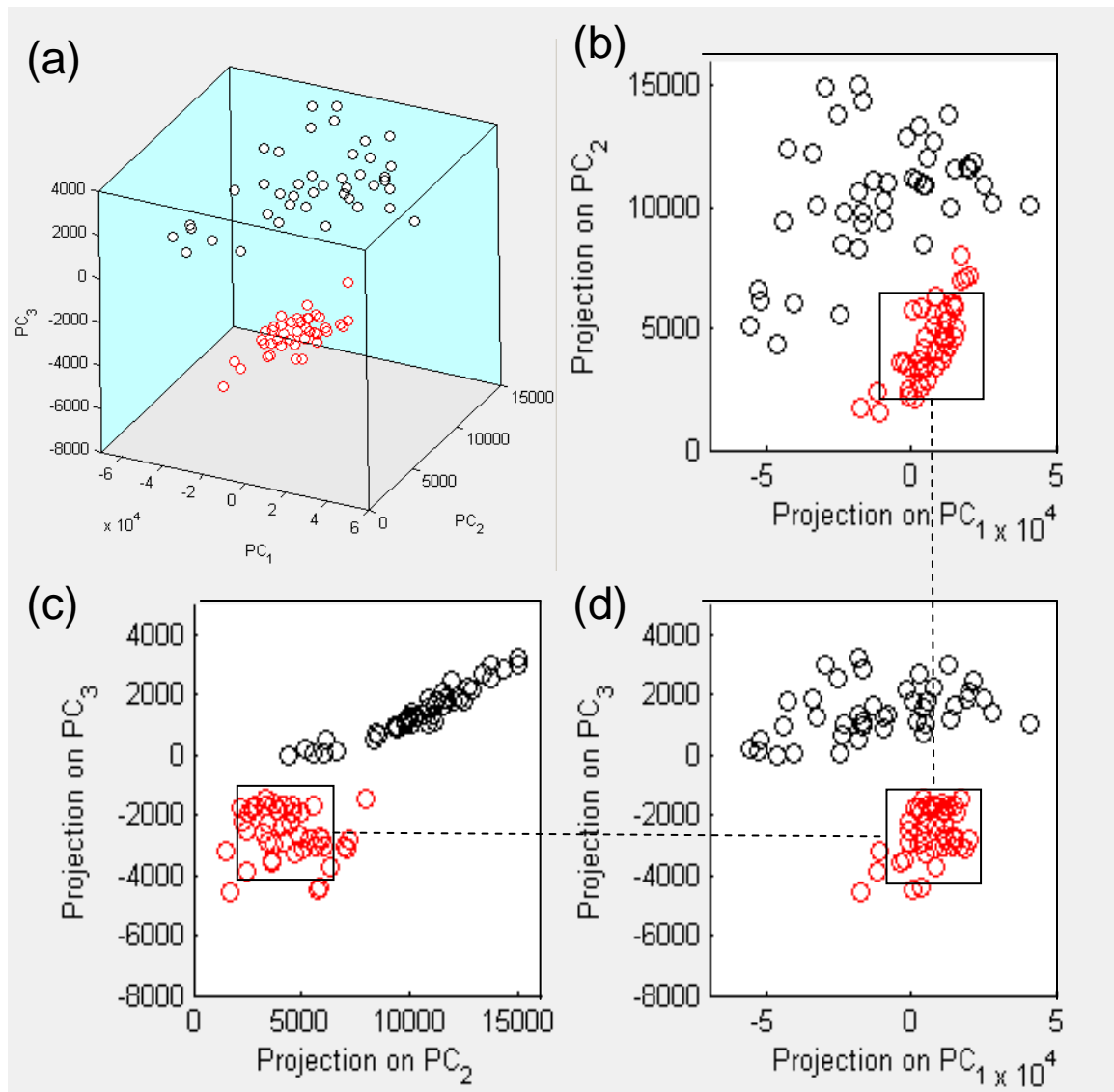


Figure 5-21. Category I and Category III scatters distinguish significantly

From the data scatter projected into the plane formed by PC3 and PC4, we found that the data scatter forms three data clouds corresponding to the categories; these are located in three distinct areas in the plane. This implies that the ranges of the design variables cover too many operating categories, which is undesired for ROM development.

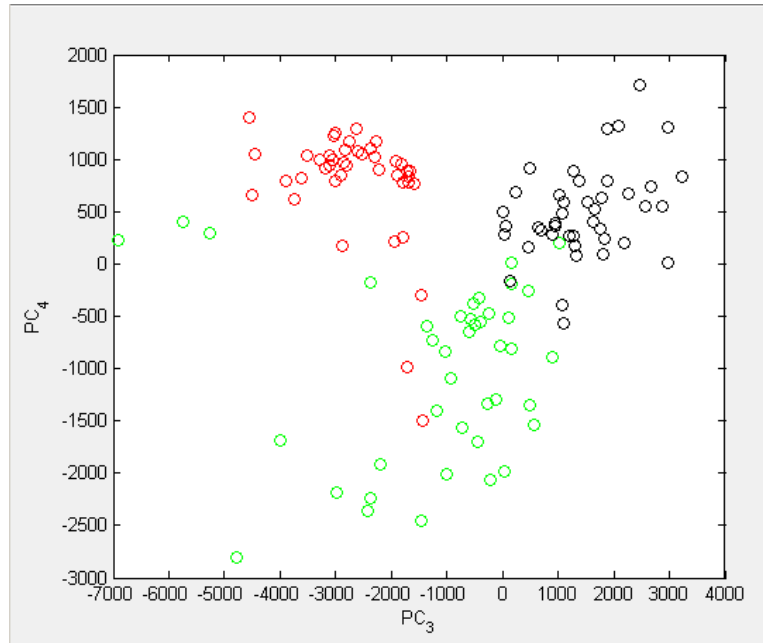


Figure 5-22. Projection scatter of the three categories of flames in the plane formed by PC₃ and PC₄ where the three data clouds concentrate in three distinguished areas.

5.2.8 Parametric ROM

Adopting the concepts of Parametric Linear Programming, we propose the concept of “parametric ROM” based on our observations of this case study and the test and evaluation of the ROM of the combustor.

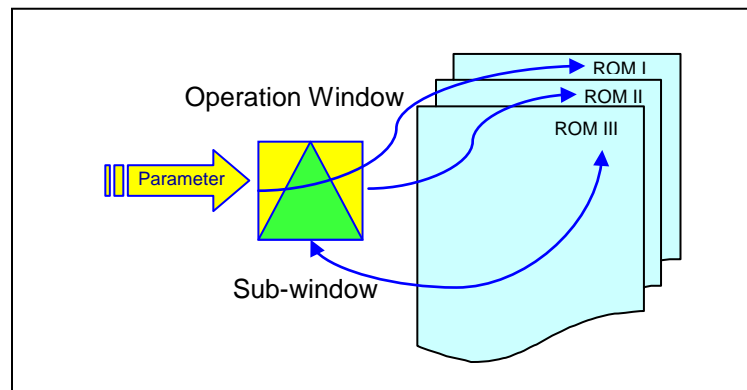
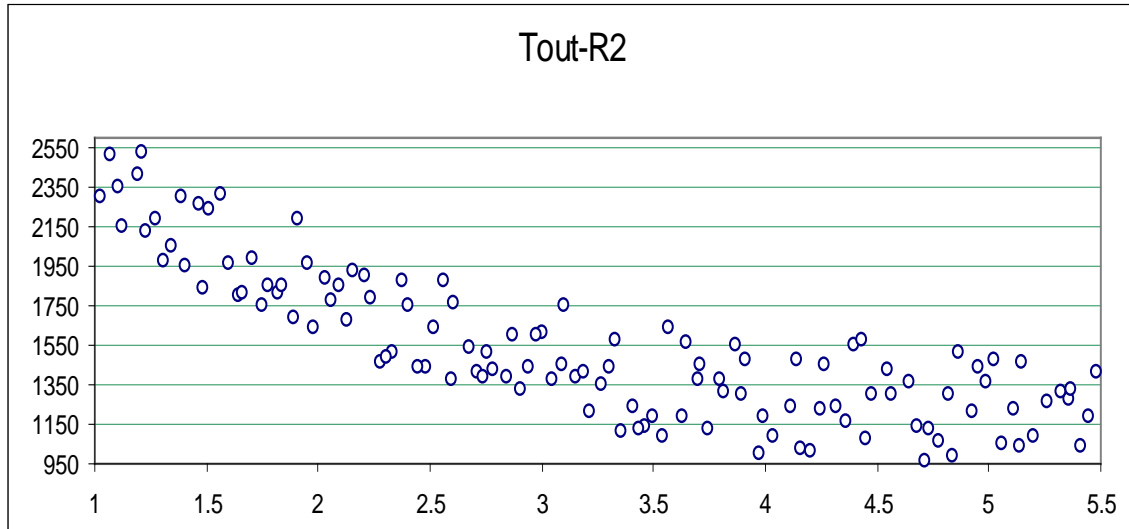


Figure 5-23. Scheme of parametric ROM

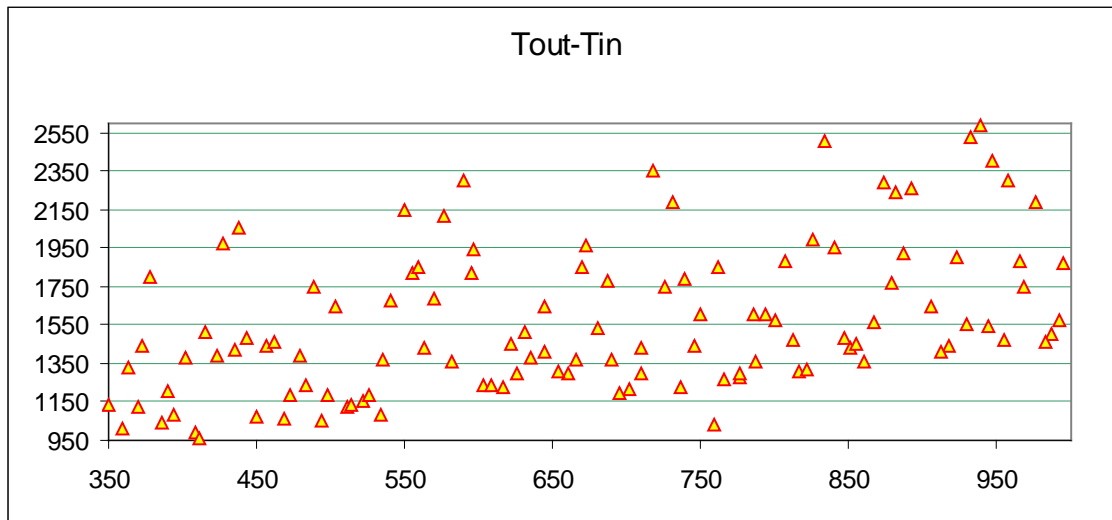
According to our observation of the flames, we can classify the data points into three equal groups with 42 points in each category (two extreme points were excluded). The visualization of data scatter reveals that if we had used three ROMs corresponding to three categories of the flame shapes instead of one ROM, we should get better results for testing and evaluation. If we could find a parameter for each ROM relating to the individual operation window producing the specific category of the flame shapes, then we could select the ROM corresponding to this parameter. This is illustrated in Figure 5-23

Also it is obvious that if we had intended specifically to develop the ROM of the combustor for the co-simulation of FutureGen power plant only, we would not consider categories II and III. Instead, we would concentrate the inputs in the range where the combustor operations are feasible.

Investigating the correlation of outlet temperature with nine design input variables, we found that the outlet temperature correlates strongly with the molar ratio of oxygen to the fuel and weakly with the inlet temperature. These two correlations are illustrated in Fig 5-24. Also, we did not find significant correlation between the outlet temperature and other seven design input variables. From Fig 5-24 (a), we can estimate that if we intend to develop a ROM valid in the operation window where its outlet temperature is lower than 1820 K (Zitney et al. (2005)) then the ratio of oxygen to the fuel should be greater than two. In that case, there are about thirty data points in our database that are unrelated. Further from Fig 5-24 (b) the inlet temperature should be higher than about 550 K which is consistent with the concept of ignition in the combustor. Therefore, certain data points in the database corresponding to the inlet temperature lower than 550 K decrease the efficiency of the database.



(a)



(b)

Figure 5-24. Correlation of the outlet temperature with (a) the molar ratio of oxygen to fuel and (b) the temperature of inlet stream.

Thus, we need to be careful to design the operating window *a priori* according to the engineering knowledge related to the combustor operation. In that case, we would have concentrated on the data points in the database that fall into the box shown in Figure 5-19. We believe that this would lead to an efficient database for developing the ROM represented by a specific parameter.

The test and evaluation of the developed ROM reveals an important fact that the accuracy of the ROM depends strongly on the density of data points in the input space, determined by the experimental design. The accuracy for the state approximation can be improved further by:

- a. Reducing the dimension of the input space;
- b. Narrowing the ranges of the input variables;
- c. Increasing number of datasets;
- d. Exploring more sophisticated techniques of experimental design.

5.2.9 Summary of case study II

This case study is based on reduced order modeling of a key process unit, the gas turbine combustor in the FutureGen flowsheet. Here we show that a ROM methodology based on PCA instead of POD is simple, effective and efficient. It removes the burden of handling PDE equations directly and takes advantage of well-developed-and-tested CFD software, e.g. FLUENT. Combined with advanced techniques of experimental design, the developed ROM can achieve reasonable accuracy with relatively small databases. On the other hand, for multi-input cases, the number of cases and the input variables for each case should be carefully designed. The case study also shows that the accuracy of ROM approximation at experimental points is very high. Moreover, for unknown points in the input space, we can achieve acceptable accuracy for the ROM as long as the density of the known points is large enough and well-distributed.

It is worth noting that there is significant CPU time reduction due to replacing the CFD model with the ROM. This is especially true for some CFD models that need several hours to converge, as ROM needs only a few CPU seconds to give the approximate solution.

Finally, it is obvious that the PCA based methodology of developing ROMs for CFD models will accelerate process simulation in APECS significantly. Replacing the CFD model with its ROM, derived from the methodology developed in this work, APECS will be more efficient and more effective.

6. Conclusions

After a thorough investigation with development of POD-based ROM, we propose a new algorithm for PCA-based ROM. The theoretical development and two Fluent-based case studies showed that the proposed methodology, implemented in MATLAB, is both effective and efficient. The method can bring into play the advantages of well-developed facilities used in Fluent. There is no need to handle the complicated procedures of converting PDE's to ODEs as required in POD-based ROM.

Since the method introduces concepts of state space that are popular in control theory, the method overcomes any restrictions due to geometry and high dimensional fields.

Also, we note that the PCA-based ROM is derived from a set of inputs and the PCs do not depend on any individual inputs. As long as the input is in the required domain, the validity of the ROM is ensured and updates of the PCs are not necessary. This is critical to integrate the ROM with process simulation and optimization of the overall flowsheet.

For broader input ranges it is strongly recommended that parametric ROMs should be developed, instead of a single ROM for the entire range.

Significant CPU time reduction due to replacing the CFD model with the ROM is obtained, as noted in the case study.

In summary, the ROM methodology and its resulting models developed in this project have the following features:

- Easy to apply
- Fewer restrictions
- Acceptable accuracy
- Wider validity
- Much faster

Our tasks in this project have been successfully implemented with a series of results.

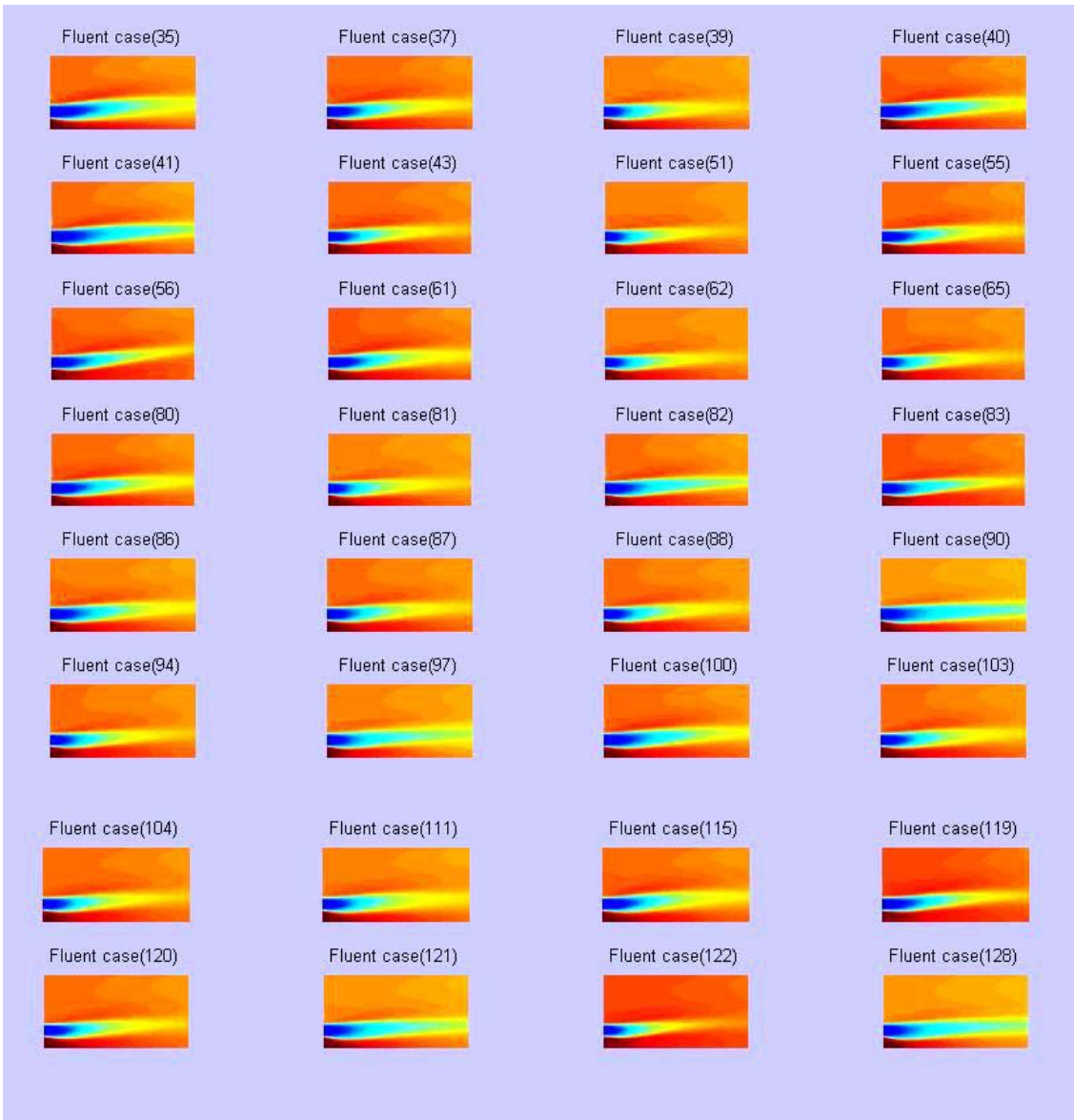
Acknowledgement:

This work was done with the support of the U.S. Department of Energy, under Award No. DE-PS26-04NT42249. However, any opinions, findings, conclusions, or recommendations expressed herein are those of the author(s) and do not necessarily reflect the views of the DOE. The authors express appreciation to the DOE project managers, Ronald W. Breault and Steve E. Zitney. The authors gratefully acknowledge the contributions of the following colleagues: Jens I. Madsen, Sorin Monteanu, Maxwell O. Osawe, Erik Ferguson and Lewis Collins for offering Fluent cases and other helpful information, advice and suggestions. Also, the authors are thankful to Shivakumar Kameswaran for his helpful discussions.

Appendix: Categories of the Flames







References

- Cazemier, W., *Proper Orthogonal Decomposition and Low Dimensional Models for Turbulent Flow*, (1968) Thesis
- Cizmas, P.G., A. Palacios, T. O'Brien, M. Syamlal, *Proper-orthogonal Decomposition of Spatio-temporal Patterns in Fluidized Beds*, Chem. Engn. Sci. 58 (2003), pp4417 - 4427
- Fahl, M., *Trust-region Methods for Flow Control Based on Reduced Order Modeling*, (2000), Thesis
- Golub, G.H. and C.F. Van Loan, *Matrix computations*, John Hopkins University Press, Baltimore, 1989
- Kragel, B., *Streamline diffusion POD Model In Optimization*, (2005) Thesis
- Kunisch, K and S. Volkwein, Control of the Burgers Equation by a Reduced-Order Approach Using Proper Orthogonal Decomposition, JOTA, 102, 2 (1999)
- Jiang, Qing, Principal Component Analysis and Neural Network Based Face Recognition, <http://people.cs.uchicago.edu/~qingj/ThesisHtml/>
- Jolliffe, I.T., *Principal Component Analysis*, Springer, 2002
- Moehils, J., T.R. Smith, P. Holmes and H. Faisst, *Models for Turbulent Plane Couette Flow Using the Proper Orthogonal Decomposition*, Physics of Fluid, 14,7 (2003)
- Morari, Manfred, Distinguished ECE Lecture: *Control of Hybrid Systems: Theory, Computation and Applications*
http://www.ece.cmu.edu/news/seminar/2007/fall/morari_11_15_07.pdf, (2007)
- Osawe, M., P. Felix, M. Syamlal, I. Lapshin, K. Joc Cleetus and S.E. Zitney, *An integrated process simulation and CFD environment using the CAPE-OPEN interface Specifications*, AIChE meeting, paper 250c, 2002
- Schlegel, M., J. Berg, W. Marquardt, O.H. Bosgra, *Projection Based Nonlinear Model Reduction for Dynamic Optimization*, Technical Report LPT-2002-22 (2002)
- Smith, T.R., J. Moehlis and P. Holmes, *Low-dimensional Model for Turbulent Plane Couette Flow in a Minimal Flow Unit*, J. Fluid Mech. 538 (2005), pp 71-110.
- Sloan, D.G., W.A. Fiveland, M.O. Osawe, S.E. Zitney and M.Syamlal, *Demonstration of coupled Cycle and CFD Simulations over a LAN*, Clearwater Coal Conference, 2005

Willcox, K.E., *Reduced-Order Aerodynamic Models for Aeroelastic control of Turbomachines*, (2000) MIT thesis.

Willcox, K., *Model Reduction for Large-Scale Applications in Computational Fluid Dynamics*,

Zitney, S.E., *CAPE-OPEN Integration of CFD and Process Simulation for Advanced Power Generation Applications*, CAPE-OPEN meeting, 2004

Software Framework for
Advanced Power Plant Simulations

TASK 5.2-A
DEMONSTRATION OF GEOMETRIC SCALING
IN AN INTEGRATED SIMULATION

Topical Report

Reporting Period Start Date: November 1, 2006
Reporting Period End Date: September 11, 2008

David Sloan, Ph.D.
Woodrow Fiveland, Ph.D.

November 30, 2009

DOE Cooperative Agreement No: DE-FC26-05NT42443

Alstom Power Inc.
US Power Plant Laboratories
P.O. Box 500, CEP 9005-1204
2000 Day Hill Road
Windsor, CT 06095

UNDER SUBCONTRACT TO
ANSYS Inc.
10 Cavendish Court, Lebanon, NH 03766

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ACKNOWLEDGEMENT STATEMENT

This document was prepared with the support of the U.S. Department of Energy, under Award No. DE-FC26-05NT42443. However, any opinions, findings, conclusions, or recommendations expressed herein are those of the author(s) and do not necessarily reflect the views of the DOE.

ACKNOWLEDGEMENTS

Financial support for this project was provided by DOE National Energy Technology Laboratory (Cooperative Agreement No. DE-FC26-05NT42443) and Alstom Power Inc. Appreciation is expressed to Ronald Breault as the DOE Project Manager and to Stephen E. Zitney (NETL) for his technical assistance and direction on the entrained flow gasifier and process model. The contributions of the many APECS project team members from ANSYS Inc. and Aspen Technology Inc. are also greatly appreciated. ANSYS Inc. is particularly acknowledged for its leadership role as the prime contractor under John Widmann. Gratitude is expressed to Sorin Munteanu, John Widmann, and others on the APECS team at ANSYS Inc. for their diligence in striving to deliver quality software.

Within the Alstom Power divisions, appreciation is expressed to a number of individuals. John L. Marion, the Alstom Power Program Manager, and Woodrow A. Fiveland, the Alstom Power Project Manager, contributed to the success of this project task.

ABSTRACT

The primary objective of this DOE project is to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of Advanced Process Engineering Co-Simulator (APECS), the simulator will link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. Generally, the simulator couples the cycle analysis software Aspen Plus® (marketed by Aspen Technology, Inc.), specialized component modeling packages, as exemplified by industrial proprietary codes (utilized by ALSTOM Power Inc.), and the FLUENT® computational fluid dynamics (CFD) code (provided by ANSYS Inc). An important enhancement to the APECS tool will be the creation of computationally efficient reduced-order models (ROM) based on information from high-fidelity computational fluid dynamics (CFD) models. Integration of the steady-state APECS tool with a virtual reality tool kit will allow process designers to navigate through a 3D virtual representation of a power plant with visualization of simulation results. A related secondary objective is to develop a prototype dynamic simulator within APECS for advanced power plants and to demonstrate a dynamic simulation that integrates plant-level and equipment-level models.

Among other responsibilities, Alstom Power has the responsibility to demonstrate the capabilities of the enhanced APECS tool to simulate proposed FutureGen power plant concepts and other relevant power generation processes and equipment. The present report encompasses only the workscope within Task 5.2a -- specifically, Alstom Power is tasked with the responsibility to demonstrate an application involving geometric parameterization and scaling. (A separate task, Addendum Task 5.2b, added later in the project, will be the subject of a different topical report.)

A demonstration case was completed in which a CFD model of an entrained flow gasifier was coupled with a FutureGen flowsheet and geometrically scaled as a function of coal throughput in order to meet a process model operational target. This report documents the case characteristics and results and discusses the viability and capabilities of the linkage/interface software for geometric scaling applications. It also discusses the challenges associated with performing an optimization study with stochastic models.

TABLE OF CONTENTS

1.0	BACKGROUND ON APECS PROGRAM	2
2.0	INTRODUCTION	2
3.0	EXECUTIVE SUMMARY	4
4.0	EXPERIMENTAL WORK.....	6
5.0	RESULTS AND DISCUSSION.....	6
5.1	Case Setup and Preparation.....	6
5.1.1	File Background and Case Preparation.....	6
5.1.2	Design and Operational Considerations.....	9
5.1.3	Scale-Up Approach	11
5.1.4	Parameterization and the Associated CAPE-OPEN Variables	12
5.1.5	Ports and Connecting Streams.....	18
5.1.6	Files and Software for the Integrated Co-Simulation.....	21
5.2	Computational Results.....	22
5.2.1	Optimization Space for CGE Variable.....	22
5.2.2	Solution Strategy and Results	25
6.0	CONCLUSIONS AND RECOMMENDATIONS	33
7.0	REFERENCES	35
8.0	DELIVERABLES.....	36

LIST OF FIGURES

Figure 1:	Flowsheet from NETL Gasification “Demo” Provided at 1 st IAB Meeting	7
Figure 2:	External Surface Grid for Baseline Case.....	8
Figure 3:	Temperature Contours (in Units of K) at Z=0 Plane for Baseline Case.	9
Figure 4:	Sketch of Gasifier Showing Parameterization.....	13
Figure 5:	CGE as a Function of Overall Stoichiometry for 3 Size Ratios.....	23
Figure 6:	Outlet Temperature as a Function of Overall Stoichiometry for 3 Size Ratios.	24
Figure 7:	Carbon Burnout as a Function of Overall Stoichiometry for 3 Size Ratios.	24
Figure 8:	CGE as a Function of Overall Stoichiometry and Iteration Count.....	26
Figure 9:	CGE as a Function of Overall Stoichiometry and Iteration Count.....	27
Figure 10:	CGE Convergence Characteristics as a Function of iteration Count	28
Figure 11:	CGE Convergence Characteristics as a Function of iteration Count (Enlarged).	29
Figure 12:	Temperature Convergence Characteristics as a Function of iteration (Enlarged).....	30
Figure 13:	Temperature Convergence Characteristics as a Function of iteration (Enlarged).....	32

LIST OF TABLES

Table 1:	GAMBIT® Journal File and Geometry Parameters.....	15
Table 2:	CAPE-OPEN Parameters As Implemented Through APECS Wizard.....	16
Table 3:	FLUENT® Block Ports and Connecting Streams.....	18
Table 4:	Species Listing and Mapping.....	20
Table 5:	Inputs for Parametric Runs	23

1.0 BACKGROUND ON APECS PROGRAM

The current APECS program is a continuation of an earlier (Vision 21) program, which was initiated under the DOE-Fluent Cooperative Agreement No. DC-FC26-00NT40954 (hereafter referred to as Phase I). Both programs have been dedicated to the development of the software infrastructure for advanced power plant simulations. The goal of the overall program work scope has been to develop an integrated suite of software tools that could be used to simulate and visualize advanced plant concepts that are being considered under the umbrella of DOE's broad CO₂ remediation and power plant efficiency programs. Advanced simulation tools are needed to evaluate new power plant concepts that will undoubtedly help to minimize costly laboratory and field trials.

The typical process simulation software that has been commercially available to industry does not meet the DOE's objective of "virtual simulation" that is needed to evaluate complex cycles. The intent of the DOE has been to improve predictive tools for cycle analysis, and to improve the component models that are used in turn to simulate the cycle. Generally, the modeled performance of various components, e.g., a gasifier, is derived from a simple set of reaction and output prescriptions that are collectively accessed by the cycle simulation software as a "black-box" library module, without any explicit impact from geometry modifications and enhancements to the fluid mechanics. These simple component models are not sufficient for component design; nor are they adequate for the evaluation of the impact of complex components on cycle performance. To meet the DOE's goal of predicting and visualizing the performance of these complex systems, there has been a need to use more sophisticated component models, by upgrading or converting library modules to CFD modules, and to couple the CFD modules directly with other equipment items in the power plant through process modelling software. The primary focus of the present APECS project has been to provide the software interface and environment that permits FLUENT® or a surrogate ROM to run in tandem with the Aspen Plus® process simulation package (see Ref. 1 for an overview of the program).

2.0 INTRODUCTION

The feasibility of using FLUENT® CFD models in concert with a process model like Aspen Plus® has been a focus of the APECS project. The role of Alstom Power Inc. (Alstom) has been to assist the project team in helping to develop and demonstrate the capabilities of the advanced APECS simulation and visualization tool, oftentimes with the use of Alstom designed and validated equipment modules (see e.g., Ref. 2 through 4). The primary Alstom responsibilities and tasks have included:

- providing its expertise and experience base in the utilization of both CFD and cycle analysis for the power generation industry,
- selecting and running cases to test and demonstrate the feasibility of the concepts,
- forming an advisory board to provide project review and feedback.

In Task 5.2a of the current APECS project, as specified in the Project Management Plan (Ref. 5), Alstom is tasked with the responsibility of geometrically parameterizing and scaling a module within a coupled simulation. The individual subtasks are to:

- 5.2.1 Develop a CFD equipment model with automated geometry control, remeshing, and interpolation of the old solution.
- 5.2.2 Demonstrate geometric scaling in an integrated simulation.
- 5.2.3 Utilize design specifications, sensitivity analysis, and/or optimization routines, as required, within Aspen Plus® to exercise the parameterized component within a process model.

One of the milestones, also stipulated in the Project Management Plan (Ref. 5), was to provide an “interim task report” for Task 5.2, which contains a description of the case simulations, the results, and an itemization of desired software modifications for future work. This report summarizes and documents the demonstration case simulations under Task 5.2a, and thus completes the requirement for the stipulated (contractual) “interim” task report. In essence, however, this report also constitutes a final topical report for the task.

The NETL was consulted in the selection of the equipment item for the geometrically scalable module from a list of candidate equipment items. The NETL (in the person of Steve Zitney) recommended an entrained flow gasifier. This choice was appropriate and convenient since:

- An Aspen Plus® module for an entrained flow gasifier had already been developed through a prior ANSYS/DOE collaborative effort. ANSYS had developed a user-defined function (UDF) for coal-gasification chemistry, as well as a corresponding entrained flow gasifier CFD case.
- The FLUENT® CFD model had been sized and calibrated to match the inputs and outputs of an existing FutureGen process model, developed at NETL. The coupled simulation involving the entrained flow gasifier had been used by NETL as an early demonstration case, showcasing various aspects of the APECS technology. Both the process model and CFD case, used in the coupled demonstrations, were provided to Alstom for use in the present project.

The gasifier case involves reacting coal particles and therefore requires the Lagrangian discrete particle model (DPM) in FLUENT®. Because of the inherent CPU burden associated with the DPM, and because the CFD case and coupled simulation would be exercised on a single processor, the CFD case had been understandably reduced in size (by NETL/ANSYS) to about 12,256 cells. The coarseness of the FLUENT® mesh for the entrained flow gasifier was helpful in promoting quick turnaround times for debugging and testing purposes, although some inherent risk is associated with coarse grids because of the possibility of a local convergence instability being induced in some regions of the computational domain where flow features are not adequately resolved or where cells have poor aspect ratios.

All of the runs were performed with Aspen Plus® as the executive software. The “sequential-modular” solution approach was used in all cases, as opposed to the “equation-oriented” solution methodology. For the present task, the primary intent and focus was to demonstrate the viability of the controller interface and the linkage between Aspen Plus® and FLUENT®. Alstom did not attempt to refine or enhance the performance of the (pre-existing) FLUENT® model or the UDFs,

except to the extent necessary to (1) achieve a reasonable compliance with the task objectives, and (2) overcome obstacles impeding the convergence of the coupled simulation.

3.0 EXECUTIVE SUMMARY

The primary objective of this DOE project is to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of Advanced Process Engineering Co-Simulator (APECS), the simulator will link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. Generally, the simulator couples the cycle analysis software Aspen Plus® (marketed by Aspen Technology, Inc.), specialized component modeling packages, as exemplified by industrial proprietary codes (utilized by Alstom Power), and the FLUENT® computational fluid dynamics (CFD) code (provided by ANSYS Inc). An important enhancement to the APECS tool is the creation of computationally efficient reduced-order models (ROMs) based on information from high-fidelity CFD models.

In this present task, the objective was to showcase an important advancement in the APECS tool kit – the addition of geometric-scaling calculations to an integrated FLUENT® CFD block execution. ANSYS provided a GAMBIT® journal file that modified the cross-sectional geometry of the gasifier as a function of feed flow rate in order to keep average velocities and residence times from changing significantly. Alstom was tasked with the responsibility of geometrically parameterizing and scaling a module within a coupled simulation. Specifically, Alstom Power was tasked with the responsibility to (a) develop a gasifier model with automated geometry control, remeshing, and interpolation of the old solution, (b) demonstrate geometric scaling in an integrated simulation, and (c) utilize design specifications, sensitivity analysis, and/or optimization routines, as required, within Aspen Plus® to exercise the parameterized component. This was accomplished.

A demonstration case was completed in which a gasifier that scaled geometrically with coal throughput was instantiated and run on an Aspen Plus® flowsheet. The geometry for the gasifier was first parameterized, so that every geometric dimension was assigned a corresponding physical parameter. The physical parameters were then given default or baseline parameters and defined in a GAMBIT® journal file (“gasifier.jou”), developed by ANSYS. In addition, at the beginning of the GAMBIT® journal file, CAPE-OPEN geometry parameters were assigned to the corresponding physical parameters. The physical parameters were then used in the GAMBIT® journal file to build a new grid with each new mandated scale-up or alteration of the geometry. This assignment (i.e., GAMBIT® physical parameter = CAPE-OPEN geometry parameter), in the journal file, allowed the executive software, Aspen Plus®, to define how the gasifier geometry scales as a function of the coal throughput variable in the flowsheet. Other Booleans and journal files assisted in the process of building the grid, interpolating the previous solution onto the new mesh, reconstructing DPM particle model source terms, changing under-relaxation factors, and generally preparing the FLUENT® case for execution. The methodology devised by ANSYS for the geometry scale-up worked flawlessly in both PC and Linux (LAN) environments.

In the present application, the FLUENT® model for the gasifier utilized multiple UDFs developed by ANSYS for the DOE. The goal was to instantiate the model on the flowsheet and to optimize the CGE at a given coal flow rate by varying the oxygen feed (or stoichiometry). However, it was found that the FLUENT® model exhibited 3 convergence problems that made it very difficult to use with optimization routines in Aspen Plus® – (a) lengthy convergence times (tens of thousands of iterations), (b) rolling or oscillating solutions, and (c) stochastic noise about the mean solution. Unfortunately, it was found that the convergence problems associated with the FLUENT® model caused the optimization routines in Aspen Plus® to bog down and fail. Therefore, a methodology was devised in which the FLUENT® model was manually sequenced to calculate the CGE at 5 selected stoichiometries, and then the optimal value of the CGE was defined as the maximum of the 5 calculated values. This worked sufficiently well to enable the flowsheet to converge.

Significant progress has been made in facilitating the direct instantiation of FLUENT® on an Aspen Plus® flowsheet, in conjunction with geometric scale-up strategies. It is hoped that the present documented experience, as well as the associated journal files and explanation of the workflow, will help other others in their efforts to utilize the geometric scale-up option in subsequent co-simulations.

Recommendations are as follows:

- (1) The lengthy convergence times associated with the present FLUENT® case precluded a fair assessment of CFD cases coupled with the optimization routines in Aspen Plus®, and this assessment, with a better case, needs to be repeated. In future NETL-affiliated efforts, one of the demonstration or application cases should involve both the (1) direct instantiation of a CFD case with the stochastic DPM model and (2) the optimization routines in Aspen Plus®. The user should be prepared to work closely with AspenTech personnel and implement a different kind of optimization routine (perhaps with Aspen Custom Modeler®) in the event that the SQP methodology in Aspen Plus® bogs down and fails (due to convergence problems, etc.). For optimization purposes, a ROM may work better than the CFD case itself; this is also something that should be tried.
- (2) The lengthy convergence times may have been caused by the inclusion of reverse rate kinetic expressions (e.g., water-gas shift) in the mechanism listing. Therefore, the gasifier model should be tested with combustion models that utilize a stiff integrator. Although stiff integrators may induce a severe CPU penalty, the cases may converge much faster.
- (3) In geometric scale-up co-simulations, after every grid is built, there should be CAPE-OPEN variables that are reported in Aspen Plus® that reflect the quality of the grid that is produced, e.g., the number of cells, the maximum cell aspect ratio, grid skewness, etc. This will help the user to know whether the grid construction methodology should be modified to produce a more agreeable grid.
- (4) A slagging model, amenable with steady-state formulations, should be added to the CFD case, so that the proper physics associated with the wall boundary conditions (in a slagging gasifier) can be utilized in the simulation.

4.0 EXPERIMENTAL WORK

There is no experimental work under this project.

5.0 RESULTS AND DISCUSSION

5.1 Case Setup and Preparation

5.1.1 File Background and Case Preparation

As part of a prior ANSYS/DOE cooperative effort over the last few years, ANSYS had constructed various incarnations of an entrained flow gasifier CFD case, as well as the requisite user-defined functions (UDFs) for coal-gasification chemistry (see Ref. 6 and 7, based on the model developed by Wen and Chaung (Ref. 8)). Sources for some of the entrained flow gasifier CFD and process model files used to prepare the coupled simulation were as follows:

- Some FLUENT® files were sent by Jens Madsen (ANSYS) to Alstom on 14 June 2007. The FLUENT® files consisted of ent-flow-gasif.cas(dat), with the accompanying scheme file gasification.scm and the UDF files energy.c, gasif-dpm.c, and gasif-dpm.h. The FLUENT® files corresponded to Version 6.3.26 (3ddp mode) and contained 12,256 cells.
- As a subcontractor to NETL, EG&G constructed an Aspen Plus® process model for a FutureGen cycle (i.e., a modified version of Case FG-3 in Ref. 9). Two modified versions of the cycle, along with the original cycle, were developed and sent to Alstom by Steve Zitney (NETL) on 27 May 2004, along with some very helpful notes (also by Steve Zitney), that summarized the major sections and features of the flowsheet.
 - ZZPSA3.bkp – Original (unmodified) FutureGen flowsheet
 - ZZPSA3_FixIn.bkp – Removed GS-HBALD design spec
 - ZZPSA3_FixIn_norcy.bkp – Removed GS-HBALD design spec and broke the GASCOOL (SOOT convergence block) and TAILGAS (TRTAIL convergence block) recycle loops. These latter modifications helped greatly to reduce the computational time of the cycle (as well as that of the FLUENT® blocks embedded within the cycle), without significantly impacting the accuracy of the cycle.
- Steve Zitney (NETL) had prepared a “demo” for the 1st Industrial Advisory Board (IAB) meeting held 21 June 2006. The demo consisted of the Aspen Plus® file APECS_FutureGen_GasCreocombSaveCW2005c.apw, modified from a version of the (above) ZZPSA3_FixIn_norcy.bkp files process model, as well as the entrained-flow gasifier FLUENT® files Gas2OxyEntslurry3d30m_MSmods_Ar_Wen_Moist_CO2_WGS_HCl_COdevol.cas(dat), in conjunction with a scheme file gasification.scm and the UDF files energy.c, gasif-law-no_collision_dfg2.c, gasif-law.h, and gasif-law-no_collision.h. The .cas (.dat) files corresponded to FLUENT® V6.2.16 and had 12,256 cells. The coupled simulation also included the FLUENT® files for a 2-D gas turbine combustor. The Aspen Plus® simulation

files (while not executed), served as a template for setting up the run for the current task. A figure from the Steve Zitney “demo” is provided in Figure 1.

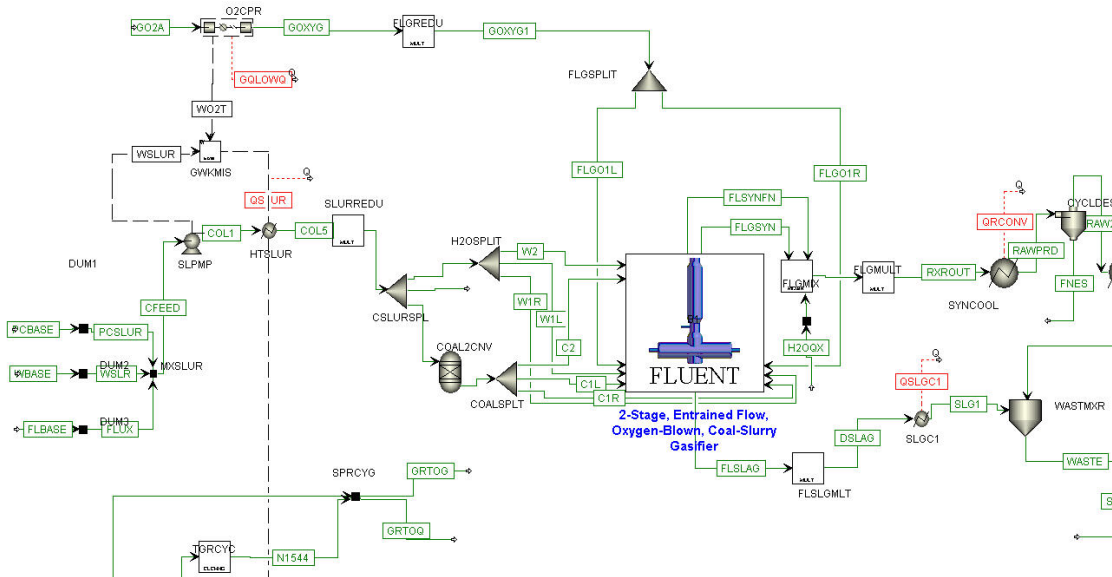


Figure 1: Flowsheet from NETL Gasification “Demo” Provided at 1st IAB Meeting

In the present task, the original ZZPSA3.bkp file was modified to allow instantiation of a FLUENT® block by following the prior process model setup work that Steve Zitney did for the IAB “demo.” Some of the modifications made to ZZPSA3.bkp were similar to those made to ZZPSA3_FixIn_norcy.bkp above.

As mentioned above, the FLUENT® model of the gasifier was derived from the ANSYS/NETL files “ent-flow-gasif.cas(dat).” Some of the CFD case characteristics and submodel options prescribed by ANSYS/NETL in the case file are itemized below:

- FLUENT® V6.3.26 was used for all of the preliminary computations, as well as the APECS-related runs.
- The grid has 12,256 cells.
- The solution algorithm is based upon the SIMPLE pressure-correction algorithm with first-order upwind differencing.
- The turbulence model is the standard k-ε turbulence model, with standard wall functions.
- For the gasification reactions, the turbulence-chemistry interaction was modeled using the composite Magnussen eddy-dissipation (mixing-limited / fast-chemistry) and Arrhenius rates (kinetically-limited) combustion submodel, in which the smaller of the two rates (eddy-dissipation or Arrhenius rates) takes local precedence. As mentioned earlier the reaction expressions were developed by ANSYS from the literature and are implemented through UDFs. Seven reactions were implemented, some of which are irreversible reactions and some of which entail reverse rate expressions (e.g., water-gas shift reaction). The reader is referred to Ref. 6 through 8 for additional details about the stoichiometry and reaction expressions.

- Custom laws, implemented through UDFs were utilized for heterogeneous moisture release, devolatilization and gasification.
- The discrete particle model in FLUENT® is a stochastic Lagrangian particle model. Surface injection was used for both coal particles and water droplets at 3 locations (two feed locations at each end of the lower first stage and the third location at the second stage feed). A Rosin-Rammler distribution of 8 particle sizes was specified for the coal particles and 5 diameters for the water droplets, along with 5 stochastic “tries”. Particle-radiation coupling was not activated.
- The radiation submodel was the Discrete Ordinates radiation model, with a mixture scattering coefficient of 0.9, and an absorption coefficient based on the “wsggm-cell-based” option.
- All of the walls were set to adiabatic.
- The inlet boundary conditions were prescribed through a mass-flow rate boundary specification. The outlet was a pressure-outlet boundary condition.

A view of the baseline geometry configuration, complete with the external surface mesh, is shown in Figure 2, and a temperature contour plot at the $Z=0$ mid-plane is provided in Figure 3.

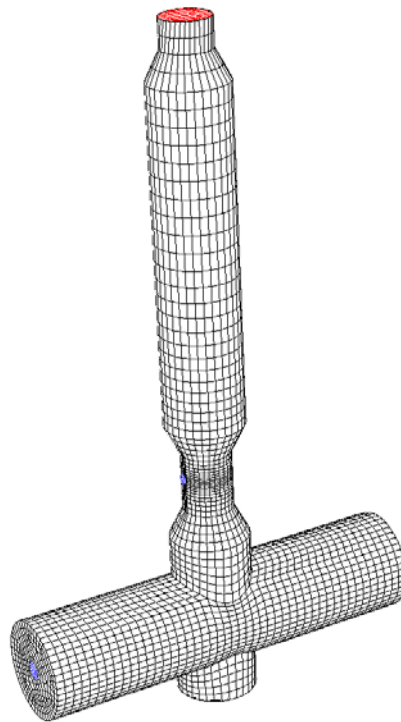


Figure 2: External Surface Grid for Baseline Case

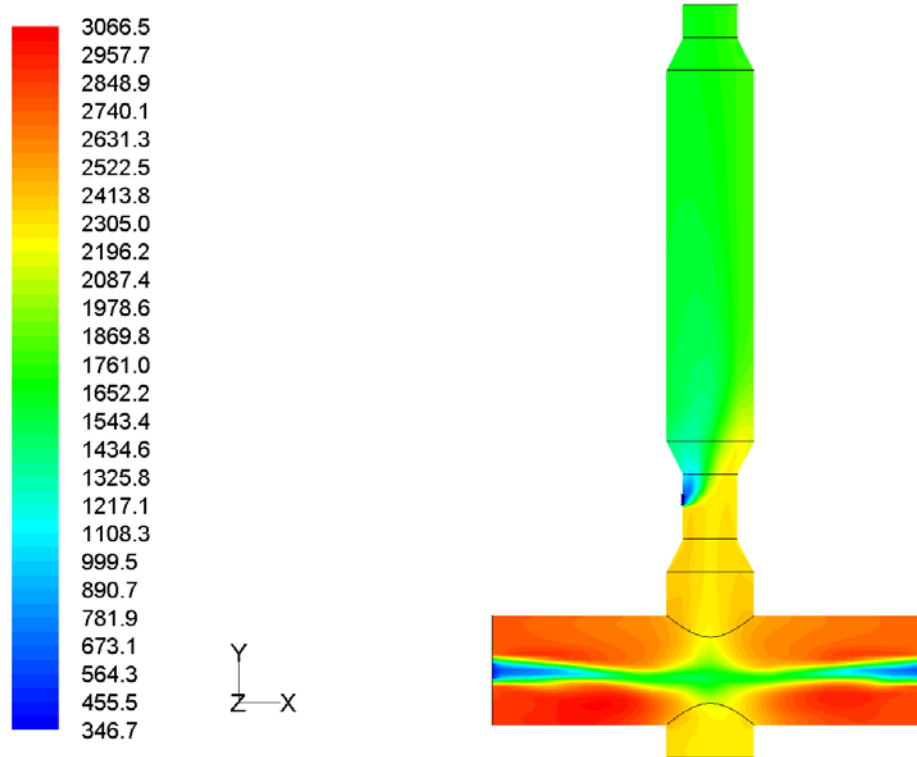


Figure 3: Temperature Contours (in Units of K) at Z=0 Plane for Baseline Case.

5.1.2 Design and Operational Considerations

General criteria for selecting the proper design and operating conditions for a gasifier have been outlined by Bockelie, et al. (Ref. 11 and 12). For the entrained flow gasifier utilized in the present project, the following assumptions and principles guided the geometric scale-up:

- Carbon conversion can be treated on a per-pass basis (gasifier alone) or on a combined gasifier-cyclone basis (where the recycle loop is included in the carbon conversion). The overall carbon conversion efficiency (based on a material balance around the gasifier and recycle loop) should be high. For the gasifier alone, the ash coming in with the coal leaves through the cyclone exhaust and through the slag tap. The cyclone efficiency may range from e.g., 90% to 99.9%. Since cyclone operation can reject almost all of the particles, this means that only a relatively small amount of char/ash mass is allowed to escape the system through the vortex finder, and that the vast majority of the char/ash will be recycled to the gasifier. As a consequence, to prevent accumulation, a mass balance dictates that most of the ash mass entering with the inlet feed coal must exit the system through the slag tap. An appropriate slagging model for the entrained flow gasifier case might contain the following features:
 - The slag model must allow the captured particles to essentially completely burn out on the wall during the transit time to the slag tap, since the amount of residual carbon in the slag is very small.

- The slag model should (ideally) allow for the formation of a layer of “frozen” or “stationary” slag next to the waterwall, and then one or more (discretized) layers of running slag. The inner surface temperature of the first stage must be high enough to enable the slag to flow freely (e.g., on the order of 1755.4 K (2700 °F) or higher, with a temperature of at least 1644.3 K (2500 °F) at the slag tap), but not so high that it becomes necessary to lower the temperature by adding steam. The maximum temperature may be dictated by the formation of eutectics between the refractory and the slag, the fluxing of the refractory off the wall, and the prevalence of a sufficient layer of the frozen slag to act as an insulating barrier.
- To recreate an accurate recycled char stream, the following parameters may need to be collected or calculated in the gasifier outlet streams and then adjusted for passage through the cyclone(s): (a) volatile content (if any remains), (b) carbon content, (c) ash content, (d) particle size distribution, (e) particle density distribution, and (f) mass flow rate.

For purposes of the present task, a slagging model was not available within the FLUENT® code, and therefore the implementation of a char recycle stream was precluded. However, it was found that the carbon burnout at the outlet of the computational domain, presuming that the calculated values are calibrated to a realistic value, was fairly high (e.g., >98.5%) over the range and that any char recycle flow rate, using the present UDF and range of inlet flows, would be a small percentage of the total inlet solids flow.

- The cold gas efficiency (CGE) should be maximized. The CGE goes down as carbon loss goes up. As one varies the oxidant and slurry reactant ratios, the CGE will adjust to correspond with the change in product species composition, according to the equation:

$$CGE = \frac{(\dot{m}_{syngas})(HHV_{syngas})}{(\dot{m}_{fuel})(HHV_{fuel})} \quad (1)$$

where the subscripted term “syngas” denotes the outlet gas mixture and the subscripted term “fuel” denotes the inlet coal. The mass flow rate for the fuel used in the denominator presumes total consumption (i.e., no unburned fuel). In the present case, the gaseous species that contribute to the CGE include CH₄, CO, H₂, and H₂S (although the H₂S that enters FLUENT® case is passed through the domain as if it were an “inert” because no H₂S chemistry is included in the FLUENT® case).

- For gasifiers that permit a wall heat loss, then as the volume increases (keeping all of the geometric ratios constant), the surface/volume (S/V) ratio goes down. If the throughput-to-volume ratio is maintained, then as the volume goes up, the fraction of the heat input that is lost through the walls goes down, and the exit temperature goes up slightly. This increased exit temperature may be offset somewhat by allowing wall heat loss and increasing the section lengths (e.g., increasing the height of the riser). As the height of the riser is extended, an increasing amount of heat is lost, the temperature decreases, and the kinetics of gasification drop off sharply with a decrease in temperature. Ordinarily, under adiabatic

conditions, as the height of the riser (and the residence time) is increased, the carbon conversion should continue to increase asymptotically (albeit with diminishing returns). Thus, the outlet carbon conversion will be dependent upon the relative magnitudes of the wall heat loss versus the riser height.

For the present task, the gasifier walls have been prescribed as adiabatic, and hence, the coal conversion and CGE are expected to increase asymptotically (i.e., with diminishing rate) as the height of the riser increases. However, from design considerations, it is not very practical or economical to continue to increase the riser height to increase the CGE; rather the height should be set, and then the system should be optimized by increasing the oxygen/coal ratio and recycling the unburned carbon.

- The velocity must be high enough to keep the particles entrained. If this constraint is met in the baseline design, then it is reasonable to assume that, as the gasifier is scaled with throughput, the cross-sectional velocities should be kept about the same. This will help to preserve the overall residence times and mixing intensities, as long as overall flow patterns and recirculation zone reattachment points are not adversely jeopardized.

5.1.3 Scale-Up Approach

Based on previous development and calibration of the baseline CFD geometry and component mass flow rates via the cooperative work between NETL and ANSYS, it was presumed that the baseline case produces the desired outlet temperature and syngas mixture. The baseline geometry ensures that:

- The desired first-stage slagging temperature is achieved
- The upward velocity is sufficient to ensure that the particles are properly entrained
- All flame stability criteria are met.
- The H₂O/coal or slurry ratio is set at the level required to effectively transport and inject the coal into the gasifier at high pressure.
- The splits in the flow rates of the constituent H₂O, coal, and O₂ streams feeding the first and second stages are set to achieve the optimal CGE for the given geometry.

Consequently, some of the operating parameters or ratios can be fixed, and the baseline geometry can be scaled as a function of the “throughput” alone (where the throughput remains to be defined). It will be presumed that the scale-up or sizing will not be a strong function of those constraints that are held constant. Some of the parameters or constraints that will be held constant include:

- The H₂O/coal ratio is set.
- The splits in the flow rates of the constituent H₂O, coal, and O₂ streams feeding the first and second stages are set.
- The relative lengths of the geometrical sections and the height of the riser are set.
- The walls will remain adiabatic.
- No char or ash recycle.

The primary factors that will be allowed to change in some fashion in order to meet desired goals include:

- The “throughput,” defined as the overall coal flow rate to the gasifier, will be varied in order to achieve a process model “design spec” target.
- The overall oxygen flow rate will be varied, through an optimization methodology, in order to achieve the maximum CGE for a given coal throughput.
- The geometry and inlet cross-sections will scale with the coal throughput in order to preserve the cross-sectional velocities and residence times.

The solution strategy utilized for modifications to the ZZPSA3.bkp flowsheet will consist of an outer and inner loop that encompasses the FLUENT® block and other auxiliary blocks associated with the gasification section:

- Outer Loop: Retain the original Design Spec BURNTEMP that varied the coal flow rate until a specific gas turbine inlet stream (XPCO3) temperature was attained.
- Inner Loop: Replace Design Spec GS-HBALD with an optimization analysis (using e.g., the Complex (Simplex) method or the default SQP methodology) that varies the O₂ flow rate until the CGE is maximized.
- As in ZZPSA3_FixIn_norcy.bkp, break the GASCOOL (SOOT convergence block for the STBLW tear stream) and TAILGAS (TRTAIL convergence block for the N1544 tear stream) recycle loops. This will eliminate prior outer loops that would force additional iterations of the CFD block.

It is conceivable that additional interesting computational experiments could be performed with this initial demonstration case for geometric scaling. Users might build in additional functionality and complexity in order to explore the ramifications of scaling up specific geometric dimensions in ways that are distinct from the rest of the geometry, such as the first-stage length and the riser height. Such exploratory design rules could be partially decoupled from the current (arbitrary) constraints governing the change in the gasifier cross sectional area, and could embrace other design spec targets (e.g., adjusting the riser height until a particular carbon conversion is attained).

5.1.4 Parameterization and the Associated CAPE-OPEN Variables

The gasifier geometry was parameterized (by Naresh Patre in Fluent-India) and the gridding of the gasifier was completely automated. A reference sketch (not to scale) showing the parameterization is provided in Figure 4.

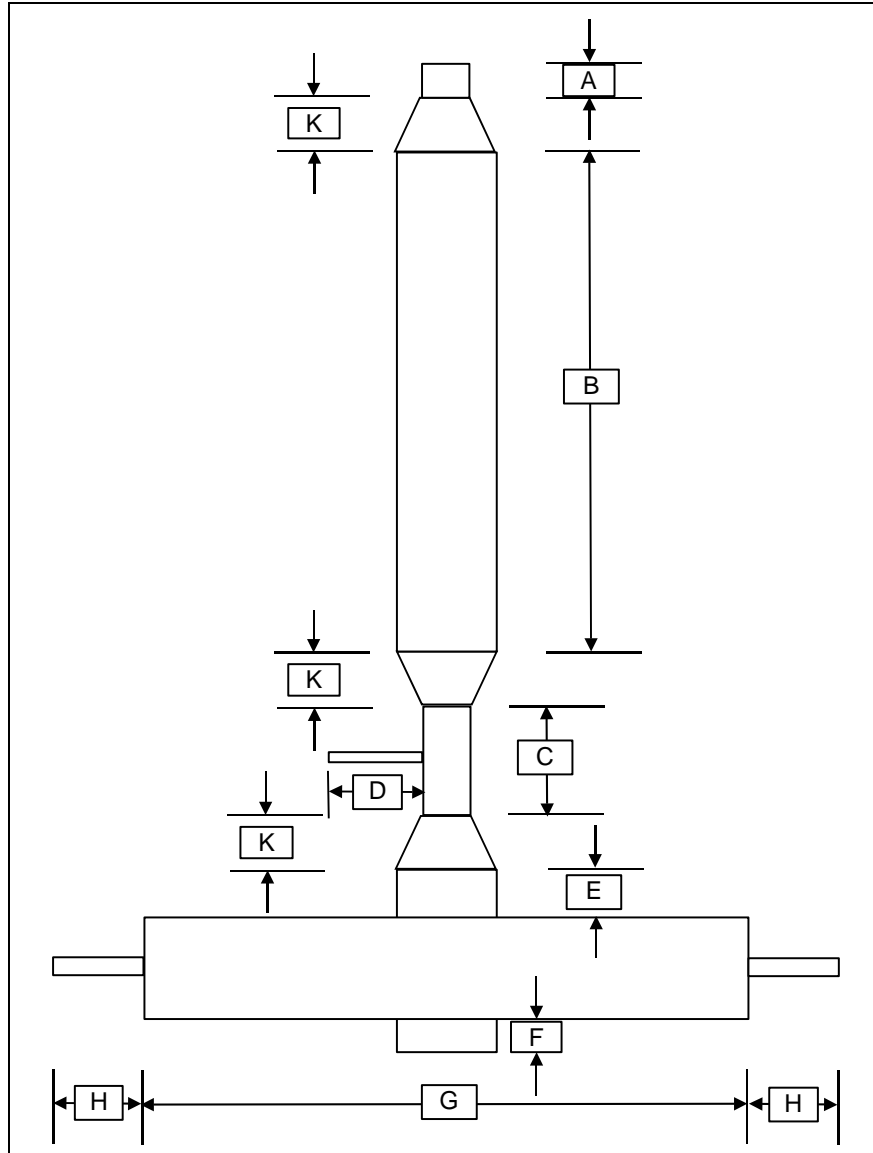


Figure 4: Sketch of Gasifier Showing Parameterization.

Each of the geometric sections was assigned a unique set of parameters for the section lengths and diameters in order to allow for maximum versatility in scaling each section.

As previously mentioned, the term “throughput” has been defined as the coal flow rate (and excludes the water required for the slurry mixture). A scaling factor (SF) may be devised for the cylindrical cross-sections, such that the “new” cross-sectional area is a simple function of the ratio of the “new” coal throughput to the original or baseline throughput, raised to the power “scaling factor (SF).”

$$\left[\frac{\text{new throughput}}{\text{baseline throughput}} \right]^{SF} = \left[\frac{\frac{\pi D_{new}^2}{4}}{\frac{\pi D_{baseline}^2}{4}} \right] = \frac{D_{new}^2}{D_{baseline}^2} \quad (2)$$

where:

- D = diameter
- L = length or height
- SF = scaling factor (power law)

The ratio may be rearranged to give the “new” diameter as:

$$D_{new} = \sqrt{D_{baseline}^2 \left[\frac{\text{new throughput}}{\text{baseline throughput}} \right]^{SF}} \quad (3)$$

The SF parameter may conceivably range from zero to any arbitrary value. Two particular values are noteworthy:

- 0: the “new” dimensions will remain unchanged relative to the baseline dimension.
- 1: the “new” dimension will change linearly with the throughput ratio.

For length or height dimensions, the length may likewise be made a function of the ratio of the “new” coal throughput to the original or baseline throughput, also raised to an arbitrary power “SF.”

$$\left[\frac{\text{new throughput}}{\text{baseline throughput}} \right]^{SF} = \frac{L_{new}}{L_{baseline}} \quad (4)$$

where L = length or height of the cylindrical section. The “new” length parameter is explicitly given as:

$$L_{new} = L_{baseline} \left[\frac{\text{new throughput}}{\text{baseline throughput}} \right]^{SF} \quad (5)$$

All of the new lengths and diameters were calculated in the Aspen Plus® calculator block C-PREUSR. In the present task, an explicit scaling factor was not employed. A value of SF = 1 was implicitly used for all of the diameter scaling factors, and a value of SF = 0 was implicitly used for all of the length scaling factors.

The GAMBIT® gridding parameters, corresponding to each of the geometry parameters, are listed in Table 1, along with some of the maximum and minimum allowable values stipulated by ANSYS personnel required to enforce topologically realistic dimensions required for the GAMBIT® gridding software.

Table 1: GAMBIT® Journal File and Geometry Parameters

Sr. No.	Names in User Journal File	Names in Parameterization Journal File	Default Values	Minimum Values	Maximum Values
1	\$section_A_diameter	\$geometry_parameter_1	1.00	> 0	Infinity
2	\$section_A_length	\$geometry_parameter_2	0.60	> 0	Infinity
3	\$section_B_diameter	\$geometry_parameter_3	1.60	> 0	Infinity
4	\$section_B_length	\$geometry_parameter_4	6.80	> 0	Infinity
5	\$section_C_diameter	\$geometry_parameter_5	1.00	> 0	Infinity
6	\$section_C_length	\$geometry_parameter_6	1.20	> \$section_D_diameter	Infinity
7	\$section_D_diameter	\$geometry_parameter_7	0.20	> 0	< \$section_C_diameter
8	\$section_D_length	\$geometry_parameter_8	0.03	> 0	Infinity
9	\$section_E_diameter	\$geometry_parameter_9	1.60	> 0	< \$section_G_diameter
10	\$section_E_length	\$geometry_parameter_10	0.80	> 0	Infinity
11	\$section_F_diameter	\$geometry_parameter_11	1.60	> 0	< \$section_G_diameter
12	\$section_F_length	\$geometry_parameter_12	0.60	> 0	Infinity
13	\$section_G_diameter	\$geometry_parameter_13	2.00	> \$section_E_diameter > \$section_F_diameter (whichever is greater)	Infinity
14	\$section_G_length	\$geometry_parameter_14	8.00	> \$section_E_diameter > \$section_F_diameter (whichever is greater)	Infinity
15	\$section_H_diameter	\$geometry_parameter_15	0.40	> 0	< \$section_G_diameter
16	\$section_H_length	\$geometry_parameter_16	0.00	0.00	Infinity
17	\$section_K_length	\$geometry_parameter_17	0.60	> 0	Infinity

The default values are the geometric dimensions (in meters) corresponding to the baseline case. The GAMBIT® journal file in this study is named “gasifier.jou.” At the beginning of the GAMBIT® journal file, the CAPE-OPEN geometry parameters (denoted “geometry_parameter_#”) are assigned to the corresponding physical parameters (denoted section_#_diameter or section_#_length). The physical parameters are then used in the GAMBIT® journal file to build a new grid with each new execution of a scale-up or alteration of the geometry. This assignment (i.e., GAMBIT® physical parameter = CAPE-OPEN geometry parameter) is illustrated with the following lines of coding from the GAMBIT® journal file (i.e., gasifier.jou):

```

/      assign CAPE-OPEN parameters to physical parameters

$section_A_diameter = $geometry_parameter_1
$section_A_length = $geometry_parameter_2
$section_B_diameter = $geometry_parameter_3
$section_B_length = $geometry_parameter_4
$section_C_diameter = $geometry_parameter_5
$section_C_length = $geometry_parameter_6
$section_D_diameter = $geometry_parameter_7
$section_D_length = $geometry_parameter_8
$section_E_diameter = $geometry_parameter_9
$section_E_length = $geometry_parameter_10
$section_F_diameter = $geometry_parameter_11
$section_F_length = $geometry_parameter_12
$section_G_diameter = $geometry_parameter_13
$section_G_length = $geometry_parameter_14
$section_H_diameter = $geometry_parameter_15

```

```
$section_H_length = $geometry_parameter_16
$section_K_length = $geometry_parameter_17
/
```

The remainder of the journal file contains GAMBIT® commands for the re-sizing and re-meshing of the geometry.

On the Aspen Plus® side, the customized CAPE-OPEN variables are itemized in Table 2.

Table 2: CAPE-OPEN Parameters As Implemented Through APECS Wizard

Parameters	Type	Description	Access Mode	Lower Bound	Upper Bound	Units	Default Value
geometry-parameterization	integer	indicates if geometry parameterization is enabled	read-write	0	1	---	0
geometry_model_name	option	base name of GAMBIT® journal file	read-only	---	---	---	gasifier
geometry_scale_factor	real	mesh scaling factor	read-only	1E-06	1E+06	---	1
geometry_parameter_1	real	section A diameter	read-write	1E-06	1000	m	1
geometry_parameter_2	real	section A length	read-write	1E-06	1000	m	0.6
geometry_parameter_3	real	section B diameter	read-write	1E-06	1000	m	1.6
geometry_parameter_4	real	section B length	read-write	1E-06	1000	m	6.8
geometry_parameter_5	real	section C diameter	read-write	1E-06	1000	m	1
geometry_parameter_6	real	section C length	read-write	1E-06	1000	m	1.2
geometry_parameter_7	real	section D diameter	read-write	1E-06	1000	m	0.2
geometry_parameter_8	real	section D length	read-write	1E-06	1000	m	0.03
geometry_parameter_9	real	section E diameter	read-write	1E-06	1000	m	1.6
geometry_parameter_10	real	section E length	read-write	1E-06	1000	m	0.8
geometry_parameter_11	real	section F diameter	read-write	1E-06	1000	m	1.6
geometry_parameter_12	real	section F length	read-write	1E-06	1000	m	0.6
geometry_parameter_13	real	section G diameter	read-write	1E-06	1000	m	2
geometry_parameter_14	real	section G length	read-write	1E-06	1000	m	8
geometry_parameter_15	real	section H diameter	read-write	1E-06	1000	m	0.4
geometry_parameter_16	real	section H length	read-write	0	1000	m	0
geometry_parameter_17	real	section K length	read-write	1E-06	1000	m	0.6
geometry_user_journal_activate	integer	indicates whether user journal file will be read in after domain interpolation	read-write	0	1	---	0
geometry_user_journal_name	option	name of user journal file	read-only	---	---	---	prep_post

The general sequence for utilization of the CAPE-OPEN variables is as follows:

- The CAPE-OPEN parameter “geometry-parameterization” is set equal to unity (in calculator block SWITCH), which will then notify the APECS controller, when Aspen Plus® executes

the FLUENT® block, to regrid the FLUENT® mesh. [Note that integer switches are used in the above sequence, because at the time of this writing, a bug exists within Aspen Plus® that precludes CAPE-BOOLEAN variables from being properly exposed to calculator blocks, sensitivity analyses, etc.]

- Whenever the parameter “geometry-parameterization” is activated (i.e., set equal to unity), then in a subsequent calculator block (C-PREUSR), the values for each “geometry_parameter_#” are overwritten based on the ratio of the current coal throughput to the baseline throughput.
- When the FLUENT® block is executed, APECS generates a boundary condition file, as well as an interpolation file from the current FLUENT® .cas and .dat files.
- The GAMBIT® journal file (i.e., gasifier.jou) then uses the newly scaled diameter and length values to construct a new mesh. The CAPE-OPEN parameter “geometry_model_name” is used for the file name for the GAMBIT® journal file, and the “geometry_parameter_#” variables provide the baseline or default geometric values. The “geometry scale factor,” not to be confused with the scaling factor in Eqns. 2 through 5, is the scale factor used within FLUENT® to scale the entire mesh “once” prior to any calculations (as e.g., when a user desires to convert a mesh constructed in units of inches to a mesh based on units of meters).
- The APECS controller reads the boundary condition file and the interpolation file into the newly created mesh file.
- A second journal file (prep_post.jou) is used to complete post-interpolation operations. A critical post-interpolation operation is exemplified in those cases that contain reacting particles. Since the FLUENT® interpolation operation does not restore particle source terms, the under-relaxation factor for the DPM model must be set to unity, and “/solve/dpm-update” must be executed as a text user interface (TUI) command. This re-establishes particle source terms within the computational domain and then the DPM under-relaxation term can be set back to a reasonable value (e.g., 0.1). The name of this second journal file is provided through the CAPE-OPEN variable “geometry_user_journal_name” and the journal file is read by the APECS controller whenever the integer switch “geometry_user_journal_activate” is set to unity. The prep_post.jou file is provided below:

```
; prep_post.jou
;
; After FLUENT grid creation, BC specification, and domain interpolation,
; this journal file performs the following operations:
;
; (1) does dpm-update
;
define user-defined execute-on-demand
"read_urf::libudf"

/solve/set/under-relaxation dpm 1.0
/solve/dpm-update

define user-defined execute-on-demand
"write_urf::libudf"
```

The “read_urf” UDF simple reads the current DPM under-relaxation factor; that factor is retained in memory until the “dpm-update” command is completed, and then that same factor is used to set the new under-relaxation factor.

It should be noted that all of the above APECS operations are performed without ever exiting FLUENT®. This helps to persist and preserve important case features, such as arbitrary (plotting) planes that the user previously created.

5.1.5 Ports and Connecting Streams

The various ports and stream connections for the FLUENT® block, all of which are associated with a single domain (Domain-1), are itemized as follows:

Table 3: FLUENT® Block Ports and Connecting Streams

FLUENT® Computational Domain Port	Inlet or Outlet Port	Description of Port	Type of Port	Corresponding Aspen Plus® Stream Name
inlet-left	inlet	1 st stage oxygen inlet on left	inlet port	FLGO1L
inlet-right	inlet	1 st stage oxygen inlet on right	inlet port	FLGO1R
injection-coal-left	inlet	coal DPM injection for 1 st stage left	DPP	C1L
injection-coal-right	inlet	coal DPM injection for 1 st stage right	DPP	C1R
injection-coal-stage-2	inlet	coal DPM injection for 2 nd stage	DPP	C2
injection-water-left	inlet	water DPM injection for 1 st stage left	DPP	W1L
injection-water-right	inlet	water DPM injection for 1 st stage right	DPP	W1R
injection-water-second	inlet	water DPM injection for 2 nd stage	DPP	W2
outlet-gas	outlet	2 nd stage syngas outlet	outlet port	FLGSYN
synslag	outlet	particles out of 2 nd stage	DPP	FLNCOT
slag	outlet	particles out of 1 st stage	DPP	SLAGOT
DPP =		discrete phase port		

The entrained flow gasifier flowsheet (ZZPSA3) has 3 substreams:

- an NC (non-conventional) substream, consisting of coal, slag, fines, and ash
- a CISOLID (conventional inert solids) substream, consisting of CARBON and SULFUR, and
- a MIXED substream consisting of 18 species: CO₂, CO, O₂, H₂, H₂O, CH₄, AR, N₂, CARBON, SULFUR, SiO₂, HCL, H₂S, CL₂, SO₂, NH₃, COS, and H₂SO₄ (where the species are displayed with the names used by Aspen Plus®).

Note that CARBON and SULFUR are in both the CISOLID and the MIXED substreams. The NC coal (entering the system through stream PCBASE) and the NC slag (entering through system stream FLBASE) added together constitute the total mass of the coal entering the system. The coal, slag, fines, and ash in the NC substream really only exchange mass with the CARBON and SiO₂ in the MIXED substream (at least in the part of the process model that surrounds the gasifier components).

It is a general requirement that all NC or CISOLID substreams must be converted into a MIXED stream before said stream reaches the FLUENT® block. This is due in part to limitations within the prevailing CAPE-OPEN standard, as well as to limitations in the implementation of the standard and solid substreams in Aspen Plus®. In like manner, the streams exhausting from the FLUENT® block are mandated to be MIXED streams only, and must be converted to NC or CISOLID streams, if such streams are desired downstream of the FLUENT® block.

Upstream of the FLUENT® block (see Figure 1), the sum of the NC coal and NC slag are passed through an RYIELD reactor (non-stoichiometric reactor based on known yield distribution) called COAL2CNV, and reactants are turned into a MIXED substream, consisting of non-zero values of CARBON, H2, N2, CL2, SULFUR, O2, and SIO2 (i.e., surrogate for ash), in proportions based on the ultimate analysis of the coal. However, the actual component yields do not matter, because only the total mass flow rate and the temperature are passed on to the FLUENT® Lagrangian particle injections through the DPM physical model ports. The resulting converted MIXED substream is split into the material streams C1L, C1R, and C2 and routed to the FLUENT® block.

At the physical model outlet ports, SYNFLAG and FLAG, a MIXED substream consisting of finite values of CARBON (surrogate for unburned carbon) and SIO2 (surrogate for ash) are passed to the connecting material streams FLNCOT (for the flyash) and FLAGOT (for the bottom slag), respectively. Again RYIELD reactors are used to convert the MIXED substreams into prescribed yields of the CISOLID component CARBON, and the NC components slag, fines, ash, etc. The material stream FLNCOT is connected to RYIELD reactor TOPCONV, and the material stream FLAGOT is connected to RYIELD reactor BOTCONV. (Neither RYIELD reactor is shown in Figure 1). A calculator block is executed immediately upstream of each RYIELD reactor in order to overwrite the prescribed component yields with the actual yields leaving the FLUENT® block.

The CAPE-OPEN variables for the two DPM physical model outlet ports (SYNSLAG and FLAG), as stored within the FLUENT® case file, along with current (or default) values (SI units), are given below:

```
(CAPE-OPEN/dpm/synslag/mass-fracs "(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)")  
(CAPE-OPEN/dpm/flag/mass-fracs "(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)")  
(CAPE-OPEN/dpm/synslag/temperature 1714.5907)  
(CAPE-OPEN/dpm/flag/temperature 2280.962)  
(CAPE-OPEN/dpm/synslag/flow-rate 2.7111783)  
(CAPE-OPEN/dpm/flag/flow-rate 0.004910129)
```

The series of mass fraction values are those corresponding to the sequence of 18 species listed in the FLUENT® case under the Materials → Species panel (see the left-hand column of Table 4) for the MIXED substream. Other than the mass fractions, only the mass flow rate and temperature are required to complete the definition of the FLUENT® block physical model outlet ports.

The species listed in both the FLUENT® case and the Aspen Plus® flowsheet, as well as their respective mappings to each other, are provided in Table 4.

Table 4: Species Listing and Mapping

	Species in the Materials Listing in FLUENT®	FLUENT® Species Exposed to the PME	Flowsheet Species
(1)	ch4	ch4	CH4
(2)	co2	co2	CO2
(3)	co	co	CO
(4)	o2	o2	O2
(5)	h2	h2	H2
(6)	h2s	h2s	H2S
(7)	h2o	h2o	H2O
(8)	cl2	cl2	CL2
(9)	carbon	carbon	CARBON
(10)	s	s	SULFUR
(11)	sio2	sio2	SIO2
(12)	hcl	hcl	HCL
(13)	ar	ar	AR
(14)	so2	so2	SO2
(15)	nh3	nh3	NH3
(16)	cos	cos	COS
(17)	h2so4	h2so4	H2SO4
(18)	n2	n2	N2

As this demonstration case was completed at an “early” point in the APECS development history, the FLUENT® case had to have the same number of public species as the PME (and thus the familiar “Zero Flow Rate Species” descriptor in the mapping procedure does not appear). However, as in other demonstration cases, the flowsheet species do need to be mapped according to the order in the FLUENT® case.

The outlet port variables for the two DPM physical model ports are set in the “DEFINE_DPM_BC(bc_slag, p, t, f, f_normal, dim)” UDF (in file gasif-dpm.c) through the lines:

```

    sprintf(slag_mass_fracs, "(0 0 0 0 0 0 0 0 0 %f 0 %f 0 0 0 0 0 0 0)", 1-x_ash,
x_ash);
    RP_Set_String("cape-open/dpm/slag/mass-fracs", slag_mass_fracs);
    RP_Set_Real("cape-open/dpm/slag/temperature", slag_part_temp);
    RP_Set_Real("cape-open/dpm/slag/flow-rate", slag_mass_flux);

```

and in “DEFINE_DPM_BC(bc_outlet, p, t, f, f_normal, dim)” UDF (in file gasif-dpm.c) through the lines:

```

    sprintf(synslag_mass_fracs, "(0 0 0 0 0 0 0 0 0 %f 0 %f 0 0 0 0 0 0 0)", 1-
x_ash, x_ash);
    RP_Set_String("cape-open/dpm/synslag/mass-fracs", synslag_mass_fracs);
    RP_Set_Real("cape-open/dpm/synslag/flow-rate", part_outlet_mass_flux);
    RP_Set_Real("cape-open/dpm/synslag/temperature", outlet_part_temp);

```

5.1.6 Files and Software for the Integrated Co-Simulation

The workscope, completed in 2008, utilized the following software versions:

- FLUENT® V6.3.26 [Microsoft Windows 2000 and Linux (32-bit)]
- GAMBIT® V2.4.6 [Microsoft Windows 2000 and Linux (32-bit)]
- Aspen Plus® V2004.1
- ANSYS APECS V1.3.0-B013 [Microsoft Windows 2000] plus kit for Linux (32-bit)
- Hummingbird Connectivity 2006 / Exceed V11.0 (for GAMBIT® applications on PC)
- ANSYS EKM® V1.1

It should be noted that since the work was completed in 2008, a full year before the completion of other demonstration subtasks in the project, the software and approach used at that time in the past will not be completely congruent with the improved APECS software and methodology available at the end of the Alstom scheduled workscope (i.e., September 2009). All of the runs were performed with Aspen Plus® as the executive process modelling software. The Aspen Plus® cycle was run in the PC Windows environment and the CFD cases were exercised on (1) a single-processor (3 GHz) PC and (2) remotely over a LAN on a single processor of a multi-processor AMD-64 Opteron box.

The FLUENT® case files (from Jens Madsen (ANSYS)) were prepared and made CAPE-OPEN and APECS compatible by sequentially performing the steps outlined in the APECS User's Manual (Ref. 10):

- Execute CAPE-OPEN Solver Configuration Wizard
- Execute APECS Model Wizard
- Upload files to the ANSYS Engineering Knowledge Manager (EKM®) database

The files uploaded to the EKM® database consist of the following:

```
energy.c
gasif-dpm.c
gasif-dpm.h
gasification.scm
gasifier.jou
gasifier_15284.cas
gasifier_15284.dat
libudf/ntx86/3ddp/libudf.dll
prep_post.jou
```

The APECS software allows for a number of journal files, which are helpful if providing increased control to the run sequence. Each of the two journal files have already been discussed. The addition of the UDFs (*.c) and header file (gasif-dpm.h) are uploaded so that recompilation can occur should the APECS block be run on a remote Linux platform.

The instantiation procedure, in general terms, is given below:

- (1) Instantiate the APECS block and rename it to e.g., "B1"
- (2) Assign the block section = DESGASIF

- (3) Make all of the stream connections in accordance with Table 3.
- (4) Update the sequence and the calculator blocks to accommodate the B1 block:

```
SEQUENCE MOPTIM C-MODBAS O2CPR MXSLUR SLPMP HTSLUR SLURREDU &  
          CSLURSPL H2OSPLIT COAL2CNV COALSPLT FLGREU FLGSPLIT &  
          C-PREUSR B1 FINESSET TOPCONV FLGMIX FLGMULT SYNCOOL &  
          CYCLDES GWKMIS SLAGSET BOTCONV FLSLGMLT SLGC1 C-PSTUSR &  
          C-CGE
```

```
CALCULATOR C-MODBAS
```

```
  DEFINE MAXITB BLOCK-VAR BLOCK=B1 VARIABLE=MAXIMUM-ITERATIONS
```

```
CALCULATOR C-PREUSR
```

```
  DEFINE GSF BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_SCALE_FACTOR"  
  DEFINE GP1 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_1"  
  DEFINE GP2 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_2"  
  DEFINE GP3 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_3"  
  DEFINE GP4 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_4"  
  DEFINE GP5 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_5"  
  DEFINE GP6 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_6"  
  DEFINE GP7 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_7"  
  DEFINE GP8 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_8"  
  DEFINE GP9 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_9"  
  DEFINE GP10 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_10"  
  DEFINE GP11 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_11"  
  DEFINE GP12 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_12"  
  DEFINE GP13 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_13"  
  DEFINE GP14 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_14"  
  DEFINE GP15 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_15"  
  DEFINE GP16 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_16"  
  DEFINE GP17 BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_PARAMETER_17"  
  DEFINE JGP BLOCK-VAR BLOCK=B1 VARIABLE=GEOMETRY-PARAMETERIZATION  
  DEFINE KJA BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_USER_JOURNAL_ACTIVATE"
```

```
CALCULATOR C-PSTUSR
```

```
  DEFINE JGP BLOCK-VAR BLOCK=B1 VARIABLE=GEOMETRY-PARAMETERIZATION  
  DEFINE KJA BLOCK-VAR BLOCK=B1 VARIABLE="GEOMETRY_USER_JOURNAL_ACTIVATE"
```

```
CALCULATOR SWITCH
```

```
  DEFINE JGP BLOCK-VAR BLOCK=B1 VARIABLE=GEOMETRY-PARAMETERIZATION
```

- (5) Complete the species mapping, as indicated by the last two columns in Table 4.

5.2 Computational Results

5.2.1 Optimization Space for CGE Variable

For optimization problems, AspenTech recommends that preliminary parametric or sensitivity analysis runs be made in order to better understand the nature of the optimization space, the constraints, and the viable range for initial estimates of the design variables. In accordance with that recommendation, a few runs were made with stand-alone FLUENT®, for 3 different geometry

scales and a range of overall stoichiometries. The grid was re-meshed each time in accordance with the GAMBIT® journal file, so that the total number of cells in the case changes.

Table 5: Inputs for Parametric Runs

Geometry	New Throughput / Baseline Throughput	DPM Coal Flow Rate (kg/s)	DPM Water Flow Rate (kg/s)	Number of Cells
B	2	56.08	23.24	10,660
A	1	28.04	11.62	14,918
C	0.5	14.02	5.81	20,180

Each geometry (A, B, and C) was run at 7 different stoichiometries from approximately 0.282 to 0.47 (where the stoichiometric mass of oxygen to the unit mass of coal = 2.13109, and where, as the stoichiometry increases, the oxygen inlet flow rate increases for a given coal and water flow rate). The initial results for the CGE, the outlet temperature, and the overall carbon burnout as a function of overall stoichiometry are shown in Figures 5 through 7.

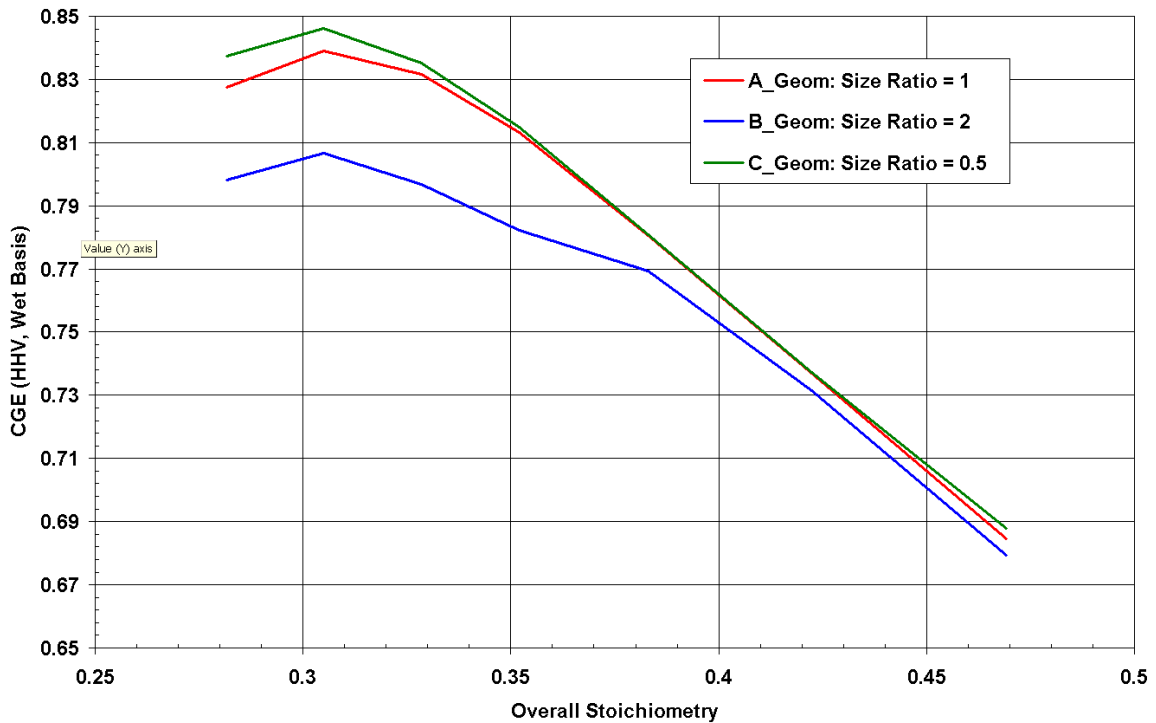


Figure 5: CGE as a Function of Overall Stoichiometry for 3 Size Ratios.

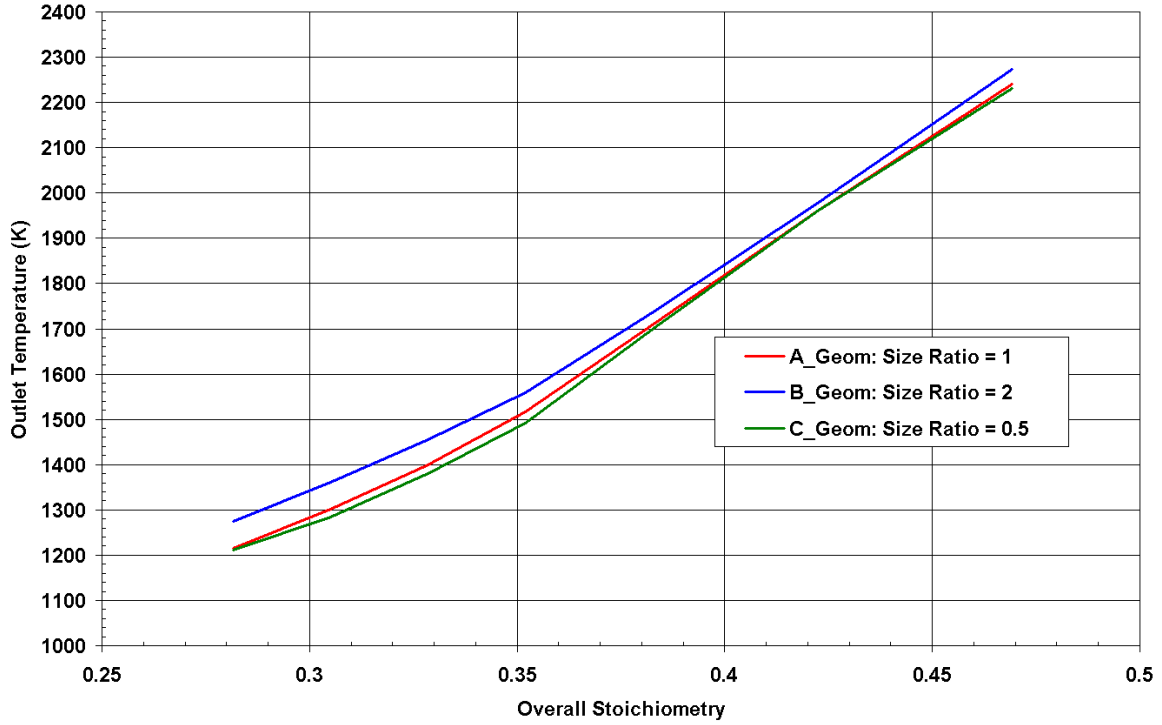


Figure 6: Outlet Temperature as a Function of Overall Stoichiometry for 3 Size Ratios.

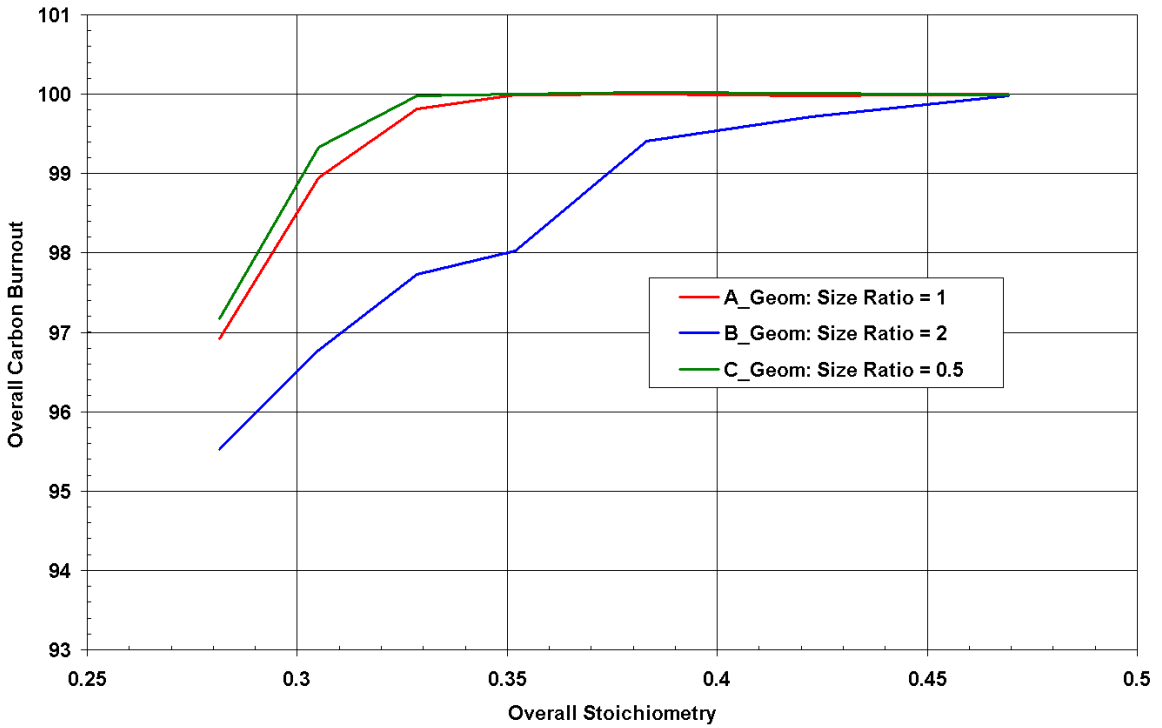


Figure 7: Carbon Burnout as a Function of Overall Stoichiometry for 3 Size Ratios.

The goal is to maximize the CGE, at a given coal mass flow rate. The objective function for the CGE should be subject to constraints. For example, the temperature in the interior of the gasifier near the walls should be kept high enough to provide a layer of running slag (e.g., about 1675 K for the present coal). In addition, the overall carbon burnout should be kept high (e.g., greater than about 97%). It can be shown that both the interior temperature (averaged over a plane below the second stage feed location; not shown) begins to be violated as a stoichiometry value of 0.28 or 0.29 is approached from a higher stoichiometry. The carbon burnout (Figure 7) constraint is a function of the geometry size ratio, and for a size ratio equal to 2.0, the constraint is violated as a stoichiometry value of 0.31 is approached from a higher stoichiometry; for smaller size ratios, the constraint is violated for stoichiometry values less than about 0.28. It is for that reason that the range over which the optimization exercise was conducted was limited to a lower value of about 0.28.

It should be noted that the curves, particularly the curve for a size ratio of 2.0, is not smoothly varying, but appears to have a “dip” in the curve at a stoichiometry of 0.35. It will be shown shortly that this is due to the fact that the cases representing the points along the curve are not necessarily converged, or are oscillating.

It should also be noted that the CGE curve is a rather shallow curve. For a change of 0.01 in the stoichiometry, the CGE (in the vicinity of the maximum value) may change by a value on the order of 0.003 to 0.005. Any oscillations or noise in the FLUENT® solution will impact the ability of the optimization routine in Aspen Plus® to converge.

5.2.2 Solution Strategy and Results

There are two optimization routines in Aspen Plus®:

- The first is the so-called COMPLEX method, which is based on a feasible path “black-box” pattern search. The method can handle inequality constraints and bounds on decision variables. The COMPLEX method frequently takes many iterations to converge, but does not require numerical derivatives.
- Sequential Quadratic Programming (SQP) method is a state-of-the-art, quasi-Newton nonlinear programming algorithm. It can converge tear streams, equality constraints, and inequality constraints simultaneously with the optimization problem. The SQP method usually converges in only a few iterations but requires numerical derivatives for all decision and tear variables at each iteration.

When these methods were attempted with the instantiated FLUENT® block, both methods bogged down and ultimately failed. The reason is because both methods require the APECS block to provide reliable, accurate, and repeatable solutions. If the APECS block solutions are not reliable or repeatable, then the optimization routines are fed inaccurate solutions; this in turn, causes the optimization routines to perform faulty line searches and to calculate erroneous search directions, thus causing the routine to ultimately fail.

Significant effort was expended in trying to understand the convergence behaviour of the FLUENT® case. Three issues can impact the convergence behaviour:

- General lack of convergence (i.e., being far from convergence)
- Rolling or oscillatory solution
- Solution “noise” (due to e.g., the stochastic particle model)

If the convergence is stable and if the “noise” is much smaller than the magnitude of the gradients in the objective function, then perhaps one or more of the tolerances associated with the optimization routine can be loosened, and this may be all that is required to get the flowsheet to converge. However, if the CFD case is subject to severe non-convergence, or when an oscillating or rolling solution is exhibited by the CFD case, then the current optimization routines in Aspen Plus® may not converge at all. Indeed, both severe convergence issues and noise were encountered in the calculation of the maximum CGE in the present task.

When several of the cases for the stand-alone FLUENT® cases were run a few thousand more iterations, it became apparent that the solution was slowly migrating and that the previous cases were not fully converged after all:

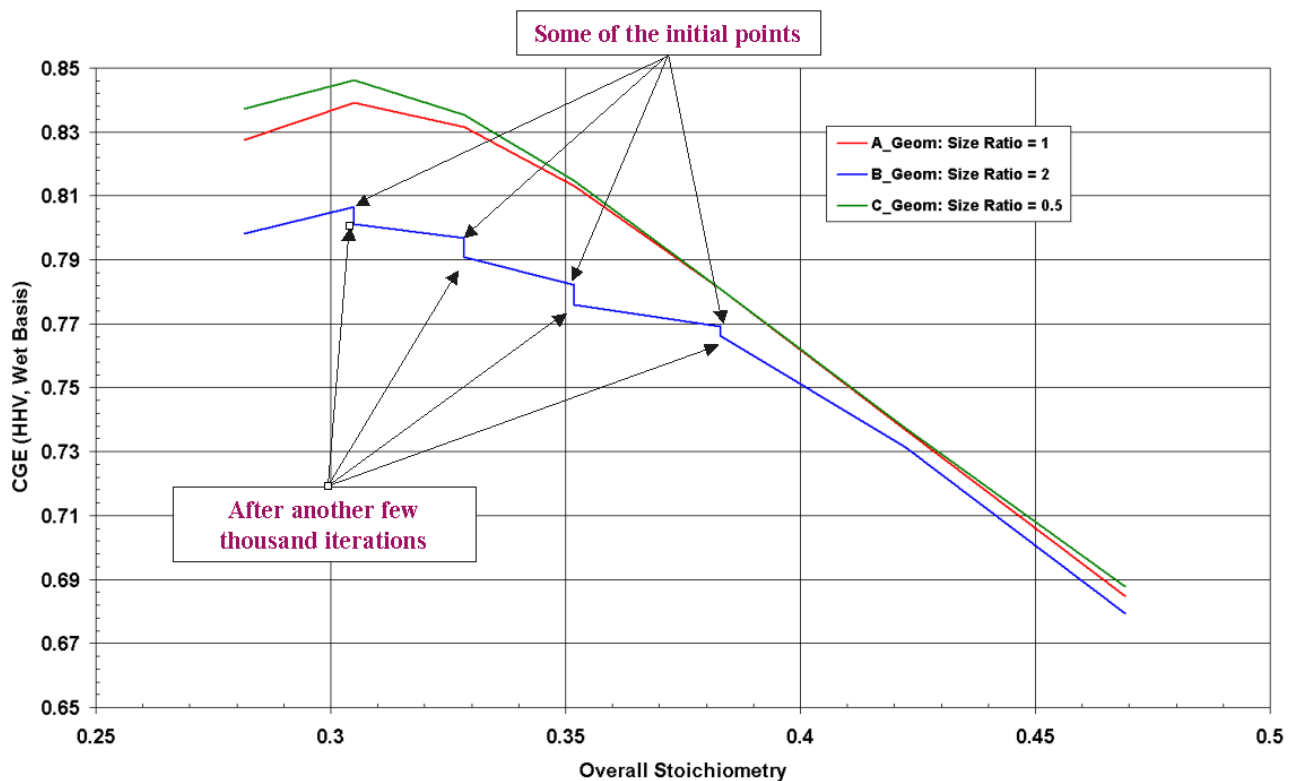


Figure 8: CGE as a Function of Overall Stoichiometry and Iteration Count.

This convergence issue was illuminated even more, when one of the restart files stored during the course of a co-simulation was harvested and subsequently exercised in stand-alone mode. In the co-simulation (see Figure 9), during one of the Aspen Plus® loops, the first FLUENT® case was run

for 4,000 iterations, and the subsequent runs associated with a change in stoichiometry, were run for 2,000 iterations. After every FLUENT® execution, the restart files were automatically saved out to an APECS temporary directory. After 10,000 total iterations, the corresponding restart file was located and extracted from the temporary directory.

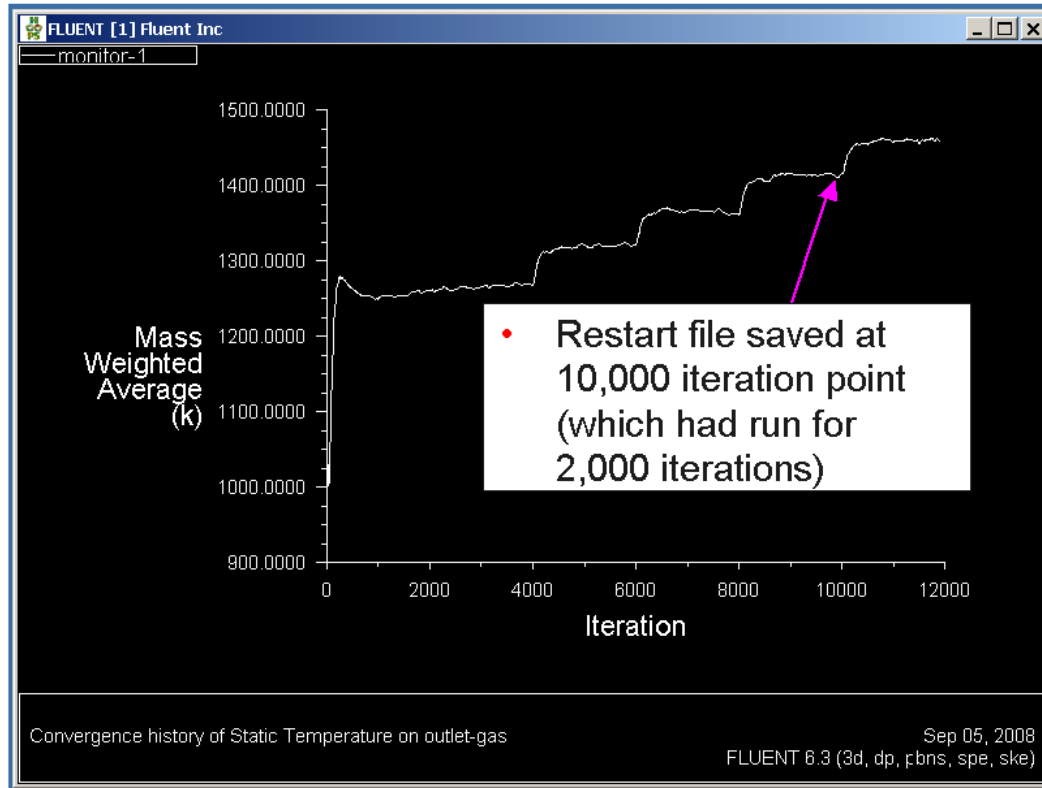


Figure 9: CGE as a Function of Overall Stoichiometry and Iteration Count.

The restart file was subsequently run in stand-alone mode for another 110,000 iterations (see the two contiguous plots in Figure 10). The under-relaxation factors were set to reasonably aggressive values. The DPM under-relaxation factor (URF) was set at 0.25, which was found to help accelerate the overall convergence (relative to a DPM URF of 0.1) without making the case unstable. The CGE values slowly climb over a course of 50,000 iterations from 0.81 to over 0.832, and then appear to adopt a rolling or oscillating convergence pattern as the 110,000 iteration mark is approached. The band of “noise” about the “rolling mean” exhibits an overall difference in magnitude of approximately 0.003. This band of “noise” appears to emanate from the stochastic particle iterations, with each particle iteration causing a spike in the CGE. The temperature also manifests a similar pattern (not shown), falling from 1415 K to about 1370 K over the same iteration range and then adopting a trend toward an apparent rolling mean. It is not known whether the rolling mean in CGE or temperature would stabilize and level out to a constant mean value if the iterations were continued.

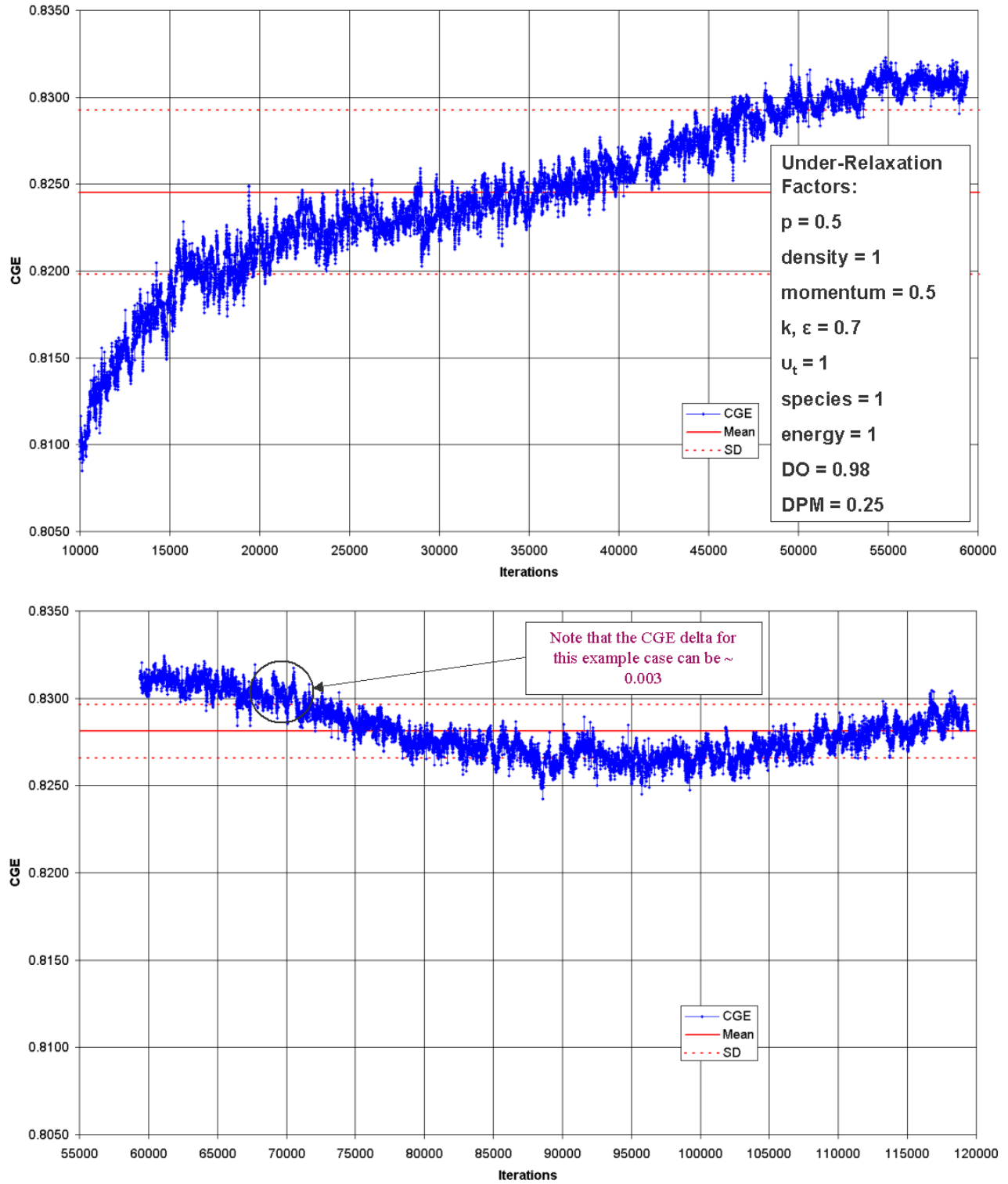


Figure 10: CGE Convergence Characteristics as a Function of iteration Count

An enlarged view of this noise pattern for the CGE (for a different run) is provided in Figure 11; a similar noise pattern is apparent for the outlet gas temperature (see Figure 12), which has a peak-to-peak amplitude of about 4 or 5 K.

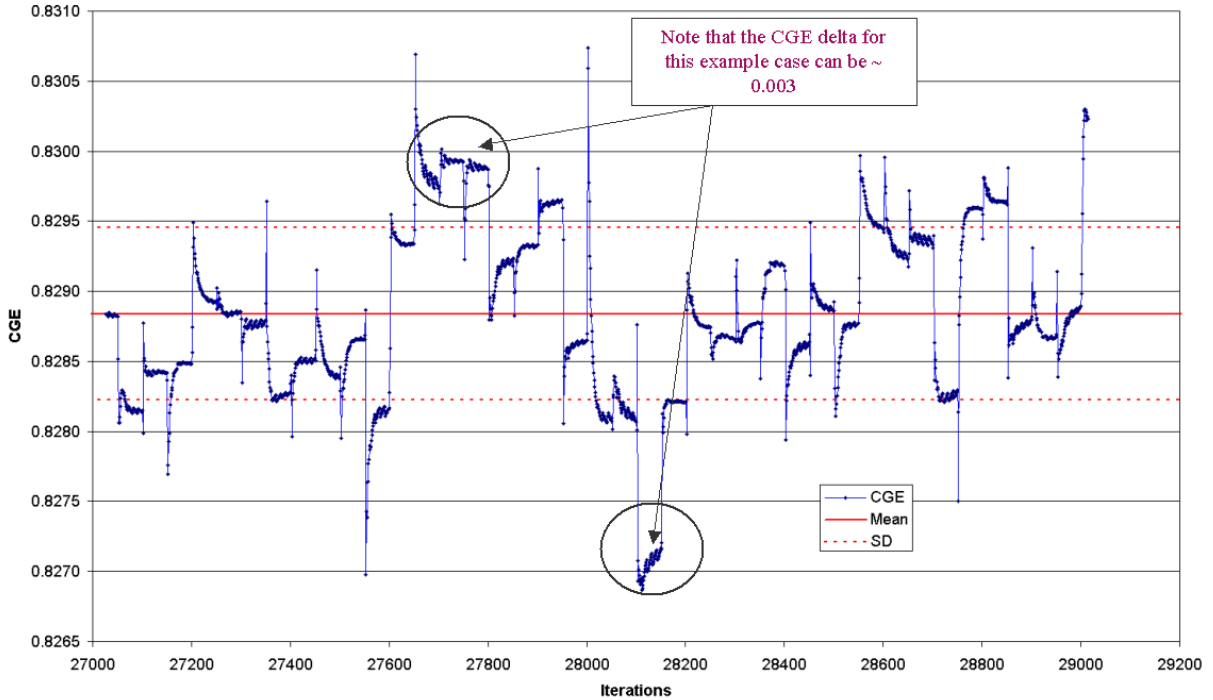


Figure 11: CGE Convergence Characteristics as a Function of iteration Count (Enlarged).

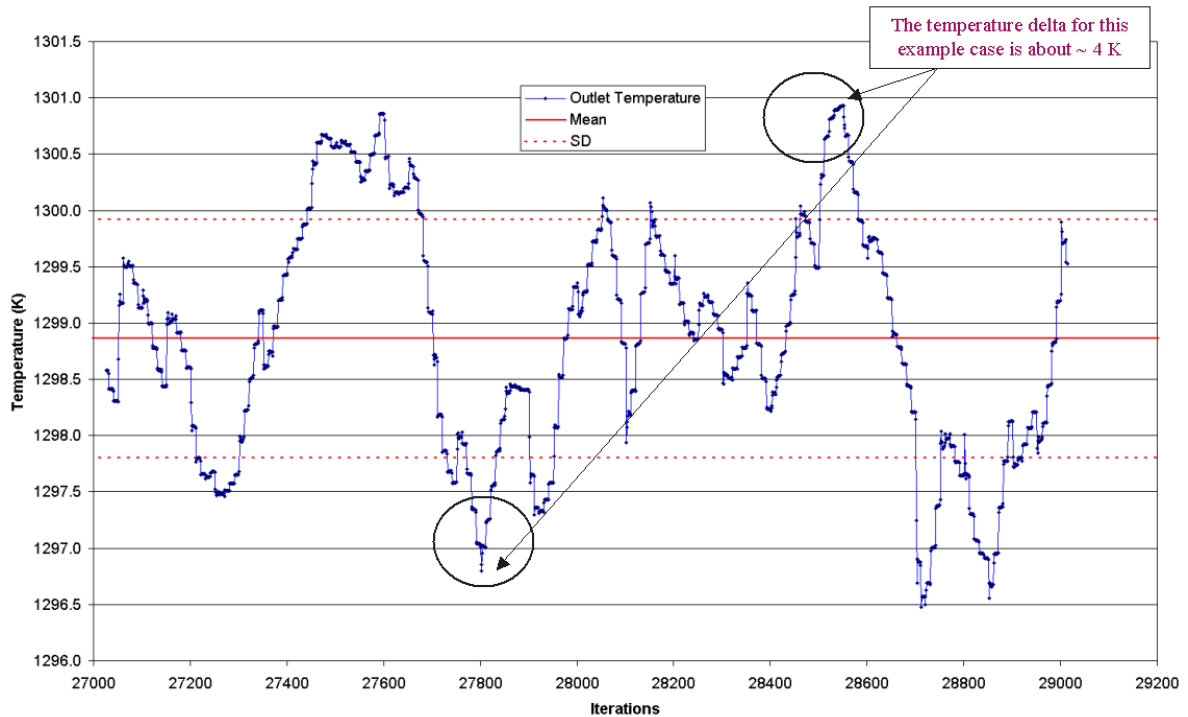


Figure 12: Temperature Convergence Characteristics as a Function of iteration (Enlarged).

Each DPM execution is marked by a CGE spike, followed by relaxation of the spike during the subsequent 50 gas-phase iterations. Again, the difference exhibited by peak-to-peak amplitudes can be on the order of 0.003. As noted earlier, for a change of 0.01 in the stoichiometry, the CGE (in the vicinity of the maximum value) may change by a value on the order of 0.003 to 0.005. Therefore, the differences in the fluctuating magnitude of the CGE caused by “noise” alone, are sufficient to induce convergence difficulties in the search patterns of the optimization algorithms in the PME.

In the present task, the coal flow rate (or throughput) is being changed, thus requiring a geometry scale-up change and a new mesh. On top of that, the optimization routine is changing the oxygen flow rate into the model. Both the geometrical and feed flow rate variations appear to require extremely lengthy convergence times (e.g., between 25,000 to more than 100,000 iterations), even though the case only has on the order of 15,000 cells. The current case is able to achieve about 530 iterations per CPU hour when run on a single processor of a PC. Thus, presuming that on the order of 50 to 100 FLUENT® block executions are required for reasonable convergence of the flowsheet, it quickly becomes apparent that the CPU time for convergence of the FLUENT® block renders the optimization exercise impractical (i.e., 50,000 iterations/530 iterations per CPU hour x 75 block executions = 7,075 hours = 295 days). Running the cases in parallel would certainly be helpful. However the grid size is so small (i.e., about 15,000 cells), that the communication time required between multiple processors would be a large part of the overall execution time and it is expected that the parallelization speed-up would be rather poor. Despite the above statements, it should be recognized that even if all of the APECS block executions could be “converged” by iterating on the order of 25,000 or 50,000 iterations for each execution, it is postulated that the resultant rolling

mean solution and the ample noise about the mean would still induce convergence difficulties in the current optimization routines in Aspen Plus®.

Part of the problem is that the SQP method in Aspen Plus® requires numerical derivatives for all decision and tear variables at each iteration, to establish a search direction, followed by line searches along that search direction. If the FLUENT® executions cannot provide a reliable and consistent numerical derivative, because of a general lack of convergence, then it is difficult to see how the SQP method can lead to a quick solution, or even find the solution at all. It is acknowledged that the current optimization routines in Aspen Plus® cannot be fairly tested by the current geometric-scaling problem. The lack of convergence achieved in each FLUENT® execution precludes a proper test of the SQP methodology, and NETL is encouraged to apply its resources in the future to formulate another task involving instantiation of a CFD block, which converges reasonably well (but which exhibits “noise”), and which can serve as a proper test subject for the SQP optimization algorithm.

It is not known why the current CFD case takes so long to converge. The UDFs developed by ANSYS for the DOE utilize the Magnussen eddy dissipation (“eddy-breakup”-type) model. A few errors were found in the UDFs, but it is not known whether the errors were benign or not, as far as the convergence times were concerned. ANSYS declined to seriously review the UDFs because most of the technology in the UDFs had already been rolled into a new version of FLUENT® that would be released in the future, and it was not cost-effective to review a set of UDFs that would no longer be used. In the opinion of the present author, the CPU penalties were probably a result of inserting reverse rate expressions into the list of mechanisms, thus creating a stiff chemistry set that required long convergence times. It is not known to what extent a stiff chemistry solver would have benefited the present CFD computations.

In order to promote flowsheet convergence, it was decided to “hardwire” a convergence algorithm. The outer loop for the design spec BURNTTEMP effectively modified the coal flow rate until a particular downstream temperature was achieved. Within that outer loop, the oxygen flow rate corresponding to the maximum CGE (associated with the prevailing coal flow rate) was required. The optimum oxygen flow rate was found simply by sequencing the calculation of the APECS block 6 times. The first FLUENT® execution was allowed to go 4,000 iterations and the remaining 5 FLUENT® executions were allowed to 2,000 iterations each. (While this number of executions is probably not sufficient for convergence, it was deemed a practical number if iterations for the present methodology.) The CGE was calculated at 5 discrete stoichiometry (STOICH) values 0.0125 apart (i.e., starting with a value of 0.28, and then sequentially 0.2925, 0.305, 0.3175, and finally 0.33). The largest CGE is determined by surveying the CGE values calculated at the 5 stoichiometry values, and then this value becomes the “optimum” value. The flowsheet repeats the calculation at that stoichiometry (STOICH) value corresponding to the “optimum” CGE value before exiting the sequence. The entire procedure is controlled through calculator blocks and sequencing. (It should be noted that a sensitivity analysis loop could not be used because a sensitivity analysis cannot be used as an inner loop; it will not reinitialize.)

The results are shown in Figure 13. Each set of 6 simulations is at a different coal flow rate; the Aspen Plus® run was stopped manually after it had run 10 outer design-spec (BURNTTEMP) loops and appeared to “converge” to a particular coal flow rate (481,764 lb_m/hr (60.7 kg/s)) within a

reasonable tolerance. At the first coal flow rate (No. 1: COAL = 170,071 lb_m/hr (21.4 kg/s)), the 6 sequenced APECS block executions are shown, as well as the corresponding number of iterations for each. In almost every case, the optimal (or largest) value of the CGE corresponds to a stoichiometry of 0.305. The difference in the CGE value calculated for both the 3rd and the 6th sequenced block execution (both at a stoichiometry of 0.305) indicates the variability in the solution due to the FLUENT® convergence issues discussed previously (i.e., (a) general lack of convergence, (b) rolling or oscillating solutions, or (c) solution noise). The third coal flow rate (No. 3: COAL = 512,583 lb_m/hr (64.6 kg/s)), represents the transition from a relatively small coal flow rate to a large coal flow rate, and therefore exhibits the largest variability between the 3rd and the 6th sequenced block executions at the optimum stoichiometry.

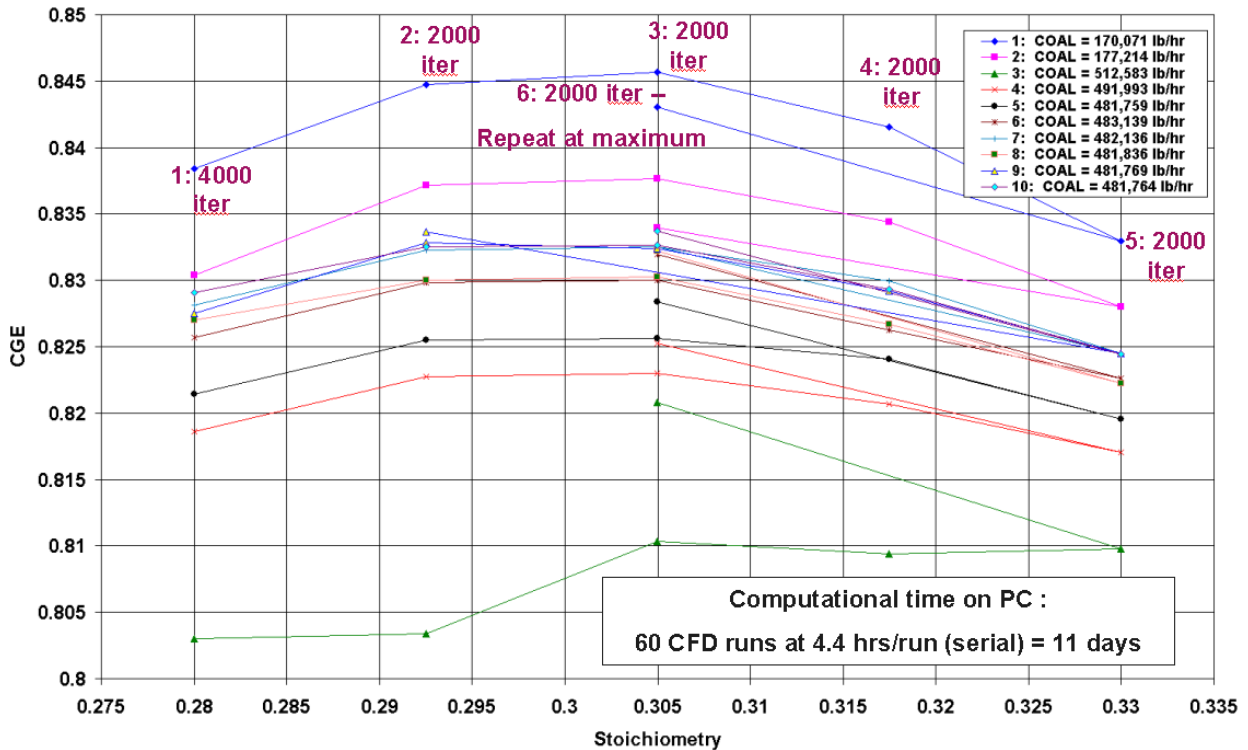


Figure 13: Temperature Convergence Characteristics as a Function of iteration (Enlarged).

The 60 CFD runs required on the order of 4.4 CPU hours/run on a serial processor of a PC, for a total elapsed time of approximately 11 days.

The above method seemed to work reasonably well in order to demonstrate the solution of the case and to produce a task deliverable, although the approach is certainly not considered to be a candidate for a general optimization algorithm. The suggestion proposed by AspenTech personnel was that, in situations where the optimization routines provided by Aspen Plus® appear to be inadequate in some way, then Aspen Custom Modeler® can be used to code and devise a different methodology that may be more suitable. If, after additional testing, the SQP methodology is found to be deficient for use with stochastic and “noisy” CFD, then perhaps other optimization approaches

more amenable with CFD can be explored: e.g., to cover the desired range with discrete calculation points and then apply a curve fit, using a regression to find the maximum or optimal value.

It should be noted that, although much of the focus in the above discussion has been on convergence issues, the geometric-scaling methodology worked flawlessly.

6.0 CONCLUSIONS AND RECOMMENDATIONS

This present task helps to showcase an important advancement in the APECS tool kit – the addition of geometric-scaling calculations to an integrated FLUENT® CFD block execution. ANSYS provided a GAMBIT® journal file that modified the cross-sectional geometry of the gasifier as a function of feed flow rate in order to keep average velocities and residence times from changing significantly. Alstom was tasked with the responsibility of geometrically parameterizing and scaling a module within a coupled simulation. Specifically, Alstom Power was tasked with the responsibility to (a) develop a gasifier model with automated geometry control, remeshing, and interpolation of the old solution, (b) demonstrate geometric scaling in an integrated simulation, and (c) utilize design specifications, sensitivity analysis, and/or optimization routines, as required, within Aspen Plus® to exercise the parameterized component. This was accomplished.

One of the milestones was to provide an “interim task report” for Task 5.2, which contains a description of the case simulations, the results, and an itemization of desired software modifications for future work. This report summarizes and documents the demonstration case simulations under this task, and thus completes the requirement for the stipulated (contractual) interim task report. In essence, however, this report also constitutes a final topical report for the task.

A demonstration case was completed in which a gasifier that scaled geometrically with coal throughput was instantiated and run on an Aspen Plus® flowsheet. The geometry for the gasifier was first parameterized, so that every geometric dimension was assigned a corresponding physical parameter. The physical parameters were then given default or baseline parameters and defined in a GAMBIT® journal file (“gasifier.jou”), developed by ANSYS. In addition, at the beginning of the GAMBIT® journal file, CAPE-OPEN geometry parameters were assigned to the corresponding physical parameters. The physical parameters were then used in the GAMBIT® journal file to build a new grid with each new mandated scale-up or alteration of the geometry. This assignment (i.e., GAMBIT® physical parameter = CAPE-OPEN geometry parameter), in the journal file, allowed the executive software, Aspen Plus®, to define how the gasifier geometry scales as a function of the coal throughput variable in the flowsheet. Other Booleans and journal files assisted in the process of building the grid, interpolating the previous solution onto the new mesh, reconstructing DPM particle model source terms, changing under-relaxation factors, and generally preparing the FLUENT® case for execution. The methodology devised by ANSYS for the geometry scale-up worked flawlessly in both PC and Linux (LAN) environments.

In the present application, the FLUENT® model for the gasifier utilized multiple UDFs developed by ANSYS for the DOE. The goal was to instantiate the model on the flowsheet and to optimize the CGE at a given coal flow rate by varying the oxygen feed (or stoichiometry). However, it was found that the FLUENT® model exhibited 3 convergence problems that made it very difficult to

use with optimization routines in Aspen Plus® – (a) lengthy convergence times (tens of thousands of iterations), (b) rolling or oscillating solutions, and (c) stochastic noise about the mean solution. Unfortunately, it was found that the convergence problems associated with the FLUENT® model caused the optimization routines in Aspen Plus® to bog down and fail. Therefore, a methodology was devised in which the FLUENT® model was manually sequenced to calculate the CGE at 5 selected stoichiometries, and then the optimal value of the CGE was defined as the maximum of the 5 calculated values. This worked sufficiently well to enable the flowsheet to converge.

Significant progress has been made in facilitating the direct instantiation of FLUENT® on an Aspen Plus® flowsheet, in conjunction with geometric scale-up strategies. It is hoped that the present documented experience, as well as the associated journal files and explanation of the workflow, will help other others in their efforts to utilize the geometric scale-up option in subsequent co-simulations.

Some additional conclusions and recommendations are as follows:

- (1) Conclusion: Optimization in the geometric scaling task was hindered by (a) lengthy convergence times, (b) rolling or oscillating pseudo-steady state behaviour, and (c) stochastic “noise” associated with particles. Problematic convergence behavior can cause significant problems with the finite-difference derivatives and search directions associated with the current optimization routine (SQP) in Aspen Plus®. The lengthy convergence times associated with the present FLUENT® case precluded a fair assessment of CFD cases coupled with the optimization routines in Aspen Plus®, and this assessment needs to be repeated.
 - [1] Recommendation: In future NETL-affiliated efforts, if industrial demonstration or application cases are pursued, then at least one or two of the demonstration cases should involve both the (1) direct instantiation of a CFD case with the stochastic DPM model and (2) the optimization routines in Aspen Plus®. The user should be prepared to work closely with AspenTech personnel and implement a different kind of optimization routine (perhaps with Aspen Custom Modeler®) in the event that the SQP methodology in Aspen Plus® bogs down and fails (due to convergence problems, etc.). Perhaps for optimization purposes, a ROM may work better than the CFD case itself; this is also something that should be tried.
- (2) Conclusion: The lengthy convergence times may have been caused by the inclusion of reverse rate kinetic expressions (e.g., water-gas shift) in the mechanism listing. The inclusion of reverse rate expressions typically require stiff integrators.
 - [2] Recommendation: The gasifier model should be tested with combustion models that utilize a stiff integrator. Although stiff integrators may induce a severe CPU penalty, the cases may converge much faster.
- (3) Conclusion: Although the methodology for the geometric scale-up worked very well, additional care should have been taken to assess the quality of the grid during the scale-up. The cross-sectional area was allowed to change, but not the length of each section. This may

produce cell aspect ratios that become rather large, or grid skewness, etc. that should be monitored.

[3-A] Recommendation: In future geometric scale-up co-simulations, after every grid is built, perhaps there should be CAPE-OPEN variables that are reported in Aspen Plus® that reflect the quality of the grid that is produced, e.g., the number of cells, the maximum cell aspect ratio, grid skewness, etc. This will help the user to know whether the grid construction methodology should be modified to produce a more agreeable grid.

[3-B] Recommendation: The user should build a number of grids manually or off-line with the geometric scale-up journal files that correspond to extremes in the solution space, in order to assess the flow domain and ensure that unexpected and undesirable flow features (e.g., recirculation at the outlet of the domain, etc.) do not appear.

(4) Conclusion: The CFD model was utilized without a slagging model.

[4] Recommendation: A slagging model, amenable with steady-state formulations, should be added to the CFD case, so that the proper physics associated with the wall boundary conditions (of slagging gasifiers) can be brought into play.

7.0 REFERENCES

1. Zitney, S. E., M. Osawe, L. Collins, E. Ferguson, D. Sloan, W. Fiveland, J. Madsen, "Advanced Process Co-Simulation of the FutureGen Power Plant," In Proc. of the 31st International Technical Conference on Coal Utilization & Fuel Systems, May 21-25, Clearwater, FL (2006).
2. Sloan, D. G., W. A. Fiveland, S. E. Zitney, and M. Syamlal, "Software Integration for Power Plant Simulations," In Proc. of the 27th International Technical Conference on Coal Utilization & Fuel Systems, March 4-7, Clearwater, FL (2002).
3. Sloan, D. G., W. A. Fiveland, S. E. Zitney, and M. Syamlal, "Power Plant Simulations Using Process Analysis Software Linked to Advanced Modules," In Proc. of the 29th International Technical Conference on Coal Utilization & Fuel Systems, April 18-22, Clearwater, FL (2004).
4. Sloan, D. G., W. A. Fiveland, M. O. Osawe, S. E. Zitney, and M. Syamlal, "Demonstrations of Coupled Cycle Analyses and CFD Simulations over a LAN," In Proc. of the 30th International Technical Conference on Coal Utilization & Fuel Systems, April 17-21, Clearwater, FL (2005).
5. "Project Management Plan, Software Framework for Advanced Power Plant Simulations, Cooperative Agreement No. DE-FC26-05NT42443", for Ronald W. Breault USDOE-NETL, compiled by John F. Widmann, representing team entities: ANSYS Inc., ALSTOM Power, Aspen Technology Inc., and Carnegie Mellon University (Department of Chemical Engineering), May 6, 2008.

6. Shi, S., S. E. Zitney, M. Shahnam, M. Syamlal, W. A. Rogers, “Modeling Coal Gasification with CFD and the Discrete Phase Method,” Proceedings of the 4th International Conference on Computational Heat and Mass Transfer (ICCHMT), Paper No. ICCHMT’05 – 273, Paris-Cachan, France (May 17-20, 2005).
7. Shi, S., “Final Report: Upgrade the Coal Gasification Model Using DPM,” Fluent Report TM-[unknown], Prepared for S. E. Zitney (DOE/NETL), May 1, 2006.
8. Wen, C. Y. and T. Z. Chung, “Entrainment Coal Gasification Modeling,” Ind. Eng. Chem. Process Des. Dev., 18(4), 684-695 (1979).
9. Shelton, W. W. and C. W. White, “The Application of an In Situ Oxygen membrane for an IGCC Process with CO₂ Capture,” Final Report, EG&G Technical Services, Inc., Morgantown, WV (August, 2004).
10. APECS: Advanced Process Engineering Co-Simulation User’s Manual, Version 1.3, Prepared by ANSYS, Inc., Lebanon, NH (July 2008).
11. Bockelie, M. J., M. K. Denison, Z. Chen, T. Linjewile, C. L. Senior, and A. F. Sarofim, “CFD Modeling for Entrained Flow Gasifiers in Vision 21 Systems,” 19th Annual International Pittsburgh Coal Conference, Pittsburgh, PA (Sep. 23-27, 2002).
12. Bockelie, M. J., M. K. Denison, Z. Chen, C. L. Senior, and A. F. Sarofim, “Using Models to Select Operating Conditions for Gasifiers,” 20th Annual International Pittsburgh Coal Conference, Pittsburgh, PA (Sep. 15-19, 2003).

8.0 DELIVERABLES

The following files are delivered with this report:

No.	Files	Description
(1)	ZZPSA3_CGE_46-PC.bkp	Aspen Plus® flowsheet
(2)	ZZPSA3_CGE_46-PC_try-10.his	Aspen Plus® history file
(3)	gasifier_15284.cas gasifier_15284.dat	FLUENT® case and data files
(4)	energy.c gasif-dpm.c gasif-dpm.h gasification.scm	UDFs
(5)	create_planar_surfaces.jou integrate_v6p3p31.jou	Auxiliary FLUENT® files (used at the discretion of the user in stand-alone mode)
(6)	gasifier_15284.CORBAsolver junk_3_lnx86.model junk_3_lnx86.txt	Files created by APECS

	solverdata.txt	
(7)	gasifier.jou	GAMBIT® journal file
(8)	prep_post.jou	Used to perform a DPM update after grid generation and flow field interpolation

Software Framework for
Advanced Power Plant Simulations

TASK 5.2-B (ADDENDUM)
DEMONSTRATION OF EMISSIONS CALCULATIONS
FOR AN OXY-FIRED BOILER
IN AN INTEGRATED SIMULATION

Topical Report

Reporting Period Start Date: November 1, 2006
Reporting Period End Date: September 11, 2009

David Sloan, Ph.D.
Woodrow Fiveland, Ph.D.

November 30, 2009

DOE Cooperative Agreement No: DE-FC26-05NT42443

Alstom Power Inc.
US Power Plant Laboratories
P.O. Box 500, CEP 9005-1204
2000 Day Hill Road
Windsor, CT 06095

UNDER SUBCONTRACT TO
ANSYS Inc.
10 Cavendish Court, Lebanon, NH 03766

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ACKNOWLEDGEMENT STATEMENT

This document was prepared with the support of the U.S. Department of Energy, under Award No. DE-FC26-05NT42443. However, any opinions, findings, conclusions, or recommendations expressed herein are those of the author(s) and do not necessarily reflect the views of the DOE.

ACKNOWLEDGEMENTS

Financial support for this project was provided by DOE National Energy Technology Laboratory (Cooperative Agreement No. DE-FC26-05NT42443) and Alstom Power Inc. Appreciation is expressed to Ronald Breault as the DOE Project Manager and to Stephen E. Zitney (NETL) for his technical input. The contributions of the many APECS project team members from ANSYS Inc. and Aspen Technology Inc. are also greatly appreciated. ANSYS Inc. is particularly acknowledged for its leadership role as the prime contractor under the direction of John Widmann. Gratitude is expressed to Sorin Munteanu, John Widmann, and others on the APECS team at ANSYS Inc. for their diligence in striving to deliver quality software. Appreciation is also expressed to Stefano Orsino for his efforts in supplying a SO_x model and mechanism.

Within the Alstom Power divisions, appreciation is expressed to a number of individuals. John L. Marion, the Alstom Power Program Manager, and Shin Kang and Woodrow A. Fiveland, the Alstom Power Project Managers, contributed to the success of this project task.

ABSTRACT

The primary objective of this DOE project is to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of Advanced Process Engineering Co-Simulator (APECS), the simulator will link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. Generally, the simulator couples the cycle analysis software Aspen Plus® (marketed by Aspen Technology, Inc.), specialized component modeling packages, as exemplified by industrial proprietary codes (utilized by ALSTOM Power Inc.), and the FLUENT® computational fluid dynamics (CFD) code (provided by ANSYS Inc). An important enhancement to the APECS tool will be the creation of computationally efficient reduced-order models (ROM) based on information from high-fidelity computational fluid dynamics (CFD) models. Integration of the steady-state APECS tool with a virtual reality tool kit will allow process designers to navigate through a 3D virtual representation of a power plant with visualization of simulation results. A related secondary objective is to develop a prototype dynamic simulator within APECS for advanced power plants and to demonstrate a dynamic simulation that integrates plant-level and equipment-level models.

Among other responsibilities, Alstom Power has the responsibility to demonstrate the capabilities of the enhanced APECS tool to simulate proposed FutureGen power plant concepts and other relevant power generation processes and equipment. The present report encompasses only the workscope within Task 5.2b, an addendum to the original Task 5.2. Specifically, Alstom Power is tasked with the responsibility to demonstrate an APECS application involving the Alstom Boiler Simulation Facility (BSF) island with oxygen firing and flue gas recycling.

A demonstration case was completed in which a CFD model of the BSF furnace was coupled with a flowsheet of the BSF facility and run in an integrated simulation. In this simulation, the flue gas recycle ratio was varied in order to match a target furnace absorption. SO_x and NO_x emissions calculations were invoked with each execution of the BSF CFD block. This report documents the case characteristics and results and discusses the viability and capabilities of the linkage/interface software for oxy-firing and emissions applications.

TABLE OF CONTENTS

1.0	BACKGROUND ON APECS PROGRAM	2
1.0	BACKGROUND ON APECS PROGRAM	2
2.0	INTRODUCTION	2
3.0	EXECUTIVE SUMMARY	3
4.0	EXPERIMENTAL WORK.....	4
5.0	RESULTS AND DISCUSSION	5
5.1	Case Setup and Preparation	5
5.1.1	FLUENT® Case Grid and Setup	5
5.1.2	SOx Model Development and Background	9
5.1.3	BSF Process Model.....	11
5.1.4	Ports and Connecting Streams	12
5.1.5	Files for Integrated Co-Simulation	17
5.2	Computational Results.....	25
5.2.1	Preliminary FLUENT® Runs	25
5.2.2	Co-Simulation Runs	28
6.0	CONCLUSIONS AND RECOMMENDATIONS	31
7.0	REFERENCES	33
8.0	DELIVERABLES.....	35

LIST OF FIGURES

Figure 1:	BSF 1994 Geometry Configuration	8
Figure 2:	BSF 1994 Geometry With Wall Heat Flux Contours.....	9
Figure 3:	BSF Process Model	12
Figure 4:	Instantiation of the BSF CFD Block	13
Figure 5:	SOx Model Convergence Behavior.....	23
Figure 6:	Absorption as a Function of the Recycle Ratio for Two Grid Sizes.....	26
Figure 7:	Absorption as a Function of the RR With Manual Recycle of the Outlet Stream	27
Figure 8:	CFD Convergence Behavior as Indicated by Monitors.....	28
Figure 9:	Comparison of the Stand-Alone FLUENT® Runs with the Co-Simulation Results	31

LIST OF TABLES

Table 1:	Principal Design Specs Used for the BSF Feeds in the Flow Sheet.....	11
Table 2:	FLUENT® Block Ports and Connecting Streams	13
Table 3:	Inlet Flow-Splitter Specification.....	14
Table 4:	Species Listing and Mapping.....	16
Table 5:	CAPE-OPEN Parameters As Implemented Through APECS Wizard	16
Table 6:	Reaction Listing.....	19
Table 7:	Co-Simulation Results.....	29

1.0 BACKGROUND ON APECS PROGRAM

The current APECS program is a continuation of an earlier (Vision 21) program, which was initiated under the DOE-Fluent Cooperative Agreement No. DC-FC26-00NT40954 (hereafter referred to as Phase I). Both programs have been dedicated to the development of the software infrastructure for advanced power plant simulations. The goal of the overall program work scope has been to develop an integrated suite of software tools that could be used to simulate and visualize advanced plant concepts that are being considered under the umbrella of DOE's broad CO₂ remediation and power plant efficiency programs. Advanced simulation tools are needed to evaluate new power plant concepts that will undoubtedly help to minimize costly laboratory and field trials.

The typical process simulation software that has been commercially available to industry does not meet the DOE's objective of "virtual simulation" that is needed to evaluate complex cycles. The intent of the DOE has been to improve predictive tools for cycle analysis, and to improve the component models that are used in turn to simulate the cycle. Generally, the modeled performance of various components is derived from a simple set of reaction and output prescriptions that are collectively accessed by the cycle simulation software as a "black-box" library module, without any explicit impact from geometry modifications and enhancements to the fluid mechanics. These simple component models are not sufficient for component design; nor are they adequate for the evaluation of the impact of complex components on cycle performance. To meet the DOE's goal of predicting and visualizing the performance of these complex systems, there has been a need to use more sophisticated component models, by upgrading or converting library modules to CFD modules, and to couple the CFD modules directly with other equipment items in the power plant through process modelling software. The primary focus of the present APECS project has been to provide the software interface and environment that permits FLUENT® or a surrogate ROM to run in tandem with the Aspen Plus® process simulation package (see Ref. 1 for an overview of the program).

2.0 INTRODUCTION

The feasibility of using FLUENT® CFD models in concert with a process model like Aspen Plus® has been a focus of the APECS project. The role of Alstom Power Inc. (Alstom) has been to assist the project team in helping to develop and demonstrate the capabilities of the advanced APECS simulation and visualization tool, oftentimes with the use of Alstom designed and validated equipment modules (see e.g., Ref. 2 to 4). The primary Alstom responsibilities and tasks have included:

- providing its expertise and experience base in the utilization of both CFD and cycle analysis for the power generation industry,
- selecting and running cases to test and demonstrate the feasibility of the concepts,
- forming an advisory board to provide project review and feedback.

Task 5.2b, of the current APECS project, was added as an addendum to Task 5.2 (see Project Management Plan (Ref. 5)). Alstom's responsibilities associated with Task 5.2b, as itemized within the Project Management Plan, are to:

- Conduct an APECS study of the Alstom Boiler Simulation Facility (BSF) island (gas side only) with oxygen firing and flue gas recycling.
- Modify the APECS software to expose the pollutant species to Aspen Plus® and invoke the special NO_x and SO_x prediction capabilities in FLUENT®.
- Characterize the impact of various flue-gas recycle and cleanup scenarios on pollutant emissions for a candidate boiler configuration.

The deliverables include a task report, as well as all model files. The task report contains a description of the case simulations, the results, and an itemization of desired software modifications for future work. This report summarizes and documents the demonstration case simulations under Task 5.2b, and thus completes the requirement for the stipulated (contractual) task report.

3.0 EXECUTIVE SUMMARY

The primary objective of this DOE project is to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of Advanced Process Engineering Co-Simulator (APECS), the simulator will link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. Generally, the simulator couples the cycle analysis software Aspen Plus® (marketed by Aspen Technology, Inc.), specialized component modeling packages, as exemplified by industrial proprietary codes (utilized by Alstom Power), and the FLUENT® computational fluid dynamics (CFD) code (provided by ANSYS Inc). An important enhancement to the APECS tool is the creation of computationally efficient reduced-order models (ROMs) based on information from high-fidelity CFD models.

In this present task, the objective was to showcase and demonstrate an important advancement in the APECS tool kit – the addition of SO_x and NO_x emissions calculations to an integrated FLUENT® CFD block execution. An oxy-fired BSF boiler was successfully run with sequential emissions calculations in an Aspen Plus® flow sheet. ANSYS modified the APECS software to expose the NO_x pollutant species to Aspen Plus®. Alstom Power was tasked with the responsibility to (a) conduct an APECS study of the Alstom Boiler Simulation Facility (BSF) island (gas side only) with oxygen firing and flue gas recycling, (b) invoke the special NO_x and SO_x prediction capabilities in FLUENT®, and (c) characterize the impact of flue-gas recycle and cleanup on pollutant emissions for a candidate boiler configuration. This was accomplished.

A demonstration case was completed in which a CFD model of the BSF furnace was coupled with a flow sheet of the BSF facility and run in an integrated simulation. In this simulation, the flue gas recycle ratio was varied within a design spec in order to match a target furnace absorption. SO_x and NO_x emissions calculations were invoked with each execution of the BSF CFD block. The NO_x calculations were generated by the NO_x submodel in FLUENT®, which is typically run in a post-processor mode. For SO_x calculations, a SO_x mechanism was solved with the laminar chemistry option in FLUENT®. To handle the different reaction models in an automated fashion, a journal file was constructed to sequentially (a) run the Eddy Dissipation model for the main combustion

species, (b) freeze the flow field and run the NO_x model for a prescribed number of iterations, (c) activate the laminar chemistry model and the SO_x species transport equations and then run the SO_x model for a prescribed number of iterations, and then (d) activate the appropriate equations and options to be able to run the Eddy Dissipation model again.

The journal file approach functioned reasonably well and allowed the FLUENT® case to be run remotely on a Linux workstation on multiple processors. Both NO_x and SO_x values were calculated at the targeted wall absorption value for oxy-firing and compared with the corresponding (stand-alone) air-fired case results. While the SO_x model was problematic (i.e., did not conserve elemental sulfur), the APECS tool kit orchestrated the solution sequence in a robust fashion.

Generally speaking, significant progress was made by ANSYS in facilitating the direct instantiation of FLUENT® on an Aspen Plus® flow sheet, in conjunction with the sequencing and running of the pollutant models. It is anticipated that the present documented experience, as well as the associated journal files and explanation of the workflow, will help other others in their project efforts to utilize the pollutant models in subsequent co-simulations.

Recommendations are as follows:

- Develop the requisite suite of pollutant-specific Cape-Open variables and Booleans that would permit a user to toggle the pollutant models off and on, specify the number of iterations for each model, etc.
- ANSYS should provide a robust and reliable SO_x mechanism and submodel. A validated and CPU-efficient mechanism should be pursued by ANSYS, industry, and interested universities.
- Provide the proper threads and software enhancements to allow the user to interface with or interact directly with FLUENT® when the FLUENT® job is spawned to a remote Linux platform.
- Permit the user to activate an option to save the FLUENT® case and data files in the “temp” directory every time it successfully executes (perhaps overwriting the previous case and data file).
- Upgrade the flow-splitter capability to include DPM inlet ports.
- Particle injections are frequently found in APECS simulations and the current methodology used in the UDFs to monitor the particle trajectory information at the outlet plane cannot be properly parallelized. ANSYS should provide a template with the proper parallelization directives.

4.0 EXPERIMENTAL WORK

There is no experimental work under this project.

5.0 RESULTS AND DISCUSSION

5.1 Case Setup and Preparation

5.1.1 FLUENT® Case Grid and Setup

For the FLUENT® model of the BSF, the BSF firing configuration, nozzle elevations, and inlet free areas correspond to Test 2d from the 1994 BSF test campaign (Ref. 6). The tests were conducted with a high-volatile-B (HvB) bituminous coal (Lynnville). A TFS-2000 windbox configuration, with 3 coal elevations and two elevations of SOFA, was employed in the tests. The CFS tips were set at 10° from the main firing angle.

The 1994 BSF test configuration was selected because it provides somewhat higher tip inlet velocities than those used in other test campaigns. In applications of oxy-firing and recirculated flue gas to boilers, the mass of N₂ normally present in the boiler in air-fired scenarios is displaced by an equivalent mass of CO₂ (approximately) in oxygen-fired scenarios. Since the molecular weight of CO₂ is greater than that of N₂ by a factor of approximately 1.57, the density of CO₂ will be correspondingly greater than that of N₂, and the velocity of the recirculated flue gas streams through the windbox nozzles will therefore be smaller (by, e.g., 36% for pure streams) than that of the air-fired case (for the same nozzle inlet free area). Since the air-fired and oxy-fired cases were to be compared in this present study on an equivalent inlet-area basis, rather than on an equivalent inlet-velocity basis, it was deemed prudent to use the BSF configuration that had the higher inlet velocities to help prevent any tendency of the particles to settle out, particularly at the bottom of the hopper.

Some of the CFD case characteristics and selected submodel options are itemized below:

- FLUENT® V6.3.31 was used for all of the preliminary BSF computations, although V12.0.16 was used for the final baseline and APECS-related runs.
- Cases with both fine and coarse grids were developed. The fine grid has 3,319,222 cells, while the coarse grid, used in the coupled APECS simulation, had 247,742 cells. The coarse grid case contained some hanging-node adaption, using boundary adaption, near the inlets. In transitioning from the fine grid to the coarse grid, some of the geometrical features were simplified. For example, the inlet ducts in the fine grid were removed for the coarse grid. In addition, the multiple SOFA tips in the USOFA and LSOFA compartments were collapsed into a single opening in the coarse grid, but no penalty was incurred in doing that because all of the tips in each of the SOFA compartments were at the same velocity anyway. Neither the fine nor coarse grids have non-conformal interfaces. The coarse grid was originally constructed with non-conformal interfaces near the inlets, but it was later found the interfaces, at least for discontinuous jumps in cells sizes on the order of 3-to-1 (or more), caused the species to not be conserved. Therefore, all non-conformal interfaces were removed in favour of a single, composite hexahedral and tetrahedral mesh. This latter mesh is subject to higher skew in some regions, but does not suffer from the blatant conservation issues of the non-conformal interfaces.

- The solution algorithm is based upon the SIMPLE pressure-correction algorithm with first-order upwind differencing.
- The turbulence model is the Realizable k- ϵ turbulence model, with non-equilibrium wall functions
- For the main combustion reactions, the turbulence-chemistry interaction was modeled using the composite Magnussen eddy-dissipation (mixing-limited / fast-chemistry) and Arrhenius rates (kinetically-limited) combustion submodel, in which the smaller of the two rates (eddy-dissipation or Arrhenius rates) takes local precedence.
- The gas-phase stoichiometry for the main combustion reactions was based upon three-step, irreversible, global reactions. The heterogeneous reactions consisted of oxidation and gasification reactions, using the multi-surface reaction model. (The reaction sets are shown subsequently in Table 6.)
- The SO_x model, discussed subsequently, was based upon a 10-step mechanism, each of the 10 forward steps having an associated reversible step.
- The species transport equations were activated or deactivated during each execution of the FLUENT® BSF case, depending upon the specific reaction model that is invoked:
 - Main combustion reactions: 9 species (coal volatile, O₂, CO, CO₂, H₂O, H₂S, H₂, SO₂, N₂).
 - NO_x model reactions: 2 species (pollutant NO, pollutant HCN).
 - SO_x model reactions: 5 species (H₂S, SO₂, SO, SH, SO₃).
 - Radical species: 3 species (O, OH, and OH). An on-demand UDF was used to solve for the radical species at each cell immediately prior to the computations of the SO_x model, so that the SO_x model mechanism would have access to those species concentrations. Transport equations were provided for the three radical species in order to hold the computed values, but were never activated.
 - Additional species required for particle injections: 3 species (C<s>, S<s>, and SiO₂). Upstream of the FLUENT® block, in integrated simulations with Aspen Plus®, the coal (NC (non-conventional)) substream is converted into a MIXED substream, consisting of various species, including C<s>, S<s>, and SiO₂. C<s> is a surrogate species for elemental carbon, S<s> is a surrogate for elemental sulfur, and SiO₂ is a proxy for ash. This MIXED substream then transfers the mass flow rate and temperature information to the boundary conditions for the discrete particle model (DPM) injections in FLUENT®. Similarly at the outlet plane of the FLUENT® block, the residual particle flow rates, temperatures, and component mass fractions of the unburned carbon, sulfur, and ash are transferred to a MIXED substream on the Aspen Plus® side. Within the MIXED substream, the C<s> receives the flow rate value of the unburned carbon, S<s> receives the flow rate value for the unconsumed sulfur, and the SiO₂ receives the flow rate value for the ash. Transport equations were provided for the three radical species, but were never activated; the values across the computational domains for these species are identically zero, except possibly at the inlet or outlet boundary where the transferral of flow rates and temperatures occur.
- The discrete particle model in FLUENT® is a stochastic Lagrangian particle model. Surface injection was used for the particles at 3 elevations of 4 corners each (for a total of 12 injection inlets). A Rosin-Rammler (logarithmic) distribution of 40 particle sizes was specified, along with a single stochastic “try,” which ultimately produced approximately

5,760 total particle injections (in the coarse grid). A single stochastic try was used because of a code bug in V12.0.16 which precluded multiple tries. Particle-radiation coupling was activated. A proprietary user-defined function (UDF) was used to calculate the particle absorption and scattering efficiencies. All of the coal moisture was presumed to have evaporated completely before entering the boiler and was given to the primary transport air.

- The devolatilization submodel was a Kobayashi-type (two-competing rates) coal devolatilization model. For the 1994 tests, the high-temperature volatile yield parameter was estimated using the commercially available Niksa Energy Associates (NEA) PC COAL LAB V3.0 software (i.e., FLASHCHAIN model).
- The radiation submodel was the Discrete Ordinates radiation model, with a mixture scattering coefficient of 0.0. A proprietary UDF was used to extract the local gas-phase absorption coefficient.
- The NO_x model calculations were based upon the existing NO_x model in FLUENT®, exercised in a post-processing mode (i.e., with all other transport equations frozen except those corresponding to the two pollutant species). Thermal, prompt, and fuel NO_x options were activated, along with the reburn option. The NO_x species were run with 2nd-order upwind numerics. The values for the volatile and char nitrogen mass fraction were determined by PC Coal Lab.
- All of the tube assemblies, banks, and exterior walls are evaporative or sub-cooled sections; the BSF does not have a superheat section. The BSF, in the 1994 configuration, has 3 pseudo wing-wall assemblies emanating from the front wall, and 3 platen assemblies directly over the arch. It also has an economizer section in the backpass. The heat absorption by the single economizer section, represented in FLUENT® by a porous medium, was specified through a volumetric heat sink term for the porous medium. The exterior walls were given a specified overall heat transfer coefficient, a specified backside (coolant) water/steam temperature, and an emissivity. The overall heat transfer coefficients for the walls were set using a calibration procedure; the overall heat transfer coefficients of the various walls are iteratively adjusted until the desired horizontal furnace outlet temperature (HFOT) is achieved. In the present case, the HFOT was arbitrarily set to be equal to 1674 K (a temperature representative of the typical HFOT in commercial boilers). This HFOT was set arbitrarily high (i.e., higher than the actual HFOT in the BSF) in order to accentuate the impact of any thermal NO_x. It should also be noted that, in reality, portions of the external walls are surfaced with refractory. For the present computations, the overall heat transfer coefficient for all of the walls was prescribed to be uniform.
- The inlet boundary conditions were prescribed through a mass-flow rate boundary specification. Where the inlet flow direction was specified to be at some angle other than at the normal to the inlet boundary plane, appropriate velocity vectors were calculated to compensate for the bias. For the BSF tests, all burners were in service.

A view of the 1994 BSF geometry configuration, based on the “fine” grid construction, is shown in Figure 1, and a semi-transparent contour plot of the total surface heat flux is provided in Figure 2.

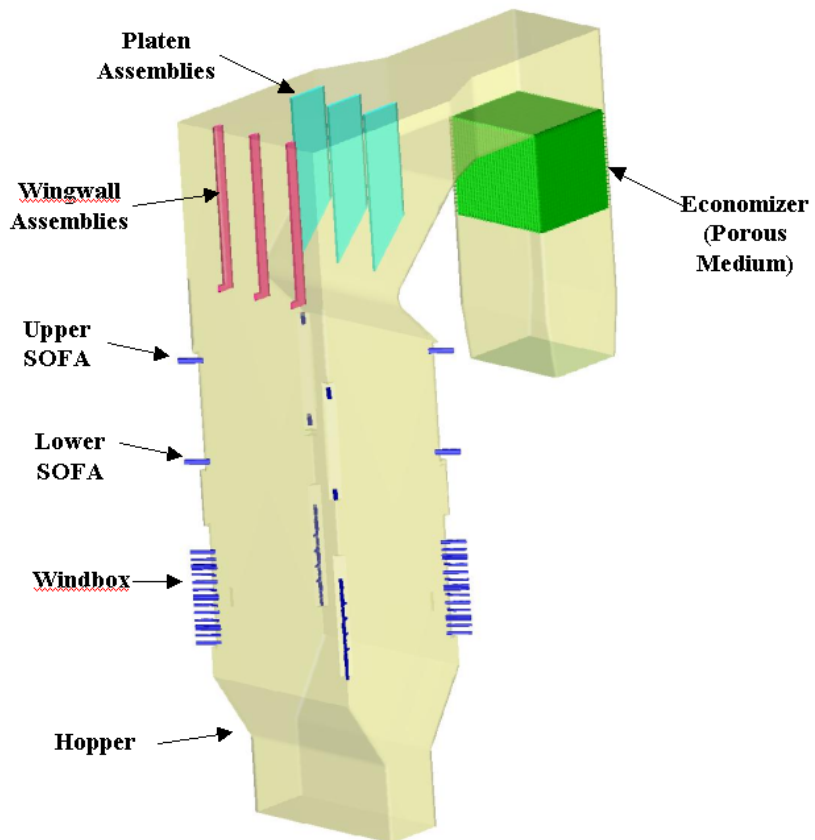


Figure 1: BSF 1994 Geometry Configuration.

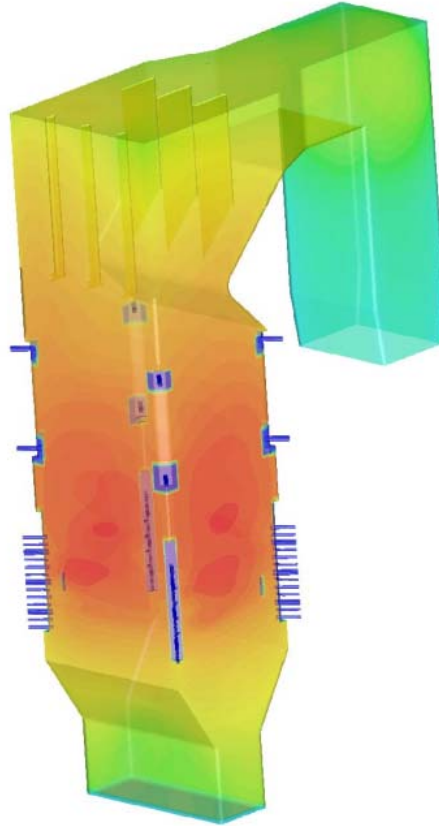


Figure 2: BSF 1994 Geometry With Wall Heat Flux Contours.

5.1.2 SO_x Model Development and Background

The FLUENT® code has a custom SO_x model, that is similar to the NO_x model, in that it can be run in a post-processing mode (i.e., it can be run by freezing the flow field and deactivating all transport equations other than those use explicitly by the SO_x models). Experimentation with the SO_x model in FLUENT® V6.3.31 in February of 2008 indicated that, for reasonable convergence times, the model was unable to predict high SO_x concentrations commensurate with levels found in pulverized-coal fired boilers. It was subsequently determined that the sulfur kinetic mechanisms were very stiff and that the SO_x model, as then constituted, failed to converge because it was not using a stiff chemistry solver.

Given the CPU-intensive nature of applying a stiff chemistry solver, some effort was made to find a simplified or global mechanism. However, other than one reference in the literature involving a PDF/equilibrium type approach and a single global kinetic step for SO₂ to SO₃ (see Ref. 7), simple global or reduced sulfur chemistry approaches, in which the kinetic rates were disclosed, were not found in the open literature. Zhuyin Ren and Stefano Orsino at ANSYS analyzed SO_x kinetic mechanisms, based on typical boiler temperatures and species concentrations, and determined that the SO_x chemistry was indeed very stiff, and that quasi steady-state or partial equilibrium assumptions, with a goal of producing some form of algebraic reduced or global mechanism, would

not be advisable. It was consequently determined that, given the task mandate of computing SO_x emissions, a SO_x mechanism would have to be implemented with a stiff chemistry solver (in conjunction with the ISAT table lookup capability) within FLUENT®.

After various discussions with ANSYS on funding approaches, a request was made by ALSTOM for ANSYS to begin work on the SO_x model in early October of 2008. This work scope had two facets:

- The workscope for the Subtask 5.2b Addendum required a reliable and robust SO_x model for the FLUENT® calculations. ANSYS proposed a sequential procedure. First, the main combustion process in the flow field would be solved with one of the existing approaches in FLUENT® (e.g., the Magnussen Eddy Dissipation method). Following the solution of the main combustion species, the flow, turbulence, energy and major species fields would be frozen, and the radical concentrations (for O, OH, and H) would be calculated based on equilibrium or partial-equilibrium assumptions. Source terms for the gaseous sulfur species, based on the coal devolatilization and char reaction rates, would be provided via the appropriate user defined functions (UDFs) if V6.3 were used, or via newly developed functionality if V12.0 could be used. SO_x mechanisms would then be converged with a stiff chemistry solver (either as part of the laminar chemistry or Eddy Dissipation Concept submodels) in FLUENT®. It was agreed that ANSYS would fund all of the enhancements required to provide a SO_x model (in order to protect and secure ANSYS intellectual property).
- Additional modifications and upgrades to the APECS software (by the ANSYS development team) were also required. Since the pollutant species in FLUENT® were not part of the “species list” exposed to the process modeller environment (PME), ANSYS developers would need to modify the FLUENT® wrapper in order to expose the pollutant species to the PME (e.g., Aspen Plus®). In principle, both the NO_x and SO_x mechanisms or submodels could be run concurrently with the main combustion species and converged with the main flow field (depending on the model used). However, it was anticipated to be more effective and CPU-efficient to execute the NO_x and SO_x submodels in post-processing mode only. In this latter instance, additional coding would be required to “freeze” and “unfreeze” the main flow field variables at the appropriate junctures in the calculation sequence, and in turn, to initiate or terminate the NO_x and SO_x submodel calculations.

For the SO_x mechanism, ANSYS selected a 10-step mechanism, each of the 10 forward steps having an associated reversible step (see details in Ref. 8 and 9). The reverse rates are prescribed explicitly or are calculated internally and automatically by FLUENT® when the user activates the “backward rate” option for a particular reaction rate. The mechanism does not include absorption of SO₃ or other sulfur species by limestone or by components in the ash (e.g., calcium), nor does it include catalytic reaction of the SO₃ with exposed metals (e.g., iron or vanadium). ANSYS prepared the mechanism and imported it into the FLUENT® BSF case. A total of five transport equations for sulfur-related species must be solved with the model – SO₂, H₂S, SO, SH, and SO₃.

ANSYS suggested the use of FLUENT V12.0.16 as the version of choice for the SO_x model, because V12.0.16 already contained many of the features required for the sulfur model (such as the accommodation of sulfur in the multiple surface reaction surface model, and the associated

devolatilization and char source terms, as well as certain chemistry solver enhancements), which obviated the need for substantial UDF development. A UDF was developed by ANSYS, which contained partial equilibrium expressions for the radicals O, OH, and H, based on equations that ANSYS had already used in other subroutines, such as the NO_x model, etc.

5.1.3 BSF Process Model

A process model was developed using the Aspen Plus® code for the BSF and the pertinent equipment items. Initially, an RGIBBS reactor was used for the BSF in order to help construct the flowsheet and to refine the convergence characteristics. A calculator block was used to ascribe to the RGIBBS reactor the appropriate overall heat absorption as a function of the recycle ratio. The principal design specs used to provide the correct boiler flows are found in Table 1.

Table 1: Principal Design Specs Used for the BSF Feeds in the Flow Sheet

Aspen Plus® Design Spec	Description
DS-O2PRI	The O ₂ injected into the primary stream is varied until the O ₂ mole fraction is equal to 0.213. (Coal flow rate is fixed.)
DS-O2TOT	The total O ₂ entering the boiler is varied until the desired O ₂ /coal mass ratio is equal to 2.50935 (corresponds to the air-fired case)
DS-RATIO	The volumetric recycle ratio is varied until a target total boiler absorption is achieved. (For the air-fired case, the total absorption is 1.0314E+07 W (3.52164E+07 Btu/hr).)
DS-TPRIM	The flow rate of the primary tempering bypass gas is varied until the temperature of the primary stream is equal to 140 °F.
DS-TRANS	The flow rate of the primary stream is varied until the desired transport gas/coal mass ratio is equal to 2.0409 (corresponds to the air-fired case).

The coal flow rate in the present process model is fixed. The total oxygen entering the boiler is based on a fixed oxygen/coal mass ratio (DS-O2TOT). Following Ref. 10, the total recycled flue gas is found by varying a volumetric recycle ratio until a target total boiler absorption is achieved (DS-RATIO). The equation for the volumetric recycle ratio is:

$$RR = \left[\frac{\text{total molar flow rate of recycle streams (prior to } O_2 \text{ addition)}}{\text{total molar flow rate of } O_2 \text{ entering boiler}} \right] \quad (1)$$

where RR = volumetric recycle ratio.

The process model is shown in Figure 3. The coal is first pulverized and then stored in silos prior to use (the pulverization mills and silos are not shown on the flow sheet). The oxygen is stored as liquid oxygen in tanks (also not shown), and then heated and then added to the recirculated flue gas streams that make up the primary and secondary boiler streams. Heat exchangers are used to cool the flue gas and to preheat the primary, secondary, and fluidization streams. The fluidization streams are used in the integrated flash dryer absorber (FDA), in which it is presumed that 90% of all sulfur species (by weight) are captured, and in the baghouse, which removes the particulate. A

condenser, used on a portion of the primary gas feed stream, removes much of the moisture, although some moisture is again added to the stream when the primary tempering gas stream is added to the stream to increase the temperature to approximately 140 °F. A wet scrubber removes additional emissions prior to being exhausted out the stack.

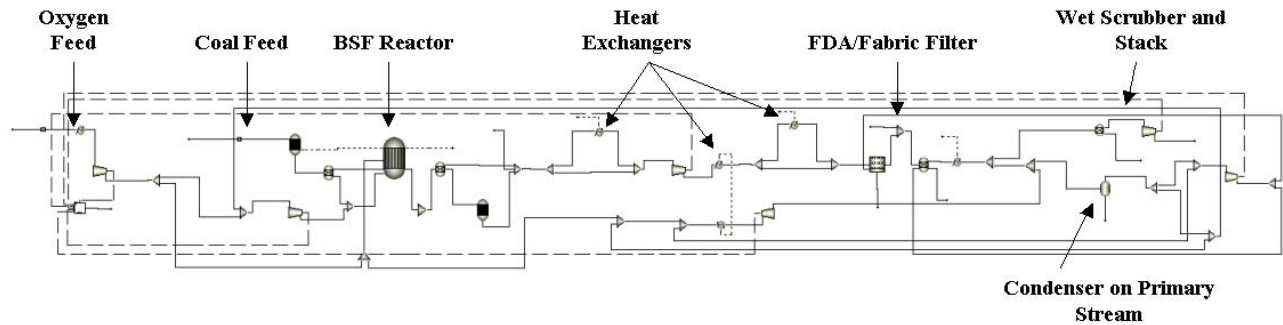


Figure 3: BSF Process Model.

The “sequential-modular” solution approach was used in all cases, as opposed to the “equation-oriented” solution methodology. All of the Design Specs are solved using the secant method (as opposed to the Broyden method). Aspen Plus® creates a “tear” stream downstream of the FDA, which is solved using the Wegstein method in the outermost loop. Fortunately, the BSF block is solved within the next inner loop (based on the DS-RATIO design spec), so that it is executed a manageable number of times. The tolerance on the tear stream greatly controls the number of BSF block executions; accordingly, to decrease the number of BSF block executions, the tolerance was changed from the default value of 0.0001 to 0.001.

5.1.4 Ports and Connecting Streams

To prepare the flow sheet to receive the FLUENT® CFD block, additional feed streams must be added and RYIELD reactors must be added to exchange mass between the MIXED and NC substreams. This new arrangement, modified for the CFD block, is shown in Figure 4.

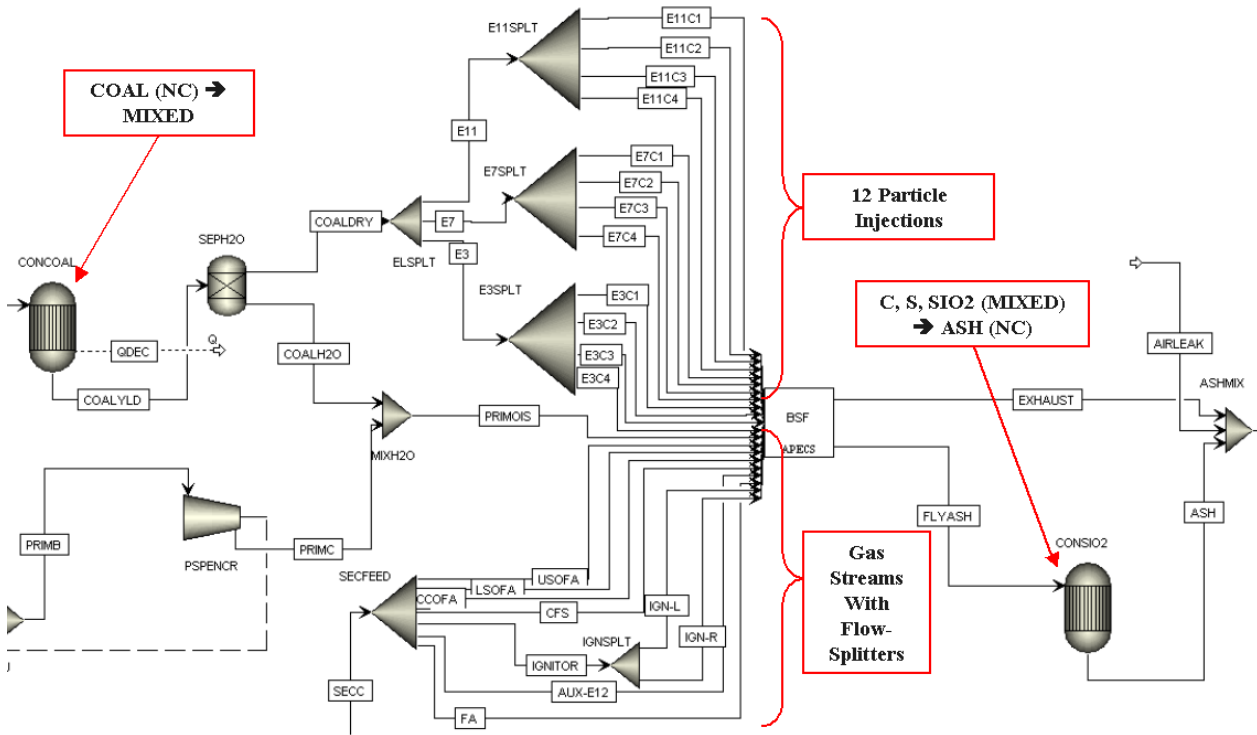


Figure 4: Instantiation of the BSF CFD Block.

The various ports and stream connections for the FLUENT® block, all of which are associated with a single domain (Domain-1), are itemized as follows:

Table 2: FLUENT® Block Ports and Connecting Streams

FLUENT Computational Domain Port	Inlet or Outlet Port	Description of Port	Type of Port	Corresponding Aspen Plus® Stream Name
COAL	inlet	primary or transport air/gas	GCP: F-S	PRIMOIS
USOFA-E15	inlet	upper separated overfire air/gas (SOFA)	GCP: F-S	USOFA
LSOFA-E14	inlet	lower separated overfire air/gas (SOFA)	GCP: F-S	LSOFA
CCOFA-E13	inlet	close-coupled overfire air/gas (CCOFA)	GCP: F-S	CCOFA
CFS	inlet	concentric firing system (CFS) air/gas	GCP: F-S	CFS
AUX-E12	inlet	auxiliary air/gas	GCP: F-S	AUX-E12
FA	inlet	fuel air/gas	GCP: F-S	FA
IGNITOR-L	inlet	ignitor (left wall) air/gas	inlet port	IGN-L
IGNITOR-R	inlet	ignitor (right wall) air/gas	inlet port	IGN-R
INJECTION-E3C4	inlet	coal DPM injection (Elev 3, Corner 4)	DPP	E3C4
INJECTION-E3C3	inlet	coal DPM injection (Elev 3, Corner 3)	DPP	E3C3
INJECTION-E3C2	inlet	coal DPM injection (Elev 3, Corner 2)	DPP	E3C2
INJECTION-E3C1	inlet	coal DPM injection (Elev 3, Corner 1)	DPP	E3C1
INJECTION-E7C4	inlet	coal DPM injection (Elev 7, Corner 4)	DPP	E7C4
INJECTION-E7C3	inlet	coal DPM injection (Elev 7, Corner 3)	DPP	E7C3

INJECTION-E7C2	inlet	coal DPM injection (Elev 7, Corner 2)	DPP	E7C2
INJECTION-E7C1	inlet	coal DPM injection (Elev 7, Corner 1)	DPP	E7C1
INJECTION-E11C4	inlet	coal DPM injection (Elev 11, Corner 4)	DPP	E11C4
INJECTION-E11C3	inlet	coal DPM injection (Elev 11, Corner 4)	DPP	E11C3
INJECTION-E11C2	inlet	coal DPM injection (Elev 11, Corner 4)	DPP	E11C2
INJECTION-E11C1	inlet	coal DPM injection (Elev 11, Corner 4)	DPP	E11C1
OUTLET	outlet	outlet for gas flow	outlet port	EXHAUST
DPM-OUTLET	outlet	outlet for DPM injections	DPP	FLYASH
DPP =	discrete phase port			
GCP: F-S =	generic custom port: flow-splitter			
Note:	At the time of case preparation, the software required the user to have at least 1 inlet port before generic custom ports could be created.			

The BSF actually has many more inlets than those prescribed in the table above. This reduction in the apparent number of inlet streams is facilitated by a very useful Generic Custom Port feature, called a “flow-splitter.” The flow-splitter feature allows a single Aspen Plus® stream, upon connection with a FLUENT® CFD block port, to be automatically partitioned or split (evenly) into any number of smaller streams, all of which are identified by having a common “prefix” name identical with that of the parent port. The flow splitter specifications are provided in Table 3.

Table 3: Inlet Flow-Splitter Specification

FLUENT® Computational Domain Port	Type of Port	Description of Port	Actual Number of FLUENT® Inlet Planes	Corresponding Names of FLUENT® Case Inlet Planes
COAL	GCP: F-S	primary or transport air/gas	12	coal-e#-c#
USOFA-E15	GCP: F-S	upper separated overfire air/gas (SOFA)	4	usofa-e15-c#
LSOFA-E14	GCP: F-S	lower separated overfire air/gas (SOFA)	4	lsofa-e14-c#
CCOFA-E13	GCP: F-S	close-coupled overfire air/gas (CCOFA)	4	ccofa-e13-c#
CFS	GCP: F-S	concentric firing system (CFS) air/gas	16	cfs-e4-c#
none (flow rate = 0)	none	auxiliary air/gas	4	aux-e2-c#
AUX-E12	GCP: F-S	auxiliary air/gas	4	aux-e12-c#
FA	GCP: F-S	fuel air/gas	12	fa-e#-c#
IGNITOR-L	inlet port	ignitor (left wall) air/gas	1	ignitor-l
IGNITOR-R	inlet port	ignitor (right wall) air/gas	1	ignitor-r
none (flow rate = 0)	none	lower auxiliary air/gas	4	laux-e1-c#
none (flow rate = 0)	none	lower auxiliary air/gas	4	laux-e5-c#
none (flow rate = 0)	none	lower auxiliary air/gas	4	laux-e9-c#
GCP: F-S =	generic custom port: flow-splitter			
Note:	For flow-splitters only: each base name is added under “Add Option” to the “flow-splitters” parameter definition.			

The BSF flowsheet has 3 substreams:

- an NC (non-conventional) substream, consisting of coal and ash

- a CISOLID (conventional inert solids) substream, consisting of CARBON and SULFUR
- a MIXED substream consisting of 20 species: CH₄, O₂, CO₂, CO, H₂O, SO₂, H₂, H₂S, CARBON, SULFUR, SiO₂, SH, SO, SO₃, N₂, NO, HCN, AR, COS, and NH₃ (where the species are displayed with the names used by Aspen Plus®).

Note that CARBON and SULFUR are in both the CISOLID and the MIXED substreams.

It is a general requirement that all NC or CISOLID substreams must be converted into a MIXED stream before said stream reaches the FLUENT® block. This is due in part to limitations within the prevailing CAPE-OPEN standard, as well as to limitations in the implementation of the standard and solid substreams in Aspen Plus®. In like manner, the streams exhausting from the FLUENT® block are mandated to be MIXED streams only, and must be converted to NC or CISOLID streams, if such streams are desired downstream of the FLUENT® block.

Upstream of the FLUENT® block, the NC coal enters the system through stream PCBASE. The NC coal is passed through an RYIELD reactor (non-stoichiometric reactor based on known yield distribution) named CONCOAL, and the coal is turned into a MIXED substream, consisting of non-zero values of CARBON, H₂O, H₂, N₂, SULFUR, O₂, and SiO₂ (i.e., surrogate for ash), in proportions based on the ultimate analysis of the coal. However, the actual component yields do not matter, because only the total mass flow rate and the temperature are passed on to the FLUENT® Lagrangian particle injections through the DPM physical model ports. The resulting converted MIXED substream is split into the material streams named for the 12 DPM injection inlet planes (INJECTION-E#C#) and routed to the FLUENT® block.

At the physical model outlet port (DPM-OUTLET), a MIXED substream consisting of finite values of CARBON (surrogate for unburned carbon), SULFUR (surrogate for unconsumed sulfur in the char), and SiO₂ (surrogate for ash) are passed to the connecting material stream FLYASH. Again an RYIELD reactor (called CONSIO₂) is used to convert the MIXED substream components (CARBON, SULFUR, and SiO₂) into the NC component ash. Hence in this flow sheet the NC ash contains the residual unburned C_s and S_s.

The CAPE-OPEN variables for the DPM physical model outlet ports (DPM-OUTLET), as defined and stored within the FLUENT® case file, along with current (or default) values (SI units), are given below:

```
(cape-open/dpm/dpm-outlet/flow-rate 0.094777271)
(cape-open/dpm/dpm-outlet/temperature 833.42957)
(cape-open/dpm/dpm-outlet/mass-fracs "(0 0 0 0 0 0 0 0 0 0.000021 0.000360
0.999632 0 0 0 0 0 0 0 0 0)")
```

The series of mass fraction values are those corresponding to the sequence of 20 species listed in the FLUENT® case under the Materials → Species panel (see the left-hand column of Table 4). Other than the mass fractions, only the mass flow rate and temperature are required to complete the definition of the FLUENT® block physical model outlet ports.

The species listed in both the FLUENT® case and the Aspen Plus® flow sheet, as well as their respective mappings to each other, are provided in Table 4.

Table 4: Species Listing and Mapping

	Species in the Materials Listing in FLUENT®	FLUENT® Species Exposed to the PME	Flow Sheet Species
(1)	vol	vol	CH4
(2)	o2	o2	O2
(3)	co2	co2	CO2
(4)	co	co	CO
(5)	h2o	h2o	H2O
(6)	so2	so2	SO2
(7)	h2	h2	H2
(8)	h2s	h2s	H2S
(9)	cs2	cs2	CARBON
(10)	ss2	ss2	SULFUR
(11)	sio2	sio2	SIO2
(12)	sh	sh	SH
(13)	so	so	SO
(14)	h (private [Note])	so3	SO3
(15)	oh (private [Note])	n2	N2
(16)	o (private [Note])	pollutant-no	NO
(17)	so3	pollutant-hcn	HCN
(18)	n2	Zero Flow Rate Species No. 1	AR
(19)	pollutant-no	Zero Flow Rate Species No. 2	COS
(20)	pollutant-hcn	Zero Flow Rate Species No. 3	NH3
Note:	The private species are not exposed to the PME (Aspen Plus®); their flow field values, in the present case, are added to those of the public species “n2”.		

The Cape-Open variables for the DPM physical model ports are set in the “DEFINE_DPM_BC(bc_outlet, p, t, f, f_normal, dim)” UDF (in file wallabs.c) through the lines:

```

sprintf(dpm_mass_fracs, "(0 0 0 0 0 0 0 0 0 %f %f %f 0 0 0 0 0 0 0 0 0)",
massFractionC, massFractionS, massFractionAsh);
RP_Set_String("cape-open/dpm/dpm-outlet/mass-fracs", dpm_mass_fracs);
RP_Set_Real("cape-open/dpm/dpm-outlet/flow-rate", part_outlet_mass_flux);
RP_Set_Real("cape-open/dpm/dpm-outlet/temperature", part_outlet_temperature);

```

The Cape-Open parameters set through the APECS Wizard are provided in Table 5.

Table 5: CAPE-OPEN Parameters As Implemented Through APECS Wizard

Parameters	Type	Description	Access Mode	Lower Bound	Upper Bound	Units	Default Value
end-user-journal-activate	integer	activates the user journal file	read-write	0	1	---	1
end-user-journal-name	option	name of user journal file	read-write	---	---	---	sxo-apecs-journal.jou
pm-energy-const	real	absorption in economizer section	read-write	-1E+10	0	W/m ³	-190,000

energy-sink-pm	real	absorption in economizer section	read-write	1	1E+10	W	1,374,640
heat-flux-wall	real	total wall absorption	read-write	1	1E+10	W	9,004,240
wall-absorption	real	total boiler absorption	read-write	1	1E+10	W	10,378,881
calculate-user-journal-activate	integer	activates the user journal file	read-write	0	1	---	1
calculate-user-journal-name	option	name of user journal file	read-only	---	---	---	prep_resid.jou
flow-splitters	option	names of flow splitters	read-write	---	---	---	aux-e12, ccofa-e13, lsofa-e14, usofa-e15, cfs, coal, fa
maximum-iterations	integer	maximum number of FLUENT® iterations	read-write	1	100000	---	1000
model-views	option	views for post-processing	read-write	---	---	---	aux-e12

The calculate-user-journal-activate and end-user-journal-activate parameters are actually Booleans, but a bug in Aspen Plus® has precluded the Booleans from being manipulated in a calculator block. Therefore, it is more convenient to treat the parameter as an integer switch (0=FALSE, 1=TRUE).

Four of the Cape-Open parameter variables are set in the “DEFINE_ADJUST(wall_abs, d)” UDF (in file wallabs.c) through the lines:

```
RP_Set_Real("cape-open/param/pm-energy-const", SVALUE);
RP_Set_Real("cape-open/param/energy-sink-pm", energy_sink);
RP_Set_Real("cape-open/param/heat-flux-wall", heat_flux);
RP_Set_Real("cape-open/param/wall-absorption", total_wall_absorp);
```

The wall absorption parameter (“wall-absorption”) is the total furnace absorption used in the design spec DS-RATIO. It is the sum of the heat flux to all of the wall surfaces (“heat-flux-wall”) plus the energy absorbed in the economizer porous medium section (“energy-sink-pm”). The energy absorbed in the economizer section is equal to the volumetric extraction rate (“pm-energy-const” in W/m³) multiplied by the volume of the economizer section (provided as a constant in the UDF).

5.1.5 Files for Integrated Co-Simulation

Software versions used in this task included the following:

- ANSYS EKM® V1.2
- FLUENT® V6.3.31 and FLUENT® V12.0.16 [Linux (64-bit)]
- Aspen Plus® V7.1 (with Cumulative Patch 1)
- ANSYS APECS V1.7.0.019 plus kit for Linux (64-bit)
- COFluent 12.0.16 Wrapper V1.7.0.004

All of the co-simulation runs were performed with Aspen Plus® as the executive process modelling software. The Aspen Plus® flow sheet was run in the PC Windows environment and the CFD cases were exercised remotely over a LAN on an AMD-64 dual quad-core (8-processor) Opteron box.

The FLUENT® case files were prepared and made CAPE-OPEN and APECS compatible by sequentially performing the steps outlined in the APECS User's Manual (Ref. 11):

- Execute FLUENT® CAPE-OPEN Configuration Wizard
- Execute APECS Model Wizard
- Upload files to the ANSYS Engineering Knowledge Manager (EKM®) database

The files uploaded to the EKM® database consist of the following:

GASTABLE_2p139_BSF.TXT
abs.c
bsf_33980.cas.gz
bsf_33980.dat.gz
calc-radicals.c
compile_udf_unix.jou
dpm_Qprop_bitum.c
gastable.txt
libudf/ntx86/3ddp/libudf.dll
sxo-apecs-journal.jou
view-0
wallabs.c
prep_resid.jou

The APECS software allows for a number of journal files, which are helpful in providing increased control to the run sequence. Each of the principal journal files, used in the present task, will be discussed in turn.

ANSYS constructed a journal file which is designed to sequentially (a) run the Eddy Dissipation model for the main combustion species, (b) freeze the flow field and run the NO_x model for a prescribed number of iterations, (c) activate the laminar chemistry model and the SO_x species transport equations and then run the SO_x model for a prescribed number of iterations, and then (d) activate the appropriate equations and options to be able to run the Eddy Dissipation model again.

ANSYS determined that it would be more efficient and versatile to turn the different combustion and mechanism models off and on via a journal file, rather than through Cape-Open variables. A Cape-Open Boolean variable does exist, i.e., "post-process-pollutant-models". However, this variable is somewhat limited in that: (a) it toggles between species that have the word "pollutant" in their name and those that do not, and (b) it uses the same number of iterations to solve the pollutants transport equations as were used to solve the main combustion reactions. Since in the present case (a) three different combustion models were being sequentially activated (main combustion, NO_x, and SO_x), (b) greater control was desired in being able to specify the number of iterations for each model, and (c) additional UDFs were executed or toggled off and on at different times, it was apparent that a journal file was required and that the Cape-Open variable for pollutants would not be used.

A knowledge of the reaction order becomes important in determining which reactions to activate and deactivate during the solution of the respective combustion, NO_x, and SO_x model computations. The 26 reactions specified in the FLUENT® case are provided in Table 6. The reactions are given in lower case font because that is how they appear in the FLUENT® case.

Table 6: Reaction Listing

Reaction Number	Reaction	Reaction Number	Reaction
(1) - forward	$\text{h}_2\text{s} + \text{h} \rightarrow \text{sh} + \text{h}_2$	(2) - reverse	$\text{sh} + \text{h}_2 \rightarrow \text{h}_2\text{s} + \text{h}$
(3) - forward	$\text{oh} + \text{h}_2\text{s} \rightarrow \text{h}_2\text{o} + \text{sh}$	(4) - reverse	$\text{h}_2\text{o} + \text{sh} \rightarrow \text{oh} + \text{h}_2\text{s}$
(5) - forward	$\text{so} + \text{oh} \rightarrow \text{h} + \text{so}_2$	(6) - reverse	$\text{h} + \text{so}_2 \rightarrow \text{so} + \text{oh}$
(7) - forward	$\text{sh} + \text{o} \rightarrow \text{so} + \text{h}$	(8) - reverse	$\text{so} + \text{h} \rightarrow \text{sh} + \text{o}$
(9) - forward	$\text{o} + \text{h}_2\text{s} \rightarrow \text{sh} + \text{oh}$	(10) - reverse	$\text{sh} + \text{oh} \rightarrow \text{o} + \text{h}_2\text{s}$
(11) - forward	$\text{so} + \text{o}_2 \rightarrow \text{so}_2 + \text{o}$	(12) - reverse	$\text{so}_2 + \text{o} \rightarrow \text{so} + \text{o}_2$
(13) - forward	$\text{h} + \text{sh} + \text{m} \rightarrow \text{h}_2\text{s} + \text{m}$	(14) - reverse	$\text{h}_2\text{s} + \text{m} \rightarrow \text{h} + \text{sh} + \text{m}$
(15) - forward	$\text{so} + \text{o} + \text{m} \rightarrow \text{so}_2 + \text{m}$	(16) - reverse	$\text{so}_2 + \text{m} \rightarrow \text{so} + \text{o} + \text{m}$
(17) - forward	$\text{so}_2 + \text{o} + \text{m} \rightarrow \text{so}_3 + \text{m}$	(18) - reverse	$\text{so}_3 + \text{m} \rightarrow \text{so}_2 + \text{o} + \text{m}$
(19) - forward	$\text{so}_3 + \text{o} \rightarrow \text{so}_2 + \text{o}_2$	[19 - reverse]	[not explicitly prescribed; backward rate option activated]
(20)	$\text{vol} + (0.974) \text{o}_2 \rightarrow (0.253) \text{co} + (0.886) \text{h}_2\text{o} + (0.024) \text{h}_2\text{s} + (0.023) \text{n}_2$		
(21)	$\text{co} + (0.5) \text{o}_2 + (0) \text{h}_2\text{o} \rightarrow \text{co}_2$		
(22)	$\text{h}_2 + (0.5) \text{o}_2 \rightarrow \text{h}_2\text{o}$		
(23)	$\text{csolid} + (0.5) \text{o}_2 \rightarrow \text{co}$ [Note]		
(24)	$\text{csolid} + \text{co}_2 \rightarrow (2) \text{co}$ [Note]		
(25)	$\text{csolid} + \text{h}_2\text{o} \rightarrow \text{h}_2 + \text{co}$ [Note]		
(26)	$\text{ssolid} + \text{o}_2 \rightarrow \text{so}_2$ [Note]		
Note:	Reactions (23) through (26) are particle surface reactions and describe reactions with the solid species C<s> and S<s>. The name csolid was used instead of C<s>, and the name ssolid was used instead of S<s>, in order to avoid any potential conflict with APECS rules (because angle-brackets are not accepted characters within the APECS environment).		

The above reaction number table is a useful reference in deciphering the text of the journal file. The journal file for activating and deactivating reactions is named “sxo-apecs-journal.jou.” It is activated via the Cape-Open Parameters:

- end-user-journal-name
- end-user-journal-activate

The listing of the text within “sxo-apecs-journal.jou” follows:

```

;
;   journal file to be called AFTER APECS solves the flow equations
;
;
;   start with original case set-up
;
/define/models/species/volumetric-reactions? yes
/define/models/species/set-turb-chem-interaction n y
    
```

```
/define/materials/change-create chemkin-import chemkin-import n n yes
"mechanism-1" all 7 "reaction-20" "reaction-21" "reaction-22" "reaction-23"
"reaction-24" "reaction-25" "reaction-26" no no no no no no no no no
/solve/set/equations/disco yes
/solve/set/equations/flow yes
/solve/set/equations/ke yes
/solve/set/equations/temperature yes
/solve/set/equations/species-0 y
/solve/set/equations/species-1 y
/solve/set/equations/species-2 y
/solve/set/equations/species-3 y
/solve/set/equations/species-4 yes
/solve/set/equations/species-5 yes
/solve/set/equations/species-6 yes
/solve/set/equations/species-7 yes
/solve/set/equations/species-8 n
/solve/set/equations/species-9 n
/solve/set/equations/species-10 no
/solve/set/equations/species-11 n
/solve/set/equations/species-12 n
/solve/set/equations/species-13 no
/solve/set/equations/species-14 n
/solve/set/equations/species-15 n
/solve/set/equations/species-16 n
/define/models/dpm/interaction/no.-of-cont-phase-iters-per-dpm-iter 40
;
;
;   set up and solve for NOx
;
/define/user-defined/function-hooks/adjust ""
/solve/set/equations/disco n
/solve/set/equations/flow n
/solve/set/equations/ke n
/solve/set/equations/pollutant-0 yes
/solve/set/equations/pollutant-1 yes
/solve/set/equations/species-0 n
/solve/set/equations/species-1 n
/solve/set/equations/species-2 n
/solve/set/equations/species-3 n
/solve/set/equations/species-4 n
/solve/set/equations/species-5 n
/solve/set/equations/species-6 n
/solve/set/equations/species-7 n
/solve/set/equations/species-8 n
/solve/set/equations/species-9 n
/solve/set/equations/species-10 n
/solve/set/equations/species-11 n
/solve/set/equations/species-12 n
/solve/set/equations/species-13 n
/solve/set/equations/species-14 n
/solve/set/equations/species-15 n
/solve/set/equations/species-16 n
/solve/set/equations/temperature no
/define/models/dpm/interaction/no.-of-cont-phase-iters-per-dpm-iter 0
/solve/iterate 30
;
;
```

```
;      set up and solve for SOx
;
/define/models/species/set-turb-chem-interaction y
/solve/set/stiff-chemistry yes 1 0.1 0.1
/define/materials/change-create chemkin-import chemkin-import no no yes
"mechanism-1" all 20 "reaction-1" "reaction-2" "reaction-3" "reaction-4"
"reaction-5" "reaction-6" "reaction-7" "reaction-8" "reaction-9" "reaction-10"
"reaction-11" "reaction-12" "reaction-13" "reaction-14" "reaction-15" "reaction-
16" "reaction-17" "reaction-18" "reaction-19" "reaction-20" no no no no no no no
no no no
/define/user-defined/execute-on-demand "calc_radicals_cor::libudf"
/solve/set/equations/disco n
/solve/set/equations/flow n
/solve/set/equations/ke n
/solve/set/equations/species-0 n
/solve/set/equations/species-1 n
/solve/set/equations/species-2 n
/solve/set/equations/species-3 n
/solve/set/equations/species-4 n
/solve/set/equations/species-5 yes
/solve/set/equations/species-6 n
/solve/set/equations/species-7 yes
/solve/set/equations/species-8 n
/solve/set/equations/species-9 n
/solve/set/equations/species-10 n
/solve/set/equations/species-11 yes
/solve/set/equations/species-12 yes
/solve/set/equations/species-13 n
/solve/set/equations/species-14 n
/solve/set/equations/species-15 n
/solve/set/equations/species-16 yes
/solve/set/equations/pollutant-0 n
/solve/set/equations/pollutant-1 n
/solve/set/equations/temperature n
/define/models/dpm/interaction/no.-of-cont-phase-iters-per-dpm-iter 0
;/define/models/species/integration-parameters yes 1e-08 1e-09 0.001 500 0.001 1
0
/solve/iterate 40
;
;
;      return case to original state
;
/define/models/species/volumetric-reactions? yes
/define/models/species/set-turb-chem-interaction n y
/define/materials/change-create chemkin-import chemkin-import n n yes
"mechanism-1" all 7 "reaction-20" "reaction-21" "reaction-22" "reaction-23"
"reaction-24" "reaction-25" "reaction-26" no no no no no no no no no no
/solve/set/equations/disco yes
/solve/set/equations/flow yes
/solve/set/equations/ke yes
/solve/set/equations/temperature yes
/solve/set/equations/species-0 y
/solve/set/equations/species-1 y
/solve/set/equations/species-2 y
/solve/set/equations/species-3 y
/solve/set/equations/species-4 yes
/solve/set/equations/species-5 yes
```



```
/solve/set/equations/species-6 yes
/solve/set/equations/species-7 yes
/solve/set/equations/species-8 n
/solve/set/equations/species-9 n
/solve/set/equations/species-10 no
/solve/set/equations/species-11 n
/solve/set/equations/species-12 n
/solve/set/equations/species-13 no
/solve/set/equations/species-14 n
/solve/set/equations/species-15 n
/solve/set/equations/species-16 n
/define/models/dpm/interaction/no.-of-cont-phase-iters-per-dpm-iter 40
/define/user-defined/function-hooks/adjust "wall_abs::libudf" ""
;
;
; return control to FLUENT Wrapper
;
```

The first section concerns the main combustion reactions and the activation of the Magnussen Eddy Dissipation model. Reactions (20) through (26) are activated in “Mechanism-1.” Following that step, various species transport equations are turned off or on, where the species numbers correspond to the species in the left-hand column of Table 4.

The second section concerns convergence of the NO_x (post-processing) submodel. The user-defined subroutine “DEFINE_ADJUST(wall_abs, d)” in file “wallabs.c” is deactivated so that the wall absorption calculations are not performed during the NO_x and SO_x model computations. The NO_x model, already set up for running, requires only that the pollutant transport equations be activated before it can be executed. Once that step is accomplished, the NO_x model is run for 30 iterations, which is usually sufficient to converge the model when the under-relaxation factors are unity.

The third section concerns convergence of the SO_x model. The laminar chemistry model is activated and the appropriate SO_x reactions are turned on (i.e., Reactions (1) through (20)). It should be noted that Reaction (20) is the “vol + (0.974) o₂ → (0.253) co + (0.886) h₂o + (0.024) h₂s + (0.023) n₂” step. It was found that this reaction step must be activated in “Mechanism-1” during the SO_x model computations in order to preserve the pyrolysis source terms feeding the production of “h₂s;” otherwise the pyrolysis source terms are zeroed out. (The preservation of the DPM source terms related to the production of SO₂ from the char is already automatically taken care of in FLUENT® and no additional user intervention is required.) The execute-on-demand UDF “calc_radicals_cor” is executed next in order to establish a flow field for the radical species O, OH, and H concentrations. The transport equations for the 5 species associated with the SO_x model are activated and the model is run approximately 40 iterations.

Following the running of the SO_x model, the Eddy Dissipation model is reactivated and control is returned to the FLUENT® wrapper.

Note on SO_x model convergence issues. A comment should be made as to why the SO_x model is only run 40 iterations, since the laminar chemistry model would typically be viewed as potentially requiring many more iterations. It was found that the SO_x model is not conserving elemental sulfur at the outlet of the computational domain. If the SO₂ and H₂S are monitored at the outlet plane, it is

found that, shortly after the SO_x model is initiated, the outlet SO₂ values transition fairly quickly to appropriate levels consistent with elemental sulfur conservation (see Figure 5). After e.g., 150 iterations, the outlet SO₂ values begin to rise slightly to a new plateau that is in error by perhaps 1 to 4%. After perhaps another 400 iterations, the outlet SO₂ almost linearly increases, so that after another 2,000 iterations, the sulfur at the outlet plane has now increased by e.g., a factor of 8, with no sign of levelling off. The SO_x species residuals are staying relatively constant or decreasing; it is not apparent from the residuals that the SO_x transport equations are becoming unstable or “blowing up.” This issue has been elevated to the FLUENT® developers. However, for purposes of this project, since the convergence of the SO_x model cannot be trusted, it was deemed necessary to limit the number of SO_x model iterations to e.g., 40 iterations, and hope that the outlet values would not go too far astray during that period of convergence. Any lack of conservation in the elemental sulfur species during the execution of the FLUENT® block will surely impact the convergence of the flow sheet (particularly the tear stream), but the extent of the error introduced by this conservation issue has not been fully explored nor characterized.

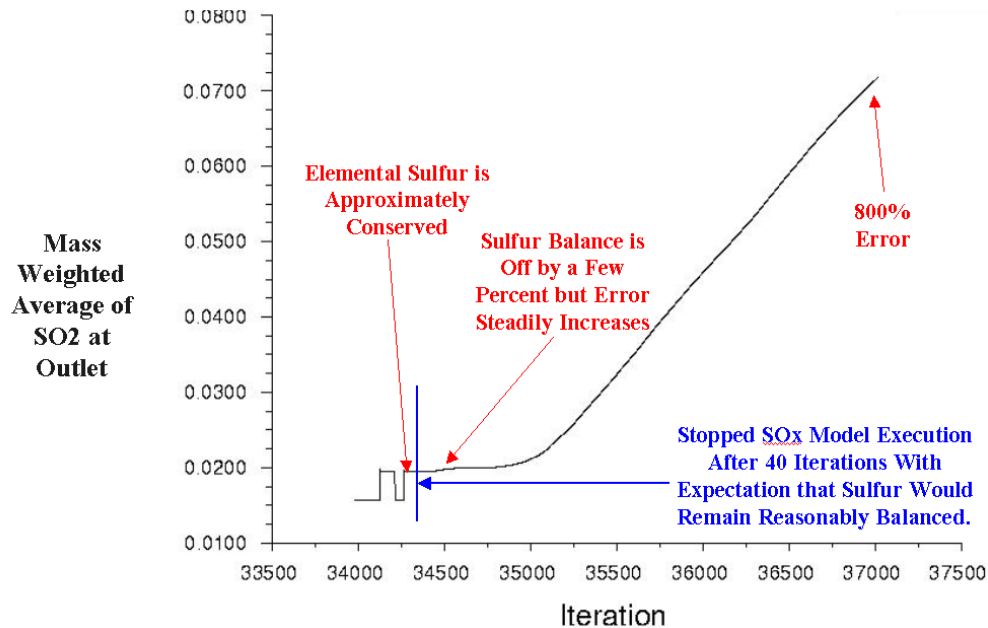


Figure 5: SO_x Model Convergence Behavior.

Another journal file provided by ANSYS is called “compile_udf_unix.jou.” Since the case and data files and associated files (including libudf.dll) were prepared and uploaded to EKM® from the PC environment, this journal file serves to recompile the UDFs at run time on the Linux machine and then load the “libudf” into the case file. The DEFINE_ADJUST UDF is also activated.

```
;  
; Journal file to compile and load UDF on LINUX  
;  
define user-defined compiled  
compile  
libudf  
yes
```

```
abs.c
calc-radicals.c
dpm_Qprop_bitum.c
wallabs.c
" "
" "
;
define user-defined compiled
load
libudf
;
;
/define/user-defined/function-hooks/adjust "wall_abs::libudf" " "
;
```

ANSYS has prepared another user journal file activated via the Cape-Open parameters:

- calculate-user-journal-name
- calculate-user-journal-activate

For the present task, this user journal file, called “prep_resid.jou,” was used to help run a minimum number of iterations by:

- (a) lowering one or more of the residual tolerances
- (b) running a specified number of iterations
- (c) and then putting the residual tolerances back where they were originally

The APECS calculate command is executed after the journal file commands are completed. If the residuals are less than the stipulated thresholds, then the case is deemed converged; otherwise, it begins to calculate the number of iterations dictated by APECS. While any need for this journal file may be obviated by judicious selection of the residual thresholds, it is sometimes difficult to tell when a case is converged merely by looking at the residuals. (Certain monitors, coupled with residuals are usually much more helpful.) Furthermore, when performing a co-simulation, it is difficult for the user to know what the ideal convergence residual levels should be, unless the user has monitored the residuals for a large number of cases. Since such monitors do not seem to be presently available when the CFD block is executed remotely over a LAN, and since it is difficult to determine the proper convergence levels for the residuals, a good compromise is to utilize a user-journal file which forces a user-specified minimum number of iterations.

```
; prep_resid.jou
;
;
k   epsilon   do      vol   o2      co2      co      cont   x      y      z      energy
      h2o      so2      h2      h2s
;
/solve/monitors/residual/convergence-criteria 1.e-09 1.e-06 1.e-06 1.e-06 1.e-06
1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06
/solve/iterate 1000
/solve/monitors/residual/convergence-criteria 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06
1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06 1.e-06
```

Since the Cape-Open parameter “Maximum Iterations” is also set to 1000, each execution of the FLUENT® block could conceivably run 2000 iterations.

The instantiation procedure, in general terms, is given below:

- (1) Instantiate the APECS block and rename it to “BSF”
- (2) Assign the block section = DEGASIF (default)
- (3) Make all of the stream connections in accordance with Table 2
- (4) Update the design specs and calculator blocks to accommodate the BSF block:

```
DESIGN-SPEC DS-RATIO
      DEFINE QQBSFO BLOCK-VAR BLOCK=BSF VARIABLE=WALL-ABSORPTION
```

```
CALCULATOR C-BSFQ
      EXECUTE BEFORE BLOCK BSF
```

```
CALCULATOR C-PSTUSR
      DEFINE QBSFWA BLOCK-VAR BLOCK=BSF VARIABLE=WALL-ABSORPTION
      DEFINE PMECON BLOCK-VAR BLOCK=BSF VARIABLE=PM-ENERGY-CONST
      DEFINE ENSKPM BLOCK-VAR BLOCK=BSF VARIABLE=ENERGY-SINK-PM
      DEFINE HTFLUX BLOCK-VAR BLOCK=BSF VARIABLE=HEAT-FLUX-WALL
      EXECUTE AFTER BLOCK BSF
```

- (5) Complete the species mapping, as indicated by the last two columns in Table 4.

5.2 Computational Results

5.2.1 Preliminary FLUENT® Runs

A number of stand-alone FLUENT® runs were made in order to understand and gauge the heat transfer characteristics of the CFD cases as a function of recycle ratios. The first set of runs consisted of an air-fired case compared with 5 oxy-fired, flue-gas recycle cases at different recycle ratios. Both the fine-grid (3,319,222 cells) cases and the coarse-grid (247,742 cells) cases were run with the same inlet and wall boundary conditions. The compositions of the inlet boundary conditions were taken from a previous study (Ref. 12), in which the overall secondary mixture consisted of leakage air, preheated oxygen, and heated secondary flue gas in specified proportions. The primary stream composition was identical to that of the secondary flue gas, without one of the streams being preferentially enriched by oxygen. It should be emphasized that in these FLUENT® runs, no effort was made to manually (and iteratively) recycle the boiler exit gases to mix with the other inlet feed streams. The results are shown in Figure 6.

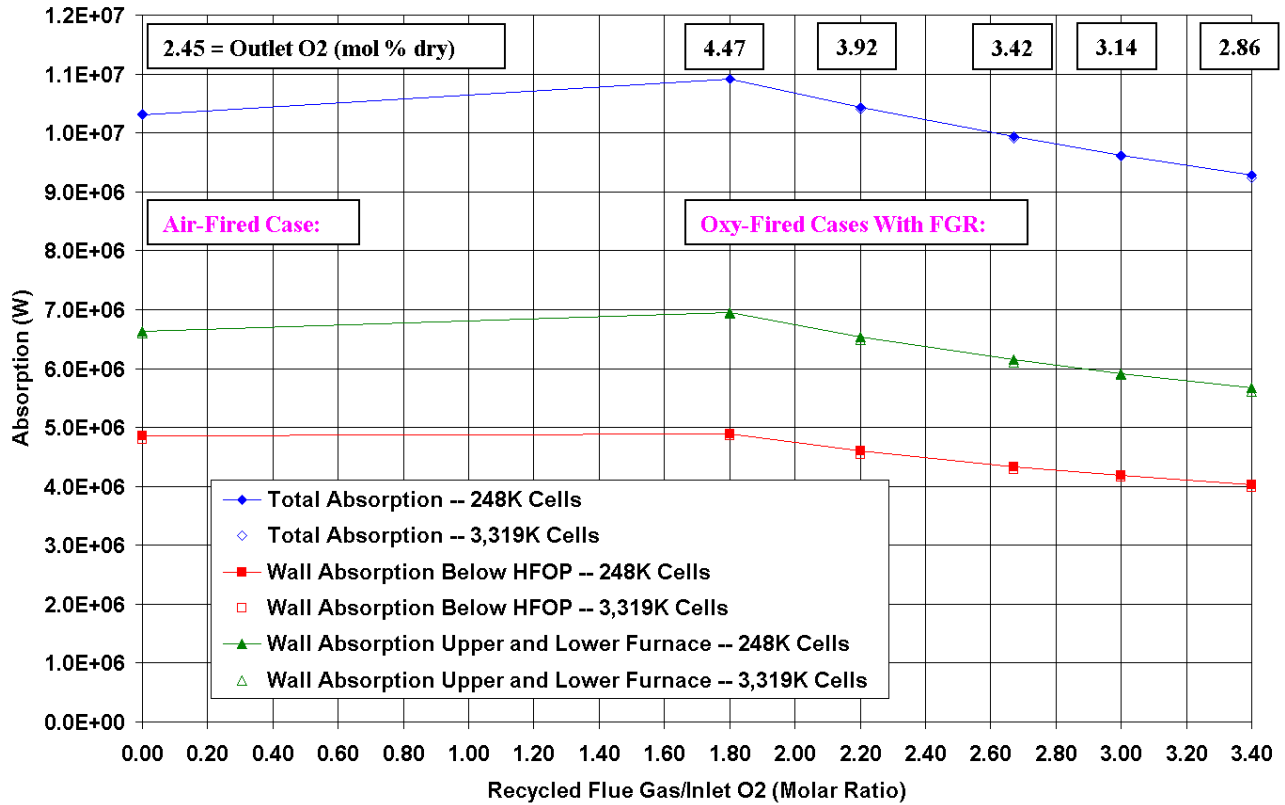


Figure 6: Absorption as a Function of the Recycle Ratio for Two Grid Sizes.

The term “total absorption” encompasses the entire computational domain -- furnace, convective pass and backpass. The term “wall absorption below the horizontal furnace outlet plane (HFOP)” is defined as meaning the total wall absorption below the nose/arch. The term “wall absorption for the upper and lower furnace” is defined herein as excluding the convective pass and backpass (but including the platens over the arch).

The intent of the above plot is to show that the absorption results for the coarse grid practically overlay the absorption results for the fine grid. A comparison of the NOx emissions predictions between the coarse- and fine-grid cases was not performed.

In the second set of runs, only the coarse-grid (247,742 cells) cases were exercised. An air-fired case was again compared with 5 oxy-fired, flue-gas recycle cases at different recycle ratios, but this time the outlet FGR stream was manually recycled to mix with the oxygen and leakage air feeds to produce a new inlet composition. The manual recycle was performed without condensing out any of the water, etc. It was found that only a few iterations were sufficient to converge the process to the point where the inlet composition stopped changing. The results are shown in Figure 7.

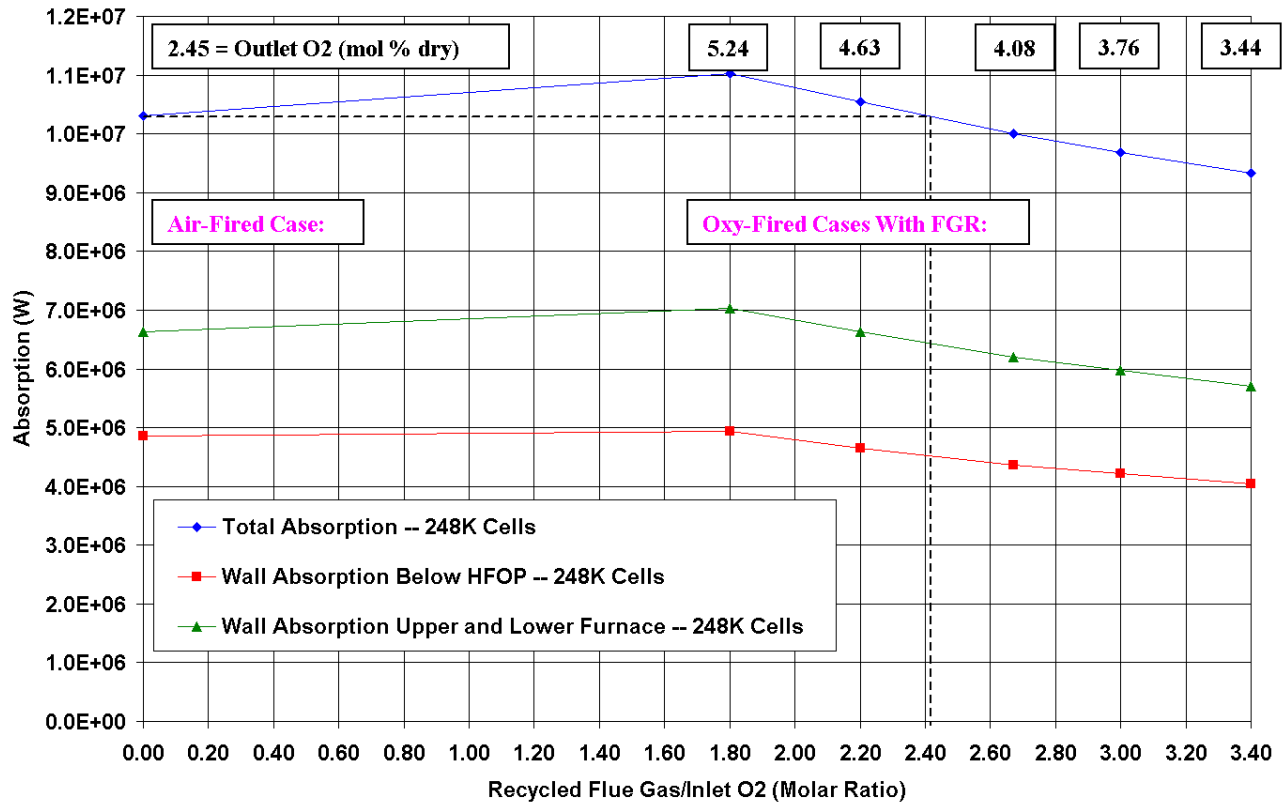


Figure 7: Absorption as a Function of the RR With Manual Recycle of the Outlet Stream.

The lines on the graph appear to be only slightly different from the lines of the previous graph (Figure 6), but the outlet oxygen concentrations have changed significantly. The recycle ratio, under which the absorption of the oxy-fired operating cases matches the absorption of the air-fired case, is approximately 2.42. It should also be noted that, if the design intent is to maintain the same overall absorption in the oxy-fired case as in the air-fired case, then it will probably not be possible to maintain desired values for other design variables, such as the exit oxygen concentration or boiler efficiency. The above cases implicitly assumed negligible leakage. If significant air leakage occurs downstream of the boiler (e.g., at the FDA/fabric filter), and if that air leakage is proportional to the flue gas flow rate, then the additional O₂ introduced into the boiler through the recirculated flue gas could be substantial and could, in fact, completely change how the boiler outlet O₂ varies with the volumetric recycle ratio.

The above graph serves as the basis for the co-simulation. In the design spec DS-RATIO, the recycle ratio is varied until the total absorption is equal to a target absorption (e.g., the absorption of the air-fired case).

The convergence behaviour of the stand-alone FLUENT® cases is relatively well behaved, but the cases can also exhibit occasional spikes or oscillatory behavior (see Figure 8) that can cause problems in the calculation of design spec or optimization search directions, and perhaps ultimately cause the flowsheet convergence to fail. The spikes appear to be less frequent or severe at the

higher recycle ratios, but must nevertheless be considered when trying to decide on convergence criteria. The tear stream convergence criterion was coarsened with the expectation that this would help the flow sheet to converge more quickly and still maintain a reasonable accuracy.

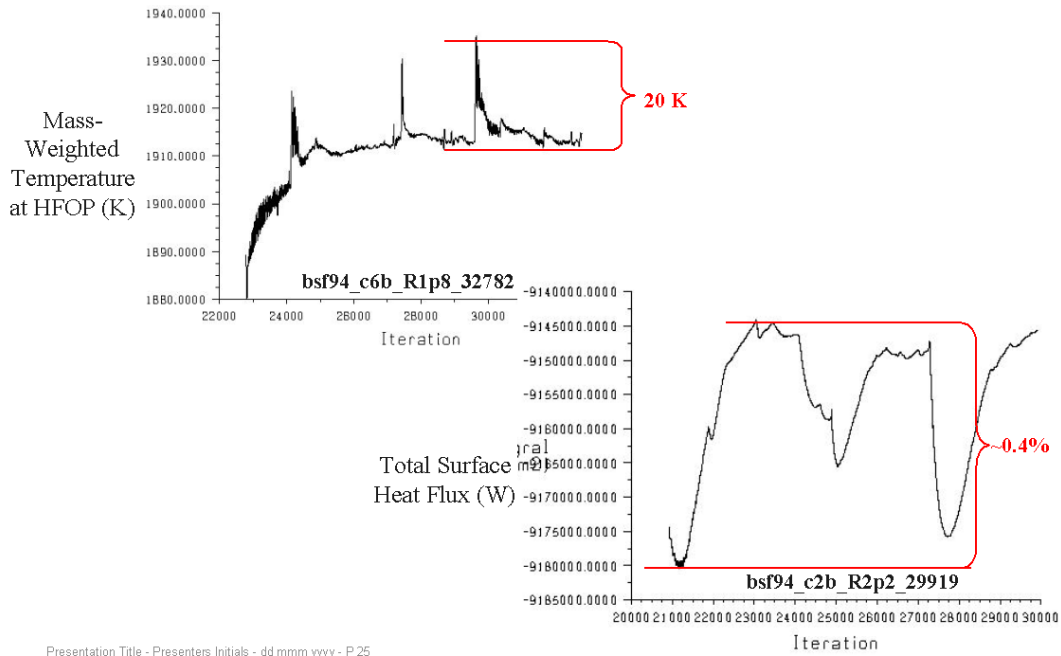


Figure 8: CFD Convergence Behavior as Indicated by Monitors.

5.2.2 Co-Simulation Runs

The results for the co-simulation will not be identical to those of the preliminary FLUENT® runs, because of some subtle differences, but the results should be similar. For example, in the co-simulation, the moisture will be removed from the primary stream and the primary stream will receive a significant infusion of oxygen, both of which did not occur in the preliminary stand-alone FLUENT® trials.

Efforts were hampered by bugs in both Aspen Plus® and FLUENT®. The apparent bugs in Aspen Plus® V7.1 include:

- The streams entering the Cape-Open block are flashed with a KODE=2 (P, T) flash. This flashing was only recently added and was not done in V6.5. This error may not impact the inlet (gas) boiler streams, but it could seriously impact e.g., a saturated water/vapour stream, producing the wrong inlet enthalpy.
- At the outlets from a Cape-Open block, the total CISOLID flow rate is given a value equal to 0.1E+36. This of course causes a huge mass imbalance across the FLUENT® block. A workaround, found quite by accident, was to print out the total CISOLID and component streams in a subsequent calculator block, which caused Aspen Plus® to recognize the discrepancy and reset the CISOLID stream flow rate.
- After the Cape-Open block has been instantiated and the streams have been connected, it is usually efficient to save out the flow sheet. However, when this is done and the flow sheet

(.apw file) has been read back in, one finds that the stream connections and the viability of the block are not persisted. AspenTech's recommendation for a temporary workaround is that one must run first before saving out the flow sheet, but this is not a workable solution for a large CFD block that may run for days. Furthermore, saving out the flowsheet more than once also leads to the inviability of the block.

Apparent FLUENT® V12.0.16 bugs include:

- Lack of elemental sulfur conservation with the SO_x mechanism and the laminar chemistry option.
- Message passing does not work with more than one stochastic try.
- The shared memory plus workpile algorithm option produces a "Divergence in the AMG solver".
- In the materials panel for the multiple surface reactions, carbon must be entered before sulfur.

With respect to the UDFs, it should be noted that the "DEFINE_DPM_BC(bc_outlet, p, t, f, f_normal, dim)" UDF requires a different approach before it can be considered properly parallelized for the "message passing" option. The UDF can only be used with the "shared memory" option.

Since the SO_x model did not appear to conserve elemental sulfur, and since its convergence was not considered reliable, it was deemed judicious to limit the number of SO_x model iterations to a value of 40 (as stated previously). The inlet and outlet values to the BSF block were monitored, and it was decided that rather than wait (perhaps several days) until the flow sheet converged, that it would be more prudent to simply interrupt or terminate the flow sheet run when the inlet and outlet values pertaining to the BSF block "settled down." Sometimes, tuning a convergence tolerance within Aspen Plus® requires experience and judgment, and to get it right may ultimately require multiple attempts to solve the flow sheet (particularly for the non-expert users). However, in the present work, inputs and outputs for the equipment items of interest are written to a history file (via calculator blocks), and these can be monitored by the user while the Aspen Plus® flow sheet is still running. When the user determines that the inputs and outputs have settled down sufficiently, the user can interrupt (or if necessary, terminate the run). This was the methodology used in the present task.

In the present case, the design spec varied the volumetric recycle ratio until a particular absorption was achieved. In Run 1, the target absorption was the same as in the air-fired case; in Run 2, the target absorption was somewhat less (9.7E+06 W). In Run 1, the flow sheet was already close to convergence, so 10 FLUENT® block executions were considered adequate to pronounce the solution effectively converged. An average of about 8 hours (elapsed time) was required per FLUENT® block execution, so that a total of 3.3 days were expended in the computation. In Run 2, over 51 iterations were run, so that almost 16.7 days were expended in the computation. The results are shown in Table 7.

Table 7: Co-Simulation Results

Parameter	Results from Oxy-Fired Flow Sheet	
	Run 1	Run 2
Target Absorption	10.314 MW (35.216 MBtu/hr)	9.7 MW (33.12 MBtu/hr)
% Recycle	69.6	74.1
Volumetric Recycle Ratio	2.48	3.13
Outlet O2 (% , Dry, by Volume)	4.65	3.99
NOx (oxy-fired) / NOx (air-fired) [Note 1]	0.18	0.25
SOx (oxy-fired) / SOx (air-fired) [Note 2]	1.13	1.1
Exit SO3 Mole Fraction	negligible	negligible
Number of FLUENT® Executions	10	51
CPU Days	3.3	16.7
Note 1:	At location downstream of recycle take-off	
Note 2:	At exit of boiler. FDA removes 90% of sulfur species from boiler exhaust before the recycle is split off.	

It is difficult to comment on the NOx without some experimental data for verification. In one literature source (Ref. 10) NOx values typically declined with an increase in recycle ratio and were on the order of 30% of the air value. The NOx values are of the correct order, but the trend is in a different direction. Since almost all of the sulfur goes to SO₂, the overall SOx concentrations at the exit of the boiler are simply those that correspond the overall mass balance, with the concentration in the recycle dictated by the percent recycle and the amount of sulfur removed in the FDA. The real species of interest was the SO₃, but the mechanism predicted negligible values for that species. The high boiler exit gas temperatures (approximately 1,000 K) contribute to that low number, but the accuracy of the mechanism needs to be validated before additional commentary can be offered. Since the NOx and SOx models are known to be lacking in quantitative accuracy, no additional commentary is offered relative to the “goodness” of the emissions values at this present time. However, for purposes of the present task, it may be stated that a workable methodology for the inclusion of pollutant models within the APECS environment has been developed and demonstrated, and that this approach, described in detail within this report, can serve as a basis for future efforts in emissions co-simulations.

The results of the co-simulation runs in the PME environment are shown on the same plot with the stand-alone FLUENT® runs. It is apparent that the trend line for the co-simulations may have shifted relative to that of the stand-alone cases. This should not be surprising since the inlet conditions to the BSF are somewhat different from those of the stand-alone cases (e.g., water removed from the primary stream, inlet temperature differences, etc.) It may be postulated that the trend line for the co-simulations may shift and change slope as leakage, recycled flue gas cleanup strategies, and design spec criteria are modified.

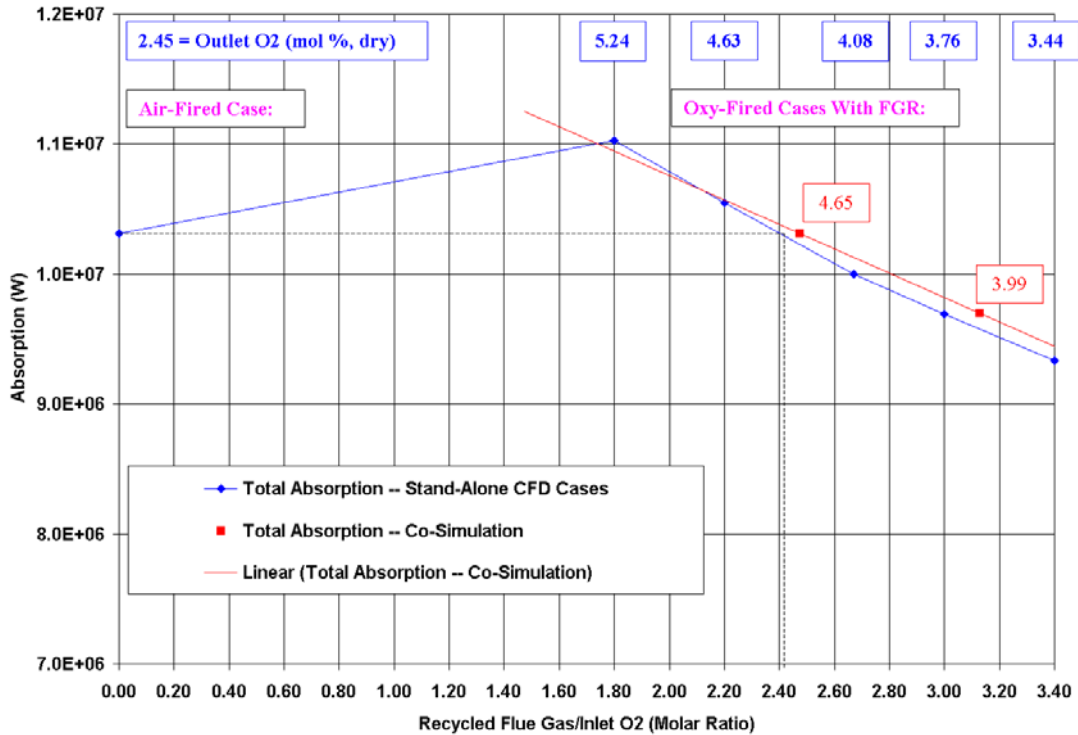


Figure 9: Comparison of the Stand-Alone FLUENT® Runs with the Co-Simulation Results.

6.0 CONCLUSIONS AND RECOMMENDATIONS

This present task helps to showcase an important advancement in the APECS tool kit – the addition of SO_x and NO_x emissions calculations to an integrated FLUENT® CFD block execution. An oxy-fired BSF boiler was successfully run with sequential emissions calculations in an Aspen Plus® flow sheet. ANSYS modified the APECS software to expose the NO_x pollutant species to Aspen Plus®. Alstom Power was tasked with the responsibility to (a) conduct an APECS study of the Alstom Boiler Simulation Facility (BSF) island (gas side only) with oxygen firing and flue gas recycling, (b) invoke the special NO_x and SO_x prediction capabilities in FLUENT®, and (c) characterize the impact of flue-gas recycle and cleanup on pollutant emissions for a candidate boiler configuration. This was accomplished.

A demonstration case was completed in which a CFD model of the BSF furnace was coupled with a flow sheet of the BSF facility and run in an integrated simulation. In this simulation, the flue gas recycle ratio was varied within a design spec in order to match a target furnace absorption. SO_x and NO_x emissions calculations were invoked with each execution of the BSF CFD block. The NO_x calculations were generated by the NO_x submodel in FLUENT®, which is typically run in a post-processor mode. The corresponding SO_x submodel in FLUENT® was found to not converge and significant effort was expended by the APECS team to find a work-around strategy that involved a stiff integrator. The final solution involved the use of a journal file that was designed to sequentially (a) run the Eddy Dissipation model for the main combustion species, (b) freeze the flow field and run the NO_x model for a prescribed number of iterations, (c) activate the laminar

chemistry model and the SO_x species transport equations and then run the SO_x model for a prescribed number of iterations, and then (d) activate the appropriate equations and options to be able to run the Eddy Dissipation model again. Although a Cape-Open Boolean variable does exist for the pollutant models, the Cape-Open variable approach is still rather limited, and ANSYS determined that the journal file would offer the desired versatility and control.

The journal file approach seemed to work reasonably well and functioned properly, even in an environment where the FLUENT® case was run remotely on a Linux workstation on multiple processors. Both NO_x and SO_x values were calculated at the targeted wall absorption value for oxy-firing and compared with the corresponding (stand-alone) air-fired case results. It should be noted that the SO_x values require further scrutiny since the current SO_x model and mechanism does not conserve elemental sulfur.

Significant progress has been made by ANSYS in facilitating the direct instantiation of FLUENT® on an Aspen Plus® flow sheet, in conjunction with the sequencing and running of the pollutant models. It is hoped that the present documented experience, as well as the associated journal files and explanation of the workflow, will help other others in their efforts to engage the pollutant models in subsequent co-simulations.

Some additional conclusions and recommendations are as follows:

(1) Conclusion: ANSYS determined that it would be more efficient and versatile to turn the different combustion and mechanism models off and on via a journal file, rather than through Cape-Open variables. A Cape-Open Boolean variable does exist, i.e., “post-process-pollutant-models”. However, this variable is somewhat limited in that: (a) it toggles between species that have the word “pollutant” in their name and those that do not, and (b) it uses the same number of iterations to solve the pollutants transport equations as were used to solve the main combustion reactions. Since in the present case (a) three different combustion models were being sequentially activated (main combustion, NO_x, and SO_x), (b) greater control was desired in being able to specify the number of iterations for each model, and (c) additional UDFs were executed or toggled off and on at different times, it was apparent that a journal file was required and that the Cape-Open variable for pollutants would not be used.

[1] Recommendation: Develop the requisite suite of Cape-Open variables and Booleans that would permit a user to toggle the pollutant models off and on, specify the number of iterations for each model, etc.

(2) Conclusion: The FLUENT® SO_x model does not presently conserve elemental sulfur. It is not known if the solution algorithm for the “laminar chemistry” model is at fault or whether the mechanism is problematic and in a state of incipient divergence.

[2] Recommendation: ANSYS should provide a robust and reliable SO_x mechanism and submodel. A validated and CPU-efficient mechanism should be sought by ANSYS, industry, and interested universities.

(3) Conclusion: When the FLUENT® block execution is spawned to the Linux platform, the user can see the information exchange in the same window or screen where the environment variables were set and where the requisite processes for APECS functionality were initiated. However, the user cannot interact with this screen and FLUENT® GUIs and panels are not “live” or available. One can apparently interact with FLUENT® in a Windows environment, but not in a Linux environment.

[3] Recommendation: Provide the proper threads and software enhancements to allow the user to interface with or interact with FLUENT® when the FLUENT® job is spawned to a remote Linux platform.

(4) Conclusion: Currently, the FLUENT® case is not automatically saved after every FLUENT® execution in the “temp” directory on the Linux platform. The FLUENT® case is not saved until the flow sheet is saved, at which time, the FLUENT® case is newly uploaded to the EKM® database. However, in cases where FLUENT® requires several hours to perform a single execution, and in cases where the Aspen Plus® run may be prematurely terminated by the user, it is oftentimes important to have a saved case and data file available, representing the last successful FLUENT® execution.

[4] Recommendation: Permit the user to activate the option to save FLUENT® in the “temp” directory every time it successfully executes, perhaps overwriting the previous case and data file.

(5) Conclusion: Flow-splitters are currently available for gas inlet ports, but they are not available for DPM injections.

[5] Recommendation: Upgrade the flow-splitter capability to include DPM inlet ports.

(6) Conclusion: The current methodology used in the UDFs to monitor the particle trajectory information at the outlet plane cannot be properly parallelized, and is therefore not amenable with the CPU-saving “message passing” option.

[6] Since particle injections are frequently found in APECS simulations, ANSYS would provide a beneficial service to users by providing a template with the proper parallelization directives.

7.0 REFERENCES

1. Zitney, S. E., M. Osawe, L. Collins, E. Ferguson, D. Sloan, W. Fiveland, J. Madsen, “Advanced Process Co-Simulation of the FutureGen Power Plant,” In Proc. of the 31st International Technical Conference on Coal Utilization & Fuel Systems, May 21-25, Clearwater, FL (2006).
2. Sloan, D. G., W. A. Fiveland, S. E. Zitney, and M. Syamlal, "Software Integration for Power Plant Simulations," In Proc. of the 27th International Technical Conference on Coal Utilization & Fuel Systems, March 4-7, Clearwater, FL (2002).

3. Sloan, D. G., W. A. Fiveland, S. E. Zitney, and M. Syamlal, "Power Plant Simulations Using Process Analysis Software Linked to Advanced Modules," In Proc. of the 29th International Technical Conference on Coal Utilization & Fuel Systems, April 18-22, Clearwater, FL (2004).
4. Sloan, D. G., W. A. Fiveland, M. O. Osawe, S. E. Zitney, and M. Syamlal, " Demonstrations of Coupled Cycle Analyses and CFD Simulations over a LAN," In Proc. of the 30th International Technical Conference on Coal Utilization & Fuel Systems, April 17-21, Clearwater, FL (2005).
5. "Project Management Plan, Software Framework for Advanced Power Plant Simulations, Cooperative Agreement No. DE-FC26-05NT42443", for Ronald W. Breault USDOE-NETL, compiled by John F. Widmann, representing team entities: ANSYS Inc., ALSTOM Power, Aspen Technology Inc., and Carnegie Mellon University (Department of Chemical Engineering), May 6, 2008.
6. Chow, O. K., Griffith, B. F., Levasseur, A. A., "TFS-2000 Slagging and Corrosion," ABB Combustion Engineering Internal Report PPL-94-CRD-4 (Oct. 1994).
7. Schneider, D. R., and Z. Bogdan, "Modeling of SO₃ Formation in the Flame of a Heavy-Oil Fired Furnace," Chem. Biochem. Eng. Q., 17(3), 175-181 (2003).
8. Perera, S. A. L., and Schowalter, David G., "A SO_x Formation Model for Industrial CFD Applications," Clearwater Coal Technology Conference (May 2006).
9. Fluent 6.3 User's Guide, "20.2.3 Reaction Mechanisms for Sulfur Oxidation," Table 20.2.1 (<http://www.fluentusers.com>).
10. Sarofim, Adel, "Oxy-Fuel Combustion: Progress and Remaining Issues," International Oxy-Combustion Research Network, 2nd Workshop, Windsor, CT (Jan 25-26, 2007).
11. APECS: Advanced Process Engineering Co-Simulation User's Manual, Version 1.6.1, Prepared by ANSYS, Inc., Lebanon, NH (June 2009).
12. Engineering Feasibility and Economics of CO₂ Capture on an Existing Coal-Fired Power Plant, Final Report, Volume II – Bench-Scale Testing and Computational Fluid Dynamics (CFD) Evaluation, Submitted by ALSTOM Power Inc., Prepared for the Ohio Department of Development (Grant No. CDO/D-98-8) and the U.S. DOE (Grant No. DE-CF26-99FT40576), (Report Submittal Date: June 29, 2001).

8.0 DELIVERABLES

The following files are delivered with this report:

No.	Files	Description
(1)	bsf_oxy_cfd_v7p1_Rev-5_sequence.bkp bsf_oxy_cfd_v7p1_Rev-5_sequence_R2p7_saved.bkp	Aspen Plus® flowsheet prior to instantiation, followed by the flowsheet saved after Run 2 was completed
(2)	bsf_oxy_cfd_v7p1_Rev-5_sequence_Run1.his bsf_oxy_cfd_v7p1_Rev-5_sequence_R2p7_saved.his	Aspen Plus® history files for Run 1, followed by the history file for Run 2
(3)	bsf_33980.cas.gz bsf_33980.dat.gz	FLUENT® 3D case and data files
(4)	abs.c dpm_Qprop_bitum.c calc-radicals.c wallabs.c gastable.txt	UDFs
(5)	bsf_33980.CORBASolver COFluentWrapper.txt	Files created by APECS
(6)	prep_resid.jou	Used to help the user to run a minimum number of iterations.
(7)	compile_udf_unix.jou	Used for automatic compilation of UDFs on Linux systems
(8)	sxo-apecs-journal.jou	Journal file that sequences the main combustion, NO _x , and SO _x reactions

Software Framework for
Advanced Power Plant Simulations

TASK 5.3-A
DEMONSTRATION OF A REGRESSION ROM
FOR A HEAT RECOVERY STEAM GENERATOR
IN AN INTEGRATED CO-SIMULATION

Topical Report

Reporting Period Start Date: November 1, 2006
Reporting Period End Date: September 11, 2009

David Sloan, Ph.D.
Woodrow Fiveland, Ph.D.

November 30, 2009

DOE Cooperative Agreement No: DE-FC26-05NT42443

Alstom Power Inc.
US Power Plant Laboratories
P.O. Box 500, CEP 9005-1204
2000 Day Hill Road
Windsor, CT 06095

UNDER SUBCONTRACT TO
ANSYS Inc.
10 Cavendish Court, Lebanon, NH 03766

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ACKNOWLEDGEMENT STATEMENT

This document was prepared with the support of the U.S. Department of Energy, under Award No. DE-FC26-05NT42443. However, any opinions, findings, conclusions, or recommendations expressed herein are those of the author(s) and do not necessarily reflect the views of the DOE.

ACKNOWLEDGEMENTS

Financial support for this project was provided by DOE National Energy Technology Laboratory (Cooperative Agreement No. DE-FC26-05NT42443) and Alstom Power Inc. Appreciation is expressed to Ronald Breault as the DOE Project Manager and to Stephen E. Zitney (NETL) for his technical input. The contributions of the many APECS project team members from ANSYS Inc. and Aspen Technology Inc. are also greatly appreciated. ANSYS Inc. is particularly acknowledged for its leadership role as the prime contractor under John Widmann. Gratitude is expressed to Sorin Munteanu, John Widmann, and others on the APECS team at ANSYS Inc. for their diligence in striving to deliver quality software.

Within the Alstom Power divisions, appreciation is expressed to a number of individuals. John L. Marion, the Alstom Power Program Manager, and Shin Kang and Woodrow A. Fiveland, the Alstom Power Project Managers, contributed to the success of this project task.

ABSTRACT

The primary objective of this DOE project is to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of Advanced Process Engineering Co-Simulator (APECS), the simulator will link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. Generally, the simulator couples the cycle analysis software Aspen Plus® (marketed by Aspen Technology, Inc.), specialized component modeling packages, as exemplified by industrial proprietary codes (utilized by Alstom Power Inc.), and the FLUENT® computational fluid dynamics (CFD) code (provided by ANSYS Inc). An important enhancement to the APECS tool will be the creation of computationally efficient reduced-order models (ROM) based on information from high-fidelity computational fluid dynamics (CFD) models. Integration of the steady-state APECS tool with a virtual reality tool kit will allow process designers to navigate through a 3D virtual representation of a power plant with visualization of simulation results. A related secondary objective is to develop a prototype dynamic simulator within APECS for advanced power plants and to demonstrate a dynamic simulation that integrates plant-level and equipment-level models.

Among other responsibilities, Alstom Power has the responsibility to demonstrate the capabilities of the enhanced APECS tool to simulate proposed FutureGen power plant concepts and other relevant power generation processes and equipment. The present report encompasses only the work scope within Subtasks 3.4.1 and 5.3a, having to do with the regression ROM for the FutureGen Equipment Model No. 1. Specifically, Alstom Power is tasked with the responsibility to develop a ROM for FutureGen Equipment Model No. 1 based on a regression methodology (Subtask 3.4.1). Subtask 5.3a requires that the ROM then be systematically tested with a simplified version of a FutureGen cycle.

The equipment item selected by the DOE for FutureGen Equipment Model No. 1 was a heat recovery steam generator (HRSG) (specifically the HRSG utilized previously in Demonstration Case 3 of the preceding Vision 21 (Phase I) program). A regression ROM was built from multiple runs of the baseline 3-D CFD case. The ROM was instantiated in the NETL flowsheet and exercised for various inlet flow rate perturbations.

In addition, a regression ROM was also constructed for a second HRSG (associated with Demonstration Case 2 of the preceding Vision 21 (Phase I) program). This case was easier to exercise than Demonstration Case 3, and served as a simple trial case to surface and eliminate software bugs. A significant difference in ROM robustness was demonstrated between a previous “on-the-fly” collection approach and the current Latin Hypercube sampling procedure in the ROM Builder.

This report documents the case characteristics and results and discusses the viability and capabilities of the linkage/interface software for regression ROMS.

TABLE OF CONTENTS

1.0	BACKGROUND ON APECS PROGRAM	3
2.0	INTRODUCTION	3
3.0	EXECUTIVE SUMMARY	5
4.0	EXPERIMENTAL WORK.....	6
5.0	RESULTS AND DISCUSSION.....	6
5.1	Demo Case 2: Case Setup and Preparation	7
5.1.1	Demo Case 2: Introduction	7
5.1.2	Demo Case 2: FLUENT® 3-D Grid and Case Setup	9
5.1.3	Demo Case 2: CO Parameters, Ports, and Connecting Streams	10
5.2	Demo Case 2: Computational ROM Results.....	13
5.2.1	Demo Case 2: “On-the-Fly” Simulation Collections and ANN ROM Results.....	13
5.2.2	Demo Case 2: Present Methodology for ROM Construction	18
5.2.2	Demo Case 2: Integrated Co-Simulation With ROM	25
5.3	Demo Case 3: Case Setup and Preparation	28
5.3.1	Demo Case 3: Introduction	28
5.3.2	Demo Case 3: FLUENT® 3-D Grid and Case Setup	29
5.3.3	Demo Case 3: CO Parameters, Ports, and Connecting Streams	31
5.4	Demo Case 3: Computational ROM Results.....	37
5.4.1	Demo Case 3: “On-the-Fly” Simulation Collections and ANN ROM Results	37
5.4.2	Demo Case 3: Present Methodology for ROM Construction	38
5.4.3	Demo Case 3: Integrated Co-Simulation With ROM	44
6.0	CONCLUSIONS AND RECOMMENDATIONS	48
7.0	REFERENCES	50
8.0	DELIVERABLES.....	51

LIST OF FIGURES

Figure 1:	HRSR Geometry for Demonstration Case 2.....	7
Figure 2:	Collection of Heat Exchanger Blocks Represented by APECS HRSR Block.....	8
Figure 3:	HRSR Portion of Flowsheet with Duplicator Blocks (in Yellow).....	9
Figure 4:	Flowsheet Showing Instantiated Demo C2 ROM.....	10
Figure 5:	Pre-Computed Results for Demo C2 ANN ROM (Model G).....	14
Figure 6:	Solution Paths for FLUENT® vs NN ROM for Demo C2 (Model G, -30,000 lb _m /hr)....	16
Figure 7:	Pre-Computed Results for Demo C2 ANN ROM (Model H).....	17
Figure 8:	Solution Paths for FLUENT® vs NN ROM for Demo C2 (Model H, -30,000 lb _m /hr)....	18
Figure 9:	FLUENT® Versus ROM Predictions for the Temperature at Port Fwecon1-out.....	22
Figure 10:	FLUENT® Versus ROM Predictions for the Temperature at Port Hpsh1-out.....	23
Figure 11:	FLUENT® vs 1 st -3 rd Order ROM for the Temperature at Hpsh1-out (Sims 1-10).....	24
Figure 12:	FLUENT® vs 1 st -3 rd Order ROM for the Temperature at Hpsh1-out (Sims 21-30).....	25
Figure 13:	HRSR Tube Banks and Geometry for Demonstration Case 3 (Expanded View).....	29

Figure 14: Exterior Grid View of the HRSG Geometry for Demonstration Case 3.....	30
Figure 15: Flowsheet Showing Instantiated Demonstration Case 3 FLUENT®/ROM Block.....	31
Figure 16: Flowsheet View (Enlarged) Showing Instantiated Demonstration Case 3 Block.....	32
Figure 17: FLUENT® and ANN ROM Convergence Comparison for Demonstration Case 3.....	38
Figure 18: FLUENT® Versus ROM Predictions for the Temperature at Port Hpsh1-out.....	42
Figure 19: FLUENT® Versus ROM Predictions for the Temperature at Port Ipsh1-out.....	43
Figure 20: FLUENT® vs 1 st -3 rd Order ROM for the Temperature at Ipsh1-out (Sims 41-50).....	44
Figure 21: FLUENT® and ROM Convergence Comparison for Demonstration Case 3.....	48

LIST OF TABLES

Table 1: FLUENT® Block Ports and Connecting Streams for Demonstration Case 2.....	10
Table 2: Species Listing and Mapping for Demonstration Case 2.....	11
Table 3: CAPE-OPEN Parameters for Demonstration Case 2.....	11
Table 4: Demo Case 2: Comparison of FLUENT® and ANN ROM Results (Model G).....	15
Table 5: Demo Case 2: Comparison of FLUENT® and ANN ROM Results (Model H).....	17
Table 6: ROM Variable Range for Demonstration Case 2.....	20
Table 7: Co-Simulation ROM Robustness Testing for Demonstration Case 2.....	27
Table 8: FLUENT® Block Ports and Connecting Streams for Demonstration Case 3.....	32
Table 9: Species Listing and Mapping for Demonstration Case 3.....	33
Table 10: CAPE-OPEN Parameters for Demonstration Case 3.....	33
Table 11: Convergence Loops for Demonstration Case 3.....	37
Table 12: ROM Variable Range for Demonstration Case 3.....	40
Table 13: Co-Simulation ROM Robustness Testing for Demonstration Case 3.....	46

1.0 BACKGROUND ON APECS PROGRAM

The current APECS program is a continuation of an earlier (Vision 21) program, which was initiated under the DOE-Fluent Cooperative Agreement No. DC-FC26-00NT40954 (hereafter referred to as Phase I). Both programs have been dedicated to the development of the software infrastructure for advanced power plant simulations. The goal of the overall program work scope has been to develop an integrated suite of software tools that could be used to simulate and visualize advanced plant concepts that are being considered under the umbrella of DOE's broad CO₂ remediation and power plant efficiency programs. Advanced simulation tools are needed to evaluate new power plant concepts that will undoubtedly help to minimize costly laboratory and field trials.

The typical process simulation software that has been commercially available to industry does not meet the DOE's objective of "virtual simulation" that is needed to evaluate complex cycles. The intent of the DOE has been to improve predictive tools for cycle analysis, and to improve the component models that are used in turn to simulate the cycle. Generally, the modeled performance of various components is derived from a simple set of reaction and output prescriptions that are collectively accessed by the cycle simulation software as a "black-box" library module, without any explicit impact from geometry modifications and enhancements to the fluid mechanics. These simple component models are not sufficient for component design; nor are they adequate for the evaluation of the impact of complex components on cycle performance. To meet the DOE's goal of predicting and visualizing the performance of these complex systems, there has been a need to use more sophisticated component models, by upgrading or converting library modules to CFD modules, and to couple the CFD modules directly with other equipment items in the power plant through process modelling software. The primary focus of the present APECS project has been to provide the software interface and environment that permits FLUENT® or a surrogate ROM to run in tandem with the Aspen Plus® process simulation package (see Ref. 1 for an overview of the program).

2.0 INTRODUCTION

The feasibility of using FLUENT® CFD models in concert with a process model like Aspen Plus® has been a focus of the APECS project. The role of Alstom Power Inc. (Alstom) has been to assist the project team in helping to develop and demonstrate the capabilities of the advanced APECS simulation and visualization tool, oftentimes with the use of Alstom designed and validated equipment modules (see e.g., Ref. 2 to 7). The primary Alstom responsibilities and tasks have included:

- providing its expertise and experience base in the utilization of both CFD and cycle analysis for the power generation industry,
- selecting and running cases to test and demonstrate the feasibility of the concepts,
- forming an advisory board to provide project review and feedback.

Among other responsibilities, Alstom Power has the responsibility to demonstrate the capabilities of the enhanced APECS tool to simulate proposed FutureGen power plant concepts and other relevant power generation processes and equipment. The present report encompasses only the work scope

within Subtasks 3.4.1 and 5.3a, having to do with the regression ROM for the FutureGen Equipment Model No. 1. Subtasks 3.4.1 and 5.3a are detailed within the Project Management Plan (Ref. 8). Alstom's responsibilities associated with those subtasks are to:

- [Subtask 3.4.1] Develop a ROM for FutureGen Equipment Model No. 1 based on a regression methodology
- [Subtask 5.3a] Systematically test the ROM with a simplified version of a FutureGen cycle.

The FutureGen Equipment Model No. 1 was selected by the DOE to be a heat recovery steam generator (HRSG), specifically the HRSG associated with Demonstration Case 3 in the Phase I program.

It should be noted that in the Phase I program, two demonstration cases, HRSGs were investigated:

- Demonstration Case 2 -- a 270 MWe, natural gas combined cycle (NGCC), consisting of a gas turbine, steam turbine, heat recovery steam generator (HRSG), etc.
- Demonstration Case 3 – a 250 MWe FutureGen integrated gasification combined cycle (IGCC), with an air separation unit (ASU), CO₂ capture, a gas turbine burning hydrogen-enriched syngas, a pressure-swing absorption (PSA) section for hydrogen stream production, a heat recovery steam generator (HRSG), and acid-gas cleanup.

In the Phase I program, engineering design specifications for the HRSG were extracted from the process models, and a CFD model was constructed in accordance with the design. Both the gas side and the steam/water side of the CFD model were calibrated to the design specifications. The CFD case was instantiated directly in the flowsheet and exercised over a range of flow rates.

The focus of the present task is to build a regression ROM for Demonstration Case 3, and exercise it in the FutureGen flowsheet. A regression ROM was built from multiple runs of the baseline 3-D CFD case. The ROM was instantiated in the NETL flowsheet and exercised for various inlet flow rate perturbations.

In addition, a regression ROM was also constructed for the HRSG associated with Demonstration Case 2. This case represented a much simpler cycle than that of Demonstration Case 3, and was therefore easier to instantiate and exercise than Demonstration Case 3, and served as a simple trial case to surface and eliminate software bugs.

The deliverables include a task report, as well as all model files. The task report contains a description of the case simulations, the results, and an itemization of desired software modifications for future work. Both Demonstration Case 2 and Demonstration Case 3 will be discussed in the report. This report summarizes and documents the demonstration case simulations under Task 5.3a, and thus completes the requirement for the stipulated (contractual) task report.

3.0 EXECUTIVE SUMMARY

The primary objective of this DOE project is to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of Advanced Process Engineering Co-Simulator (APECS), the simulator will link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. Generally, the simulator couples the cycle analysis software Aspen Plus® (marketed by Aspen Technology, Inc.), specialized component modeling packages, as exemplified by industrial proprietary codes (utilized by ALSTOM Power Inc.), and the FLUENT® computational fluid dynamics (CFD) code (provided by ANSYS Inc). An important enhancement to the APECS tool is the creation of computationally efficient reduced-order models (ROMs) based on information from high-fidelity CFD models.

The demonstrated results from this APECS project helps to showcase a novel and unique advancement in the field of process modeling and design – the application and use of regression ROMs built using a Latin Hypercube sampling procedure in the new APECS ROM Builder. Alstom Power was tasked with the responsibility to (a) develop a ROM for FutureGen Equipment Model No. 1 (HRSG) based on a regression methodology (Subtask 3.4.1), and to (b) systematically test the ROM with a simplified version of a FutureGen cycle (Subtask 5.3a). These tasks were accomplished. This report documents the case characteristics and results and discusses the viability and capabilities of the APECS controller and wrapper software for regression ROMS.

The equipment item selected by the DOE for FutureGen Equipment Model No. 1 was a heat recovery steam generator (HRSG) (specifically the HRSG utilized previously in Demonstration Case 3 of the preceding Vision 21 (Phase I) program). A regression ROM was built from multiple runs of the baseline 3-D CFD case. The ROM was instantiated in the NETL flowsheet and exercised for various inlet flow rate perturbations.

In addition, a regression ROM was also constructed for a second HRSG (associated with Demonstration Case 2 of the preceding Vision 21 (Phase I) program). This case was easier to exercise than Demonstration Case 3, and served as a simple trial case to surface and eliminate software bugs. A significant difference in ROM robustness was demonstrated between a previous “on-the-fly” collection approach and the current Latin Hypercube sampling procedure in the ROM Builder.

Substantial progress has been since project inception by ANSYS in facilitating the direct instantiation of FLUENT® on an Aspen Plus® flowsheet, as well in building FLUENT®-derived ROMs. It is hoped that the APECS tool kit will continue to be developed and enhanced. Many bugs have been surfaced and eliminated and the software has been made more user-friendly and transparent. The documentation is maturing and is a great help in assisting the user to understand the procedures and work flow.

Recommendations are as follows:

- In the ROM Builder, provide good statistical indicators that quantify the overall superiority or “goodness of fit” of one regression order over another (e.g., 1st-order regression versus 2nd- or 3rd-order regression).
- In the ROM Builder, allow the user the option to see the ROM versus FLUENT® prediction behavior for all of the runs simultaneously as opposed to the default 10 runs.
- FLUENT® users often utilize convergence monitors (e.g., mass-weighted T, O₂ at the outlet plane). Currently, in the ROM Builder, there is really no way for the user to check how well each of the cases is converging. It may be a good idea for the monitors to be saved, just like PCA plots are saved for each simulation, so that the user can call them up later (after all the simulations are done) and look at them so see how well each case behaved.
- One of the issues with ROM building is that the user must prescribe maxima and minima for each of the independent inlet variables. For a large number of inlet variables, this can become an ambiguous and tedious task. It is recommended that ANSYS find the “best” way for the user to prescribe the Min/Max range for each ROM variable and automate as much as possible. For example, is it possible to work with AspenTech and tag an equipment item in a flowsheet, such that the minima and maxima of each stream and CAPE-OPEN variable, associated the equipment item, are calculated and reported when the flowsheet is converged?
- The FLUENT® tube bank heat exchanger model required a steam-water property calculator. Since so many equipment items require steam-water properties, it would be advisable to have a steam-water property calculator that would always be available to the FLUENT® tube bank heat exchanger model, both in the direct-instantiation mode, and in the ROM Builder mode (independent of the platform that it is running on).

4.0 EXPERIMENTAL WORK

There is no experimental work under this project.

5.0 RESULTS AND DISCUSSION

Candidate models for the FutureGen Equipment Model No. 1 included a gasifier, steam turbine (low to high pressure), air preheater, syngas cooler (convective or radiative), etc. The NETL was consulted in the selection of the Equipment Model No. 1 from the list of candidate equipment items. The FutureGen Equipment Model No. 1 was selected by the DOE (in the person of Steve Zitney) in 2007 to be a heat recovery stream generator (HRSG), specifically the HRSG associated with Demonstration Case 3 in the Phase I program.

Demonstration Case 2 (from the Phase I program; see e.g., Ref. 3, 4, 5, and 7) is considered to be a much simpler HRSG case than Demonstration Case 3, as evidenced by a simpler cycle and fewer material streams connected to the APECS block. Therefore, Demonstration Case 2 was used to debug and test the APECS tool kit and ROM procedure before the methodology was applied to

Demonstration Case 3. Furthermore, Demonstration Case 2 was used to test some early ROM building efforts. Since the Demonstration Case 2 results are germane to this effort, this case will also be discussed in some detail.

5.1 Demo Case 2: Case Setup and Preparation

5.1.1 Demo Case 2: Introduction

The natural gas combined cycle (NGCC) power plant associated with Demonstration Case 2 consists of an advanced gas turbine, steam turbine, generator, and HRSG, all supplied by ALSTOM Power. In combined cycle mode, the power plant operates at a net efficiency of 57.5%. The gas turbine generates approximately 2/3 of the 270 MW of electrical output from the NGCC power plant. The gas turbine generator has an efficiency of 38.5% when firing natural gas fuel. The exhaust gas exits the gas turbine at a temperature around 923 K where it enters an HRSG. The HRSG contains both high and low pressure evaporative and superheat surface as well as HP reheat.

The HP feed pump also takes water from the LP steam drum, a small part of which is sent to the gas turbine (GT) cooler. Most of the HP feedwater flows through the HP economizer and then into the “once-through” (i.e., no drum) evaporator section where it exits as slightly superheated steam. The steam is then sent to the HP separator where it is mixed with superheated steam from the GT cooler. The steam is then superheated and conditioned in the HP desuperheater and sent to the HP steam turbine. From the steam turbine outlet, the steam passes through a reheat (RH) section and into the RH desuperheater.

At the maximum continuous rating (MCR), the HRSG is designed to provide a superheat outlet steam flow of approximately 60 kg/s at 838 K and 16.5 MPa. The design reheat steam flow is approximately 59 kg/s at 836 K and 3.6 MPa. The HRSG component and “island” are the focus of the CFD and ROM co-simulations. The HRSG geometry is represented in Figure 1.

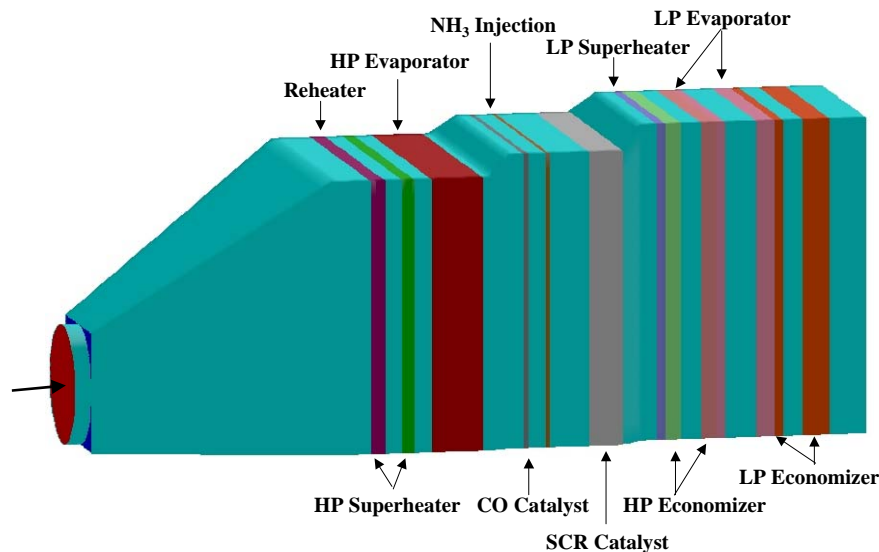


Figure 1: HRSG Geometry for Demonstration Case 2.

The CAPE-OPEN (CO) block, representing the CFD module or the ROM module, is instantiated on the flowsheet and represents the collection of heat exchanger modules shown in Figure 2:

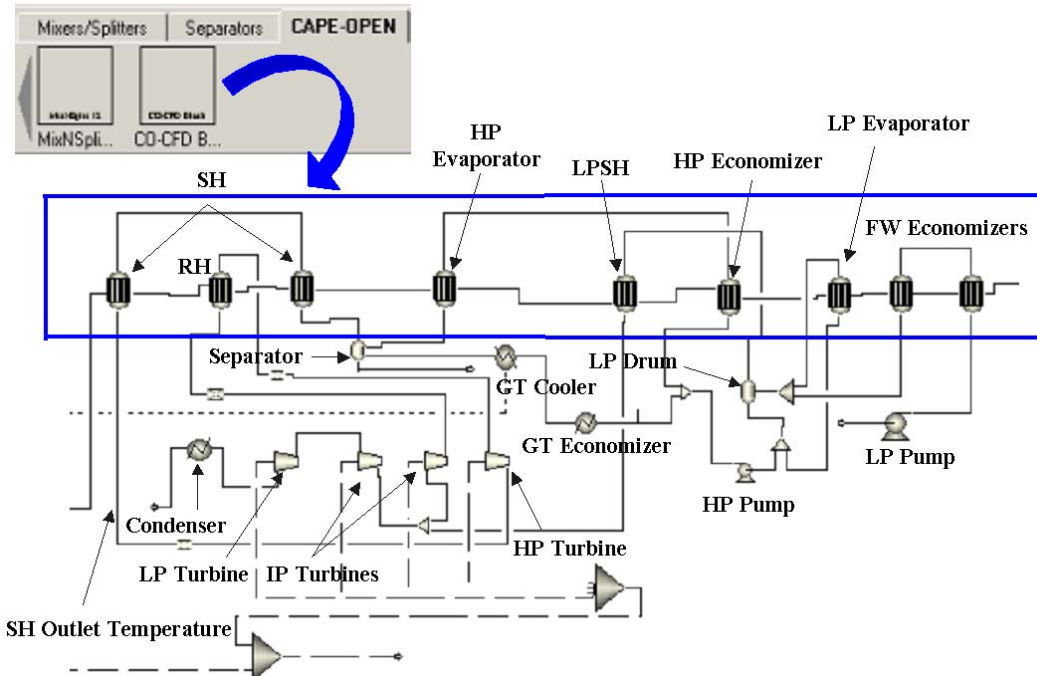


Figure 2: Collection of Heat Exchanger Blocks Represented by APECS HRSG Block.

It has been found, that to enhance flowsheet convergence, the instantiation of the APECS block should not simply replace the designated heat exchanger modules. Each tube bank in the HRSG (or group of tube banks in series) is represented by a single heat exchanger (HX) icon on the Aspen Plus flowsheet. The inlet stream to each HX icon is duplicated (by a “duplicator block”) and connected to the FLUENT® block icon via physical model ports (see Figure 3). Although physical model ports for each tube bank outlet stream are also constructed and connected to the FLUENT® icon, they are not accessed. Instead, CO variables are defined for the outlet parameters of each tube bank and such “informational” CO variables are passed back to Aspen Plus and are used to overwrite the corresponding HX icon variables. The HX icons are solved prior to the FLUENT® block, thus allowing tube bank inlet conditions to be transferred to FLUENT® which satisfy global mass and energy balances (based on the assigned heat exchanger characteristics). Subsequently, the FLUENT® block is converged, and the updated tube bank outlet conditions are passed back to the Aspen Plus HX icons for the next iteration. If the HX icons were not retained on the flowsheet, then some of the inlet physical model port streams attached to FLUENT®, which depend on upstream flows which have not yet passed through FLUENT®, would not satisfy global mass and energy balances, and overall convergence might be adversely impacted. In the present strategy, mass and energy balances are ensured for all of the HRSG HX icons collectively, prior to the transfer of input stream information to the tube bank models in the FLUENT® block. It is believed that the current

strategy is advantageous for those CFD modules or blocks that are simultaneously connected to many parts of the cycle.

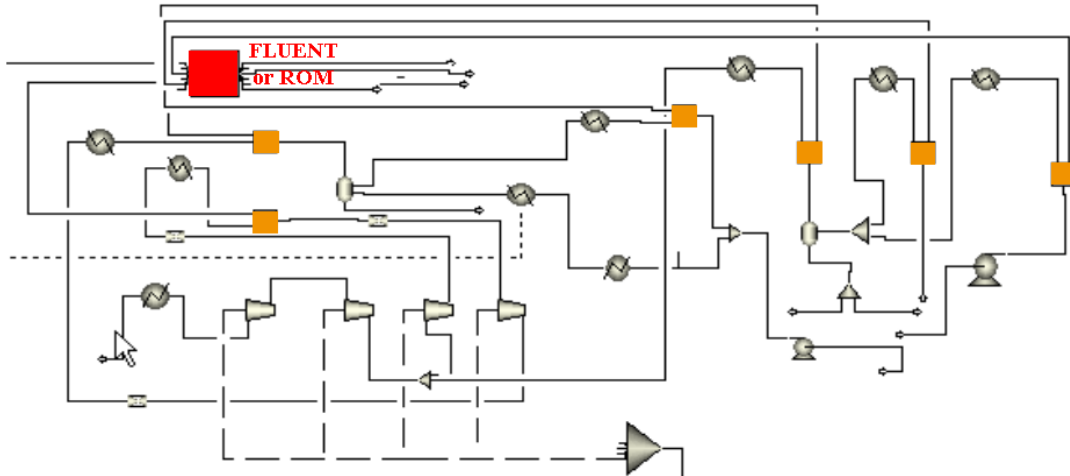


Figure 3: HRSG Portion of Flowsheet with Duplicator Blocks (in Yellow).

Following the control strategy for once-through HRSGs, an Aspen Plus “design spec” utility, based on the Broyden method, manipulates the HP flow rate until the desired superheat outlet temperature is achieved. The Broyden algorithm also simultaneously converges the CO variables (denoted as “Fortran tears”) transferred between the FLUENT® block and Aspen Plus. For Demonstration Case 2, the six groups of heat exchangers (single or in series) required 24 CO variables (principally for monitoring purposes), of which only 12 were transferred from FLUENT® to Aspen Plus (2 per HX icon), and which were manipulated by the Broyden algorithm as “Fortran tears.”

5.1.2 Demo Case 2: FLUENT® 3-D Grid and Case Setup

During the Phase I program, the FLUENT® 3-D case was generated and the heat exchanger tube banks surface effectiveness factors were calibrated to the design information from an in-house HRSG design code. To expedite calculations, a grid with approximately 40,000 cells was used.

Some of the other CFD case characteristics and selected submodel options are itemized below:

- FLUENT® V6.3.26 (on a PC) was used for all of the CFD investigatory computations, as well as for the ROM preparation runs.
- The solution algorithm is based upon the SIMPLE pressure-correction algorithm with standard discretization for pressure and first-order upwind differencing for all of the other transport equations.
- The turbulence model is the Realizable k- ϵ turbulence model, with standard wall functions.
- The external walls are adiabatic.
- Species transport equations were activated for 5 species (CH₄, O₂, CO₂, H₂O, N₂), without reaction.
- The radiation submodel was turned off.

Domain Inlet and Outlet ID	Physical Model Inlet Port	Corresponding Aspen Stream Name	Physical Model Outlet Port	Corresponding Aspen Stream Name
2	fwecon2_group	LPECIN	fwecon1-out	ECON1OT
3	hpecon_group	HPECEVIN	hpevap-out	HPEVAPOT
4	hpsh2_group	HPSHIN	hpsh1-out	HPSH1OT
5	lpevap_group	LPEVAPIN	lpevap-out	LPEVAPOT
6	lpsh_group	LPSHIN	lpsh-out	LPSHOT
7	rh_group	RHIN	rh-out	RHOT

The species listed in both the FLUENT® case and the Aspen Plus® flowsheet, as well as their respective mappings to each other, are provided in the following table:

Table 2: Species Listing and Mapping for Demonstration Case 2

	Species in the Materials Listing in FLUENT®	FLUENT® Species Exposed to the PME	Flowsheet Species
(1)	ch4	ch4	CH4
(2)	o2	o2	O2
(3)	co2	co2	CO2
(4)	h2o	h2o	H2O
(5)	n2	n2	N2

The CAPE-OPEN parameters set through the APECS Wizard are provided in the following table:

Table 3: CAPE-OPEN Parameters for Demonstration Case 2

Parameters	Type	Description	Access Mode	Lower Bound	Upper Bound	Units	Default Value	Default Value in SI Units
fwecon1-temp-out	real		read-write	77	2,060	F	320.00	433.15
fwecon1-pres-out	real		read-write	58	14,504	psi	109.845	757,354.61
fwecon1-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	563,701.00	71.03
fwecon1-qual-out	real		read-write	0	1	---	0.00	0.00
hpevap-temp-out	real		read-write	77	2,060	F	740.00	666.48
hpevap-pres-out	real		read-write	58	14,504	psi	2,513.860	17,332,454.54
hpevap-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	419,531.00	52.86
hpevap-qual-out	real		read-write	0	1	---	1.00	1.00
hpsh1-temp-out	real		read-write	77	2,060	F	1,049.00	838.15
hpsh1-pres-out	real		read-write	58	14,504	psi	2,405.040	16,582,167.05
hpsh1-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	474,135.00	59.74
hpsh1-qual-out	real		read-write	0	1	---	1.00	1.00
lpevap-temp-out	real		read-write	77	2,060	F	334.00	440.93
lpevap-pres-out	real		read-write	58	14,504	psi	109.845	757,354.61
lpevap-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	2,000,000.00	252.00
lpevap-qual-out	real		read-write	0	1	---	0.05	0.05
lpsh-temp-out	real		read-write	77	2,060	F	611.00	594.82
lpsh-pres-out	real		read-write	58	14,504	psi	105.845	729,775.58
lpsh-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	89,566.00	11.29
lpsh-qual-out	real		read-write	0	1	---	1.00	1.00

rh-temp-out	real		read-write	77	2,060	F	1,046.00	836.48
rh-pres-out	real		read-write	58	14,504	psi	574.941	3,964,078.65
rh-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	474,135.00	59.74
rh-qual-out	real		read-write	0	1	---	1.00	1.00
calculate-user-journal-activate	integer	activates the user journal file	read-write	0	1	---	1	1
calculate-user-journal-name	option	name of user journal file	read-only	---	---	---		prep_resid.jou
maximum-iterations	integer	maximum number of FLUENT® iterations	read-write	1	100,000	---		1,000

The calculate-user-journal-activate parameter is actually a Boolean, but a bug in Aspen Plus® has precluded such Booleans from being manipulated in a calculator block. Therefore, it is more convenient to treat the parameter as an integer switch (0=FALSE, 1=TRUE).

Each HX block is associated with 4 CAPE-OPEN variables, but as mentioned earlier, only 2 of the variables (e.g., T, p) for each HX block are passed from FLUENT® to Aspen Plus, and are manipulated and converged by the Broyden algorithm as “Fortran tears.”

The CO parameter variables associated with heat exchanger tube banks are set in the “void report_data_exchange()” UDF (in file hxc.c) through the lines:

```
RP_Set_Real("fwecon1-temp-out", tout[1]*1.8 - 459.67);
RP_Set_Real("fwecon1-pres-out", pout[1]/101325.0*14.6959488);
RP_Set_Real("fwecon1-mdot-out", mout[1]/0.45359237*3600.0);
RP_Set_Real("fwecon1-qual-out", xout[1]);

RP_Set_Real("cape-open/hxc/fwecon1-out/flow-rate", mout[1]);
RP_Set_Real("cape-open/hxc/fwecon1-out/temperature", tout[1]);
RP_Set_Real("cape-open/hxc/fwecon1-out/pressure", pout[1]);
RP_Set_Real("cape-open/hxc/fwecon1-out/outlet-quality", xout[1]);

RP_Set_Real("hpevap-temp-out", tout[3]*1.8 - 459.67);
RP_Set_Real("hpevap-pres-out", pout[3]/101325.0*14.6959488);
RP_Set_Real("hpevap-mdot-out", mout[3]/0.45359237*3600.0);
RP_Set_Real("hpevap-qual-out", xout[3]);

RP_Set_Real("cape-open/hxc/hpevap-out/flow-rate", mout[3]);
RP_Set_Real("cape-open/hxc/hpevap-out/temperature", tout[3]);
RP_Set_Real("cape-open/hxc/hpevap-out/pressure", pout[3]);
RP_Set_Real("cape-open/hxc/hpevap-out/outlet-quality", xout[3]);

RP_Set_Real("hpsh1-temp-out", tout[6]*1.8 - 459.67);
RP_Set_Real("hpsh1-pres-out", pout[6]/101325.0*14.6959488);
RP_Set_Real("hpsh1-mdot-out", mout[6]/0.45359237*3600.0);
RP_Set_Real("hpsh1-qual-out", xout[6]);

RP_Set_Real("cape-open/hxc/hpsh1-out/flow-rate", mout[6]);
RP_Set_Real("cape-open/hxc/hpsh1-out/temperature", tout[6]);
RP_Set_Real("cape-open/hxc/hpsh1-out/pressure", pout[6]);
RP_Set_Real("cape-open/hxc/hpsh1-out/outlet-quality", xout[6]);
```

```
RP_Set_Real("lpevap-temp-out", tout[7]*1.8 - 459.67);
RP_Set_Real("lpevap-pres-out", pout[7]/101325.0*14.6959488);
RP_Set_Real("lpevap-mdot-out", mout[7]/0.45359237*3600.0);
RP_Set_Real("lpevap-qual-out", xout[7]);

RP_Set_Real("cape-open/hxc/lpevap-out/flow-rate", mout[7]);
RP_Set_Real("cape-open/hxc/lpevap-out/temperature", tout[7]);
RP_Set_Real("cape-open/hxc/lpevap-out/pressure", pout[7]);
RP_Set_Real("cape-open/hxc/lpevap-out/outlet-quality", xout[7]);

RP_Set_Real("lpsh-temp-out", tout[4]*1.8 - 459.67);
RP_Set_Real("lpsh-pres-out", pout[4]/101325.0*14.6959488);
RP_Set_Real("lpsh-mdot-out", mout[4]/0.45359237*3600.0);
RP_Set_Real("lpsh-qual-out", xout[4]);

RP_Set_Real("cape-open/hxc/lpsh-out/flow-rate", mout[4]);
RP_Set_Real("cape-open/hxc/lpsh-out/temperature", tout[4]);
RP_Set_Real("cape-open/hxc/lpsh-out/pressure", pout[4]);
RP_Set_Real("cape-open/hxc/lpsh-out/outlet-quality", xout[4]);

RP_Set_Real("rh-temp-out", tout[8]*1.8 - 459.67);
RP_Set_Real("rh-pres-out", pout[8]/101325.0*14.6959488);
RP_Set_Real("rh-mdot-out", mout[8]/0.45359237*3600.0);
RP_Set_Real("rh-qual-out", xout[8]);

RP_Set_Real("cape-open/hxc/rh-out/flow-rate", mout[8]);
RP_Set_Real("cape-open/hxc/rh-out/temperature", tout[8]);
RP_Set_Real("cape-open/hxc/rh-out/pressure", pout[8]);
RP_Set_Real("cape-open/hxc/rh-out/outlet-quality", xout[8]);
```

5.2 Demo Case 2: Computational ROM Results

5.2.1 Demo Case 2: “On-the-Fly” Simulation Collections and ANN ROM Results

In the last half of 2006, a significant effort was initiated with ANSYS (directed by Maxwell Osawe) in generating ROMs for Demonstration Case 2 based on the Artificial Neural Network (ANN) methodology. Although the methodology for the collection of the simulations for the “on-the-fly” training set, described in this section, was known by ANSYS to be mathematically less sophisticated and robust than that to be ultimately developed a couple of years later for the regression ROM Builder, ANSYS desired to make an initial foray into the ANN ROM arena using this simpler collection technique. A discussion of this past methodology and the corresponding results will help to compare and contrast the progress made over that period of time.

The ANN was essentially based on computing the optimum set of weights that minimizes the network error. The implementation interleaves the simulated annealing and classical non-linear Polak-Ribière conjugate gradient techniques. The annealing option helps to avoid false minima. Although both the simulated annealing methods and the conjugate gradient methods were investigated in the present work, it was found that the annealing method produced a ROM that appeared to be more robust over a wider range than the conjugate gradient method.

FLUENT® was instantiated on the flowsheet and then the economizer inlet flow rate was perturbed over a range. During each of the Aspen Plus / FLUENT® co-simulations (each associated with a specific economizer inlet flow rate condition), the input and output results from the FLUENT® block were saved “on the fly” in a MySQL database (the EKM® database had not yet been released for use) each time that FLUENT® was executed within the Aspen Plus flowsheet. The collection of input and output results from each FLUENT® block execution constituted the “simulation data set” or the “table of pre-computed results” on which the ANN was trained to construct a ROM. The ANN was used to construct ROMs trained on a total of between 76 to 233 FLUENT® executions, depending on the flowsheet configuration.

An example of one set of pre-computed results (corresponding to the democ2_model_G configuration) is displayed in Figure 5. Stream 1 (FS1) is perturbed from $-40,000$ to $+20,000$ lb_m/hr from the initial baseline value in the flowsheet, and the resultant superheat outlet temperatures are plotted.

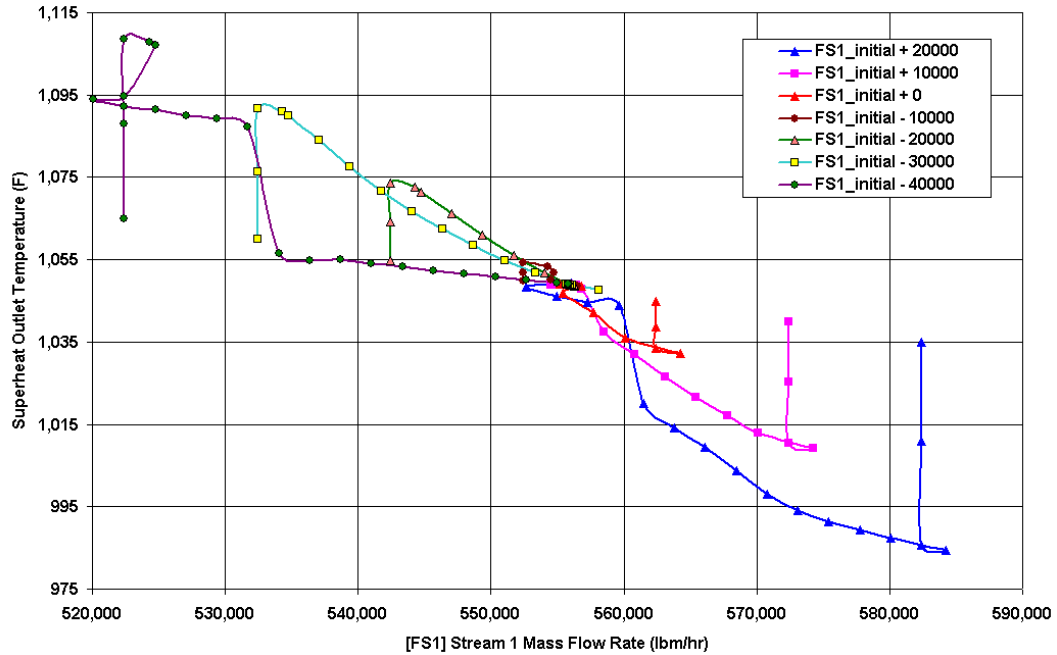


Figure 5: Pre-Computed Results for Demo C2 ANN ROM (Model G).

An ANN ROM was constructed using 10 neurons and the simulated annealing approach. The number of FLUENT® executions required for convergence at each perturbation condition, as well as the corresponding number of the instantiated ANN ROM executions at the same condition, are provided in the following table:

Table 4: Demo Case 2: Comparison of FLUENT® and ANN ROM Results (Model G).

Initial Perturbation to Economizer Flow Rate (lbm/hr)	Number of Executions Required for Convergence in Co-Simulations with Aspen Plus	
	FLUENT®	ANN ROM from "On the Fly" Simulations
20,000	21	73
10,000	19	24
0	12	16
-10,000	9	9
-20,000	15	16
-30,000	20	27
-40,000	26	> 99
Total:	122	---

The ANN ROM was trained on 122 “on-the-fly” simulations. It can be seen that the number of FLUENT® or ANN ROM executions increase as the initial flow rate perturbation deviates from the initial baseline value. The ANN ROM seems to perform well throughout the range, except at the range limits, where the number of executions increases significantly. The ANN ROM did not converge at the $-40,000 \text{ lb}_m/\text{hr}$ condition. It can be surmised that the ANN ROM did not perform well (or extrapolate well) near the extreme limits of the range of the pre-computed results over which it was trained.

Since the number of executions required by the ANN ROM, during flowsheet convergence, is different from that of FLUENT®, it can be supposed that the solution surface is also different. A (primitive) indication of the solution surface can be seen by following the solution path of different flow rate variables; an example is provided in Figure 6:

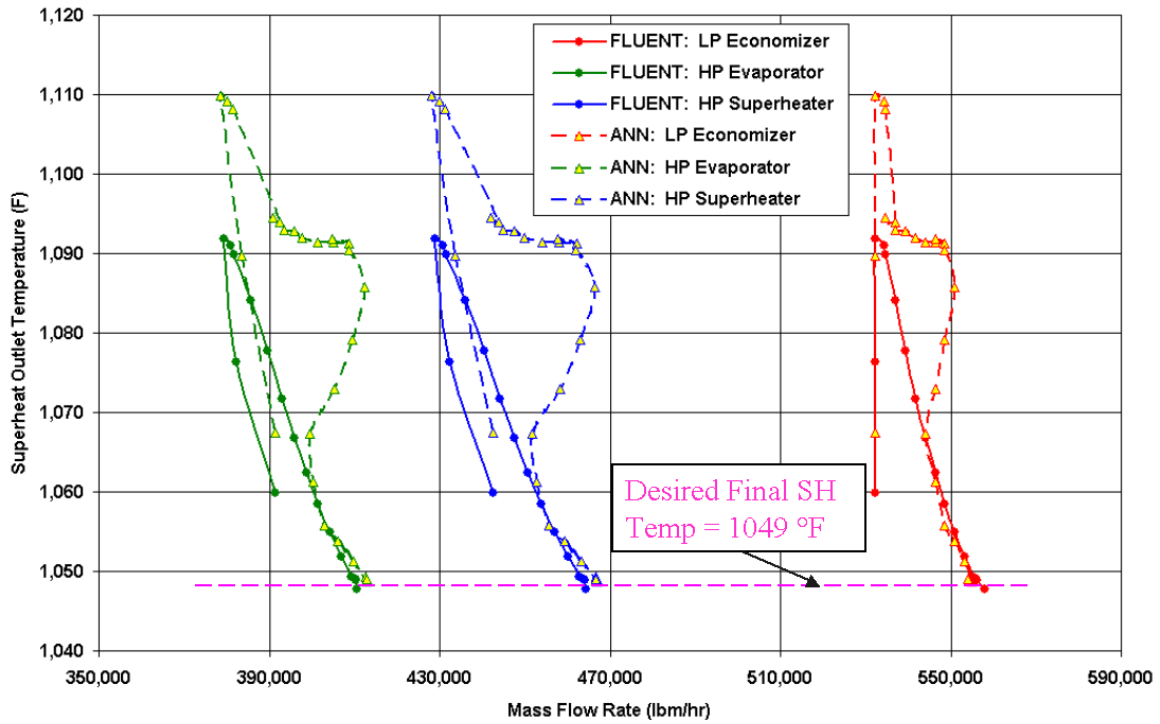


Figure 6: Solution Paths for FLUENT® vs NN ROM for Demo C2 (Model G, -30,000 lb_m/hr).

No statement is made relative to whether the deviation from the FLUENT® solution path is benign or not. The observation is simply made that the paths can be significantly different from each other. It is also not known at this time whether any “so-called” higher-order ROMs would be more aligned with the FLUENT® solution path.

The flowsheet was then modified by transferring the CAPE-OPEN parameters (P, Heat Duty) from FLUENT® to the heater block HXHPECEV, instead of the parameters (P,T). This new flowsheet configuration is denoted as the democ2_model_H configuration. The set of pre-computed results (corresponding to the democ2_model_H) is displayed in Figure 7. Again, Stream 1 (FS1) is perturbed from -40,000 to +20,000 lb_m/hr from the initial baseline value in the flowsheet, and the resultant superheat outlet temperatures are plotted.

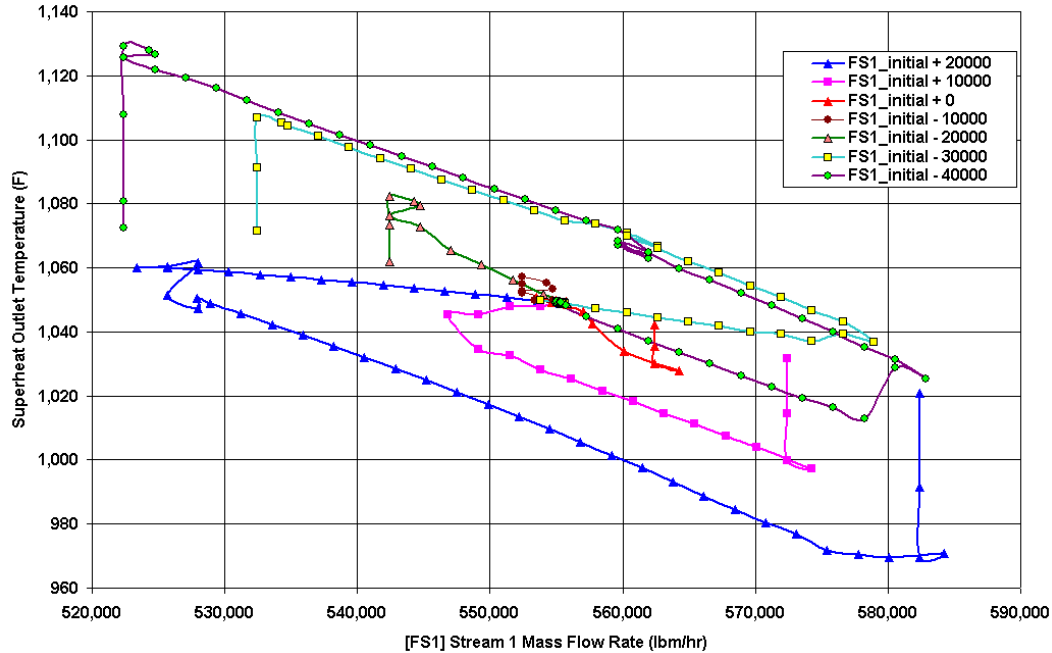


Figure 7: Pre-Computed Results for Demo C2 ANN ROM (Model H).

The Model-H configuration solution paths appear to be very different from those of Model-G. An ANN ROM was constructed using 10 neurons and the simulated annealing approach. The number of FLUENT® executions required for convergence at each perturbation condition, as well as the corresponding number of the instantiated ANN ROM executions at the same condition, are provided in the following table:

Table 5: Demo Case 2: Comparison of FLUENT® and ANN ROM Results (Model H).

Initial Perturbation to Economizer Flow Rate (lbm/hr)	Number of Executions Required for Convergence in Co-Simulations with Aspen Plus	
	FLUENT®	ANN ROM from "On the Fly" Simulations
20,000	55	84
10,000	22	21
0	21	14
-10,000	17	12
-20,000	15	54
-30,000	49	39
-40,000	54	> 99
Total:	233	---

The ANN ROM was trained on 233 “on-the-fly” simulations. Again, the ANN ROM seems to perform reasonably well throughout the range, except at the range limits, where the number of executions increases significantly. Even with 233 simulations in the training set, the ANN ROM did not converge at the $-40,000 \text{ lb}_m/\text{hr}$ condition.

As before, the solution paths of different flow rate variables can be tracked as an indicator of how different the solution paths might be between FLUENT® and the ANN ROM. An example is shown in Figure 8:

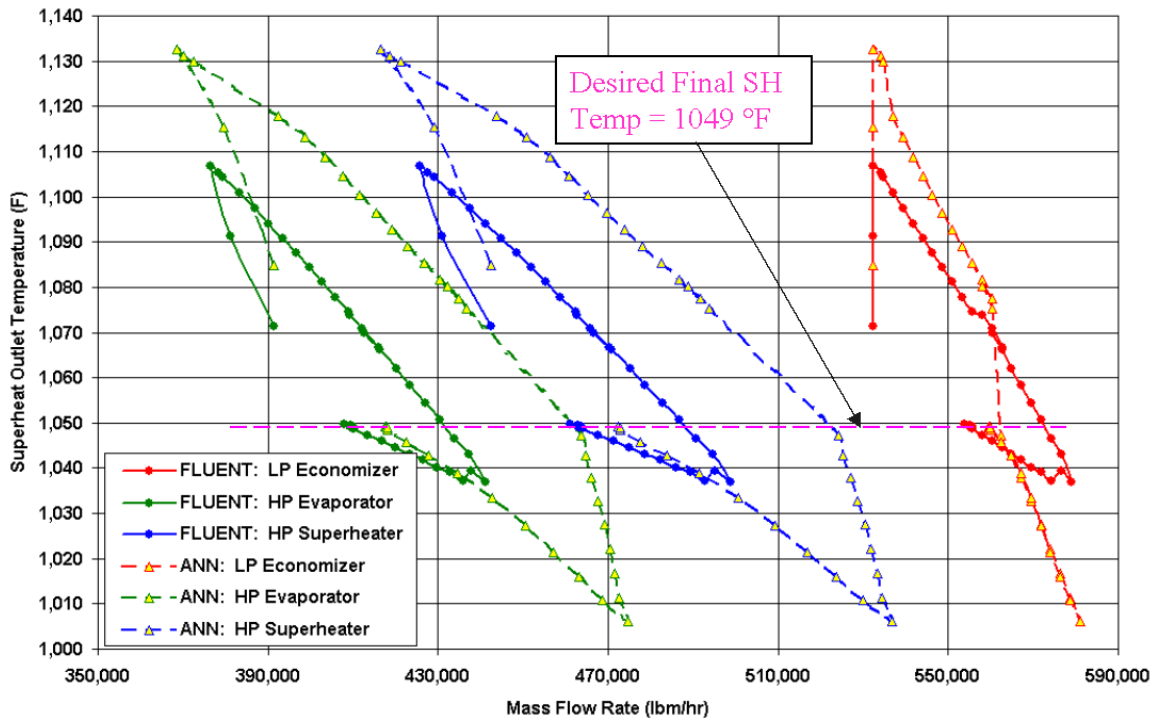


Figure 8: Solution Paths for FLUENT® vs NN ROM for Demo C2 (Model H, $-30,000 \text{ lb}_m/\text{hr}$).

The results from the “on-the-fly” collection technique and resultant ANN ROM will be compared with the results from the ROM Builder and the regression ROM in subsequent sections.

5.2.2 Demo Case 2: Present Methodology for ROM Construction

The 3-D CFD case for the HRSG was used in the ROM Builder. It was run on a Windows PC with the following software:

- ANSYS EKM® V1.2
- FLUENT® V6.3.26
- ANSYS APECS V1.7.0.019
- COfluent.6.3.26 Wrapper V1.7.0.015

The FLUENT® case files were prepared and made CAPE-OPEN and APECS compatible by sequentially performing the steps outlined in the APECS User's Manual (Ref. 9):

- Execute FLUENT® CAPE-OPEN Configuration Wizard
- Execute APECS Model Wizard
- Upload files to the ANSYS Engineering Knowledge Manager (EKM®) database
- Launch the APECS User Interface to build the ROM

The files uploaded to the EKM® database consist of the following:

CpofPH.dat
ETAofPT.dat
HofPT.dat
SATPropsOfPsat.dat
TcofPT.dat
TofPH.dat
XofPH.dat
demo_c2_7300_CO.cas
demo_c2_7300_CO.dat
drum.scm
libudf/ntx86/3d/libudf.dll
mixing-constants.scm (← not used)
prep_resid.jou

The APECS software allows for a number of journal files, which are helpful in providing increased control to the run sequence. ANSYS has prepared a user journal file activated via the CAPE-OPEN parameters:

- calculate-user-journal-name
- calculate-user-journal-activate

For the present task, this user journal file, called “prep_resid.jou,” was used to help run a minimum number of iterations by:

- (a) lowering one or more of the residual tolerances
- (b) running a specified number of iterations
- (c) and then putting the residual tolerances back where they were originally

The APECS calculate command is executed after the journal file commands are completed. If the residuals are less than the stipulated thresholds, then the case is deemed converged; otherwise, it begins to calculate the number of iterations dictated by APECS. While any need for this journal file may be obviated by judicious selection of the residual thresholds, it is sometimes difficult to tell when a case is converged merely by looking at the residuals. Furthermore, it is difficult to know what the residual thresholds should be for operating conditions removed from those of the baseline case. Since it is difficult to determine the proper convergence levels for the residuals, a good compromise is to utilize a user-journal file, which forces a user-specified minimum number of iterations:

```
; prep_resid.jou  
;
```

```

;
k epsilon ch4 o2 co2 h2o cont x y z energy
;
/solve/monitors/residual/convergence-criteria 1.e-09 2.e-06 2.e-06 1.e-05 2.e-06
2.e-05 2.e-05 0.001 1.e-07 1.e-07 1.e-07
/solve/iterate 200
/solve/monitors/residual/convergence-criteria 3.e-05 2.e-06 2.e-06 1.e-05 2.e-06
2.e-05 2.e-05 0.001 1.e-07 1.e-07 1.e-07

```

Under the “Regression ROM” tab, the following options were selected:

- Order of Regression = 1
- Save Every Instance = Yes
- Mass Balance Tolerance = 0.001 (default)

ROM inputs were supplied for all of the ports, as well as the CAPE-OPEN parameters. Since the flowsheet had previously been run for a number of perturbed economizer flow rates for the on-the-fly data collection, the on-the-fly results were used as a guide for the maximum and minimum values for each of the independent variables. The range for the ROM Builder was made slightly larger than that of the on-the-fly data collection. It was very difficult to know how much larger the range could be made for each independent variable. Since it was known, from past experience, that the steam-water property lookup tables used in the present CFD case were not very robust, it was decided to error on the side of caution and choose conservative values for the maxima and minima. A condensed version of the variable range table for the ROM construction is provided below:

Table 6: ROM Variable Range for Demonstration Case 2

	Units	Results from Previous Runs			Input for ROM Builder			
		Minimum	Maximum	Base Values	Include	Minimum	Maximum	Base Values
hpecon_group								
Temperature	(K)	444.149	444.149	444.149	No	100	1000	444.149
Pressure	(Pa)	17353200	17353200	17353200	No	101325	2.00E+07	17353200
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	45.88795343	56.25884421	51.69952197	Yes	43	58	51.69952197
rh_group								
Temperature	(K)	612.118	647.989	629.728	Yes	600	660	629.728
Pressure	(Pa)	3943590	3943590	3943590	No	101325	5.00E+06	3943590
Phase Fraction	Quality	1	1	1	No	0	1	1
H2O	(kg/s)	51.88616089	63.61252585	58.45743631	Yes	48	66	58.45743631
inlet								
Temperature	(K)	923.303	923.303	923.303	No	100	1000	923.303
Pressure	(Pa)	104939	104939	104939	No	101325	2.00E+05	104939
CH4	(kg/s)	0	0	0	No	0	1000	0
O2	(kg/s)	47.66810952	47.66810952	47.66810952	No	0	1000	47.66810952
CO2	(kg/s)	25.73264962	25.73264962	25.73264962	No	0	1000	25.73264962
H2O	(kg/s)	25.84684668	25.84684668	25.84684668	No	0	1000	25.84684668
N2	(kg/s)	282.5850127	282.5850127	282.5850127	No	0	1000	282.5850127
hpsh2_group								
Temperature	(K)	629.74	763.561	667.169	Yes	617	775	667.169
Pressure	(Pa)	16692500	16692500	16692500	No	101325	2.00E+07	16692500
Phase Fraction	Quality	1	1	1	No	0	1	1

H2O	(kg/s)	51.88616089	63.61252585	58.45743631	Yes	51.6	63.9	58.45743631
lpsh_group								
Temperature	(K)	441.301	441.301	441.301	No	100	1000	441.301
Pressure	(Pa)	757355	757355	757355	No	101325	8.00E+05	757355
Phase Fraction	Quality	1	1	1	No	0	1	1
H2O	(kg/s)	10.00433265	13.93654092	11.58047674	Yes	7	16	11.58047674
fwecon2_group								
Temperature	(K)	311.571	311.571	311.571	No	100	1000	311.571
Pressure	(Pa)	757355	757355	757355	No	101325	8.00E+05	757355
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	65.52953818	73.61698461	70.03787701	Yes	63	76	70.03787701
lpevap_group								
Temperature	(K)	441.302	441.302	441.302	No	100	1000	441.302
Pressure	(Pa)	757355	757355	757355	No	101325	8.00E+05	757355
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	251.9967744	251.9967744	251.9967744	No	0	1000	251.9967744

The present case has only 7 independent inlet variables (as indicated by “Yes” under the “Include” column). A total of 100 pre-computed simulations was specified in the ROM Builder -- 1 simulation with the baseline values, 14 simulations corresponding to maximum and minimum values (for the 7 independent inlet variables varied), and 85 user-specified simulations (using the “Design-of-Experiments” option based on the Latin Hypercube sampling procedure).

Each of the 100 simulations ran an initial 200 iterations (as stipulated by prep_resid.jou), followed by an additional 1,000 potential iterations (as stipulated by the “maximum-iterations” parameter in Table 3), presuming that the convergence criteria were not achieved, for a total possible maximum number of iterations of 1,200 per simulation. The ROM Builder completed the 100 simulations in approximately 24 hours elapsed time on a single processor of a PC. The mass balance was achieved for all 100 simulations; only one run did not achieve the stipulated convergence residual levels.

As examples, a couple of the ROM Builder plot results are shown in Figures 9 and 10, corresponding to the temperature at Port fwecon1-out and the temperature at Port hpsh1-out, respectively. It should be noted that only the first 40 out of the possible 100 simulations are plotted in each Figure.

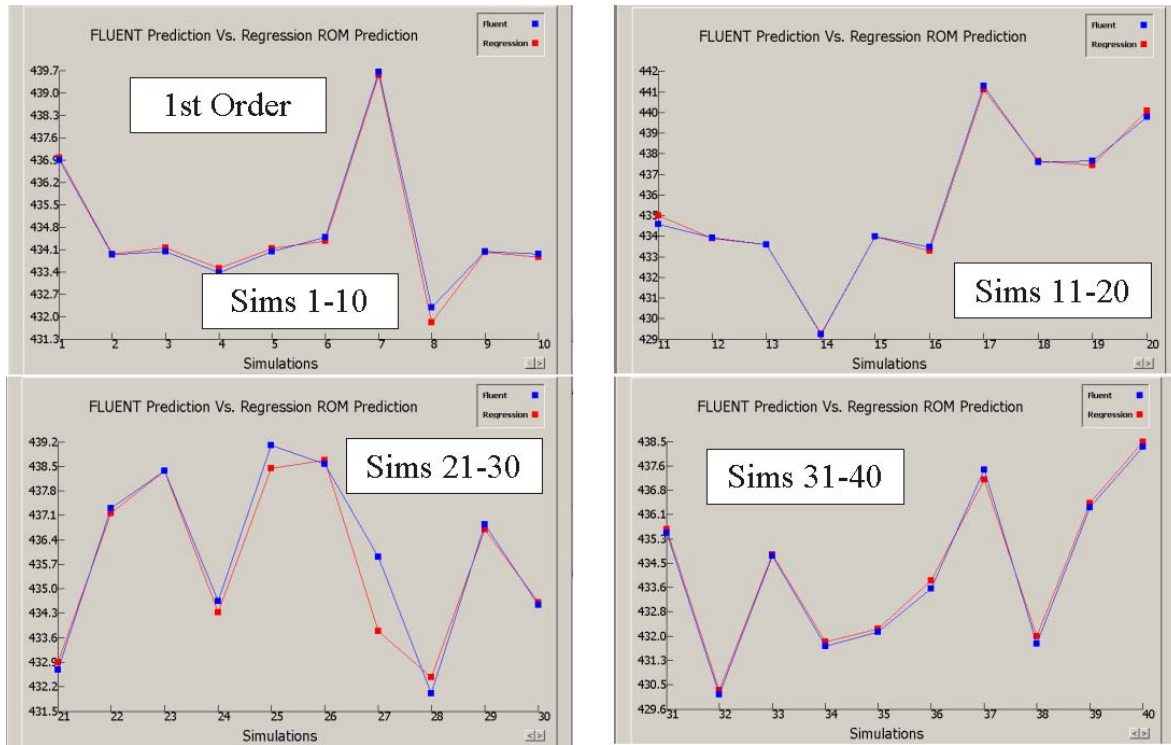


Figure 9: FLUENT® Versus ROM Predictions for the Temperature at Port Fwecon1-out.

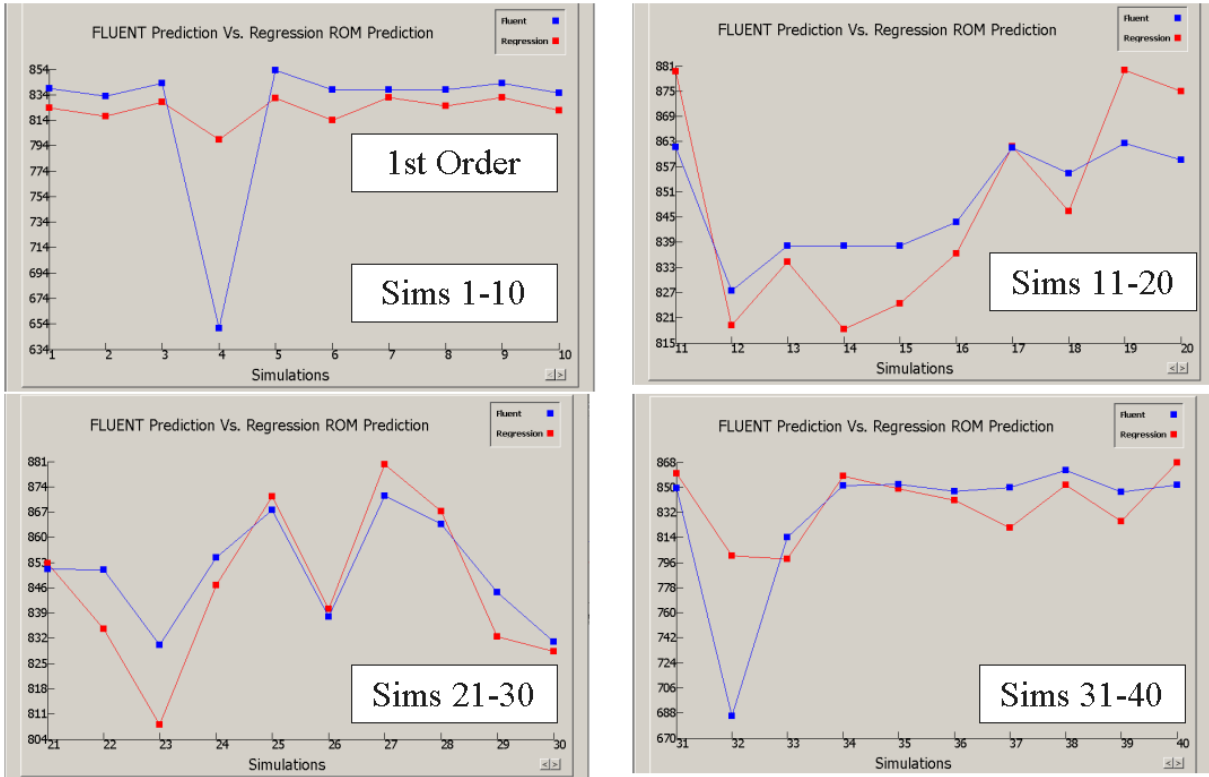


Figure 10: FLUENT® Versus ROM Predictions for the Temperature at Port Hpsh1-out.

The “goodness of fit” does not appear to be as good for the higher hpsh1-out outlet port temperature, as it is for the fwecon1-out outlet port temp. The HP superheater banks are interleaved by a reheater bank, and thus might be more sensitive to upstream (coolant side) conditions impacting the banks that feed it, than the constant-initial-temperature feedwater stream feeding the economizer bank.

Plots of the ROM predictions for 1st-, 2nd-, and 3rd-order regressions of the temperature at Port hpsh1-out are shown in Figure 11 for Simulations 1 through 10, and in Figure 12 for Simulations 21 through 30. In Figure 11, it is clear that the “goodness of fit” for Simulation No. 4 steadily improves as the order of the regression increases. This may be said for the other simulations in the figure as well. However, in Figure 12, it appears that the fit for Simulations 23 and 26 actually appears to be getting worse as the order of the regression increases. Surely the “goodness of fit” is getting better “overall” as the order of the regression increases, although the fit for individual simulation points may be getting worse. It is difficult for the casual industrial user to judge which regression order he should use merely by looking at these graphical plots. It is also difficult to understand quantitatively how much better a given order of regression is than another. For example, a 2nd-order regression ROM may be better than a 1st-order regression ROM, but by how much is it better? The casual user might ask: When does one reach a point of diminishing returns in the performance of successive orders of regression ROMs? Such questions are difficult to answer without reliable statistical measures of performance. It is therefore imperative that

quantitative statistical indicators be provided in the ROM Builder so that a user may be able to discriminate between the various ROMs and to select the “best” ROM for the application.

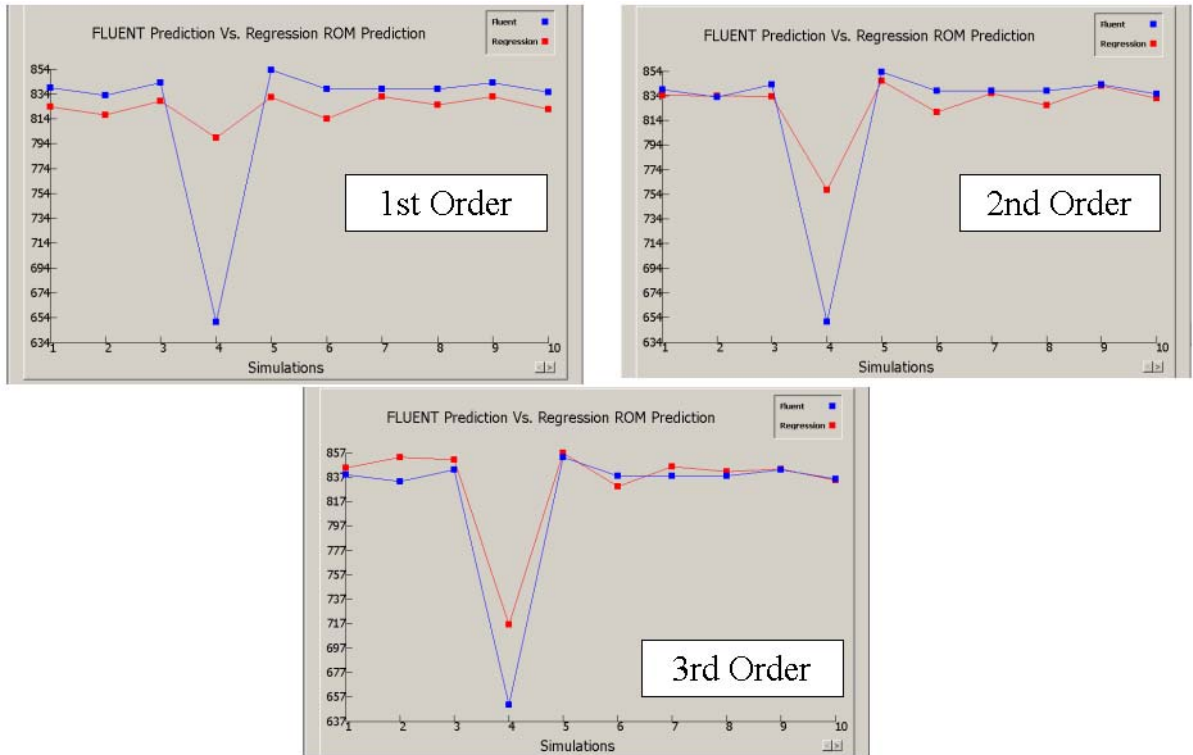


Figure 11: FLUENT® vs 1st-3rd Order ROM for the Temperature at Hpsh1-out (Sims 1-10).

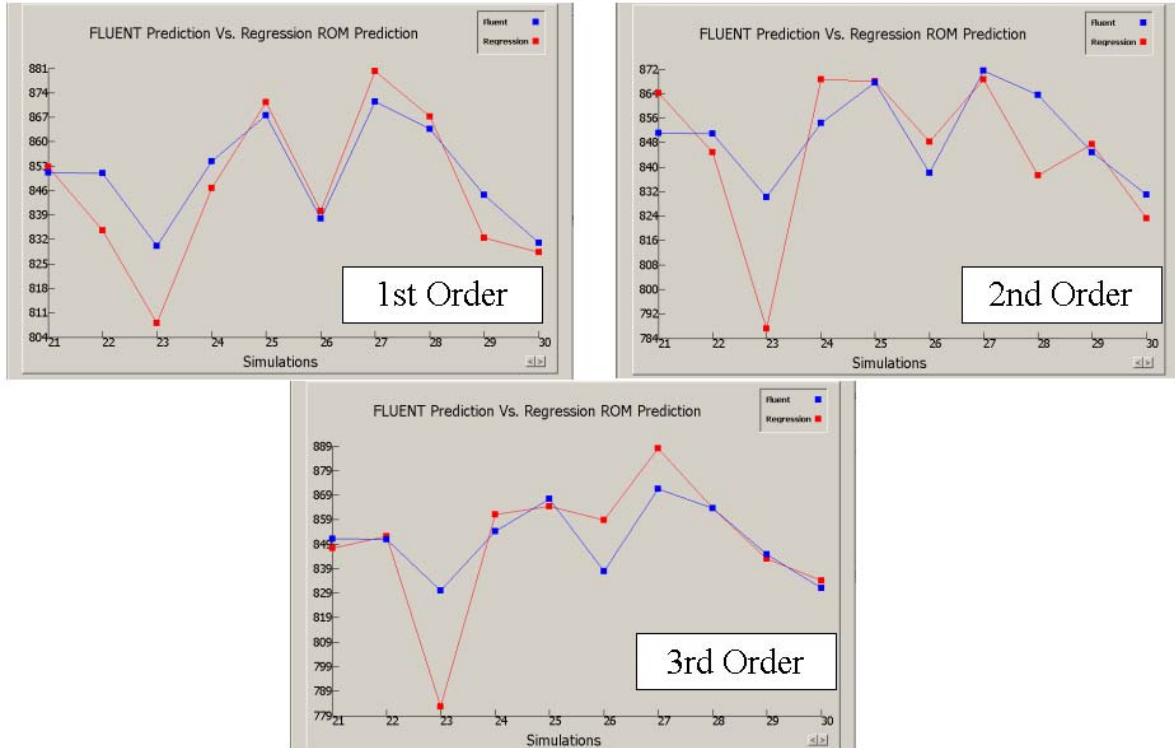


Figure 12: FLUENT® vs 1st-3rd Order ROM for the Temperature at Hpsh1-out (Sims 21-30).

5.2.2 Demo Case 2: Integrated Co-Simulation With ROM

The integrated co-simulation was run on a different PC than that of the ROM Builder. The software versions used in the co-simulation included the following:

- ANSYS EKM® V1.2
- Aspen Plus® V7.1 (with Cumulative Patch 1)
- ANSYS APECS V1.7.0.018
- COFluent 12.0.16 Wrapper V1.7.0.004

All of the co-simulation runs were performed with Aspen Plus® as the executive process modelling software. The Aspen Plus® flowsheet was run in the PC Windows environment and the ROM was run in the same environment.

The original flowsheet was created and run with Aspen Plus® Version 2004.1. However, it was found, after some experimentation, that the newly instantiated ROMs did not work in either V2004.1 or in V2006.5; the ROMs only worked properly in V7.1. Therefore, all of the co-simulation work was migrated to Aspen Plus® V7.1.

However, it was found that V7.1 presently exhibits the following CAPE-OPEN related problems:

- Aspen Plus® V7.1 does not persist the stream connections to the CAPE-OPEN block until the flowsheet has converged; only then can the user save out a working .apw file. This is a problematic in the present task, because in order to start each perturbation run from exactly the same initial conditions, the instantiation procedure must be repeated every run.
- Aspen Plus® V7.1 is flashing the inlet streams (KODE = 2 (i.e., T-P flash)) in order to facilitate making available additional properties in the inlet streams, so that the CAPE-OPEN unit operation could make use of said properties if necessary. For the present task, this inlet stream flashing has a relatively benign effect because the inlet gas stream is (typically) not in a 2-phase regime (but it could potentially be a serious problem if the inlet stream is in the 2-phase regime).

The instantiation procedure, in general terms, is given below:

- (1) Instantiate the APECS block and rename it to, e.g., “B1.”
- (2) Assign the block section = GAS (rather than STEAM)
- (3) Make all of the stream connections in accordance with Table 1.
- (4) Update the appropriate convergence blocks and calculator blocks to accommodate the ROM block:

Convergence → Sequence → S-LOOP: Position Unit Operation B1 between C-
PREUSR and C-PSTUSR:

```
SEQUENCE S-LOOP C-PREUSR B1 C-PSTUSR
```

```
CALCULATOR C-PSTUSR
```

```
DEFINE TOEC1 BLOCK-VAR BLOCK=B1 VARIABLE=FWECON1-TEMP-OUT  
DEFINE POEC1 BLOCK-VAR BLOCK=B1 VARIABLE=FWECON1-PRES-OUT  
DEFINE FOEC1 BLOCK-VAR BLOCK=B1 VARIABLE=FWECON1-MDOT-OUT  
DEFINE QOEC1 BLOCK-VAR BLOCK=B1 VARIABLE=FWECON1-QUAL-OUT  
DEFINE TOHPEV BLOCK-VAR BLOCK=B1 VARIABLE=HPEVAP-TEMP-OUT  
DEFINE FOHPEV BLOCK-VAR BLOCK=B1 VARIABLE=HPEVAP-PRES-OUT  
DEFINE FOHPEV BLOCK-VAR BLOCK=B1 VARIABLE=HPEVAP-MDOT-OUT  
DEFINE QOHPEV BLOCK-VAR BLOCK=B1 VARIABLE=HPEVAP-QUAL-OUT  
DEFINE TOSH1 BLOCK-VAR BLOCK=B1 VARIABLE=HPSH1-TEMP-OUT  
DEFINE POSH1 BLOCK-VAR BLOCK=B1 VARIABLE=HPSH1-PRES-OUT  
DEFINE FOSH1 BLOCK-VAR BLOCK=B1 VARIABLE=HPSH1-MDOT-OUT  
DEFINE QOSH1 BLOCK-VAR BLOCK=B1 VARIABLE=HPSH1-QUAL-OUT  
DEFINE TOLPEV BLOCK-VAR BLOCK=B1 VARIABLE=LPEVAP-TEMP-OUT  
DEFINE POLPEV BLOCK-VAR BLOCK=B1 VARIABLE=LPEVAP-PRES-OUT  
DEFINE FOLPEV BLOCK-VAR BLOCK=B1 VARIABLE=LPEVAP-MDOT-OUT  
DEFINE QOLPEV BLOCK-VAR BLOCK=B1 VARIABLE=LPEVAP-QUAL-OUT  
DEFINE TOLPSH BLOCK-VAR BLOCK=B1 VARIABLE=LPSH-TEMP-OUT  
DEFINE POLPSH BLOCK-VAR BLOCK=B1 VARIABLE=LPSH-PRES-OUT  
DEFINE FOLPSH BLOCK-VAR BLOCK=B1 VARIABLE=LPSH-MDOT-OUT  
DEFINE QOLPSH BLOCK-VAR BLOCK=B1 VARIABLE=LPSH-QUAL-OUT  
DEFINE TORH BLOCK-VAR BLOCK=B1 VARIABLE=RH-TEMP-OUT  
DEFINE PORH BLOCK-VAR BLOCK=B1 VARIABLE=RH-PRES-OUT  
DEFINE FORH BLOCK-VAR BLOCK=B1 VARIABLE=RH-MDOT-OUT  
DEFINE QORH BLOCK-VAR BLOCK=B1 VARIABLE=RH-QUAL-OUT
```

- (5) Complete the species mapping, as indicated by the last two columns in Table 2.

The ROM, with a regression order of 1, was first uploaded into EKM®. Subsequently, it was instantiated on the flowsheet and run with different initial economizer flow rates to see if the ROM performed in a robust manner over the desired range from -40% to +40% of the baseline values. The results of the regression ROM, built using the Design-of-Experiments option, are compared in the following table with the ANN ROM, constructed from the “on-the-fly” collection of simulations.

Table 7: Co-Simulation ROM Robustness Testing for Demonstration Case 2

Initial Perturbation to Economizer Flow Rate (lbm/hr)	Number of Executions Required for Convergence in Co-Simulations with Aspen Plus		
	FLUENT®	ANN ROM from "On the Fly" Simulations	Regression ROM from ROM Builder Simulations
40,000	---	---	25 [Note]
30,000	---	---	---
20,000	21	73	56 [Note]
10,000	19	24	---
0	12	16	10
-10,000	9	9	---
-20,000	15	16	10
-30,000	20	27	---
-40,000	26	> 99	18
Total:	122	---	---
Note:	2 physical solutions possible with once-through separator; additional control added to prevent the “undesirable” solution.		

At the +20,000 lb_m/hr (+2.52 kg/s) to the +40,000 lb_m/hr (+5.04 kg/s) perturbation condition, the “once-through” separator HRSG has, in principle, two physical solutions. To prevent the occurrence of the “undesirable” solution in actual HRSG systems, a control system is implemented that precludes the stream temperature entering the separator from approaching the dew-point temperature. Computationally, the target expression in the design spec (DS-TSHOU) was modified to quickly change the magnitude of the target expression when saturation conditions were reached, but this represented a discontinuity, and Aspen Plus® does not handle discontinuities well. Consequently, the design spec struggled and the results bounced around a little before the stream temperature was again raised above the saturation conditions and the cycle could again move towards the “desirable” solution and convergence. This convergence difficulty is evident from the regression ROM results at the +20,000 lb_m/hr (+2.52 kg/s) and the +40,000 lb_m/hr (+5.04 kg/s) perturbation condition, and masks any apparent benefit garnered by the regression ROM over the ANN ROM constructed from the “on-the-fly” training set.

However, at the -40,000 lb_m/hr (-5.04 kg/s) perturbation condition, the regression ROM clearly performs better than the ANN ROM, and requires only 18 ROM executions to converge. It is true that the range between the maximum and minimum for each independent inlet variable was slightly larger in the ROM Builder than in the “on-the-fly” training set. However, it is also accepted that the

Latin Hypercube sampling procedure, currently in the ROM Builder, offers a more mathematically sound approach to sampling than the on-the-fly approach, by varying the independent inlet variables in a more controlled and uniform manner, such that the resulting simulations more closely reflect partial derivatives in which one or more inlet variables are varied while other inlet variables are left constant. The more sound and fundamentally based sampling procedure utilized in the ROM Builder gives any ROM constructed from the associated pre-computed simulations a distinct advantage over the “on-the-fly” methodology and any associated ROMs built from it.

Each of the above cases converged to approximately the same solution (within the convergence tolerances). The final converged co-simulation results, achieved with the instantiated ROM, compared to target values taken from the original (Phase I) flowsheet with the instantiated FLUENT® case, are easily within 0.3% error or less.

5.3 Demo Case 3: Case Setup and Preparation

As explained earlier, candidate models for the FutureGen Equipment Model No. 1 included a gasifier, steam turbine (low to high pressure), air preheater, syngas cooler (convective or radiative), etc. The NETL was consulted in the selection of the Equipment Model No. 1 from the list of candidate equipment items. The FutureGen Equipment Model No. 1 was selected by the DOE (in the person of Steve Zitney) in 2007 to be a heat recovery stream generator (HRSG), specifically the HRSG associated with Demonstration Case 3 in the Phase I program. Previous co-simulations of Demonstration Case 3 performed with both FLUENT® and an in-house, proprietary, industrial code, have been reported on previously (see e.g., Ref. 4, 6, and 7).

5.3.1 Demo Case 3: Introduction

From a historical perspective, a modified Aspen Plus® FutureGen IGCC cycle named “ZZPSA3_FixIn_norcy.bkp” file was prepared by Stephen Zitney at NETL (based on the original cycle discussed in Ref. 10) and sent to ALSTOM on May 27, 2004. The cycle consists of a 250 MWe FutureGen IGCC with CO₂ capture, a gas turbine burning hydrogen enriched syngas, a pressure-swing absorption (PSA) section for hydrogen stream production, an air separation unit (ASU), a heat recovery steam generator (HRSG), and acid-gas cleanup. The HRSG is a three-pressure system (i.e., low pressure (LP), intermediate pressure (IP), and high pressure (HP)), with each line splitting off from the common output of a deaerator and being pressurized with dedicated pumps. Each line consists of various economizer tube banks, one or more evaporative banks with a drum, and a superheat section. The HRSG also has two reheat tube banks. Various fractions of the economizer flows are split off and recycled to other sections of the flowsheet, including the shift reactors, where heat is absorbed and then returned to the HRSG. The HRSG module, shown in Figure 13 with its 18 tube banks, is the focus of Demonstration Case 3.

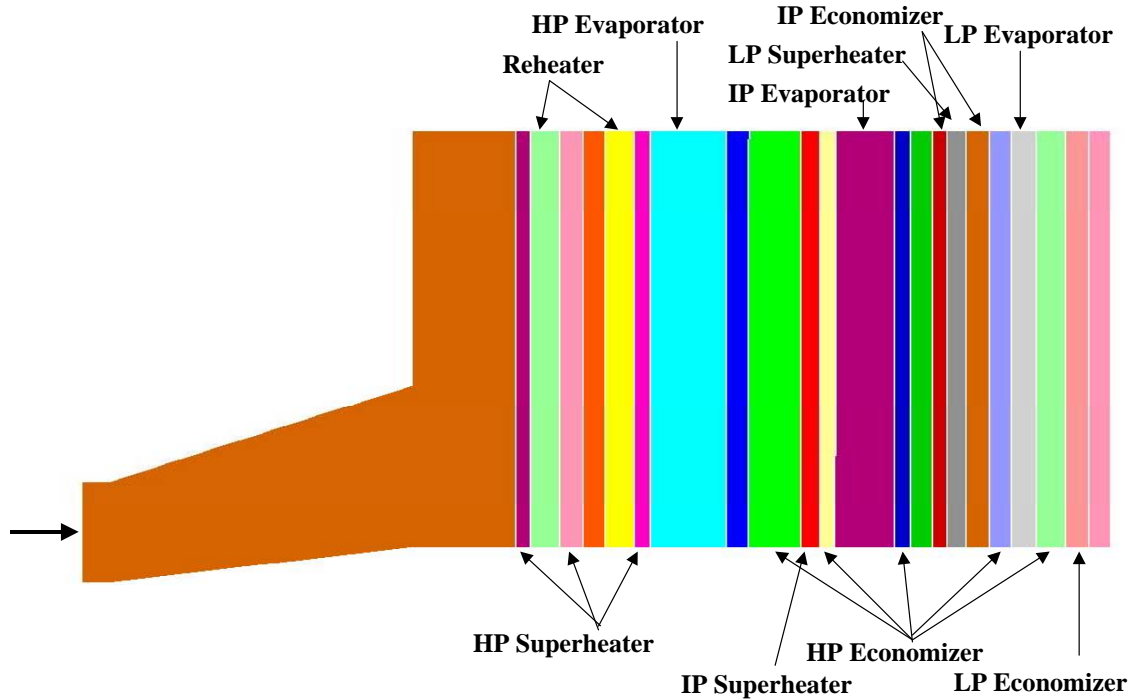


Figure 13: HRSG Tube Banks and Geometry for Demonstration Case 3 (Expanded View).

In an HRSG with drums, the usual control strategy, and hence the present solution strategy, is focused on balancing the circulation around the drums. Each of the inlet streams to the various HX blocks are duplicated and the duplicate stream is connected to the corresponding inlet of the tube bank in the FLUENT® model. CO variables for the outlet quantities from each tube bank in the FLUENT® model are passed back to Aspen Plus® and associated with their corresponding “Fortran tear” variables in a calculator block. The evaporative section consists of a recycle stream and the evaporator HX block. Presuming forced circulation for the recycle stream, the control strategy typically involves setting the circulation flow rate through the evaporator (based on certain criteria) and then modifying the economizer flow rate until the recycle rates are balanced. Accordingly, two design specs (using the secant method) were constructed as external convergence loops -- one which manipulated the total deaerator flow until the circulation was balanced on the IP line, and the other which manipulated the split to the HP pump/line until the circulation for the HP line was balanced. The LP line had a relatively low flow rate and the balancing of its circulation rate was ignored in the present calculation (although a design spec could also be developed for the LP line).

5.3.2 Demo Case 3: FLUENT® 3-D Grid and Case Setup

During the Phase I program, a conceptualized HRSG was engineered using ALSTOM’s design code to match the constraints and boundary conditions of the HRSG tube banks in the FutureGen cycle, as represented on the Aspen Plus® flowsheet. A corresponding FLUENT® model of the HRSG was constructed with approximately 40,000 cells. The cell-count was kept low in order to keep computational turnaround times low for debugging purposes. The exterior grid is shown in Figure 14.

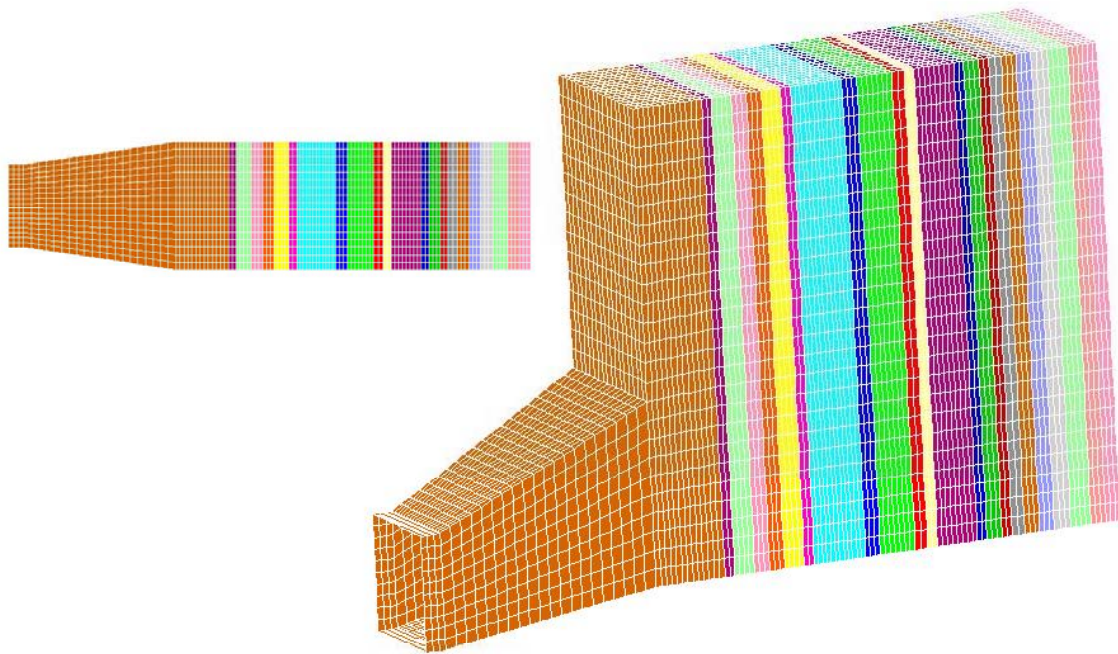


Figure 14: Exterior Grid View of the HRSG Geometry for Demonstration Case 3.

Since the FLUENT® code did not contain the HRSG design standards and the semi-empirical heat transfer relationships for finned-tubes contained within the ALSTOM design code, the FLUENT® case had to be calibrated to the results of the design code by adjusting a simple surface effectiveness factor.

Some of the other CFD case characteristics and selected submodel options are itemized below:

- FLUENT® V6.3.26 (on a PC) was used for all of the CFD investigatory computations, as well as for the ROM preparation runs.
- The solution algorithm is based upon the SIMPLE pressure-correction algorithm with standard discretization for pressure and first-order upwind differencing for all of the other transport equations.
- The turbulence model is the standard k- ϵ turbulence model, with standard wall functions.
- The external walls are adiabatic.
- Species transport equations were activated for 6 species (O_2 , CO_2 , H_2O , SO_2 , Ar, N_2), without reaction.
- The radiation submodel was turned off.
- The tube banks are treated as porous media. The pseudo 1-D FLUENT® heat exchanger tube bank model was used to compute the energy exchange and temperature differences between the gas-side and steam-side flows.
- The inlet boundary condition was prescribed through a mass-flow rate boundary specification. The outlet flow boundary condition was set as a pressure outlet.

5.3.3 Demo Case 3: CO Parameters, Ports, and Connecting Streams

Since the ROM is built directly from multiple runs of the 3-D case, the 3-D CFD case must also be prepared with all of the necessary CO parameters and port connectivity information required for ROM instantiation.

The FLUENT® HRSG model was constructed with 18 tube banks. Since some of the tube banks were considered to be in series with each other without intervening junctions (i.e., without flows being split off or added to the streams), some of them were conceptually grouped together to form a composite heat exchanger on the flowsheet. Consequently, the 18 tube banks in the HRSG were conceptually collapsed to only 12 HX groups, and 12 corresponding HX icons were placed on the flowsheet. For each composite heat exchanger, 4 CO variables were defined for the tube bank outlet values (mass flow rate, temperature, pressure, and vapor fraction), principally for monitoring purposes. Of these 48 CO variables, 24 were designated as “Fortran tear” variables (half of which were pressure quantities which didn’t change).

The conceptual methodology used for flowsheet preparation was introduced previously in reference to Figure 3. This arrangement, enlarged to show a few of the duplicator blocks and stream connections associated with the HRSG portion of the overall process model, is shown in Figure 15.

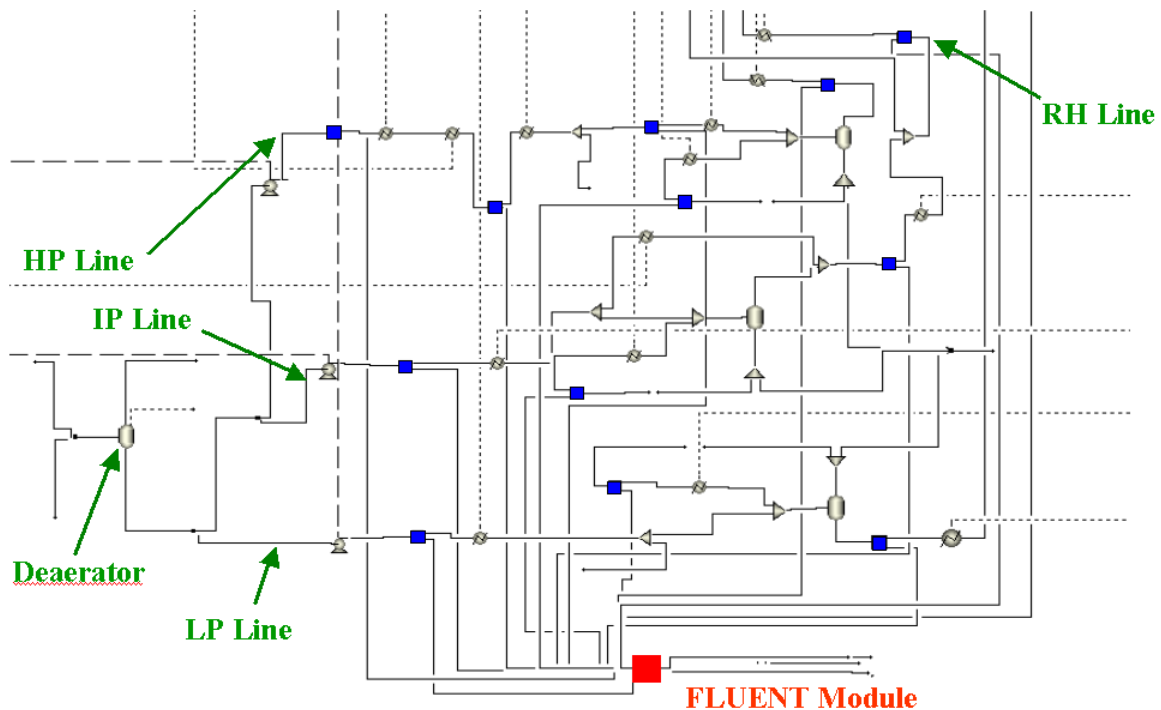


Figure 15: Flowsheet Showing Instantiated Demonstration Case 3 FLUENT®/ROM Block.

An enlarged view of the ROM block and the connecting streams is provided in Figure 16.

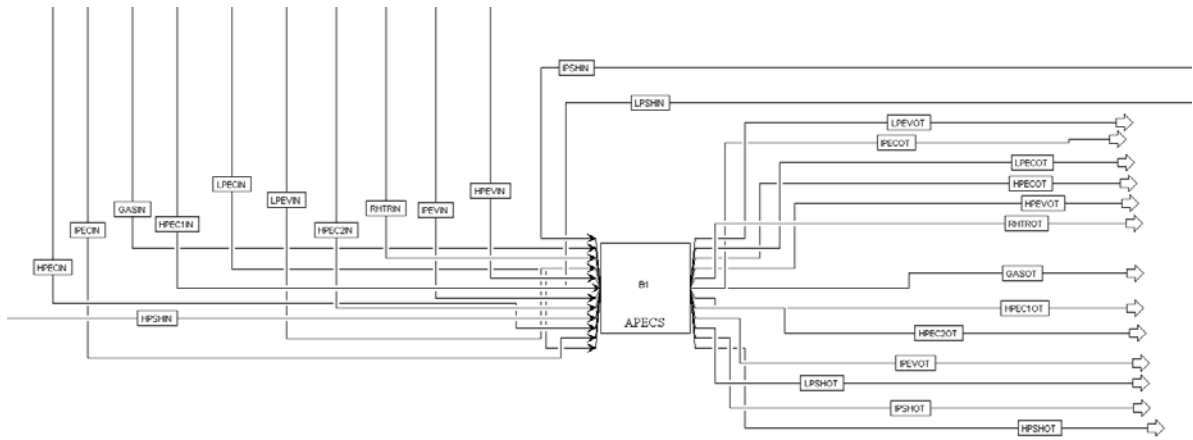


Figure 16: Flowsheet View (Enlarged) Showing Instantiated Demonstration Case 3 Block.

The various ports and stream connections for the FLUENT®-derived ROM block are itemized in the table below. The block only has 2 materials stream connections, representing the gas-side (inlet and outlet). The tube banks are associated with physical model (heat exchanger) ports, with each tube bank (or groups of tube banks in series) corresponding to a distinct domain (Domains 2 through 13).

Table 8: FLUENT® Block Ports and Connecting Streams for Demonstration Case 3

Domain Inlet and Outlet ID	Computational Domain Inlet Port	Corresponding Aspen Stream Name	Computational Domain Outlet Port	Corresponding Aspen Stream Name
1	inlet	GASIN	outlet	GASOT
Domain Inlet and Outlet ID	Physical Model Inlet Port	Corresponding Aspen Stream Name	Physical Model Outlet Port	Corresponding Aspen Stream Name
2	lpecon_group	LPECIN	lpecon	LPECOT
3	hpecon6_group	HPECIN	hpecon4	HPECOT
4	lpevap_group	LPEVIN	lpevap	LPEVOT
5	ipecon2_group	IPECIN	ipecon1	IPECOT
6	lpsh_group	LPSHIN	lpsh	LPSHOT
7	ipevap_group	IPEVIN	ipevap	IPEVOT
8	hpecon2_group	HPEC2IN	hpecon2	HPEC2OT
9	ipsh1_group	IPSHIN	ipsh1	IPSHOT
10	hpecon1_group	HPEC1IN	hpecon1	HPEC1OT
11	hpevap_group	HPEVIN	hpevap	HPEVOT
12	hpsh3_group	HPSHIN	hpsh1	HPSHOT
13	rhtr2_group	RHTRIN	rhtr1	RHTRTROT

The species listed in both the FLUENT® case and the Aspen Plus® flowsheet, as well as their respective mappings to each other, are provided in the following table:

Table 9: Species Listing and Mapping for Demonstration Case 3

	Species in the Materials Listing in FLUENT®	FLUENT® Species Exposed to the PME	Flowsheet Species
(1)	o2	o2	O2
(2)	co2	co2	CO2
(3)	h2o	h2o	H2O
(4)	so2	so2	SO2
(5)	ar	ar	AR
(6)	n2	n2	N2
(7)	---	Zero Flow Rate Species No. 1	H2
(8)	---	Zero Flow Rate Species No. 2	CO
(9)	---	Zero Flow Rate Species No. 3	CH4
(10)	---	Zero Flow Rate Species No. 4	H2S
(11)	---	Zero Flow Rate Species No. 5	CARBON
(12)	---	Zero Flow Rate Species No. 6	CL2
(13)	---	Zero Flow Rate Species No. 7	HCL
(14)	---	Zero Flow Rate Species No. 8	NH3
(15)	---	Zero Flow Rate Species No. 9	COS
(16)	---	Zero Flow Rate Species No. 10	H2SO4
(17)	---	Zero Flow Rate Species No. 11	SULFUR

The CAPE-OPEN parameters set through the APECS Wizard are provided in the following table:

Table 10: CAPE-OPEN Parameters for Demonstration Case 3

Parameters	Type	Description	Access Mode	Lower Bound	Upper Bound	Units	Default Value	Default Value in SI Units
lpecon-temp-out	real		read-write	77	2,060	F	284.00	413.15
lpecon-pres-out	real		read-write	58	14,504	psi	326.70	2,252,517.20
lpecon-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	279,411.73	35.21
lpecon-qual-out	real		read-write	0	1	---	0.00	0.00
lpevap-temp-out	real		read-write	77	2,060	F	309.11	427.10
lpevap-pres-out	real		read-write	58	14,504	psi	76.70	528,827.88
lpevap-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	305,000.00	38.43
lpevap-qual-out	real		read-write	0	1	---	0.05	0.05
lpsh-temp-out	real		read-write	77	2,060	F	426.00	492.04
lpsh-pres-out	real		read-write	58	14,504	psi	74.70	515,038.37
lpsh-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	14,823.25	1.87
lpsh-qual-out	real		read-write	0	1	---	1.00	1.00
ipecon1-temp-out	real		read-write	77	2,060	F	427.00	492.59
ipecon1-pres-out	real		read-write	58	14,504	psi	1,235.70	8,519,851.57
ipecon1-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	205,952.62	25.95
ipecon1-qual-out	real		read-write	0	1	---	0.00	0.00
hpecon4-temp-out	real		read-write	77	2,060	F	370.00	460.93
hpecon4-pres-out	real		read-write	58	14,504	psi	2,299.70	15,855,873.32
hpecon4-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	773,471.18	97.46
hpecon4-qual-out	real		read-write	0	1	---	0.00	0.00
ipevap-temp-out	real		read-write	77	2,060	F	453.08	507.08
ipevap-pres-out	real		read-write	58	14,504	psi	435.70	3,004,045.75

NETL (DOE)-Fluent Inc Cooperative Research and Development Agreement DE-FC26-05NT42443
ALSTOM Power Inc. Subcontract Agreement No. FY04012ALS

ipevap-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	818,430.00	103.12
ipevap-qual-out	real		read-write	0	1	---	0.10	0.10
hpecon2-temp-out	real		read-write	77	2,060	F	424.00	490.93
hpecon2-pres-out	real		read-write	58	14,504	psi	2,285.70	15,759,346.72
hpecon2-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	773,471.18	97.46
hpecon2-qual-out	real		read-write	0	1	---	0.00	0.00
ipsh1-temp-out	real		read-write	77	2,060	F	526.00	547.59
ipsh1-pres-out	real		read-write	58	14,504	psi	429.70	2,962,677.20
ipsh1-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	204,936.58	25.82
ipsh1-qual-out	real		read-write	0	1	---	1.00	1.00
hpecon1-temp-out	real		read-write	77	2,060	F	612.00	595.37
hpecon1-pres-out	real		read-write	58	14,504	psi	2,271.70	15,662,820.12
hpecon1-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	400,238.84	50.43
hpecon1-qual-out	real		read-write	0	1	---	0.00	0.00
hpevap-temp-out	real		read-write	77	2,060	F	626.22	603.27
hpevap-pres-out	real		read-write	58	14,504	psi	1,867.70	12,877,338.17
hpevap-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	2,405,000.0	303.02
hpevap-qual-out	real		read-write	0	1	---	0.17	0.17
rhtr1-temp-out	real		read-write	77	2,060	F	1,050.00	838.71
rhtr1-pres-out	real		read-write	58	14,504	psi	377.70	2,604,149.83
rhtr1-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	769,334.68	96.93
rhtr1-qual-out	real		read-write	0	1	---	1.00	1.00
hpsh1-temp-out	real		read-write	77	2,060	F	1,050.00	838.71
hpsh1-pres-out	real		read-write	58	14,504	psi	1,808.70	12,470,547.50
hpsh1-mdot-out	real		read-write	7,937	7,936,641	lbm/hr	396,133.34	49.91
hpsh1-qual-out	real		read-write	0	1	---	1.00	1.00
calculate-user-journal-activate	integer	activates the user journal file	read-write	0	1	---	1	1
calculate-user-journal-name	option	name of user journal file	read-only	---	---	---		prep_resid.jou
maximum-iterations	integer	maximum number of FLUENT® iterations	read-write	1	100,000	---		400

The calculate-user-journal-activate parameter is actually a Boolean, but a bug in Aspen Plus® has precluded such Booleans from being manipulated in a calculator block. Therefore, it is more convenient to treat the parameter as an integer switch (0=FALSE, 1=TRUE).

Each HX block is associated with 4 CAPE-OPEN variables, but as mentioned earlier, only 2 of the variables (e.g., T, p) for each HX block are passed from FLUENT® to Aspen Plus, and are manipulated and converged by the Broyden algorithm as “Fortran tears.”

The CO parameter variables associated with heat exchanger tube banks are set in the “void report_data_exchange()” UDF (in file hxc.c) through the lines:

```
RP_Set_Real("lpecon-temp-out", tout[0]*1.8 - 459.67);
RP_Set_Real("lpecon-pres-out", pout[0]/101325.0*14.6959488);
RP_Set_Real("lpecon-mdot-out", mout[0]/0.45359237*3600.0);
RP_Set_Real("lpecon-qual-out", xout[0]);

RP_Set_Real("cape-open/hxc/lpecon-out/flow-rate", mout[0]);
RP_Set_Real("cape-open/hxc/lpecon-out/temperature", tout[0]);
RP_Set_Real("cape-open/hxc/lpecon-out/pressure", pout[0]);
RP_Set_Real("cape-open/hxc/lpecon-out/outlet-quality", xout[0]);
```

```
RP_Set_Real("hpecon4-temp-out", tout[3]*1.8 - 459.67);
RP_Set_Real("hpecon4-pres-out", pout[3]/101325.0*14.6959488);
RP_Set_Real("hpecon4-mdot-out", mout[3]/0.45359237*3600.0);
RP_Set_Real("hpecon4-qual-out", xout[3]);

RP_Set_Real("cape-open/hxc/hpecon4-out/flow-rate", mout[3]);
RP_Set_Real("cape-open/hxc/hpecon4-out/temperature", tout[3]);
RP_Set_Real("cape-open/hxc/hpecon4-out/pressure", pout[3]);
RP_Set_Real("cape-open/hxc/hpecon4-out/outlet-quality", xout[3]);

RP_Set_Real("hpsh1-temp-out", tout[6]*1.8 - 459.67);
RP_Set_Real("hpsh1-pres-out", pout[6]/101325.0*14.6959488);
RP_Set_Real("hpsh1-mdot-out", mout[6]/0.45359237*3600.0);
RP_Set_Real("hpsh1-qual-out", xout[6]);

RP_Set_Real("cape-open/hxc/hpsh1-out/flow-rate", mout[6]);
RP_Set_Real("cape-open/hxc/hpsh1-out/temperature", tout[6]);
RP_Set_Real("cape-open/hxc/hpsh1-out/pressure", pout[6]);
RP_Set_Real("cape-open/hxc/hpsh1-out/outlet-quality", xout[6]);

RP_Set_Real("rhtr1-temp-out", tout[8]*1.8 - 459.67);
RP_Set_Real("rhtr1-pres-out", pout[8]/101325.0*14.6959488);
RP_Set_Real("rhtr1-mdot-out", mout[8]/0.45359237*3600.0);
RP_Set_Real("rhtr1-qual-out", xout[8]);

RP_Set_Real("cape-open/hxc/rhtr1-out/flow-rate", mout[8]);
RP_Set_Real("cape-open/hxc/rhtr1-out/temperature", tout[8]);
RP_Set_Real("cape-open/hxc/rhtr1-out/pressure", pout[8]);
RP_Set_Real("cape-open/hxc/rhtr1-out/outlet-quality", xout[8]);

RP_Set_Real("hpevap-temp-out", tout[9]*1.8 - 459.67);
RP_Set_Real("hpevap-pres-out", pout[9]/101325.0*14.6959488);
RP_Set_Real("hpevap-mdot-out", mout[9]/0.45359237*3600.0);
RP_Set_Real("hpevap-qual-out", xout[9]);

RP_Set_Real("cape-open/hxc/hpevap-out/flow-rate", mout[9]);
RP_Set_Real("cape-open/hxc/hpevap-out/temperature", tout[9]);
RP_Set_Real("cape-open/hxc/hpevap-out/pressure", pout[9]);
RP_Set_Real("cape-open/hxc/hpevap-out/outlet-quality", xout[9]);

RP_Set_Real("hpecon1-temp-out", tout[10]*1.8 - 459.67);
RP_Set_Real("hpecon1-pres-out", pout[10]/101325.0*14.6959488);
RP_Set_Real("hpecon1-mdot-out", mout[10]/0.45359237*3600.0);
RP_Set_Real("hpecon1-qual-out", xout[10]);

RP_Set_Real("cape-open/hxc/hpecon1-out/flow-rate", mout[10]);
RP_Set_Real("cape-open/hxc/hpecon1-out/temperature", tout[10]);
RP_Set_Real("cape-open/hxc/hpecon1-out/pressure", pout[10]);
RP_Set_Real("cape-open/hxc/hpecon1-out/outlet-quality", xout[10]);

RP_Set_Real("ipsh1-temp-out", tout[11]*1.8 - 459.67);
RP_Set_Real("ipsh1-pres-out", pout[11]/101325.0*14.6959488);
RP_Set_Real("ipsh1-mdot-out", mout[11]/0.45359237*3600.0);
RP_Set_Real("ipsh1-qual-out", xout[11]);

RP_Set_Real("cape-open/hxc/ipsh1-out/flow-rate", mout[11]);
RP_Set_Real("cape-open/hxc/ipsh1-out/temperature", tout[11]);
```

```
RP_Set_Real("cape-open/hxc/ipsh1-out/pressure", pout[11]);
RP_Set_Real("cape-open/hxc/ipsh1-out/outlet-quality", xout[11]);

RP_Set_Real("hpecon2-temp-out", tout[12]*1.8 - 459.67);
RP_Set_Real("hpecon2-pres-out", pout[12]/101325.0*14.6959488);
RP_Set_Real("hpecon2-mdot-out", mout[12]/0.45359237*3600.0);
RP_Set_Real("hpecon2-qual-out", xout[12]);

RP_Set_Real("cape-open/hxc/hpecon2-out/flow-rate", mout[12]);
RP_Set_Real("cape-open/hxc/hpecon2-out/temperature", tout[12]);
RP_Set_Real("cape-open/hxc/hpecon2-out/pressure", pout[12]);
RP_Set_Real("cape-open/hxc/hpecon2-out/outlet-quality", xout[12]);

RP_Set_Real("ipevap-temp-out", tout[13]*1.8 - 459.67);
RP_Set_Real("ipevap-pres-out", pout[13]/101325.0*14.6959488);
RP_Set_Real("ipevap-mdot-out", mout[13]/0.45359237*3600.0);
RP_Set_Real("ipevap-qual-out", xout[13]);

RP_Set_Real("cape-open/hxc/ipevap-out/flow-rate", mout[13]);
RP_Set_Real("cape-open/hxc/ipevap-out/temperature", tout[13]);
RP_Set_Real("cape-open/hxc/ipevap-out/pressure", pout[13]);
RP_Set_Real("cape-open/hxc/ipevap-out/outlet-quality", xout[13]);

RP_Set_Real("ipecon1-temp-out", tout[15]*1.8 - 459.67);
RP_Set_Real("ipecon1-pres-out", pout[15]/101325.0*14.6959488);
RP_Set_Real("ipecon1-mdot-out", mout[15]/0.45359237*3600.0);
RP_Set_Real("ipecon1-qual-out", xout[15]);

RP_Set_Real("cape-open/hxc/ipecon1-out/flow-rate", mout[15]);
RP_Set_Real("cape-open/hxc/ipecon1-out/temperature", tout[15]);
RP_Set_Real("cape-open/hxc/ipecon1-out/pressure", pout[15]);
RP_Set_Real("cape-open/hxc/ipecon1-out/outlet-quality", xout[15]);

RP_Set_Real("lpsh-temp-out", tout[16]*1.8 - 459.67);
RP_Set_Real("lpsh-pres-out", pout[16]/101325.0*14.6959488);
RP_Set_Real("lpsh-mdot-out", mout[16]/0.45359237*3600.0);
RP_Set_Real("lpsh-qual-out", xout[16]);

RP_Set_Real("cape-open/hxc/lpsh-out/flow-rate", mout[16]);
RP_Set_Real("cape-open/hxc/lpsh-out/temperature", tout[16]);
RP_Set_Real("cape-open/hxc/lpsh-out/pressure", pout[16]);
RP_Set_Real("cape-open/hxc/lpsh-out/outlet-quality", xout[16]);

RP_Set_Real("lpevap-temp-out", tout[17]*1.8 - 459.67);
RP_Set_Real("lpevap-pres-out", pout[17]/101325.0*14.6959488);
RP_Set_Real("lpevap-mdot-out", mout[17]/0.45359237*3600.0);
RP_Set_Real("lpevap-qual-out", xout[17]);

RP_Set_Real("cape-open/hxc/lpevap-out/flow-rate", mout[17]);
RP_Set_Real("cape-open/hxc/lpevap-out/temperature", tout[17]);
RP_Set_Real("cape-open/hxc/lpevap-out/pressure", pout[17]);
RP_Set_Real("cape-open/hxc/lpevap-out/outlet-quality", xout[17]);
```

5.4 Demo Case 3: Computational ROM Results

5.4.1 Demo Case 3: “On-the-Fly” Simulation Collections and ANN ROM Results

As discussed previously with respect to Demonstration Case 2, a significant effort was initiated with ANSYS (directed by Maxwell Osawe) in 2006 to generating ROMs based on the Artificial Neural Network (ANN) methodology and the “on-the-fly” training sets. This methodology was also applied to Demonstration Case 3, although only for the baseline case. Thus the ANN ROM was not tested as thoroughly as the ANN ROM in Demonstration Case 2 was tested. The results based on the ANN ROM methodology will be compared and contrasted with the more recent regression ROM constructed from the ROM Builder and the Latin Hypercube sampling procedure.

FLUENT® was instantiated on the flowsheet and then the baseline case was run. During the single Aspen Plus / FLUENT® co-simulation, the input and output results from the FLUENT® block were saved “on the fly” in a MySQL database (the EKM® database had not yet been released for use) each time that FLUENT® was executed within the Aspen Plus flowsheet. The collection of input and output results from each FLUENT® block execution constituted the “simulation data set” or the “table of pre-computed results” on which the ANN was trained to construct a ROM. The ANN was used to construct a ROM trained on total of 114 FLUENT® executions.

An ANN ROM was constructed using 10 neurons and the simulated annealing approach. At the baseline condition, the ANN ROM required 85 ROM executions to converge the flowsheet, compared to the 114 FLUENT® executions. The number of convergence loops (described in the following table) required for convergence was also different between the FLUENT® and the ANN ROM runs, and are illustrated pictorially in the subsequent figure:

Table 11: Convergence Loops for Demonstration Case 3

Convergence Loop	Solution Method	Corresponding Design Spec	Variable Manipulated	Target
ST-SPEC4 (outer loop)	Secant	HPIPS	Vary fractional flow split through block SP-PMPS (to HP line)	Until HP circulation is balanced
ST-SPEC1 (middle loop)	Secant	STACKG	Vary flow rate to deaerator	Until IP circulation is balanced
ST-FTEAR (inner loop)	Broyden	[none]	Converge “Fortran tears”	

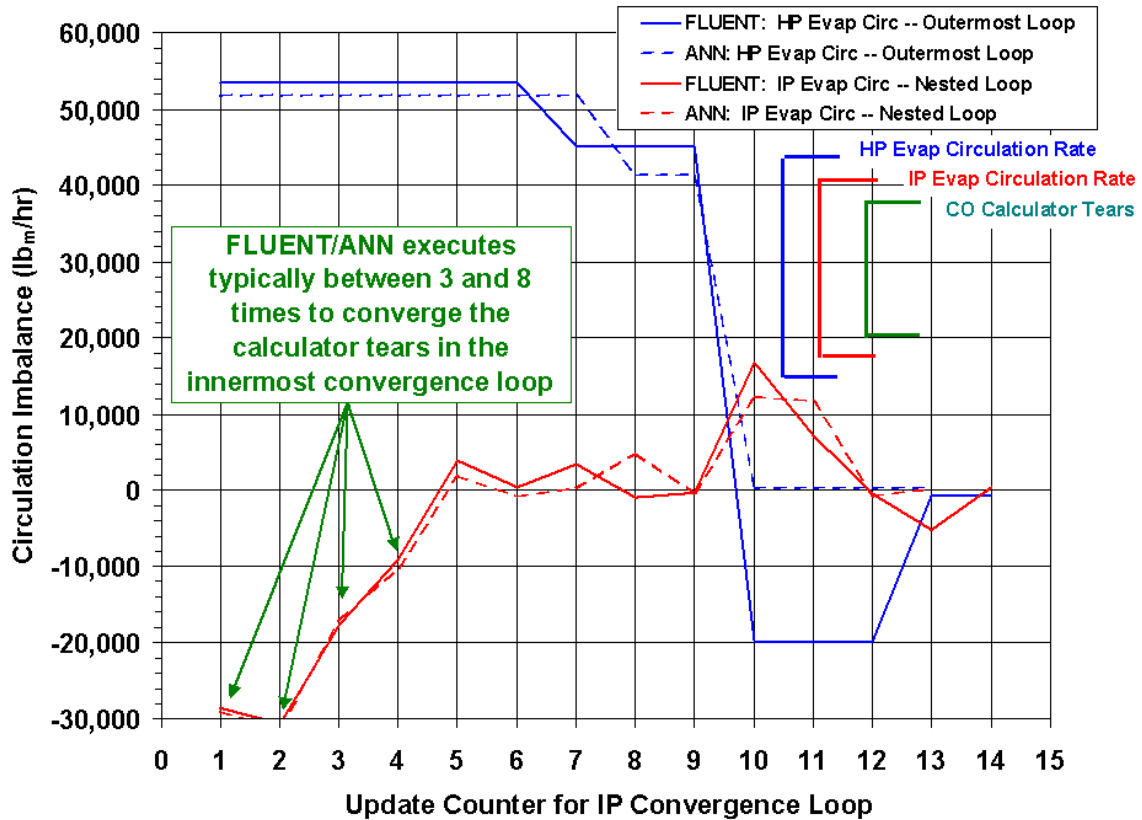


Figure 17: FLUENT® and ANN ROM Convergence Comparison for Demonstration Case 3.

The solution paths for the baseline case are somewhat different, but both FLUENT® and the ANN ROM converged to within 0.5% of each other. The results from the “on-the-fly” collection technique and resultant ANN ROM will be compared with the results from the ROM Builder and the regression ROM in subsequent sections.

5.4.2 Demo Case 3: Present Methodology for ROM Construction

The 3-D CFD case for the HRSG was used in the ROM Builder. It was run on a Windows PC with the following software:

- ANSYS EKM® V1.2
- FLUENT® V6.3.26
- ANSYS APECS V1.7.0.019
- COFluent.6.3.26 Wrapper V1.7.0.015

The FLUENT® case files were prepared and made CAPE-OPEN and APECS compatible by sequentially performing the steps outlined in the APECS User’s Manual (Ref. 9):

- Execute FLUENT® CAPE-OPEN Configuration Wizard
- Execute APECS Model Wizard

- Upload files to the ANSYS Engineering Knowledge Manager (EKM®) database
- Launch the APECS User Interface to build the ROM

The files uploaded to the EKM® database consist of the following:

CpofPH.dat
ETAofPT.dat
HofPT.dat
SATPropsOfPsat.dat
TcofPT.dat
TofPH.dat
XofPH.dat
demo_c3_2640_CO.cas
demo_c3_2640_CO.dat
drum.scm
hxc.c
interpolate.c
interpolate.h
libudf/ntx86/3d/libudf.dll
mixing-constants.scm (← not used)
prep_resid.jou
sg_p1.c

The ROM was prepared on the PC and the libudf.dll was used instead without recompiling the UDF files; consequently it was not necessary to upload hxc.c, interpolate.c, interpolate.h, and sg_p1.c to the EKM® database.

The APECS software allows for a number of journal files, which are helpful if providing increased control to the run sequence. ANSYS has prepared a user journal file activated via the CAPE-OPEN parameters:

- calculate-user-journal-name
- calculate-user-journal-activate

As explained earlier, this user journal file, called “prep_resid.jou,” was used to help run a minimum number of iterations by:

- (d) lowering one or more of the residual tolerances
- (e) running a specified number of iterations
- (f) and then putting the residual tolerances back where they were originally

The APECS calculate command is executed after the journal file commands are completed. If the residuals are less than the stipulated thresholds, then the case is deemed converged; otherwise, it begins to calculate the number of iterations dictated by APECS. Since it is difficult to determine the proper convergence levels for the residuals, a good compromise is to utilize a user-journal file, which forces a user-specified minimum number of iterations:

```
; prep_resid.jou  
;
```



```

;
k  epsilon  o2  co2  h2o  so2  ar  cont  x  y  z  energy
;
/solve/monitors/residual/convergence-criteria 1.e-09 1.e-07 1.e-07 5.e-06 2.e-06
2.e-05 2.e-05 1.e-07 1.e-07 1.e-07 1.e-07 1.e-07
/solve/iterate 200
/solve/monitors/residual/convergence-criteria 3.e-05 1.e-07 1.e-07 5.e-06 2.e-06
2.e-05 2.e-05 1.e-07 1.e-07 1.e-07 1.e-07 1.e-07

```

Under the “Regression ROM” tab, the following options were selected:

- Order of Regression = 1
- Save Every Instance = Yes
- Mass Balance Tolerance = 0.001 (default)

ROM inputs were supplied for all of the ports, as well as the CAPE-OPEN parameters. Since the flowsheet had previously been run for the baseline case for the on-the-fly data collection, the on-the-fly results were used as a guide for the maximum and minimum values for each of the independent variables. The range for the ROM Builder was made slightly larger than that of the on-the-fly data collection. It was very difficult to know how much larger the range could be made for each independent variable. A condensed version of the variable range table for the ROM construction is provided below:

Table 12: ROM Variable Range for Demonstration Case 3

	Units	Results from Previous Runs			Input for ROM Builder			
		Minimum	Maximum	Base Values	Include	Minimum	Maximum	Base Values
rhtr2_group								
Temperature	(K)	586.339	588.71	587.936	Yes	579	606	587.936
Pressure	(Pa)	2688960	2688960	2,688,955.340	No	101325	2.00E+07	2,688,955.340
Phase Fraction	Quality	1	1	1	No	0	1	1
H2O	(kg/s)	95.27916989	101.4667186	95.39735052	Yes	93	103	95.39735052
ipecon2_group								
Temperature	(K)	377.545	377.619	377.593	Yes	370	385	377.593
Pressure	(Pa)	8568110	8568110	8,568,114.874	No	101325	2.00E+07	8,568,114.874
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	23.825107	28.5593582	28.5593582	Yes	21	31	28.5593582
hpsh3_group								
Temperature	(K)	603.228	603.228	603.228	No	100	1000	603.228
Pressure	(Pa)	12877300	12877300	12,877,338.175	No	101325	2.00E+07	12,877,338.175
Phase Fraction	Quality	1	1	1	No	0	1	1
H2O	(kg/s)	48.84571196	50.85298114	49.68000235	Yes	46	53	49.68000235
hpecon2_group								
Temperature	(K)	488.302	496.189	490.367	Yes	481	504	490.367
Pressure	(Pa)	15835200	15835200	15,835,189.049	No	101325	2.00E+07	15,835,189.049
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	90.55626835	99.54358102	98.37078238	Yes	88	102	98.37078238
lpsh_group								
Temperature	(K)	427.107	427.107	427.107	No	100	1000	427.107
Pressure	(Pa)	528828	528828	528,827.884	No	101325	2.00E+07	528,827.884
Phase Fraction	Quality	1	1	1	No	0	1	1
H2O	(kg/s)	1.684675696	2.092391664	1.68836884	Yes	0.4	4	1.68836884
lpecon_group								
Temperature	(K)	376.426	376.434	376.429	Yes	369	384	376.429

NETL (DOE)-Fluent Inc Cooperative Research and Development Agreement DE-FC26-05NT42443
ALSTOM Power Inc. Subcontract Agreement No. FY04012ALS

Pressure	(Pa)	2293890	2293890	2,293,885.748	No	101325	2.00E+07	2,293,885.748
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	33.33432406	36.2108334	36.2108334	Yes	31	39	36.2108334
lpevap_group								
Temperature	(K)	427.107	427.107	427.107	No	100	1000	427.107
Pressure	(Pa)	528828	528828	528,827.884	No	101325	2.00E+07	528,827.884
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	38.42906221	38.42906221	38.42906221	No	0	1000	38.42906221
hpecon1_group								
Temperature	(K)	490.363	498.409	492.458	Yes	483	506	492.458
Pressure	(Pa)	15759300	15759300	15,759,346.719	No	101325	2.00E+07	15,759,346.719
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	43.53101	52.51832	51.34552	Yes	41	55	51.34552053
hpevap_group								
Temperature	(K)	603.227	603.227	603.227	No	100	1000	603.227
Pressure	(Pa)	12877300	12877300	12,877,338.175	No	101325	2.00E+07	12,877,338.175
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	303.0216219	303.0216219	303.0216219	No	0	1000	303.0216219
inlet								
Temperature	(K)	903.522	903.522	903.522	No	100	1000	903.522
Pressure	(Pa)	103835	103835	103,835.045	No	101325	2.00E+07	103,835.045
O2	(kg/s)	61.49343639	61.49343639	61.49343639	No	0	1000	61.49343639
CO2	(kg/s)	9.636582093	9.636582093	9.636582093	No	0	1000	9.636582093
H2O	(kg/s)	49.02238882	49.02238882	49.02238882	No	0	1000	49.02238882
SO2	(kg/s)	0.003325893	0.003325893	0.003325893	No	0	1000	0.003325893
AR	(kg/s)	6.763973112	6.763973112	6.763973112	No	0	1000	6.763973112
N2	(kg/s)	321.0584409	321.0584409	321.0584409	No	0	1000	321.0584409
hpecon6_group								
Temperature	(K)	378.551	378.595	378.56	Yes	371	386	378.56
Pressure	(Pa)	18841300	18841300	18,841,303.224	No	101325	2.00E+07	18,841,303.224
Phase Fraction	Quality	0	0	0	No	0	1	0
H2O	(kg/s)	90.55626835	99.54358102	98.37078238	Yes	88	102	98.37078238
ipsh1_group								
Temperature	(K)	507.411	511.732	510.072	Yes	500	519	510.072
Pressure	(Pa)	3004050	3004050	3,004,045.748	No	101325	2.00E+07	3,004,045.748
Phase Fraction	Quality	1	1	1	No	0	1	1
H2O	(kg/s)	24.48915243	29.69882845	24.55598934	Yes	22	32	24.55598934
ipevap_group								
Temperature	(K)	507.091	507.091	507.091	No	100	1000	507.091
Pressure	(Pa)	3004050	3004050	3,004,045.748	No	101325	2.00E+07	3,004,045.748

The present case has 16 independent inlet variables. A total of 133 pre-computed simulations were specified in the ROM Builder -- 1 simulation with the baseline values, 32 simulations corresponding to maximum and minimum values (for the 16 independent inlet variables varied), and 100 user-specified simulations (using the “Design-of-Experiments” option based on the Latin Hypercube sampling procedure).

Each of the 133 simulations ran an initial 200 iterations (as stipulated by prep_resid.jou), followed by an additional 400 iterations (as stipulated by the “maximum-iterations” parameter in Table 10), presuming that the convergence criteria were not achieved, for a total possible maximum number of iterations of 600 per simulation. The ROM Builder completed the 133 simulations in approximately 65 hours elapsed time on a single processor of a PC. The (default) mass balance was achieved for all 133 simulations; 44 runs did not achieve the stipulated convergence residual levels. All of the simulations were used in constructing the ROM.

As examples, a couple of the ROM Builder plot results are shown in Figures 18 and 19, corresponding to the temperature at Port hpsh1-out and the temperature at Port ipsh1-out, respectively. It should be noted that only the first 40 out of the possible 100 simulations are plotted in each figure.

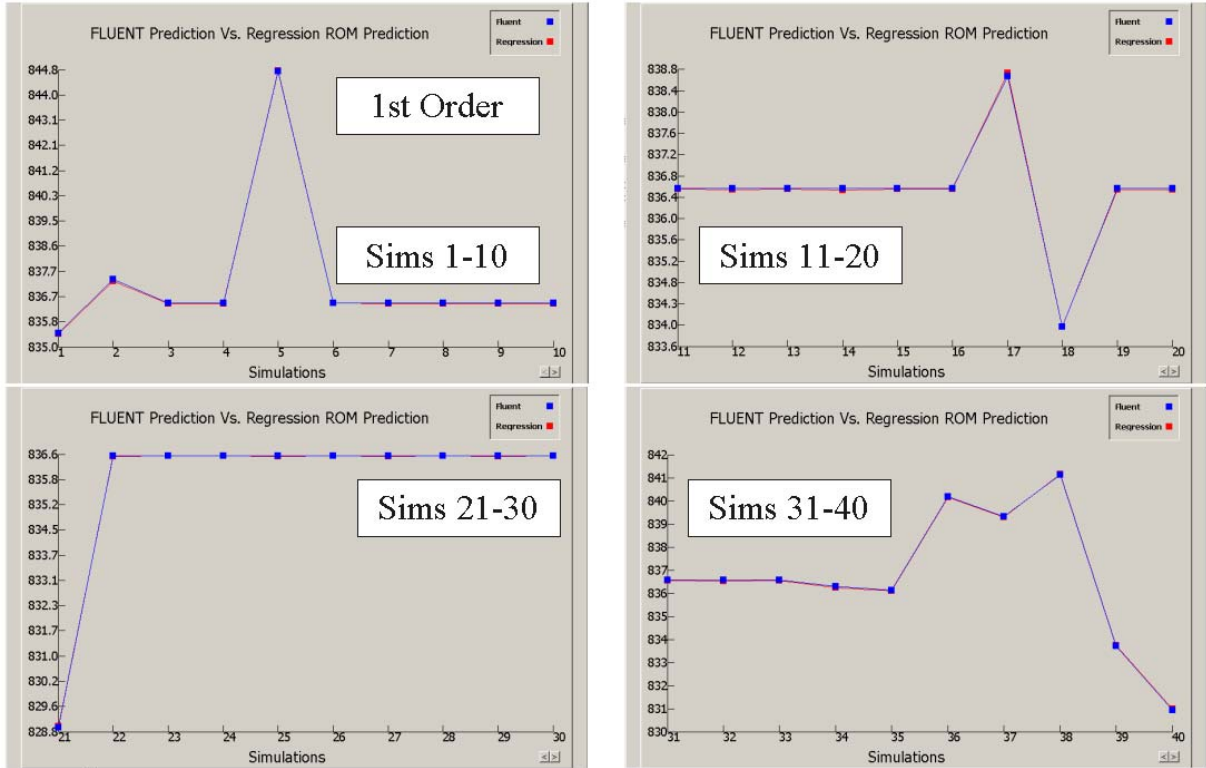


Figure 18: FLUENT® Versus ROM Predictions for the Temperature at Port Hpsh1-out.

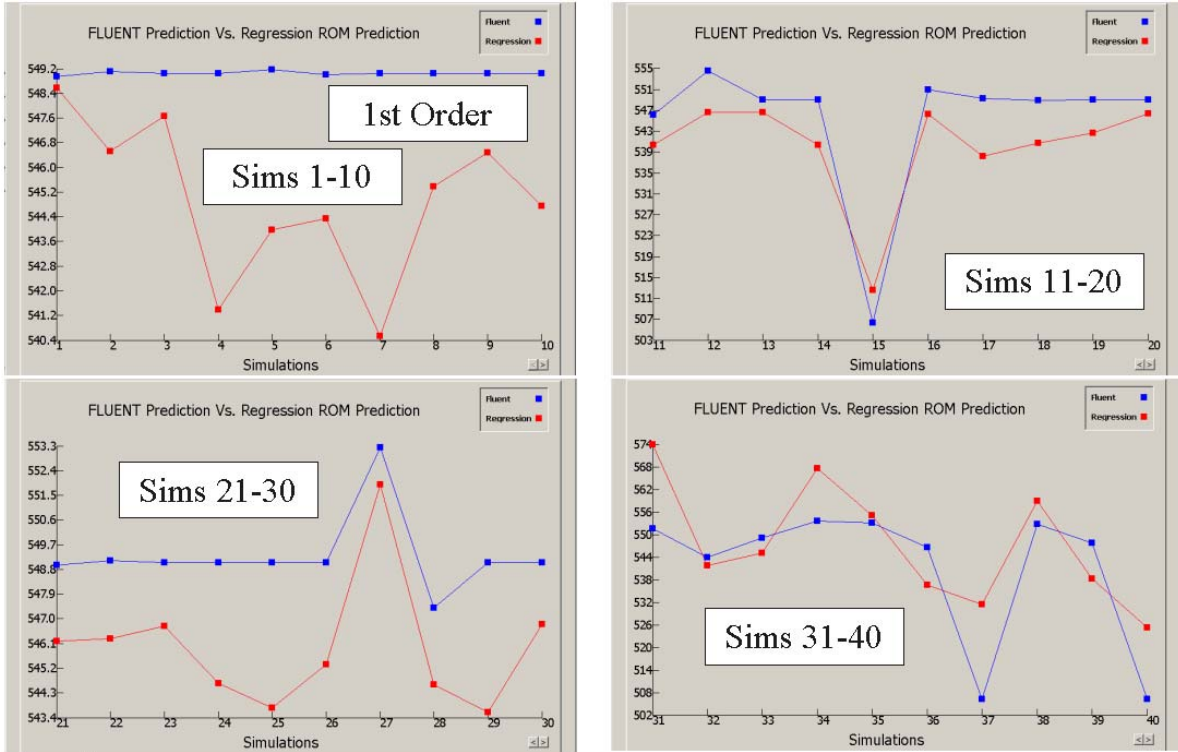


Figure 19: FLUENT® Versus ROM Predictions for the Temperature at Port Ipsh1-out.

The “goodness of fit” does not appear to be as good for the ipsh1-out outlet port temperature, as it is for the hpsh1-out outlet port temp. It is not immediately clear why one port exhibits more sensitivity than another.

Plots of the ROM predictions for 1st-, 2nd-, and 3rd-order regressions of the temperature at Port ipsh1-out are shown in the following figure for Simulations 41 through 50. In the figure, it is clear that the “goodness of fit” for Simulation No. 49 steadily improves as the order of the regression increases. This may be said for most of the other simulations in the figure as well.

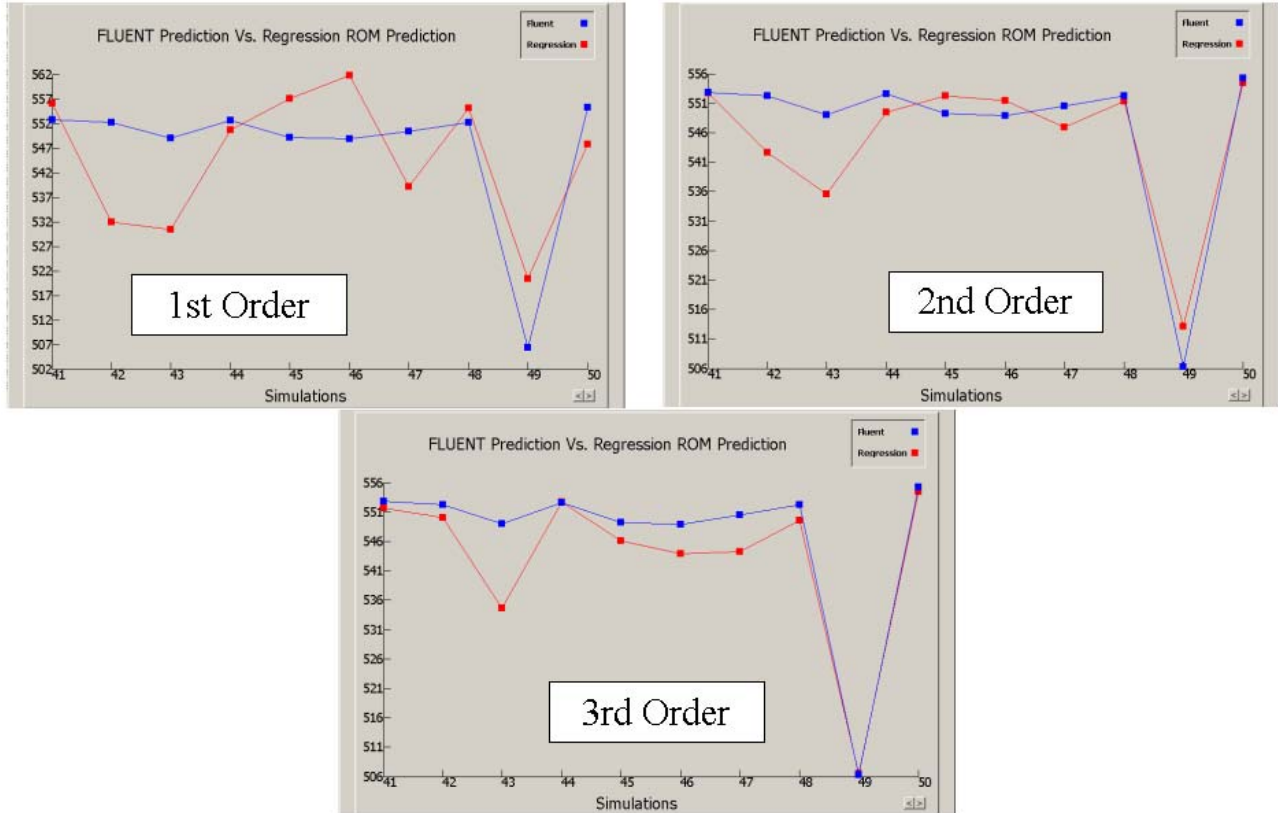


Figure 20: FLUENT® vs 1st-3rd Order ROM for the Temperature at Ipsh1-out (Sims 41-50).

5.4.3 Demo Case 3: Integrated Co-Simulation With ROM

The integrated co-simulation was run on a different PC than that of the ROM Builder. The software versions used in the co-simulation included the following:

- ANSYS EKM® V1.2
- Aspen Plus® V7.1 (with Cumulative Patch 1)
- ANSYS APECS V1.7.0.018
- COFluent 12.0.16 Wrapper V1.7.0.004

All of the co-simulation runs were performed with Aspen Plus® as the executive process modelling software. The Aspen Plus® flowsheet was run in the PC Windows environment and the ROM was run in the same environment.

The original flowsheet was created and run with Aspen Plus® Version 2004.1. However, it was found, after some experimentation, that the newly instantiated ROMs did not work in either V2004.1 or in V2006.5; the ROMs only worked properly in V7.1. Therefore, all of the co-simulation work was migrated to Aspen Plus® V7.1. The CAPE-OPEN related problems discussed earlier for Demonstration Case 2 also apply to Demonstration Case 3.

The instantiation procedure, in general terms, is given below:

1. Instantiate the APECS block and rename it to, e.g., "B1."
2. Assign the block section = GASTURB
3. Make all of the stream connections in accordance with Table 8.
4. Update the appropriate convergence blocks and calculator blocks to accommodate the ROM block:

Convergence → Sequence → STM1: Position Unit Operation B1 between
TMLPECON and C-PSTUSR:

```
SEQUENCE STM1 . . . TMLPECON B1 C-PSTUSR . . .
```

CALCULATOR C-PSTUSR

```
DEFINE TOLEC BLOCK-VAR BLOCK=B1 VARIABLE=LPECON-TEMP-OUT
DEFINE POLEC BLOCK-VAR BLOCK=B1 VARIABLE=LPECON-PRES-OUT
DEFINE FOLEC BLOCK-VAR BLOCK=B1 VARIABLE=LPECON-MDOT-OUT
DEFINE QOLEC BLOCK-VAR BLOCK=B1 VARIABLE=LPECON-QUAL-OUT
DEFINE TOLEV BLOCK-VAR BLOCK=B1 VARIABLE=LPEVAP-TEMP-OUT
DEFINE POLEV BLOCK-VAR BLOCK=B1 VARIABLE=LPEVAP-PRES-OUT
DEFINE FOLEV BLOCK-VAR BLOCK=B1 VARIABLE=LPEVAP-MDOT-OUT
DEFINE QOLEV BLOCK-VAR BLOCK=B1 VARIABLE=LPEVAP-QUAL-OUT
DEFINE TOLSH BLOCK-VAR BLOCK=B1 VARIABLE=LPSH-TEMP-OUT
DEFINE POLSH BLOCK-VAR BLOCK=B1 VARIABLE=LPSH-PRES-OUT
DEFINE FOLSH BLOCK-VAR BLOCK=B1 VARIABLE=LPSH-MDOT-OUT
DEFINE QOLSH BLOCK-VAR BLOCK=B1 VARIABLE=LPSH-QUAL-OUT
DEFINE TOIEC1 BLOCK-VAR BLOCK=B1 VARIABLE=IPECON1-TEMP-OUT
DEFINE POIEC1 BLOCK-VAR BLOCK=B1 VARIABLE=IPECON1-PRES-OUT
DEFINE FOIEC1 BLOCK-VAR BLOCK=B1 VARIABLE=IPECON1-MDOT-OUT
DEFINE QOIEC1 BLOCK-VAR BLOCK=B1 VARIABLE=IPECON1-QUAL-OUT
DEFINE TOHEC4 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON4-TEMP-OUT
DEFINE POHEC4 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON4-PRES-OUT
DEFINE FOHEC4 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON4-MDOT-OUT
DEFINE QOHEC4 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON4-QUAL-OUT
DEFINE TOIEV BLOCK-VAR BLOCK=B1 VARIABLE=IPEVAP-TEMP-OUT
DEFINE POIEV BLOCK-VAR BLOCK=B1 VARIABLE=IPEVAP-PRES-OUT
DEFINE FOIEV BLOCK-VAR BLOCK=B1 VARIABLE=IPEVAP-MDOT-OUT
DEFINE QOIEV BLOCK-VAR BLOCK=B1 VARIABLE=IPEVAP-QUAL-OUT
DEFINE TOHEC2 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON2-TEMP-OUT
DEFINE POHEC2 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON2-PRES-OUT
DEFINE FOHEC2 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON2-MDOT-OUT
DEFINE QOHEC2 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON2-QUAL-OUT
DEFINE TOISH1 BLOCK-VAR BLOCK=B1 VARIABLE=IPSH1-TEMP-OUT
DEFINE POISH1 BLOCK-VAR BLOCK=B1 VARIABLE=IPSH1-PRES-OUT
DEFINE FOISH1 BLOCK-VAR BLOCK=B1 VARIABLE=IPSH1-MDOT-OUT
DEFINE QOISH1 BLOCK-VAR BLOCK=B1 VARIABLE=IPSH1-QUAL-OUT
DEFINE TOHEC1 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON1-TEMP-OUT
DEFINE POHEC1 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON1-PRES-OUT
DEFINE FOHEC1 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON1-MDOT-OUT
DEFINE QOHEC1 BLOCK-VAR BLOCK=B1 VARIABLE=HPECON1-QUAL-OUT
DEFINE TOHEV BLOCK-VAR BLOCK=B1 VARIABLE=HPEVAP-TEMP-OUT
DEFINE POHEV BLOCK-VAR BLOCK=B1 VARIABLE=HPEVAP-PRES-OUT
DEFINE FOHEV BLOCK-VAR BLOCK=B1 VARIABLE=HPEVAP-MDOT-OUT
DEFINE QOHEV BLOCK-VAR BLOCK=B1 VARIABLE=HPEVAP-QUAL-OUT
DEFINE TORHT1 BLOCK-VAR BLOCK=B1 VARIABLE=RHTR1-TEMP-OUT
```

```

DEFINE PORHT1 BLOCK-VAR BLOCK=B1 VARIABLE=RHTR1-PRES-OUT
DEFINE FORHT1 BLOCK-VAR BLOCK=B1 VARIABLE=RHTR1-MDOT-OUT
DEFINE QORHT1 BLOCK-VAR BLOCK=B1 VARIABLE=RHTR1-QUAL-OUT
DEFINE TOHSH1 BLOCK-VAR BLOCK=B1 VARIABLE=HPSH1-TEMP-OUT
DEFINE POHSH1 BLOCK-VAR BLOCK=B1 VARIABLE=HPSH1-PRES-OUT
DEFINE FOHSH1 BLOCK-VAR BLOCK=B1 VARIABLE=HPSH1-MDOT-OUT
DEFINE QOHSH1 BLOCK-VAR BLOCK=B1 VARIABLE=HPSH1-QUAL-OUT

```

5. Complete the species mapping, as indicated by the last two columns in Table 9.

The ROM, with a regression order of 1, was first uploaded into EKM®. Subsequently, it was instantiated on the flowsheet and run with different initial deaerator (material stream TODER) flow rates to see if the ROM performed in a robust manner over the desired range from -30% to +30% of the baseline values. The results of the regression ROM, built using the Design-of-Experiments option, are compared in the following table with the ANN ROM, constructed from the “on-the-fly” collection of simulations. The ST-SPEC1 convergence loop is an inner loop for the STACKG design spec (based on the secant method) that is used to balance the IP circulation. (The ST-SPEC4 convergence loop is the outer loop that balances the HP circulation.) The sequential numbers in the “ST-SPEC1 Loops” column, delimited by commas, indicate the number of iterations that the loop passes through, each time that it is called, prior to convergence.

Table 13: Co-Simulation ROM Robustness Testing for Demonstration Case 3

Percent Change to Stream TODER (%)	TODER Flow Rate (lb _m /hr)	Number of Executions Required for Convergence in Co-Simulations with Aspen Plus®					
		Direct FLUENT® Simulations		ANN ROM from "On the Fly" Simulations		Regression ROM from ROM Builder Simulations	
		FLUENT Executions	ST-SPEC1 Loops	ROM Executions	ST-SPEC1 Loops	ROM Executions	ST-SPEC1 Loops
+30%	1,328,484.4					62	10,2,3,2
+20%	1,226,293.2					58	7,2,4,2
+10%	1,124,102.1					40	3,2,3,2
+0%	1,021,911	114	6,3,3,2	85	7,2,4	52	5,2,3,2
-10%	919,719.9					61	8,2,3,2
-20%	817,528.8					74	12,2,3,2
-30%	715,337.7					126	15,2,3,2

Typically, the ST-SPEC1 loop is “called” 4 times and iterates between 3 to 15 times for the first loop, 2 times the second loop, 3 times the third loop, and 2 times the fourth loop. The primary difference between each of the ROM and FLUENT® simulations is the number of iterations in the first ST-SPEC1 loop and the total number of ROM/FLUENT® executions. For the baseline condition, the regression ROM only requires 52 executions to converge; the FLUENT® block requires more than double that number to converge, again illustrating a difference in the solution pathway surfaces.

As the deviation from the baseline case increases, the number of regression ROM executions goes up and the ST-SPEC1 iterations increases (only during the first “call” loop, i.e., during the first execution of ST-SPEC4). At the –30% perturbation condition, the number of ROM executions appears to be going up rapidly, and is perhaps becoming strained, but still manages to converge. The regression ROM performed well over the perturbation test range (even when some of the variables attained values above or below the specified variable range, e.g., the HPECON stream ~ 68 kg/s when the corresponding ROM Builder range for that variable was only from 88 to 102 kg/s). Thus, some extrapolation was occurring for some of the variables, but still the 1st-order regression ROM did not falter. If the maximum and minimum values for the independent inlet variables in the ROM Builder had been based upon a wider perturbation range, rather than using just the baseline case as a guide, the regression ROM would surely have performed even better (i.e., fewer ROM executions required at the limits).

The above comparison between the ANN ROM and the regression ROM does not clearly show the superiority of the regression ROM (as it does in the instance of the Demonstration Case 2 comparisons), because the ANN ROM was not thoroughly tested over the same range of perturbations. However, while both ROMs work reasonably well at the baseline condition, it is believed that the regression ROM is probably superior overall, simply because of the Latin Hypercube sampling. It cannot be unequivocally stated, from the limited experimentation presented here, whether the ANN ROM, trained on the same data set as the regression ROM, would be better or worse than various orders of the regression ROM. Additional work remains in order to compare and delineate differences between the various types of ROMs.

A comparison of the convergence behaviour of the regression ROM for the baseline case, built with the Latin Hypercube sampling procedure inherent in the ROM Builder, with the ANN ROM and FLUENT® is provided pictorially in the following figure. Again, the solution paths are somewhat different, although all converged to essentially the same answers (all within 0.5%).

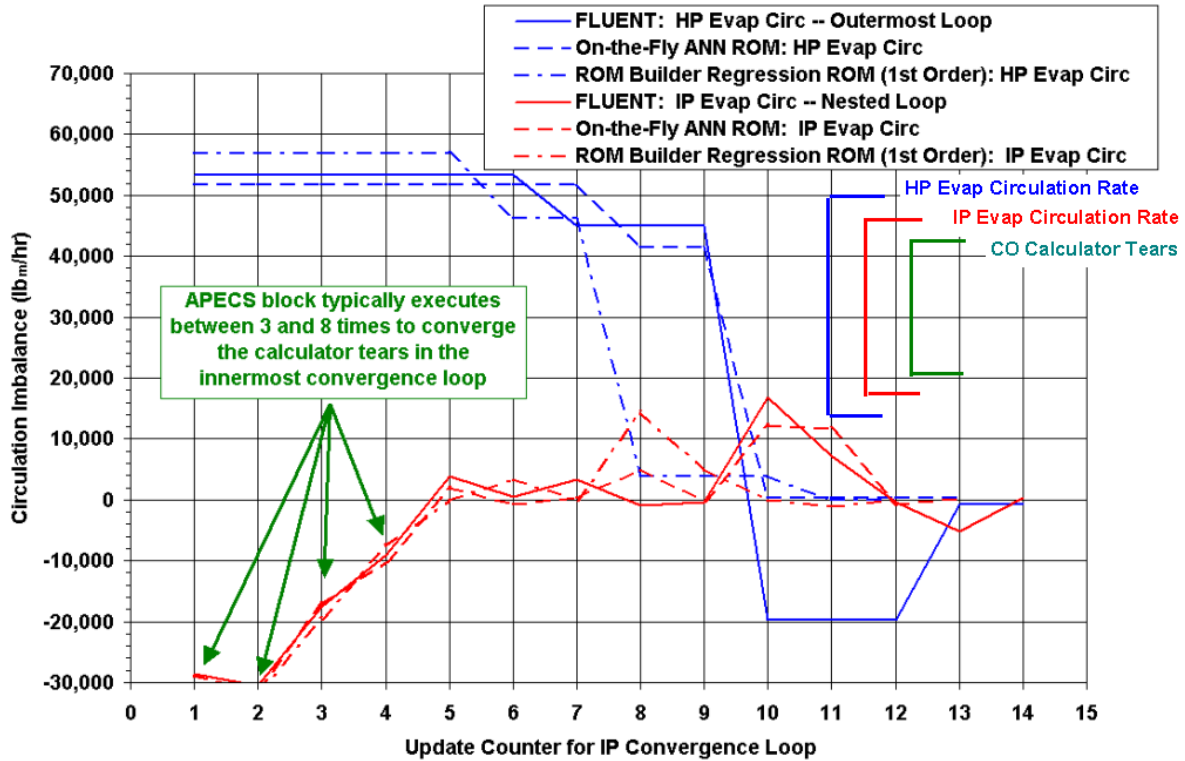


Figure 21: FLUENT® and ROM Convergence Comparison for Demonstration Case 3.

6.0 CONCLUSIONS AND RECOMMENDATIONS

The demonstrated results from this APECS project helps to showcase a novel and unique advancement in the field of process modeling and design – the application and use of regression ROMs built using a Latin Hypercube sampling procedure in the new APECS ROM Builder. Alstom Power was tasked with the responsibility to (a) develop a ROM for FutureGen Equipment Model No. 1 (HRSG) based on a regression methodology (Subtask 3.4.1), and to (b) systematically test the ROM with a simplified version of a FutureGen cycle (Subtask 5.3a). These tasks were accomplished. This report documents the case characteristics and results and discusses the viability and capabilities of the APECS controller and wrapper software for regression ROMS.

The equipment item selected by the DOE for FutureGen Equipment Model No. 1 was a heat recovery steam generator (HRSG) (specifically the HRSG utilized previously in Demonstration Case 3 of the preceding Vision 21 (Phase I) program). A regression ROM was built from multiple runs of the baseline 3-D CFD case. The ROM was instantiated in the NETL flowsheet and exercised for various inlet flow rate perturbations.

In addition, a regression ROM was also constructed for a second HRSG (associated with Demonstration Case 2 of the preceding Vision 21 (Phase I) program). This case was easier to exercise than Demonstration Case 3, and served as a simple trial case to surface and eliminate

software bugs. A significant difference in ROM robustness was demonstrated between a previous “on-the-fly” collection approach and the current Latin Hypercube sampling procedure in the ROM Builder.

Substantial progress has been since project inception by ANSYS in facilitating the direct instantiation of FLUENT® on an Aspen Plus® flowsheet, as well in building FLUENT®-derived ROMs. It is hoped that the APECS tool kit will continue to be developed and enhanced. Many bugs have been surfaced and eliminated and the software has been made more user-friendly and transparent. The documentation is maturing and is a great help in assisting the user to understand the procedures and work flow.

Some additional conclusions and recommendations are as follows:

- (1) Conclusion: It is difficult for the casual industrial user to determine, based on the present graphical comparison of the regression prediction compared to the original FLUENT® simulation available in the ROM Builder, which regression order (e.g., 1st-order versus 2nd-order) should be used for the ROM.
 - [1] Recommendation: Provide good statistical indicators that quantify the overall superiority or “goodness of fit” of one regression order over another (e.g., 1st-order regression versus 2nd- or 3rd-order regression).
- (2) Conclusion: Currently, when the user plots the ROM versus FLUENT® predictions in the ROM Builder, the user is only permitted to see 10 simulations at one time. If the user has e.g., over 100 sample simulations, then the user must page through all of these simulations for each variable. It would be beneficial to allow the user to have the option to see all of the simulations at one time.
 - [2] Recommendation: In the ROM Builder, allow the user the option to see the ROM versus FLUENT® prediction behavior for all of the runs simultaneously as opposed to the default 10 runs (at a time).
- (3) Conclusion: FLUENT® users often utilize convergence monitors (e.g., mass-weighted T, O₂ at the outlet plane). Currently, in the ROM Builder, there is really no way for the user to check how well each of the cases is converging.
 - [3] Recommendation: Perhaps it would be a good idea for the monitors to be saved, just like PCA plots are saved for each simulation, so that the user can call them up later (after all the simulations are done) and look at them so see how well each case behaved.
- (4) Conclusion: One of the issues with ROM building is that the user must prescribe maxima and minima for each of the independent inlet variables. For a large number of inlet variables, this can become an ambiguous and tedious task. It is difficult for the user to determine what these maxima and minima should be, particularly since the minimum and maximum for each variable should be somewhat realistic and balanced with the other independent inlet variables.

If the maximum and minimum of one variable is extreme, then the case may enter an unrealistic regime or even diverge. One way to determine proper maxima and minima is to perturb the flowsheet inlet conditions and run the flowsheet a large number of times, collecting the stream results for the equipment item of interest. The user can then post-process the results to determine the maximum and minimum values experienced by each independent variable over the course of convergence (at each perturbation inlet condition). However, this can be a somewhat tedious (albeit necessary) chore.

- [4] Recommendation: Find the “best” way for the user to prescribe the Min/Max range for each ROM variable and automate as much as possible. For example, is it possible to work with AspenTech and tag an equipment item in a flowsheet, such that the minima and maxima of each stream and CAPE-OPEN variable, associated the equipment item, are calculated and reported when the flowsheet is converged?
- (5) Conclusion: The FLUENT® tube bank heat exchanger model required a steam-water property calculator. Alstom has proprietary packages for stream-water properties, but could not use those packages for intellectual property (IP) reasons. The steam-water property calculator in the present work was a done through UDFs (created by Alstom Power) and functioned via a tabular lookup methodology. Consequently, it was not very robust. Since so many equipment items require steam-water properties, it would be advisable to have a steam-water property calculator that would always be available to the FLUENT® tube bank heat exchanger model (independent of the platform that it is running on).
- [5] Recommendation: The calculator should be able to run on any platform. If, for example, FLUENT® can be made to access the Aspen Plus property package in order to do the cell-by-cell steam-water property calculations, both in the direct-instantiation mode, and in the ROM Builder mode, that would perhaps be acceptable. Advancing to the next level of sophistication and ensuring that both FLUENT® and Aspen Plus always access the same general property package would make the runs more consistent and accurate.

7.0 REFERENCES

1. Zitney, S. E., M. Osawe, L. Collins, E. Ferguson, D. Sloan, W. Fiveland, J. Madsen, “Advanced Process Co-Simulation of the FutureGen Power Plant,” In Proc. of the 31st International Technical Conference on Coal Utilization & Fuel Systems, May 21-25, Clearwater, FL (2006).
2. Sloan, D. G., W. A. Fiveland, S. E. Zitney, and M. Syamlal, "Software Integration for Power Plant Simulations," In Proc. of the 27th International Technical Conference on Coal Utilization & Fuel Systems, March 4-7, Clearwater, FL (2002).
3. Sloan, D. G., W. A. Fiveland, S. E. Zitney, and M. Syamlal, "Power Plant Simulations Using Process Analysis Software Linked to Advanced Modules," In Proc. of the 29th International Technical Conference on Coal Utilization & Fuel Systems, April 18-22, Clearwater, FL (2004).

4. Sloan, D. G., W. A. Fiveland, M. O. Osawe, S. E. Zitney, and M. Syamlal, " Demonstrations of Coupled Cycle Analyses and CFD Simulations over a LAN," In Proc. of the 30th International Technical Conference on Coal Utilization & Fuel Systems, April 17-21, Clearwater, FL (2005).
5. Sloan, D. G., W.A. Fiveland, "Final Simulation Results for Demonstration Cases 1 and 2," Development of Technologies and Analytical Capabilities for Vision 21 Energy Plants, DOE Cooperative Agreement No. DE-FC26-00NT40954 (October 15, 2003).
6. Sloan, D. G. and W.A. Fiveland, "Description of Demonstration Case 3 Deliverable," Development of Technologies and Analytical Capabilities for Vision 21 Energy Plants, DOE Cooperative Agreement No. DE-FC26-00NT40954 (February 28, 2005).
7. Syamlal, M., M. Osawe, S. Zitney, L. Collins, D. Sloan, W. Fiveland, F. Joop, P. Simon , and K. J. Cleetus, "Final Technical Report: October 2000 – December 2004," Development of Technologies and Analytical Capabilities for Vision 21 Energy Plants, DOE Cooperative Agreement No. DE-FC26-00NT40954 (March 2005).
8. "Project Management Plan, Software Framework for Advanced Power Plant Simulations, Cooperative Agreement No. DE-FC26-05NT42443", for Ronald W. Breault USDOE-NETL, compiled by John F. Widmann, representing team entities: ANSYS Inc., ALSTOM Power, Aspen Technology Inc., and Carnegie Mellon University (Department of Chemical Engineering), May 6, 2008.
9. APECS: Advanced Process Engineering Co-Simulation User's Manual, Version 1.6.1, Prepared by ANSYS, Inc., Lebanon, NH (June 2009).
10. Shelton, W. W. and C. W. White, "The Application of an In Situ Oxygen Membrane for an IGCC Process with CO2 Capture," EG&G Final Report, August (2004).

8.0 DELIVERABLES

Although both Demonstration Case 2 and Demonstration Case 3 are discussed in the present report, the deliverables are associated only with Demonstration Case 3. The following files are delivered with this report:

No.	Files	Description
(1)	democ3_rom-prep_rev18_v7p1.bkp democ3_rom-prep_rev18_v7p1_saved.bkp	Aspen Plus® flowsheet, both prior to instantiation of ROM and after instantiation.
(2)	democ3_rom-prep_rev18_v7p1_saved.his democ3_rom-prep_rev18_v7p1_saved_reinit_+0.his democ3_rom-prep_rev18_v7p1_saved_reinit_todeaer_-10.his democ3_rom-prep_rev18_v7p1_saved_reinit_todeaer_-20.his	Aspen Plus® history files

	democ3_rom-prep_rev18_v7p1_saved_reinit_todeaer_-30.his democ3_rom-prep_rev18_v7p1_saved_reinit_todeaer_+10.his democ3_rom-prep_rev18_v7p1_saved_reinit_todeaer_+20.his democ3_rom-prep_rev18_v7p1_saved_reinit_todeaer_+30.his	
(3)	demo_c3_2640_CO.cas demo_c3_2640_CO.dat	FLUENT® case and data files
(4)	hxc.c interpolate.c interpolate.h sg_p1.c	UDFs [Note: These 4 files are <i>Alstom proprietary</i> and are not included in the deliverable.]
(5)	XofPH.dat CPofPH.dat ETAofPT.dat HofPT.dat SatPropsOfPsat.dat TCofPT.dat TofPH.dat	Steam tables (tabular lookup) [Note: These 7 files are <i>Alstom proprietary</i> and are not included in the deliverable.]
(5)	create_planar_surfaces.jou	Auxiliary FLUENT® files (used at the discretion of the user in stand-alone mode)
(6)	demo_c3_2640_CO.CORBASolver	Files created by APECS
(7)	drum.scm	Scheme file that defines RPVAR variables (that are also defined as CO parameters)
(8)	mixing-constants.scm	Example CO scheme file read in by drum.scm
(9)	prep_resid.jou	Used to help the user to run a minimum number of iterations.
(10)	Copy of FLUENT_log.trn	Log file of FLUENT® runs during the running of the ROM Builder from the APECS Temp directory.
(11)	democ3_B3_win_complete-133.APECSData democ3_B3_win_complete-133_2.ioData democ3_B3_win_initial-100-DOE-sims.APECSData	Initial and final APECSData files for ROM Builder, as well as the final ioData file.
(12)	demo_c3_win_rom.COInfo demo_c3_win_rom.CORBASolver demo_c3_win_rom.GridConnectivity demo_c3_win_rom.Mapp demo_c3_win_rom.Matrix	Files created by APECS upon construction of the Regression ROM

	demo_c3_win_rom.Modes demo_c3_win_rom.NodeCoordinates	
(13)	view_iso_a view_side_a view_top_a	Assorted FLUENT® view files

Software Framework for
Advanced Power Plant Simulations

TASK 5.3-B
DEMONSTRATION OF A CALIBRATED ROM
FOR A RADIANT SYNGAS COOLER
IN AN INTEGRATED CO-SIMULATION

Topical Report

Reporting Period Start Date: November 1, 2006
Reporting Period End Date: September 11, 2009

David Sloan, Ph.D.
Woodrow Fiveland, Ph.D.

November 30, 2009

DOE Cooperative Agreement No: DE-FC26-05NT42443

Alstom Power Inc.
US Power Plant Laboratories
P.O. Box 500, CEP 9005-1204
2000 Day Hill Road
Windsor, CT 06095

UNDER SUBCONTRACT TO
ANSYS Inc.
10 Cavendish Court, Lebanon, NH 03766

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ACKNOWLEDGEMENT STATEMENT

This document was prepared with the support of the U.S. Department of Energy, under Award No. DE-FC26-05NT42443. However, any opinions, findings, conclusions, or recommendations expressed herein are those of the author(s) and do not necessarily reflect the views of the DOE.

ACKNOWLEDGEMENTS

Financial support for this project was provided by DOE National Energy Technology Laboratory (Cooperative Agreement No. DE-FC26-05NT42443) and Alstom Power Inc. Appreciation is expressed to Ronald Breault as the DOE Project Manager and to Stephen E. Zitney (NETL) for his technical input. The contributions of the many APECS project team members from ANSYS Inc. and Aspen Technology Inc. are also greatly appreciated. ANSYS Inc. is particularly acknowledged for its leadership role as the prime contractor under John Widmann. Gratitude is expressed to Sorin Munteanu, John Widmann, and others on the APECS team at ANSYS Inc. for their diligence in striving to deliver quality software. Appreciation is also expressed to Alstom Power Energy Recovery GmbH for their kindness in supplying an engineering design for a radiant syngas cooler.

Within the Alstom Power divisions, appreciation is expressed to a number of individuals. John L. Marion, the Alstom Power Program Manager, and Shin Kang and Woodrow A. Fiveland, the Alstom Power Project Managers, contributed to the success of this project task. The contributions of Yen-Ming Chen and Chris Johnson for their work on the radiant syngas cooler are also noted.

ABSTRACT

The primary objective of this DOE project is to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of Advanced Process Engineering Co-Simulator (APECS), the simulator will link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. Generally, the simulator couples the cycle analysis software Aspen Plus® (marketed by Aspen Technology, Inc.), specialized component modeling packages, as exemplified by industrial proprietary codes (utilized by Alstom Power Inc.), and the FLUENT® computational fluid dynamics (CFD) code (provided by ANSYS Inc). An important enhancement to the APECS tool will be the creation of computationally efficient reduced-order models (ROM) based on information from high-fidelity computational fluid dynamics (CFD) models. Integration of the steady-state APECS tool with a virtual reality tool kit will allow process designers to navigate through a 3D virtual representation of a power plant with visualization of simulation results. A related secondary objective is to develop a prototype dynamic simulator within APECS for advanced power plants and to demonstrate a dynamic simulation that integrates plant-level and equipment-level models.

Among other responsibilities, Alstom Power has the responsibility to demonstrate the capabilities of the enhanced APECS tool to simulate proposed FutureGen power plant concepts and other relevant power generation processes and equipment. The present report encompasses only the work scope within Subtasks 3.4.2, 5.1 and 5.3, having to do with the calibrated ROM for the FutureGen Equipment Model No. 2. Specifically, Alstom Power is tasked with the responsibility to develop a ROM for FutureGen Equipment Model No. 2 based on both regression and principal component analysis (PCA) (Subtask 3.4.2), and to calibrate the ROM based on the equivalent CFD component model (Subtask 5.1.1), verifying adequate predictive accuracy over the parameter space required for convergence of the FutureGen cycle at the full load condition (Subtask 5.1.2). Subtask 5.3 requires that the ROM then be systematically tested with a simplified version of a FutureGen cycle.

The equipment item selected by the DOE for FutureGen Equipment Model No. 2 was a radiant syngas cooler. For a specific DOE process model, an engineering design for a radiant syngas cooler was completed and a CFD model was constructed in accordance with the design. The CFD model was calibrated by adjusting the wall heat transfer boundary conditions until the desired absorption was achieved. The CFD case was exercised over a range of flow rates. Since the APECS PCA capability for ROMs is presently only available for 2-D cases, a 2-D CFD case was also constructed. A regression ROM was built from multiple runs of the baseline 2-D CFD case, and the PCA capability was verified by reviewing the associated 2-D contour plots. The ROM was instantiated in the DOE flowsheet and exercised for various inlet flow rate perturbations. This report documents the case characteristics and results and discusses the viability and capabilities of the linkage/interface software for regression ROMS with the PCA-generated contour plots.

TABLE OF CONTENTS

1.0	BACKGROUND ON APECS PROGRAM	3
2.0	INTRODUCTION	3
3.0	EXECUTIVE SUMMARY	4
4.0	EXPERIMENTAL WORK.....	6
5.0	RESULTS AND DISCUSSION.....	6
5.1	Case Setup and Preparation.....	6
5.1.1	Module Selection and Design Specification.....	6
5.1.2	FLUENT® 3-D Grid and Case Setup.....	16
5.1.3	2-D FLUENT® Case Calibration and Setup	23
5.1.4	CO Parameters, Ports, and Connecting Streams.....	31
5.2	Computational ROM Results.....	35
5.2.1	ROM Construction.....	35
5.2.2	Integrated Co-Simulation With ROM.....	42
6.0	CONCLUSIONS AND RECOMMENDATIONS	48
7.0	REFERENCES	50
8.0	DELIVERABLES.....	51

LIST OF FIGURES

Figure 1:	Gasifier with Downstream RSC and Slag Crusher (www.SHG-SCHACK.com).....	7
Figure 2:	Portion of the Process Model Surrounding the Radiant Syngas Cooler.....	9
Figure 3:	Portion of the Process Model Associated With the RSC Energy Stream.	10
Figure 4:	Typical Quench System Downstream of RSC (Figure 6 Cited in Ref. 7).....	11
Figure 5:	RSC with Tube Cage Internals (Plan View).	13
Figure 6:	RSC with Dimensional Heights of Tube Cage and Internals.....	14
Figure 7:	Gas-Phase Absorption Coefficient for RSC Case.....	16
Figure 8:	RSC 45°-Sector Geometry.....	18
Figure 9:	RSC 45°-Sector Grid with 102,000 Cells.....	18
Figure 10:	RSC 22.5°-Sector Grid with 60,000 Cells.....	19
Figure 11:	Pathlines Showing RZ in RSC 45°-Sector Grid (1.9 Million Cells (Gas Only)).	20
Figure 12:	Vertical Velocity Contours for RSC 45° Sector Case (1.9 Million Cells, Gas Only).	21
Figure 13:	Pathlines for RSC 22.5° Sector Case (60,000 Cells, With Particles).	22
Figure 14:	Domed Inlet Section for RSC (Ref. 7).	23
Figure 15:	Outlet Temperature as a Function of Inlet Flow Rate for 3-D Case.	25
Figure 16:	Volumetric Energy Sink Constant for Porous Medium in 2-D Case.	25
Figure 17:	Monitored Outlet Temperature for the 3-D Baseline Case.	26
Figure 18:	Monitored Outlet Temperature for the 3-D -15% Case.....	26
Figure 19:	Monitored Outlet Temperature for the 3-D +15% Case.....	27
Figure 20:	Z-Velocity Contour Plots for 3-D Cases.....	28
Figure 21:	Temperature Contour Plots for 3-D Cases.....	29

Figure 22: Axial Velocity Contour Plots for 2-D Cases.....	30
Figure 23: Temperature Contour Plots for 2-D Cases.	30
Figure 24: Preparation of the Flowsheet for Instantiation of the RSC ROM.	31
Figure 25: FLUENT® Versus ROM Predictions for Temperature at Port Outlet.....	38
Figure 26: FLUENT® Versus ROM Predictions for External Wall Heat Flux.	39
Figure 27: FLUENT® vs. ROM Contour Plots for Axial Velocity for Simulation #5.	39
Figure 28: FLUENT® vs. ROM Contour Plots for Radial Velocity for Simulation #5.....	40
Figure 29: FLUENT® vs. ROM Contour Plots for Velocity Magnitude for Simulation #5.....	40
Figure 30: FLUENT® vs. ROM Contour Plots for Temperature for Simulation #5.	41
Figure 31: Simulation Status for the ROM Builder Runs.....	42
Figure 32: PCA Axial Velocity Results from the Co-Simulation.....	46
Figure 33: PCA Radial Velocity Results from the Co-Simulation.....	46
Figure 34: PCA Velocity Magnitude Results from the Co-Simulation.....	47
Figure 35: PCA Temperature Results from the Co-Simulation.....	47

LIST OF TABLES

Table 1: Stream Results for the RSC from the Converged Flowsheet	11
Table 2: Stream Results for the RSC from the Converged Flowsheet	24
Table 3: FLUENT® Block Ports and Connecting Streams	32
Table 4: Species Listing and Mapping.....	33
Table 5: CAPE-OPEN Parameters As Implemented Through APECS Wizard.....	34
Table 6: ROM Variable Range	37
Table 7: Co-Simulation ROM Robustness Testing	45
Table 8: Comparison of Co-Simulation Results with Target Values	45

1.0 BACKGROUND ON APECS PROGRAM

The current APECS program is a continuation of an earlier (Vision 21) program, which was initiated under the DOE-Fluent Cooperative Agreement No. DC-FC26-00NT40954 (hereafter referred to as Phase I). Both programs have been dedicated to the development of the software infrastructure for advanced power plant simulations. The goal of the overall program work scope has been to develop an integrated suite of software tools that could be used to simulate and visualize advanced plant concepts that are being considered under the umbrella of DOE's broad CO₂ remediation and power plant efficiency programs. Advanced simulation tools are needed to evaluate new power plant concepts that will undoubtedly help to minimize costly laboratory and field trials.

The typical process simulation software that has been commercially available to industry does not meet the DOE's objective of "virtual simulation" that is needed to evaluate complex cycles. The intent of the DOE has been to improve predictive tools for cycle analysis, and to improve the component models that are used in turn to simulate the cycle. Generally, the modeled performance of various components is derived from a simple set of reaction and output prescriptions that are collectively accessed by the cycle simulation software as a "black-box" library module, without any explicit impact from geometry modifications and enhancements to the fluid mechanics. These simple component models are not sufficient for component design; nor are they adequate for the evaluation of the impact of complex components on cycle performance. To meet the DOE's goal of predicting and visualizing the performance of these complex systems, there has been a need to use more sophisticated component models, by upgrading or converting library modules to CFD modules, and to couple the CFD modules directly with other equipment items in the power plant through process modelling software. The primary focus of the present APECS project has been to provide the software interface and environment that permits FLUENT® or a surrogate ROM to run in tandem with the Aspen Plus® process simulation package (see Ref. 1 for an overview of the program).

2.0 INTRODUCTION

The feasibility of using FLUENT® CFD models in concert with a process model like Aspen Plus® has been a focus of the APECS project. The role of Alstom Power Inc. (Alstom) has been to assist the project team in helping to develop and demonstrate the capabilities of the advanced APECS simulation and visualization tool, oftentimes with the use of Alstom designed and validated equipment modules (see e.g., Ref. 2 to 4). The primary Alstom responsibilities and tasks have included:

- providing its expertise and experience base in the utilization of both CFD and cycle analysis for the power generation industry,
- selecting and running cases to test and demonstrate the feasibility of the concepts,
- forming an advisory board to provide project review and feedback.

Among other responsibilities, Alstom Power has the responsibility to demonstrate the capabilities of the enhanced APECS tool to simulate proposed FutureGen power plant concepts and other relevant power generation processes and equipment. The present report encompasses only the work scope

within Subtasks 3.4.2, 5.1 and 5.3b, having to do with the calibrated ROM for the FutureGen Equipment Model No. 2. Specifically, Alstom Power is tasked with the responsibility to develop a ROM for FutureGen Equipment Model No. 2 based on principal component analysis (PCA) (Subtask 3.4.2), and to calibrate the ROM based on the equivalent CFD component model (Subtask 5.1.1), verifying adequate predictive accuracy over the parameter space required for convergence of the FutureGen cycle at the full load condition (Subtask 5.1.2). Subtask 5.3 requires that the ROM then be systematically tested with a simplified version of a FutureGen cycle.

The equipment item selected by the DOE for FutureGen Equipment Model No. 2 was a radiant syngas cooler. For a specific DOE process model, an engineering design for a radiant syngas cooler was completed and a CFD model was constructed in accordance with the design. Since the APECS PCA capability for ROMs is presently only available for 2-D cases, a 2-D CFD case was also constructed. A regression ROM was built from multiple runs of the baseline 2-D CFD case, and the PCA capability was verified by reviewing the associated 2-D contour plots. The ROM was instantiated in the DOE flowsheet and exercised for various inlet flow rate perturbations. This report documents the case characteristics and results, and discusses the viability and capabilities of the linkage/interface software for regression ROMS with the PCA-generated contour plots.

Subtasks 3.4.2, 5.1 and 5.3b are detailed within the Project Management Plan (Ref. 5), and embrace a work scope associated with a calibrated ROM for the FutureGen Equipment Model No. 2. Alstom's responsibilities associated with those subtasks are to:

- [Subtask 3.4.2] Develop a ROM for FutureGen Equipment Model No. 2 based on principal component analysis (PCA).
- [Subtask 5.1.1] Calibrate the ROM based on multiple solutions of the equivalent CFD component model.
- [Subtask 5.1.2] Verify adequate predictive accuracy over the parameter space required for convergence of the FutureGen cycle at the full load condition.
- [Subtask 5.3b] Systematically test the ROM with a simplified version of a FutureGen cycle.

The deliverables include a task report, as well as all model files. The task report contains a description of the case simulations, the results, and an itemization of desired software modifications for future work. This report summarizes and documents the demonstration case simulations under Task 5.3b, and thus completes the requirement for the stipulated (contractual) task report.

3.0 EXECUTIVE SUMMARY

The primary objective of this DOE project is to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of Advanced Process Engineering Co-Simulator (APECS), the simulator will link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. Generally, the simulator couples the cycle analysis software Aspen Plus® (marketed by Aspen Technology, Inc.), specialized component modeling packages, as exemplified by industrial proprietary codes (utilized by Alstom Power Inc.), and the FLUENT® computational fluid dynamics (CFD) code (provided by ANSYS Inc). An important

enhancement to the APECS tool is the creation of computationally efficient reduced-order models (ROMs) based on information from high-fidelity CFD models.

In the present work scope, Alstom Power is tasked with the responsibility to (a) develop a ROM for a radiant syngas cooler based on both regression and principal component analysis (PCA), (b) to calibrate the ROM based on the equivalent CFD component model, verifying adequate predictive accuracy over the parameter space required for convergence of a NETL FutureGen process model at the full load condition, and to (c) systematically test the ROM with a simplified version of a FutureGen cycle.

For a specific DOE process model, an engineering design for a radiant syngas cooler was completed and a CFD model was constructed in accordance with the design. The CFD model was calibrated by adjusting the wall heat transfer boundary conditions until the desired absorption was achieved. The CFD case was exercised over a range of flow rates. Since the APECS PCA capability is presently only available for 2-D cases, a 2-D CFD case was also constructed. A regression ROM was built from multiple runs of the CFD case, and the PCA capability was verified by reviewing the associated 2-D contour plots. The ROM was instantiated in the DOE flowsheet and exercised for various inlet flow rate perturbations. The demonstrated results from this APECS project represent a novel and unique advancement in the field of process modeling and design – the application and use of PCA technology for ROM contour plots. All of the runs were completed successfully.

The CFD runs exposed a recirculation zone on the symmetry axis, with multiple, complex reverse flow zone patterns downstream of the recirculation zone. Such aerodynamic patterns do not exist in an actual Alstom RSC. However, Alstom Power Energy Recovery GmbH would not provide the additional engineering design support or information necessary to eradicate such patterns from the computational domain because of the likelihood that proprietary or sensitive information would be disclosed. Consequently, the RSC CFD model provided as a deliverable with this report, is only a gross approximation of a properly operating RSC. Nevertheless, the procedure used here -- to construct a ROM with regression and PCA capability, to instantiate it on a flowsheet, and to run it in a co-simulation -- may be followed for any subsequent syngas cooler design investigated by NETL and its partners.

During this project, significant progress was made in surfacing and eradicating bugs in the APECS software tool kit (as well as bugs in FLUENT® and Aspen Plus®). Recommendations for future enhancements might include:

- Develop 3-D PCA transformations for the APECS tool kit. Rather than carry the burden of a full 3-D flow field transformation, an alternative or intermediate approach might be to find a way to do 2-D PCA transformations on selected 2-D plotting planes defined by the user within the 3-D domain of the FLUENT® case.
- Provide greater interactive capability in manipulating the PCA-generated plots. For example, the user might desire to (a) change scales on the plot, or (b) enlarge and expand certain areas of the plot to view certain details better, or (b) clip values. Many of the interactive post-processing features currently in FLUENT®, such as pathlines, etc. might

also be put to good advantage in both the ROM Builder and the CFD Viewer (or its equivalent).

- When specifying inputs for the individual simulations in the ROM Builder, allow the user to toggle between the options of (a) putting in the individual species flow rates or (b) inserting the total mass flow rate plus the individual species mass fractions.
- Perform a sorting and pre-mapping based on similar names so that much, or all, of the species mapping is already done for the user.
- Modify the Simulation Status panel to (a) reflect the exceptions or errors that are thrown when the FLUENT® case is terminated, and (b) provide the “monitor” results for each run so that the user can better assess the convergence.

4.0 EXPERIMENTAL WORK

There is no experimental work under this project.

5.0 RESULTS AND DISCUSSION

5.1 Case Setup and Preparation

5.1.1 Module Selection and Design Specification

Candidate models for the FutureGen Equipment Model No. 2 included a gasifier, steam turbine (low to high pressure), air preheater, syngas cooler (convective or radiative), etc. The NETL was consulted in the selection of the Equipment Model No. 2 from the list of candidate equipment items. In January of 2008, the NETL (in the person of Steve Zitney) recommended a radiant syngas cooler. Specifically, the NETL was interested in obtaining a FLUENT® CFD model that reasonably represents a GE radiant syngas cooler that sits under a downflow GE gasifier. A typical ALSTOM design for such an arrangement is provided in Figure 1:

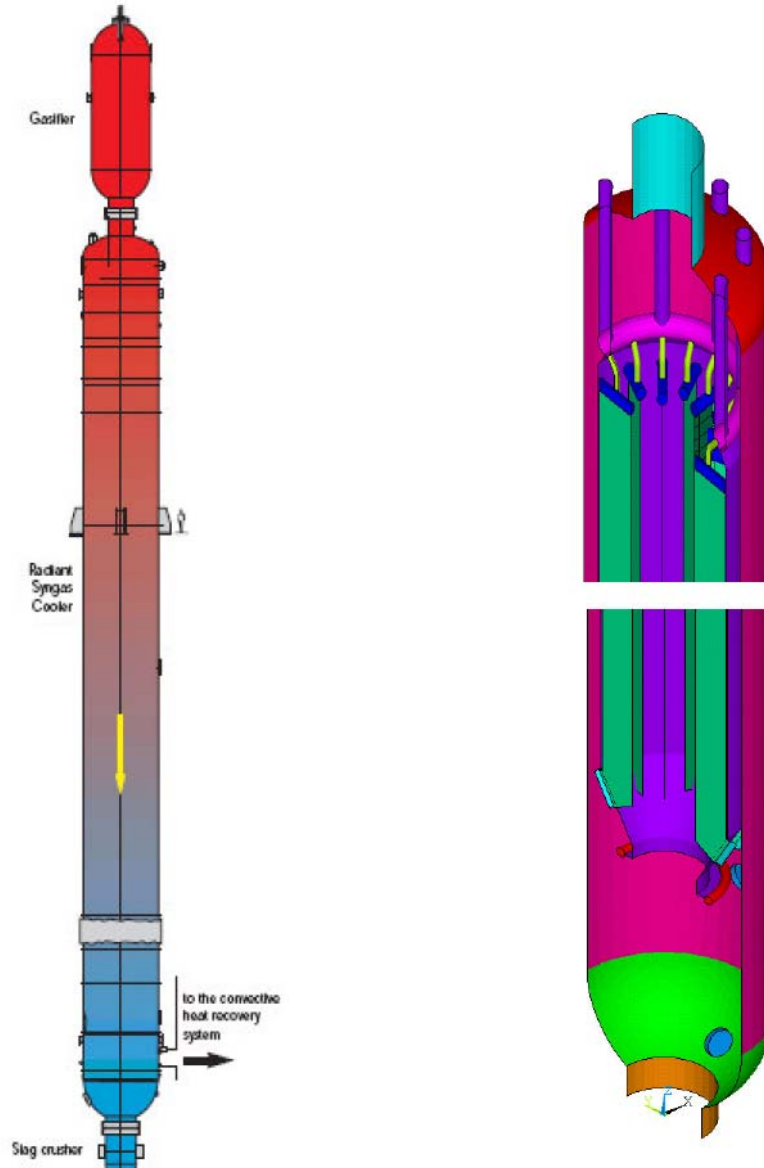


Figure 1: Gasifier with Downstream RSC and Slag Crusher (www.SHG-SCHACK.com)

The operating and feed/exhaust conditions for the radiant syngas cooler were extracted from Ref. 6. In the referenced report, General Electric Energy (GEE) offers three design configurations for the GEE gasifier island, but finally selects the “radiant only” configuration as the best design:

“Radiant Only: In this configuration, the hot syngas exiting the gasifier passes through a radiant syngas cooler where it is cooled from about 1316°C (2400°F) to 816°C (1500°F), then through a water quench where the syngas is further cooled to about 204°C (400°F) prior to entering the syngas scrubber. Relative to the quench configuration, the radiant only design offers increased output, higher efficiency, improved reliability/availability, and results in the lowest cost of electricity. This configuration was chosen by GEE and Bechtel for the design of their reference plant.”

The referenced report also focused on two separate cases: (1) Case 1 – GEE IGCC without CO₂ capture, and (2) Case 2 – GEE IGCC with CO₂ capture. The NETL requested that the present project focus exclusively on Case 2 – GEE IGCC with CO₂ capture. The plant configuration consists of two gasifier trains, two advanced F Class turbines, two HRSGs, and one steam turbine. The report provides the following details:

- The net plant power is 556 MWe. The gasification plant utilizes two gasification trains to process a total of 5,448 tonnes/day (6,005 TPD) of Illinois No. 6 coal. Each of the 2 x 50 percent gasifiers operates at maximum capacity. The slurry feed pump takes suction from the slurry run tank, and the discharge is sent to the feed injector of the GEE gasifier. The air separation plant supplies 4,635 tonnes/day (5,110 TPD) of 95 mole percent oxygen to the gasifiers and the Claus plant. Carbon conversion in the gasifier is assumed to be 98 percent including the fines recycle stream. The gasifier vessel is a refractory-lined, high-pressure combustion chamber. The coal slurry feedstock and oxygen are fed through a fuel injector at the top of the gasifier vessel. The coal slurry and the oxygen react in the gasifier at 5.6 MPa (815 psia) and 1,316°C (2,400°F) to produce syngas. The syngas consists primarily of hydrogen and carbon monoxide, with lesser amounts of water vapour and carbon dioxide, and small amounts of hydrogen sulphide, carbonyl sulphide, methane, argon, and nitrogen. The heat in the gasifier liquefies coal ash. Hot syngas and molten solids from the reactor flow downward into a radiant heat exchanger where the syngas is cooled.
- The syngas is cooled from 1316°C (2,400°F) to 593°C (1,100°F) in the radiant synthesis gas cooler (SGC) and the molten slag solidifies in the process. The solids collect in the water sump at the bottom of the gasifier and are removed periodically using a lock hopper system. The waste heat from this cooling is used to generate high-pressure steam. Boiler feedwater in the tubes is saturated, and then steam and water are separated in a steam drum. Approximately 548,122 kg/h (1,208,400 lb/h) of saturated steam at 13.8 MPa (2,000 psia) is produced.
- The syngas exiting the radiant cooler is directed downwards by a dip tube into a water sump. Most of the entrained solids are separated from the syngas at the bottom of the dip tube as the syngas goes upwards through the water. The syngas exits the quench chamber saturated at a temperature of 210°C (410°F).

On January 21, 2008, Steve Zitney (NETL) provided the Aspen Plus® process model corresponding to Case 2: “IGCC005_GEWCO2.zip,” which contained the Aspen Plus® template file “IGCC005.apr,” as well as 10 RDS proprietary .obj files requiring linking (see Ref. 6). The original flowsheet was run with Aspen Plus® Version 2004.1. The “sequential-modular” solution approach was used in all cases, as opposed to the “equation-oriented” solution methodology. A picture of the portion of the converged flowsheet surrounding the radiant syngas cooler is given in Figure 2.

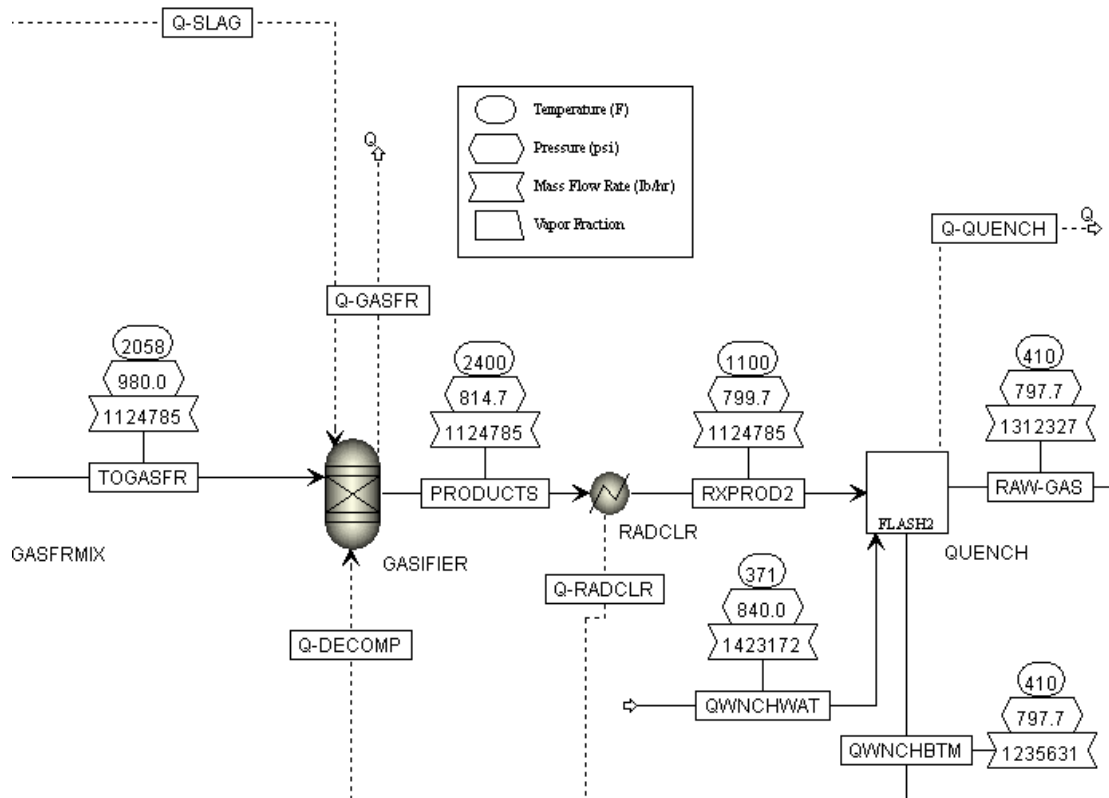


Figure 2: Portion of the Process Model Surrounding the Radiant Syngas Cooler.

In the above process model, the gasifier, radiant syngas cooler, and quench section constitute three separate modules on the flowsheet.

- The amount of wet coal fed to the gasifier island is based upon achieving a particular power output of the system (through the Design Spec “POWER”). The flow rates of the accompanying slurry and oxygen feeds are both computed directly as a linear function of the wet coal feed (via calculator blocks). The gasifier module (GASIFIER) is an RGIBBS reactor with the specified outlet pressure and temperature indicated in Figure 2. The product distribution was computed and calibrated using the calculation option “restrict chemical equilibrium,” along with a specified temperature approach. Consequently, the outlet product species distribution, temperature, and pressure at the gasifier outlet are fixed, no matter how much the wet coal feed rate varies at any point during the convergence.
- The module for the radiant syngas cooler (RADCLR) is a simple heat exchanger model, in which the outlet temperature of 1,100°F (866.48 K) has been specified, as well as a pressure drop across the module of 15 psi (103,421 Pa). As shown below in Figure 3, the calculated heat duty for the cooler (energy stream QRADCLR) is used to calculate the “net” feedwater (RAW-EVAP) flow rate required to raise the syngas cooler feedwater (slightly subcooled) quality to a vapor fraction of unity (via the Design Spec “STM2”). The outlet (saturated-steam) stream leaving the syngas cooler is called RAWEVAP2.

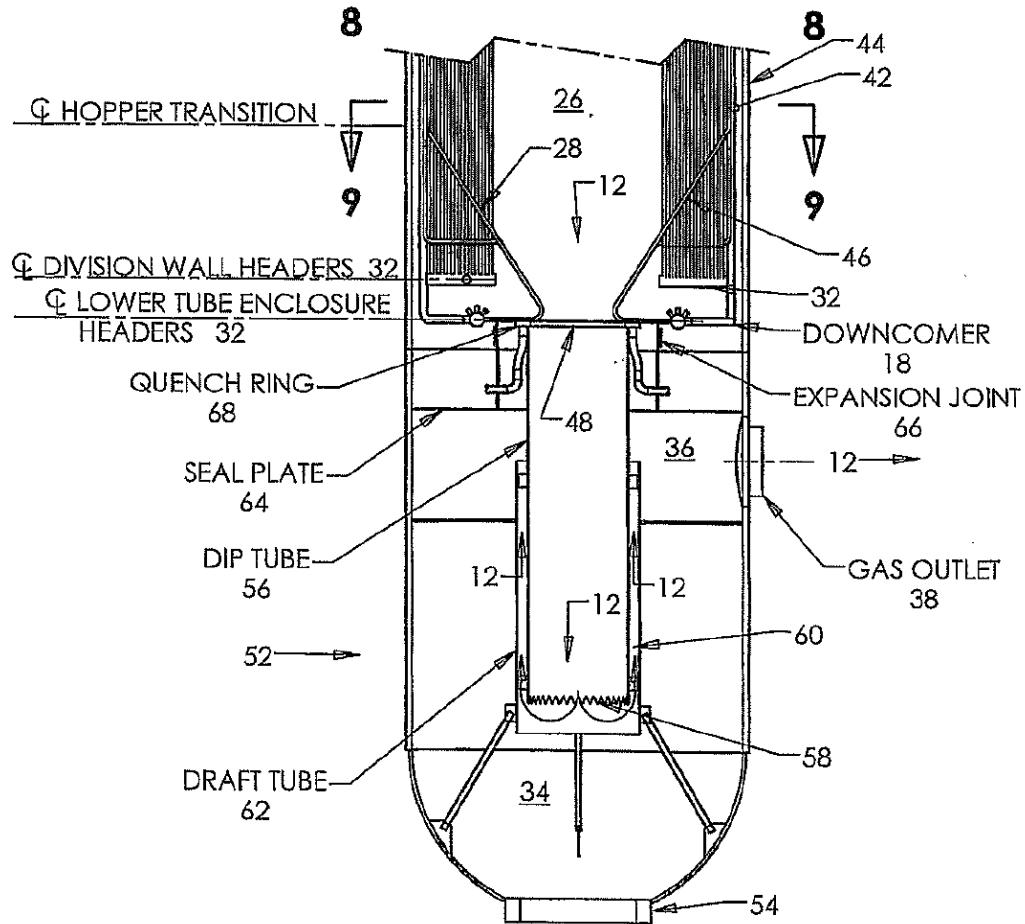


Figure 4: Typical Quench System Downstream of RSC (Figure 6 Cited in Ref. 7).

On March 17, 2008, a request was made to Alstom Power Energy Recovery GmbH (APER GmbH) to assess the time and effort required to provide a geometric design for the radiant syngas cooler, based on the following values from the converged Aspen Plus® flowsheet:

Table 1: Stream Results for the RSC from the Converged Flowsheet

	PRODUCTS	RXPROD2	RAW-EVAP	RAWEVAP2
Substream: MIXED				
Mass Fraction				
H2O	0.1261335	0.1261335	1	1
AR	0.0155514	0.0155514		
CO2	0.325803	0.325803		
O2	1.56E-12	1.56E-12		
N2	0.0122325	0.0122325		
CH4	7.76E-04	7.76E-04		
CO	0.4723418	0.4723418		
COS	5.57E-04	5.57E-04		

H2	0.0330809	0.0330809		
H2S	0.0121438	0.0121438		
NH3	1.38E-03	1.38E-03		
Total Flow (lb/hr)	1.0699E+06	1.0699E+06	1.2084E+06	1.2084E+06
Total Flow (kg/s)	134.80	134.80	152.25	152.25
Temperature (°F)	2400	1100	595	635.8167
Temperature (K)	1588.7	866.5	585.9	608.6
Pressure (psi)	814.7	799.7	2000.7	2000.7
Pressure (Pa)	5,617,159	5,513,737	13,794,341	13,794,341
Vapor Fraction	1	1	0	1
Liquid Fraction	0	0	1	0
Enthalpy (Btu/lb)	-1,811.68	-2,389.675	-6,259.17	-5,728.214
Enthalpy (J/kg)	4,211,148	5,554,665	-14,549,089	-13,314,912
Density (lb/ft ³)	0.5418025	0.9751099	43.32313	5.311201
Density (kg/m ³)	8.6788	15.6198	693.9700	85.0773
Average MW	20.40872	20.40872	18.01528	18.01528
Dew Point (°F)	341.1132	339.7052	635.8167	635.8167
Substream: NC				
Mass Fraction				
SLAG	1	1		
Total Flow (lb/hr)	54,925.23	54,925.23		
Total Flow (kg/s)	6.92046	6.92046		
Solid Fraction	1	1		
Enthalpy (Btu/lb)	662.7756	239.9786		
Enthalpy (J/kg)	1,540,585	557,817		
Note:	The above results are for a dual train RSC. For a single train RSC, mass flow rates should be divided by a factor of 2.			
Note:	SLAG is defined in the flowsheet as consisting of 11.66% carbon and 88.34% ash.			
Note:	The total heat transferred (QCALC) from the composite RSC to the energy stream Q-RADCLR is equal to 641,596,385 Btu/hr (187,907,541.3 W). For a single train RSC, this value should be divided by a factor of 2.			

APER GmbH agreed to size the radiant syngas cooler. However, in an effort to protect Intellectual Property (IP), APER GmbH determined that it would not provide RSC geometry or detailed information on process engineering that included any IP. Furthermore, APER GmbH required a secrecy agreement that reserved the right to approve all documents (associated with the present task) for publication. APER GmbH signed the secrecy agreement on May 21, 2008, and a purchase order (PO) was subsequently requisitioned on June 5, 2008.

APER GmbH provided the final engineering design on August 29, 2008. In accordance with a series of communications with NETL, the inlet diameter to the RSC was fixed at 0.8 m in order to accommodate the exhaust diameter of the single-stage, downflow, entrained gasifier being developed by the Vishwamitra Research Institute (VRI), a task funded under the direction of NETL.

The design for the properly sized geometry is provided in Figures 5 and 6. It should be noted that the final tube cage is not exactly cylindrical. Due to fabrication reasons, often a few tubes with fins are welded together in a longitudinal direction to form a straight “board” or wall section in a pre-fabrication step. Subsequently, these “boards” joined together to form an n-sided regular polygon, where the value “n” depends on the number of discrete “boards” in the tube cage.

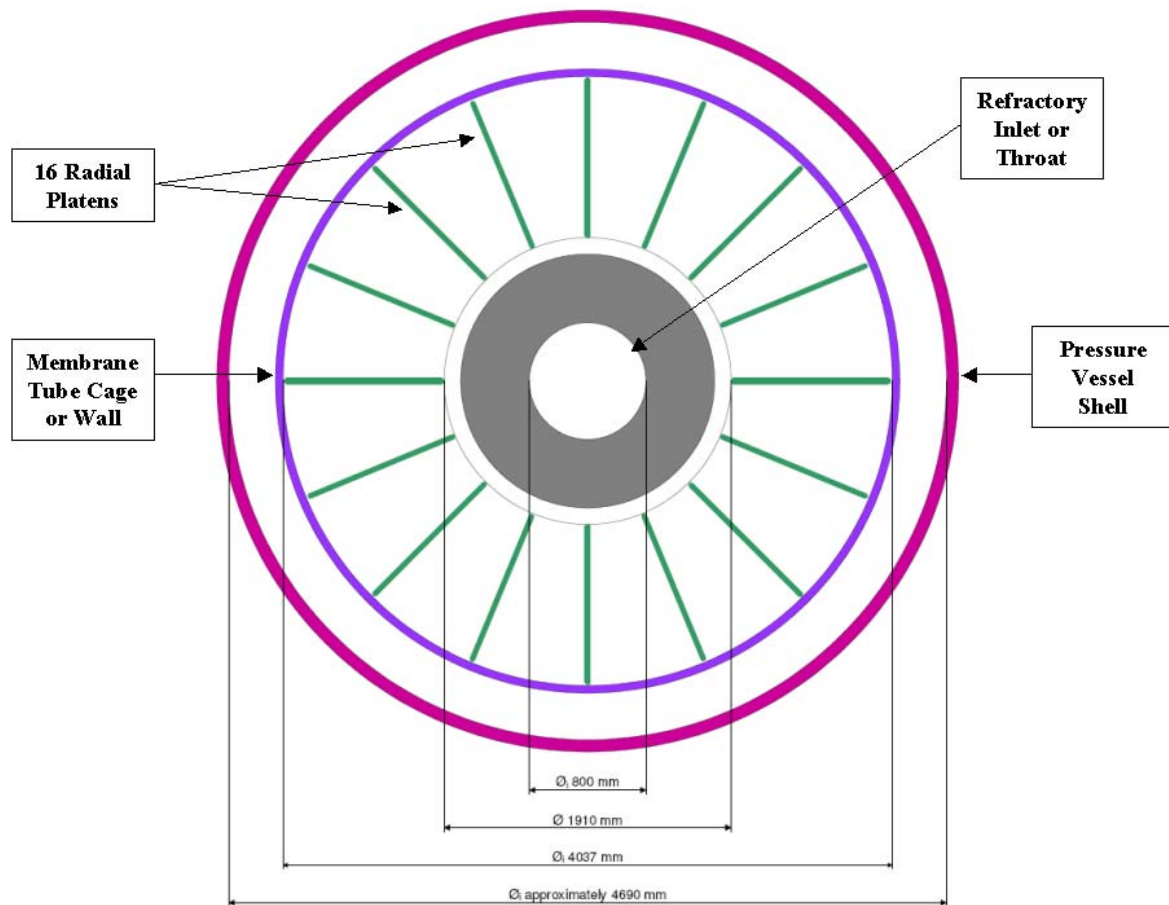


Figure 5: RSC with Tube Cage Internals (Plan View).

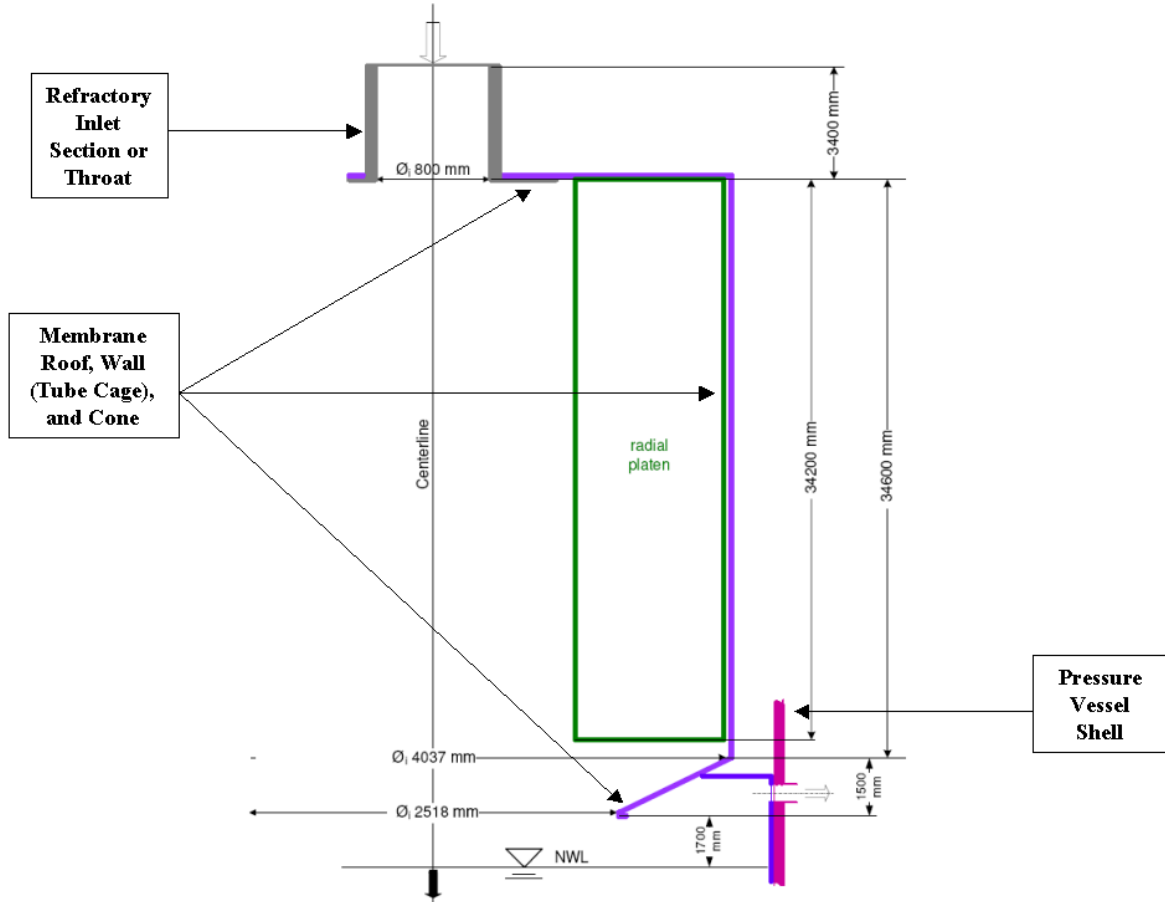


Figure 6: RSC with Dimensional Heights of Tube Cage and Internals.

Hot synthesis gas with slag enters the RSC via a refractory equipped inlet section (called a throat) connecting the gasifier and the RSC. The main section for heat transfer is surrounded by a membrane roof on top, a cylindrical membrane wall, and a membrane cone. Radially arranged vertical platens are located inside the heat transfer section to enlarge the heat transfer surface. Membranes as well as platens are of a tube-fin-tube design. The center part of the main section for heat transfer is free of internals to allow the vertical pass of slag into the water filled sump in the bottom section of the RSC. Slag is cooled down in the water bath in the sump section and withdrawn via a bottom flange into a slag crusher and a lock hopper system.

Circulation water coming from a steam drum is heated up and partly evaporated inside the tubes of the membranes and platens. The mixture of steam and water is routed back into a steam drum, where the saturated steam is separated and exported. Boiler feedwater is added to the steam drum to keep the water level constant.

The sizing was optimized with an in-house, proprietary design codes at APER GmbH. With reference to Figures 5 and 6, the sizing results are as follows:

- The free diameter for syngas flow from the gasifier down into the RSC is 0.800 m (as requested as requested by NETL to match the outlet geometry of the gasifier). The length of this section from the gasifier connection to the RSC downward: 3.400 m. With the given process data a syngas velocity of approximately 15.6 m/s is calculated. This velocity is within the range that is normally used for process design of this section of the RSC. (Access from the gasifier through this free 0.8 m diameter into the RSC has to be checked for erection and installation of the refractory in this section.)
- Free (inner) diameter of the tube cage (membrane wall cage, polyangular cross-section): 4.037 m.
- Horizontal membrane (roof of the tube cage) runs from the above-mentioned inlet section to the vertically arranged tube cage. With the given process data an average syngas downflow velocity of approximately 1.0 m/s is calculated in the top section of the tube cage. This velocity is within the range normally used for process design of this section of the RSC.
- Inside the tube cage there are 16 radially arranged platens (tube-fin-tube design) set in place.
- The inner section of the tube cage with a diameter of 1.910 m has to be free of internals to allow the slag to fall down through the center of the RSC.
- Height of the radially arranged platen from top of the tube cage downwards is 34.200 m.
- Height of the vertically arranged tube cage from top of the tube cage downwards is 34.600 m.
- The tube cage continues downwards in a cone with a free (inner) diameter at the lower end of 2.518 m. The height of this cone is calculated to be approximately 1.5 m.
- Distance from lower end of the cone of the tube cage to Normal Water Level (NWL) in the water bath (sump) is chosen to be 1.7 m.
- Inner diameter of the vessel shell is calculated to be approximately 4.690 m. The tube cage is a solid wall, consisting of metal tubes and fins, welded together. A small amount of purge gas is added into the RSC head and in the annulus section between the membrane cage and the pressure vessel shell. The purge gas is at a temperature similar to the saturation temperature of the water/steam mixture inside the (tube cage) tubes.
- A membrane wall as a thermal barrier protects the inner side of the vessel shell above the level of the water bath up to the outside of the tube cage cone as well. Cooled down syngas passes the membrane cone and a quench ring and enters a dip tube containing the water sump. The syngas makes a U-turn and bubbles up through a draft tube into a plenum area, from which it exhausts into horizontally arranged transfer line(s) that penetrate through the vessel shell. The heavy slag particles and fly ash drop into the water bath. Perhaps some of the fine fly ash/slag particles are entrained with the bubbling syngas and follow the flow path of the syngas into the transfer line(s).
- Dimensions are based on the requested syngas outlet temperature and a fouling behaviour of a raw syngas-out formed from a coal like Pittsburgh No. 8 or Illinois No. 6. The heat loss to ambient is assumed to 1 % of transferred heat rate and the continuous blowdown from the steam drum is fixed to 1 % of boiler feedwater supply rate.

For CFD case calibration purposes, the throat section, made out of refractory, can be considered to be adiabatic. For the current configuration, a certain fraction of the total absorption comes from the radial platen; the remainder comes from the external walls, i.e., the tube cage, membrane cone, and the membrane roof.

5.1.2 FLUENT® 3-D Grid and Case Setup

Using the design information provided by APER GmbH, 3-D CFD cases were prepared to investigate the flow characteristics and convergence behaviour of the radiant syngas cooler. It was decided to take advantage of the geometric symmetry associated with the 16 radial platens in the RSC and focus on both a 45°-sector model (containing 3 radial platens), as well as a 22.5°-sector model (bounded by 2 radial platens).

The gas-phase absorption coefficient was calculated from (a) a proprietary, in-house code (based on an Edwards Wide Band Model) for the given inlet mole fractions (corresponding to Table 1), (b) an average RSC static pressure of 5,565,450 Pa, and (c) a FLUENT®-calculated path length of 1.017 m. The calculated gas-phase absorption coefficient (see Figure 7), as a function of temperature, was curve-fit with a quadratic expression over the expected temperature range; the quadratic expression was then utilized explicitly in the FLUENT® materials panel. The gas-phase scattering coefficient was assumed to be identically zero.

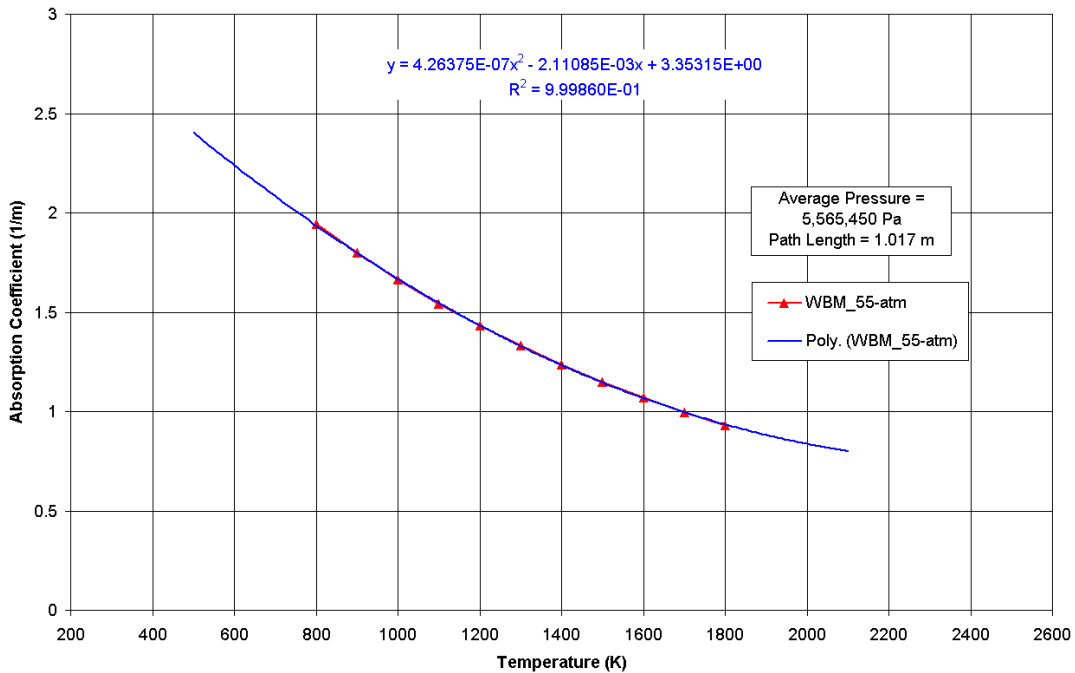


Figure 7: Gas-Phase Absorption Coefficient for RSC Case.

Some of the other CFD case characteristics and selected submodel options are itemized below:

- FLUENT® V6.3.31 (Linux) was used for all of the RSC CFD investigatory computations and the ROM preparation runs used FLUENT® V6.3.26 on a PC with a Windows OS.
- The solution algorithm is based upon the SIMPLE pressure-correction algorithm with first-order (or second-order) upwind differencing.
- The turbulence model is the Realizable k-ε turbulence model, with standard wall functions.

- To help reduce the possibility of reverse flow at the domain exit, a small porous media zone was placed near the exit of the computational domain, with inertial resistance factors of 100 m^{-1} in each coordinate direction.
- Species transport equations were activated for 11 species (H_2O , Ar, CO_2 , O_2 , CH_4 , CO, COS, H_2 , H_2S , NH_3 , C<s>, S<s>, SiO_2 , N_2), without reaction, along with a diffusion energy source. The species C<s>, S<s>, and SiO_2 were added to prepare the FLUENT® case for ROM construction and for interfacing with the flowsheet.
- The discrete particle model in FLUENT® is a stochastic Lagrangian particle model. Surface injection was used for the particles at the single inlet plane. A Rosin-Rammler particle size distribution was specified for the “slag” particles; the Rosin-Rammler values selected were identical to those utilized by James Spenik (Ref. 8) in his CFD case for the single-stage, downflow, entrained gasifier (K9.cas/data). A total of 7 particle diameters were specified, along with 20 stochastic tries. Particle-radiation coupling was activated. A proprietary user-defined function (UDF) was used to calculate the particle absorption and scattering efficiencies. The particle density was calculated to be approximately 175 kg/m^3 , based on (a) adherence to the standard FLUENT® “constant-particle-diameter” and “varying-particle-density” model, (b) the specified slag composition of 11.66% carbon and 88.34% ash, (c) a knowledge of the initial coal composition, and (d) an initial dry coal density of $1,420.74 \text{ kg/m}^3$. The Discrete Random Walk option was activated, but the Random Eddy Lifetime option was deactivated. The number of continuous phase iterations per DPM iteration was set equal to 50.
- The radiation submodel was the Discrete Ordinates radiation model, with an azimuthal and polar control angle pixelation of 3×3 (as recommended by the FLUENT® Users’ Documentation for 3-D sector problems involving symmetry boundary conditions). A single flow iteration was performed per radiation iteration.
- All of the platens and exterior walls are evaporative or sub-cooled sections; the RSC does not have a superheat section. When the computational domain included both sides of a radial platen, the platen was treated as an “infinitely-thin” surface and then slit (using the TUI command “slit-face-zones”) to create a separate heat transfer surface on each side of the wall. The convergence or “cone,” the dip tube, the longitudinal or side wall, and the platens were all given a specified overall heat transfer coefficient ($222 \text{ W/m}^2\text{-K}$), a specified backside (coolant) water/steam temperature (608.7056 K), and an emissivity (0.8). The overall heat transfer coefficients for the walls were set using a calibration procedure; the overall heat transfer coefficients of the various walls were iteratively adjusted until the desired overall wall absorption at the design condition was achieved. For the present computations, the overall heat transfer coefficient for all of the convective walls was prescribed to be uniform. The walls for the inlet tube and the roof were assigned adiabatic boundary conditions (i.e., heat flux = 0).
- The inlet boundary condition was prescribed through a mass-flow rate boundary specification. The outlet flow boundary condition was set as a pressure outlet.

Grids for both 45° and 22.5° sectors were constructed. In an effort to try to understand physical phenomena and convergence issues, both coarse and fine grids were prepared. The geometry and mesh for a 45° sector are shown in Figures 8 and 9.

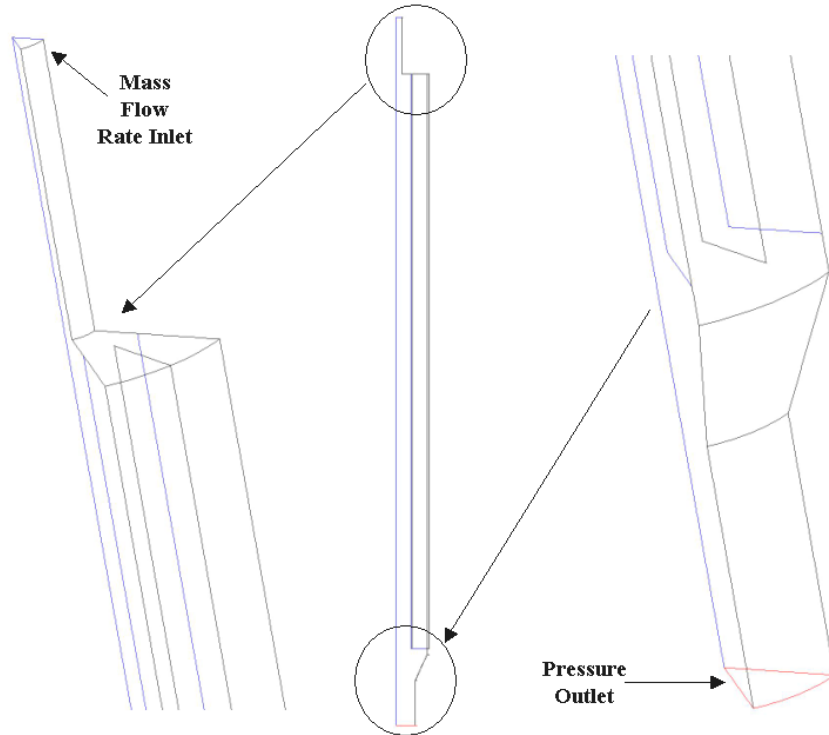


Figure 8: RSC 45°-Sector Geometry.

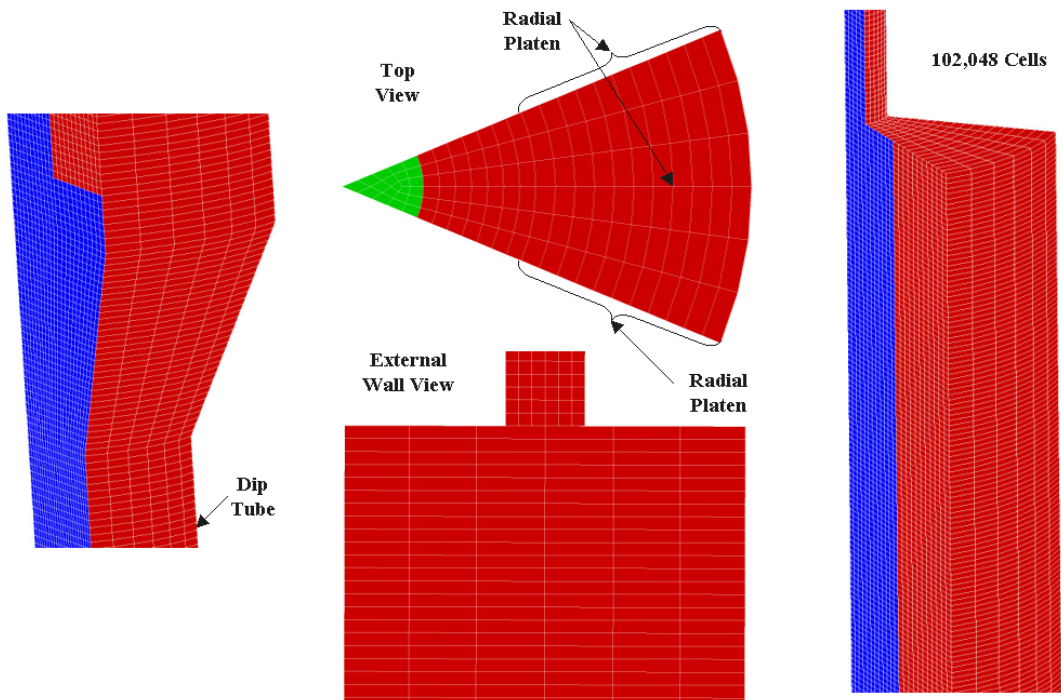


Figure 9: RSC 45°-Sector Grid with 102,000 Cells.

The geometry and mesh for a corresponding 22.5-degree sector case are shown in Figure 10.

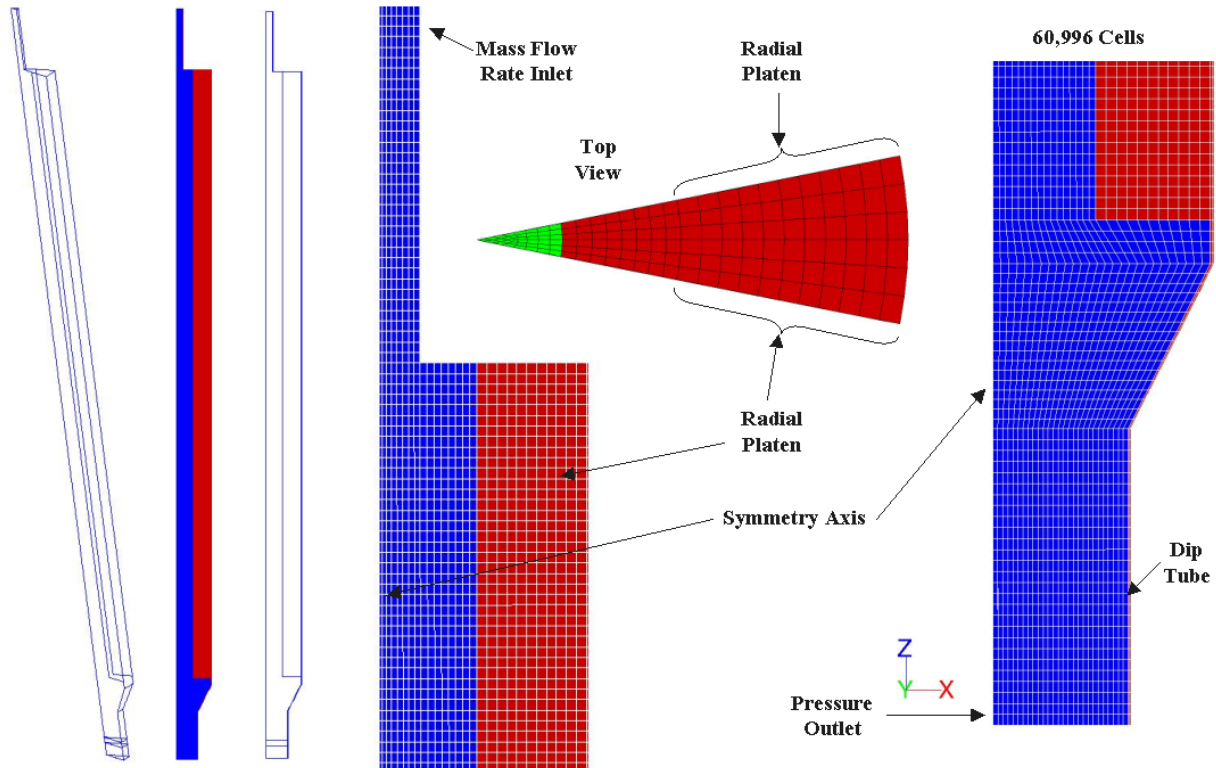


Figure 10: RSC 22.5°-Sector Grid with 60,000 Cells.

Early in the CFD runs, it was found that no matter which geometry configuration (22.5° sector or 45° sector) or grid size (e.g., 60,000 to 1,900,000 cells) is used for the computation, a centerline recirculation zone (RZ) near the symmetry axis is established, which induces low mass flow in the core region with corresponding “eddy-like” structures in the downstream core region. (The words “eddy-like structures” are indicative only, and may not properly reflect the physics of the phenomenon.) The corner recirculation zone patterns may also be complex, with more than one corner recirculation zone. An example is shown in Figure 11 for a 1.9-million cell case (without particles).

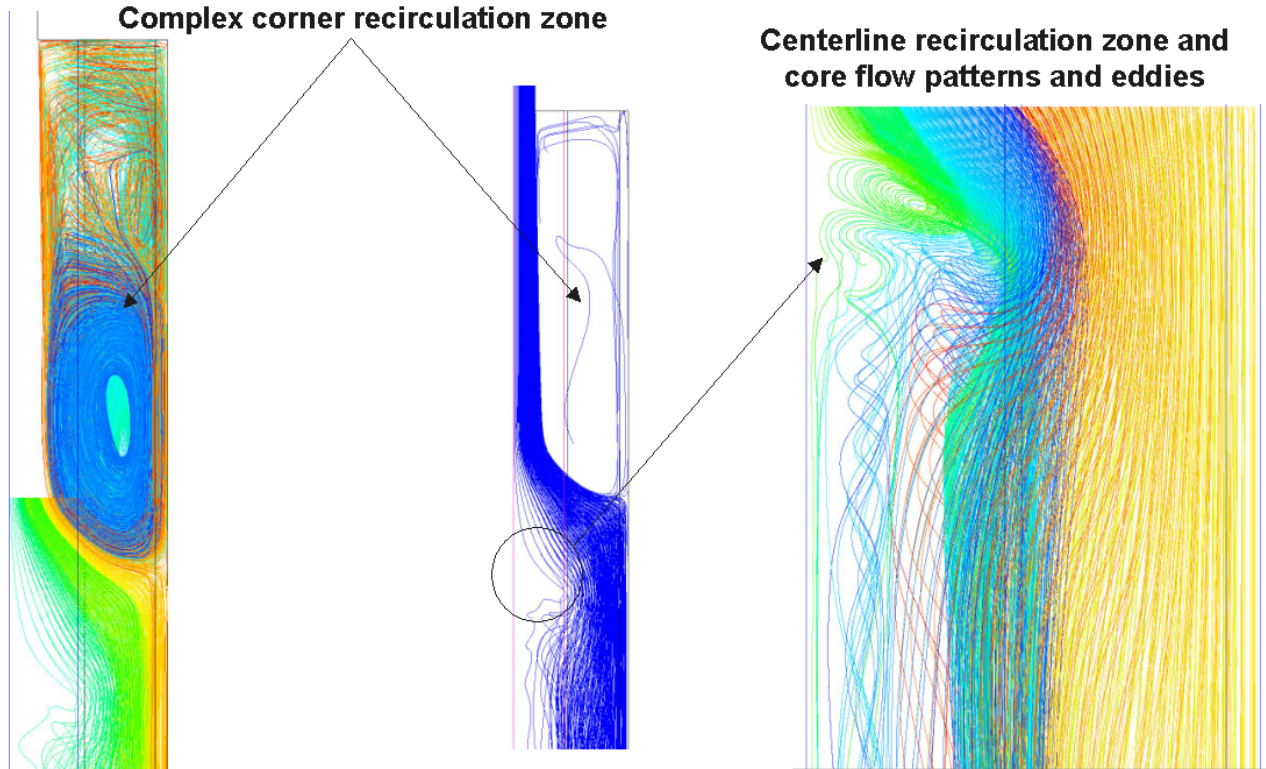


Figure 11: Pathlines Showing RZ in RSC 45°-Sector Grid (1.9 Million Cells (Gas Only)).

The corresponding longitudinal velocity contour plots are shown in Figure 12. The plots show the persistence of a low velocity core all the way to the exit convergence or cone. These computations also seem to be subject to azimuthal asymmetry (i.e., at certain elevations, the average velocity between a pair of platens may be different from that between an adjacent pair of platens).

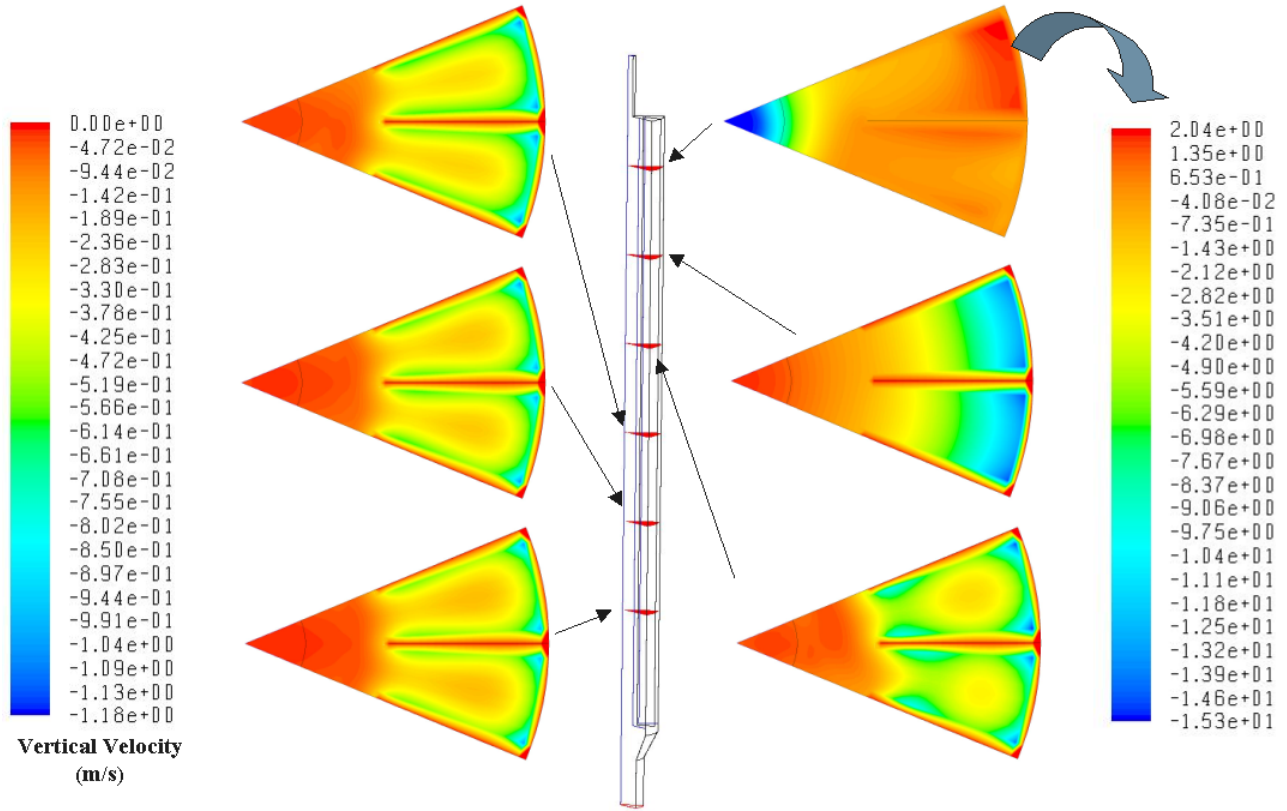


Figure 12: Vertical Velocity Contours for RSC 45° Sector Case (1.9 Million Cells, Gas Only).

All of the CFD runs were steady state. Generally speaking, there was an apparent “eddy-shedding” effect downstream of the recirculation zone, apparently inducing multiple sequential regions of reverse velocity. The introduction of particles seems to exacerbate the appearance of these multiple regions or patterns of reverse velocity. An example is shown in Figure 13. The flow fields appeared to indicate that transient, large-eddy simulations might be required to ultimately understand and resolve such phenomena. Indeed, it should be noted that in an APER GmbH brochure (Ref. 9), a CFD computation (by GE Energy) is shown that appears to rely on large eddy simulations or similar advanced computational technologies.

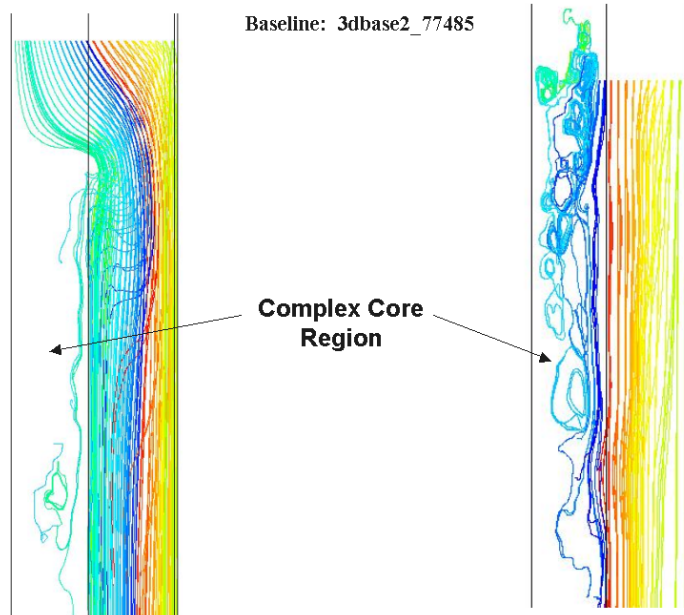


Figure 13: Pathlines for RSC 22.5° Sector Case (60,000 Cells, With Particles).

Since correspondence with APER GmbH indicated that a centerline recirculation zone does not typically occur in their simulations, a number of different grids and submodel options were run to try and eliminate the recirculation zone and induce greater symmetry in the results, but all to no avail:

- Variations in grid refinement
- Enhanced accuracy, through e.g., 2nd order discretization, double precision, and modified multigrid settings
- Reduction of the Y+ variable near the external wall (from several thousand down to a few hundred), as suggested by ANSYS support personnel.
- Activation of the water-gas shift reaction.
- Change in the FLUENT® code version from V6.3.31 to V12.0.16 (although the latter version may yield slightly more symmetric results).

Since, in general, heat extraction induces an adverse pressure gradient, attempts were also made to custom modify the heat extraction profiles. However, this did not prove to be beneficial, and even a completely adiabatic case appeared to be on the verge of forming an incipient centerline RZ.

APER GmbH has firmly and justifiably maintained the stance that if any additional computational recommendations or conceptual design details (either geometrical or mechanical) were to be provided to help resolve the RZ issues, that the mandated Alstom deliverable to the DOE would potentially provide sensitive and proprietary information to Alstom's competitors. The only information that APER GmbH would volunteer is that, in their simulations, they do not see any reverse flow zones in the core.

The remaining time and budget on the task did not permit a sensitivity analysis on various geometric perturbations that might beneficially impact the flow field and remove the centreline RZ. Should the NETL desire to investigate the issue more fully, the NETL may wish to review patent applications and perform a sensitivity analysis of potential geometric features that might remove the RZ. For example, a geometric feature in a patent application (Ref. 7) that was not included in the present design includes a domed transition section that provides an initial divergence somewhat upstream of the rearward-facing step (see Figure 14).

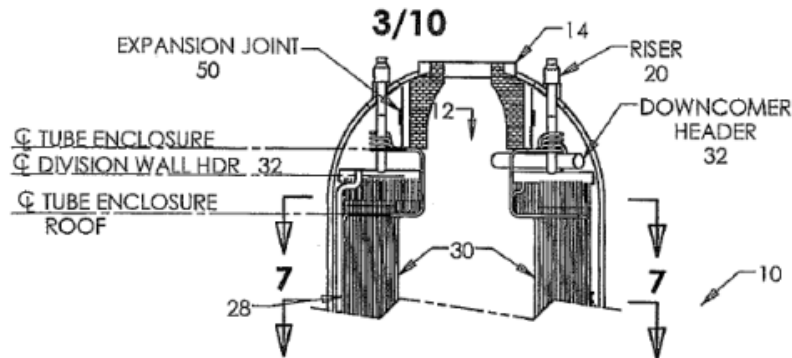


Figure 14: Domed Inlet Section for RSC (Ref. 7).

Consequently, rather than continue to hunt for a solution to eliminate the centerline RZ problem, it was decided to go ahead and develop the ROM based on the present computational status. Although the centerline RZ and the subsequent reverse flow zones downstream of the RZ tend to cause significant unsteadiness in the convergence behaviour of the cases, it was hoped that the unsteadiness would not be sufficient to jeopardize the construction of the ROM.

5.1.3 2-D FLUENT® Case Calibration and Setup

One of the main purposes of the present task is to develop a ROM that can take advantage of the Principal Component Analysis (PCA) technology available in APECS. This technology was investigated and showcased by Carnegie Mellon University under the present project, and subsequently implemented by ANSYS in the ROM Builder. The purpose of this technology is to perform a linear transformation on the flow field nodal information of the sampled CFD runs and to provide contour plots at any given operating or sample condition. However, at the present time, the PCA technology in APECS is only available for 2-D cases; extension of the capability to 3-D cases will not be available for perhaps another year. Accordingly, it was determined that, although 3-D cases of the RSC would be built to investigate and characterise the flow field, the ROM development effort would ultimately rely on and be limited to 2-D cases. It became necessary, in order to meet the terms of the task deliverable and provide 2-D ROM contour plots built with PCA technology, to find a way to construct a calibrated 2-D surrogate for the 3-D RSC model over the desired operating range.

The flowsheet provided by NETL (in IGCC005_GEWCO2.zip) controlled the flow rate to the gasifier and syngas cooler through the input NC (coal) flow rate for the stream “WET-COAL.” The oxygen and slurry flow rates, which are mixed with the coal, are linear functions of the WET-

COAL flow rate (as prescribed through calculator blocks). Therefore, it was decided, somewhat arbitrarily, that for the ROM to be effective, it should be functional over e.g., $\pm 30\%$ of the initial WET-COAL flow rate. The “baseline” WET-COAL flow rate corresponds to the initial WET-COAL flow rate, as found in the original flowsheet (as received from NETL). It may be justifiably argued that the baseline or reference WET-COAL flow rate should correspond to the final flow rate calculated for the WET-COAL stream at convergence, and so it probably should. However, in the original/baseline flowsheet, the final converged flow rate corresponds to -3.79% deviation from the initial value, and is thus sufficiently close that a distinction between the initial baseline and final converged values need not be made for purposes of ROM range performance.

To construct a 2-D (axisymmetric) CFD that would be equivalent to the 3-D case, care must be taken to extract the all of the energy corresponding to the total 3-D wall absorption. The main problem is that the 2-D case has a wall surface area that is much reduced from that of the 3-D case. Attempts to artificially augment the external wall absorption in the 2-D case to match that of the 3-D case, through a UDF, failed. To resolve this problem, 2-D (axisymmetric) CFD cases were constructed such that the platen walls were “transformed” into a circumferential porous medium region from which heat could be extracted through the specification of a volumetric energy sink term.

Five 3-D (22.5° sector) cases were run, corresponding to the baseline flow rate and to a -30% , -15% , $+15\%$, and $+30\%$ deviation from the baseline flow rate (see Table 2 and Figure 15). The outlet temperature was noted for each run. Then a series of 2-D runs were executed, corresponding to the same flow rates as the parent 3-D cases, and the porous medium volumetric sink term was changed until the desired outlet flow temperature (and total absorption) was achieved (see Figure 16).

Table 2: Stream Results for the RSC from the Converged Flowsheet

Case	3-D 22.5° Sector Case (60,996 Cells)			2-D Axisymmetric Case (10,004 Cells)		
	3-D Inlet Gas Mass Flow Rate	3-D Inlet Particle Mass Flow Rate	3-D Outlet Temp	2-D Inlet Gas Mass Flow Rate	2-D Inlet Particle Mass Flow Rate	2-D Energy Constant for Platen Porous Medium
	(kg/s)	(kg/s)	(K)	(kg/s)	(kg/s)	(W/m3)
-30%	2.949	0.1513825	759	47.180	2.422120	-151,000
-15%	3.581	0.1838210	812	57.290	2.941136	-177,000
Baseline	4.213	0.2162600	868	67.400	3.460160	-191,000
+15%	4.844	0.2486990	902	77.510	3.979184	-212,000
+30%	5.476	0.2811380	931	87.620	4.498208	-234,000

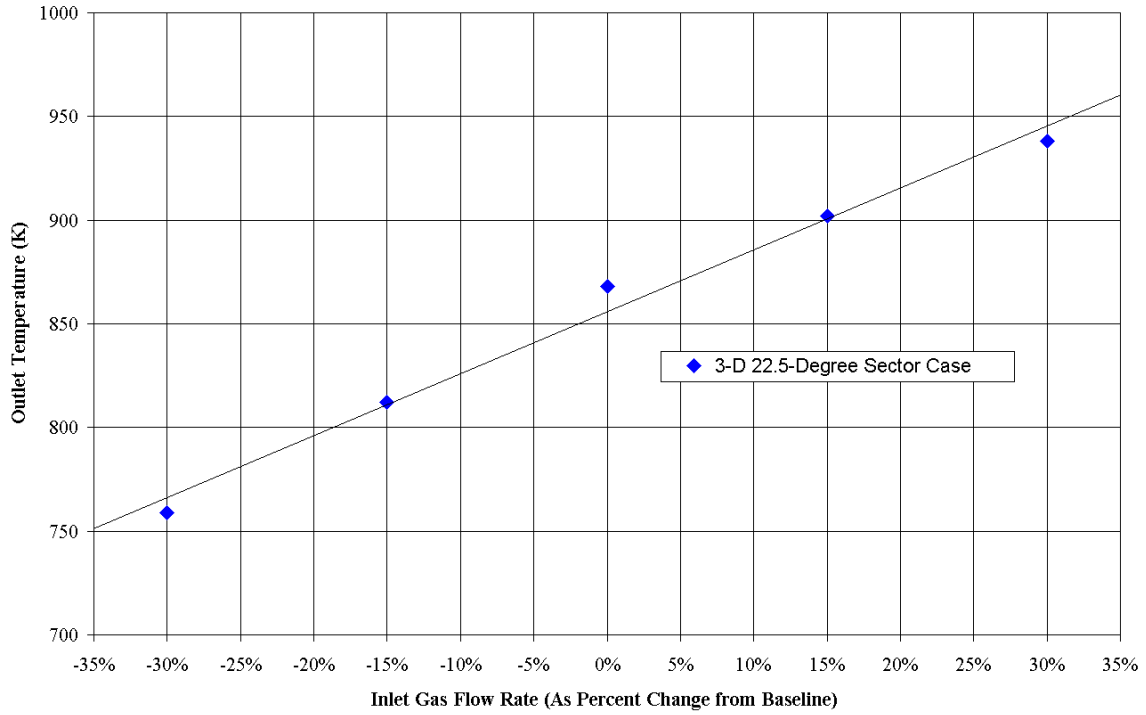


Figure 15: Outlet Temperature as a Function of Inlet Flow Rate for 3-D Case.

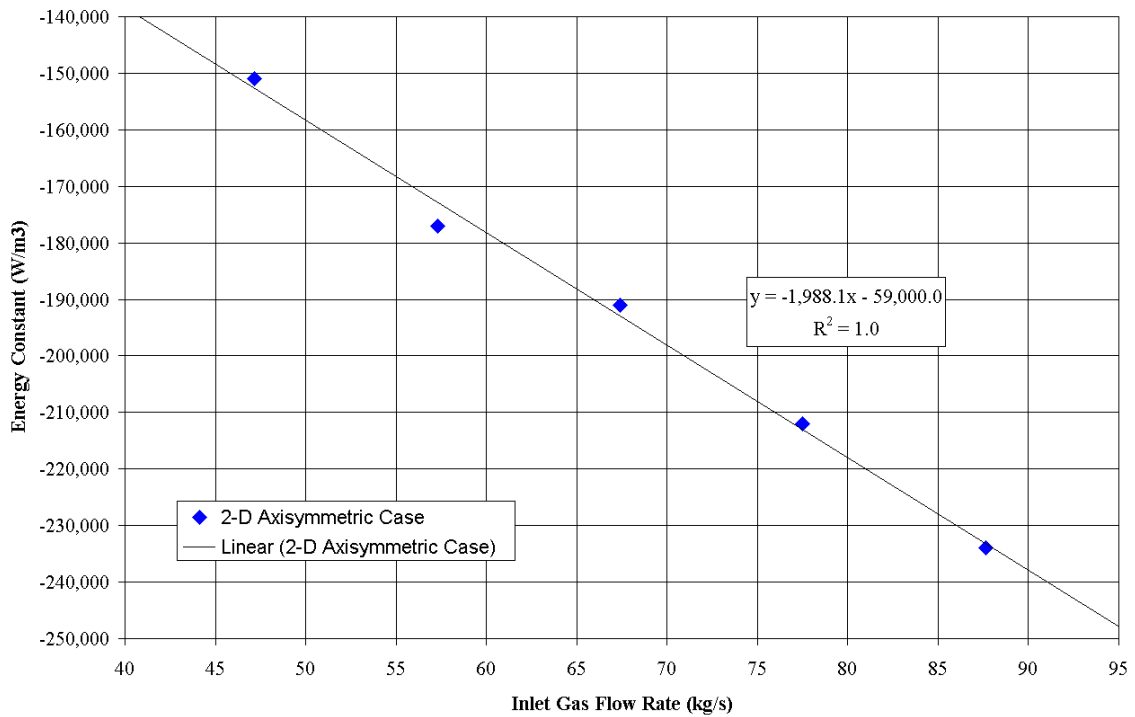


Figure 16: Volumetric Energy Sink Constant for Porous Medium in 2-D Case.

The correlation provided in the above figure was used in a DEFINE_SOURCE UDF to specify the volumetric sink term in the platen porous medium as a function of the gas mass flow rate at the inlet of the computational domain. Assuming a functionality other than linear may not be justified because the sampling size is small and the monitored 3-D (mass-weighted) outlet temperature may vary by ± 10 to ± 20 K during the convergence, as shown in Figures 17 through 19.

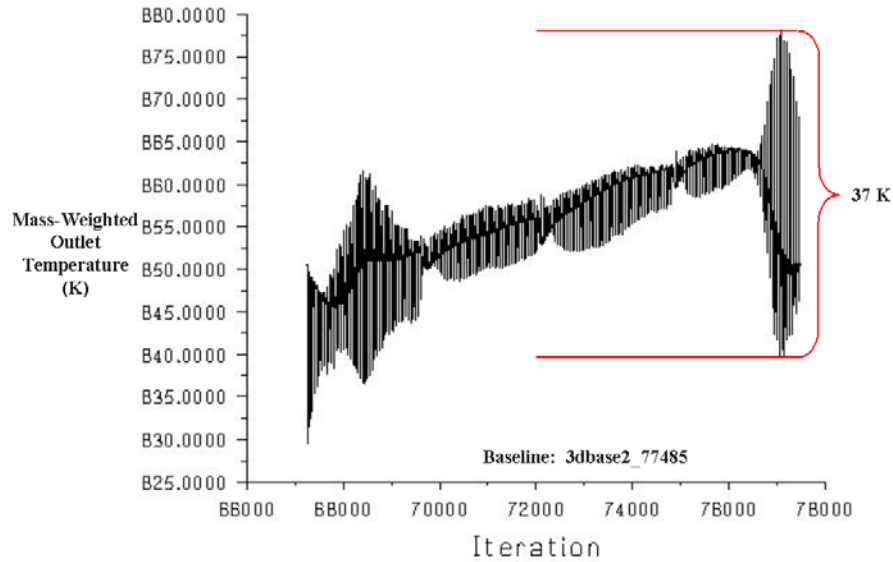


Figure 17: Monitored Outlet Temperature for the 3-D Baseline Case.

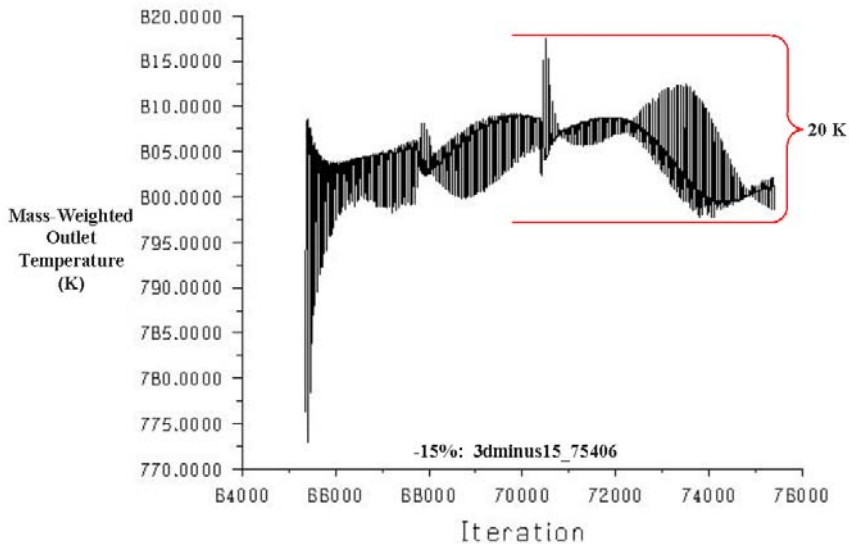


Figure 18: Monitored Outlet Temperature for the 3-D -15% Case.

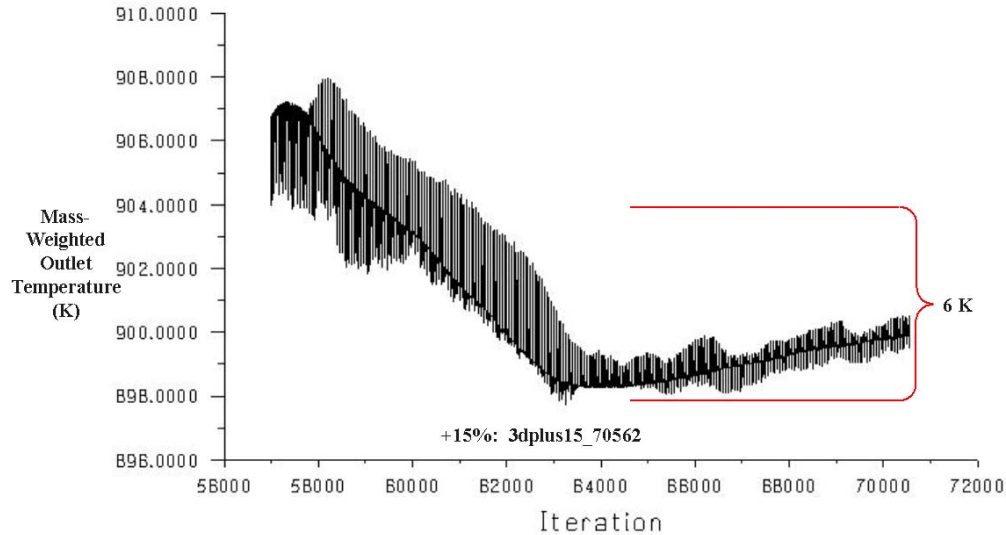


Figure 19: Monitored Outlet Temperature for the 3-D +15% Case.

Contour plots of the longitudinal velocities (clipped at zero velocity to visualize the reverse flow pattern) at a $y=0$ mid-plane for the five 3-D cases are shown in Figure 20. The reverse flow zones are more severe and pronounced at the lower flow rates (e.g., -30% and -15%) than at the higher flow rates. At the -30% case, one reverse flow zone is very long and persists almost to the exit. The corresponding temperatures are shown in Figure 21. The inlet temperatures are identical in each case. As the inlet flow rate decreases for the same wall heat transfer resistance and surface area, the outlet temperature correspondingly decreases. (The species concentrations are uniform throughout the computational domain.)

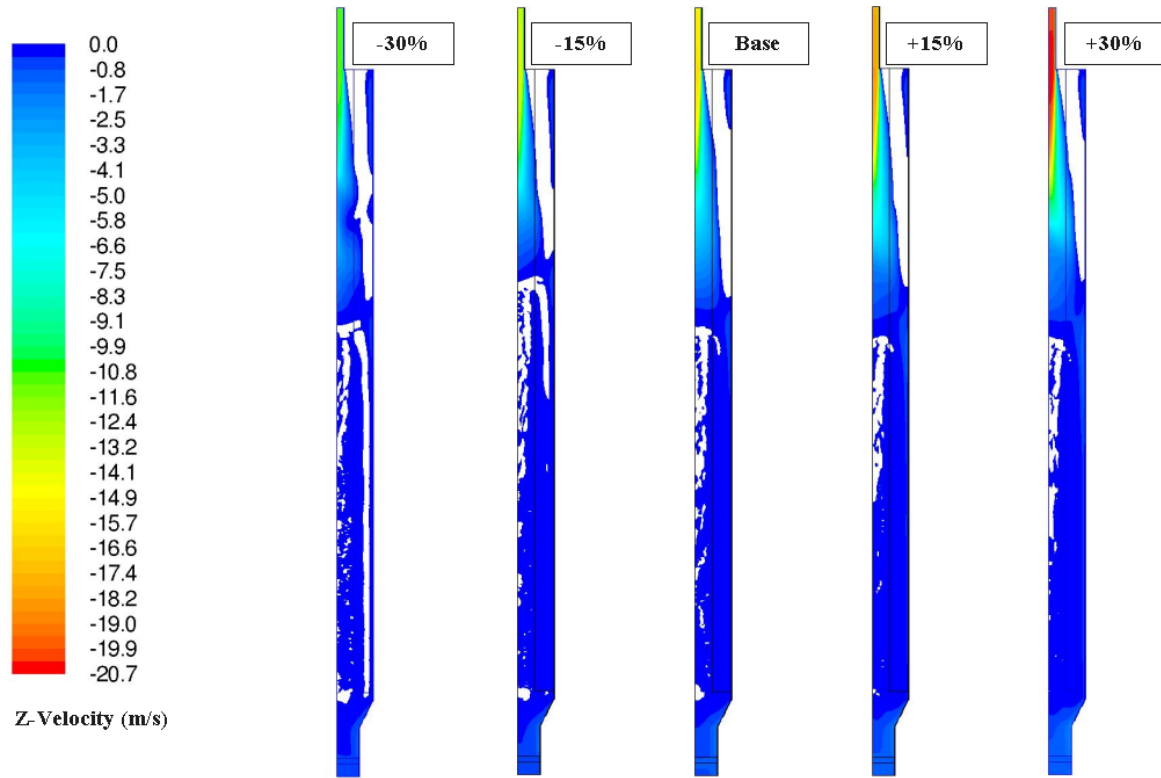


Figure 20: Z-Velocity Contour Plots for 3-D Cases.

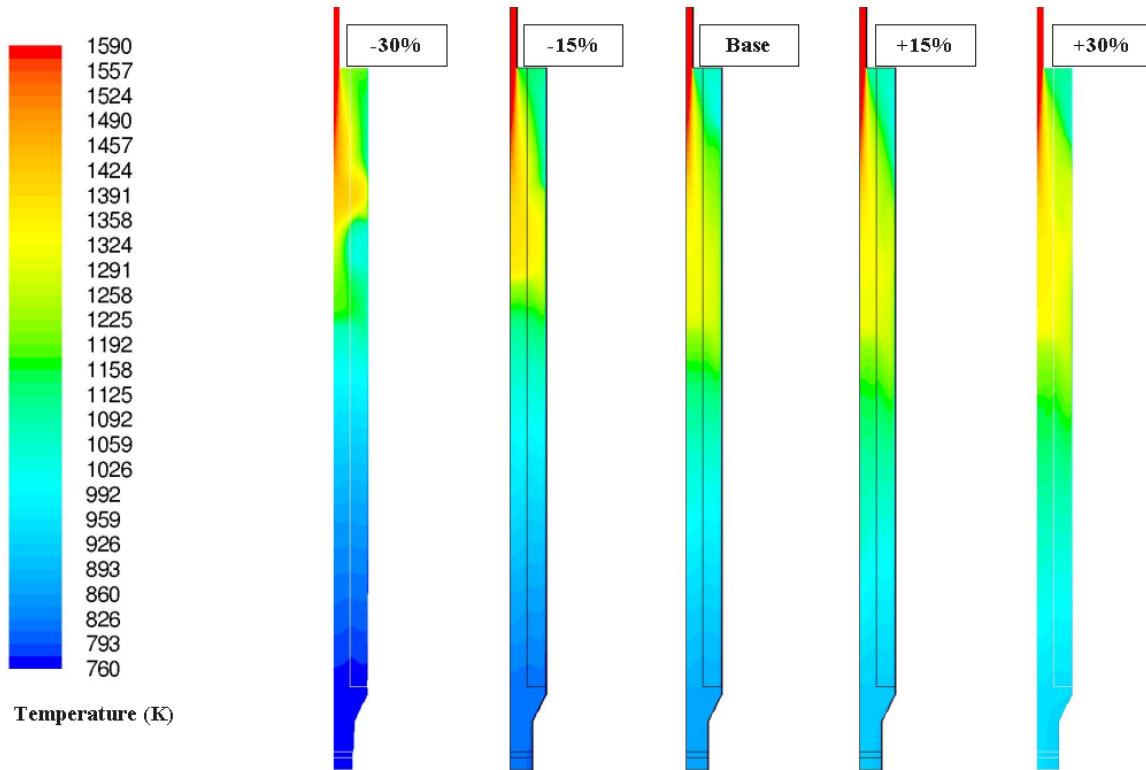


Figure 21: Temperature Contour Plots for 3-D Cases.

The clipped axial velocity contours and the corresponding temperature contours for the 2-D, axisymmetric case are shown in Figures 22 and 23. As with the 3-D cases, the number, size, and extent of the reverse flow zones in the core are more severe as the inlet flow rate decreases. In the 2-D cases, the length of the corner recirculation zone decreases with a decrease in the inlet mass flow rate; this trend is not as apparent in the 3-D cases, which shows a more complex flow structure.

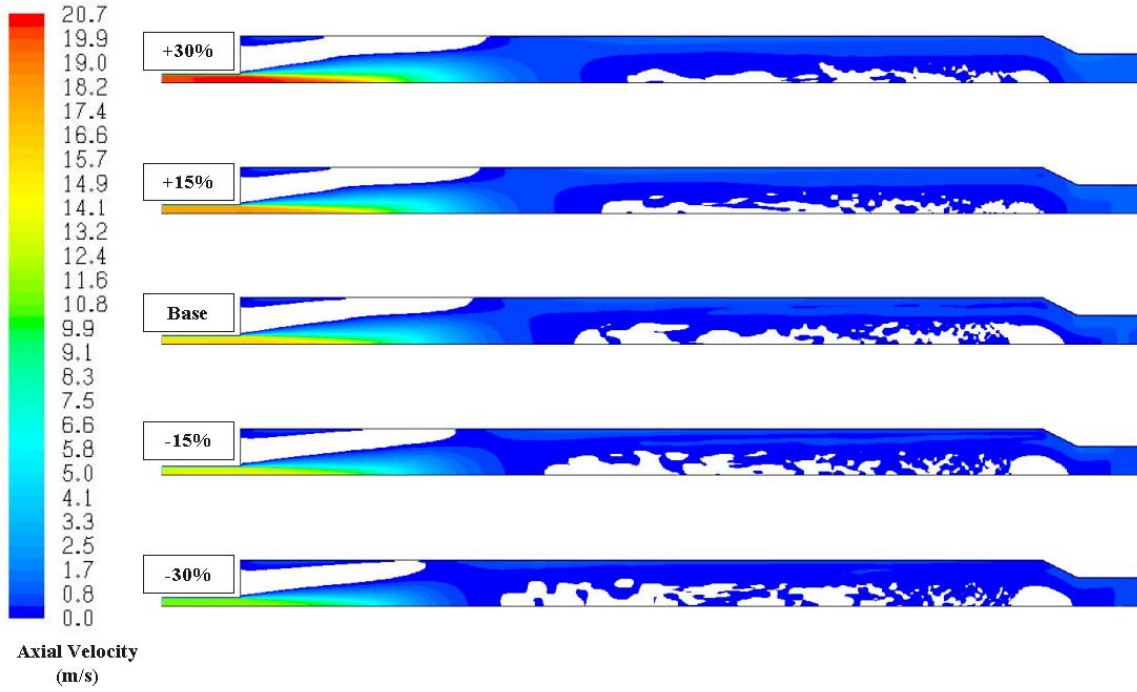


Figure 22: Axial Velocity Contour Plots for 2-D Cases.

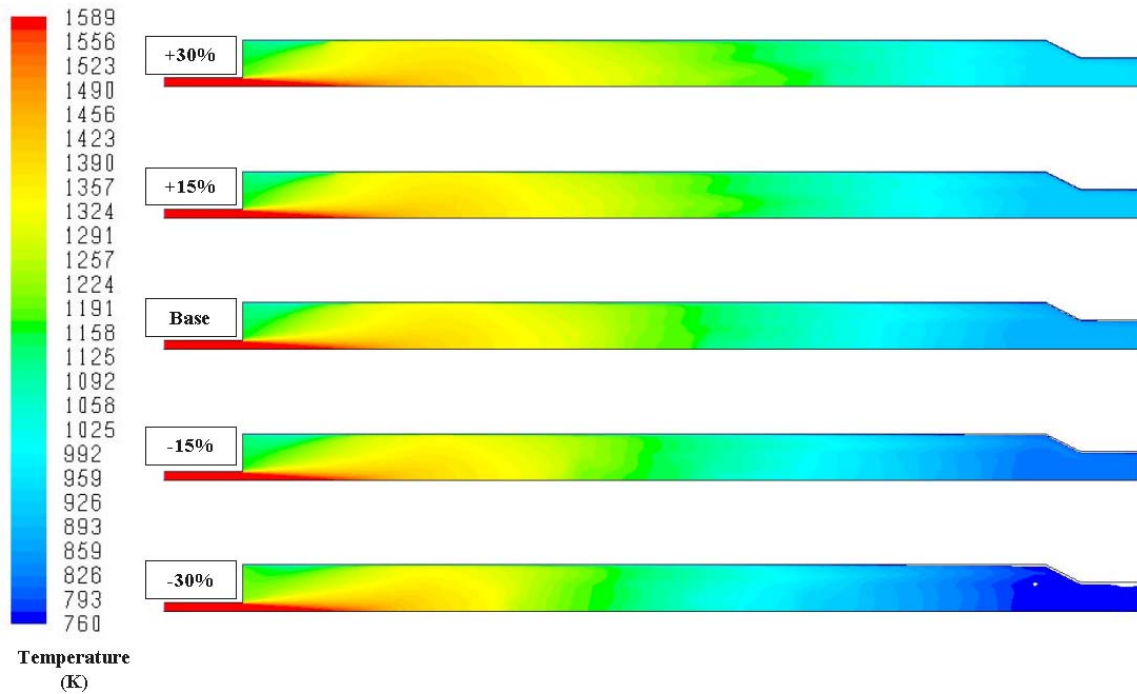


Figure 23: Temperature Contour Plots for 2-D Cases.

5.1.4 CO Parameters, Ports, and Connecting Streams

Since the ROM is built directly from multiple runs of the 2-D case, the 2-D CFD case must also be prepared with all of the necessary CO parameters and port connectivity information required for ROM instantiation.

To prepare the flowsheet to receive the RSC ROM block, additional feed and product streams must be added and RYIELD reactors must be added to exchange mass between the MIXED and NC substreams. This new arrangement, modified for the ROM block, is shown in Figure 24.

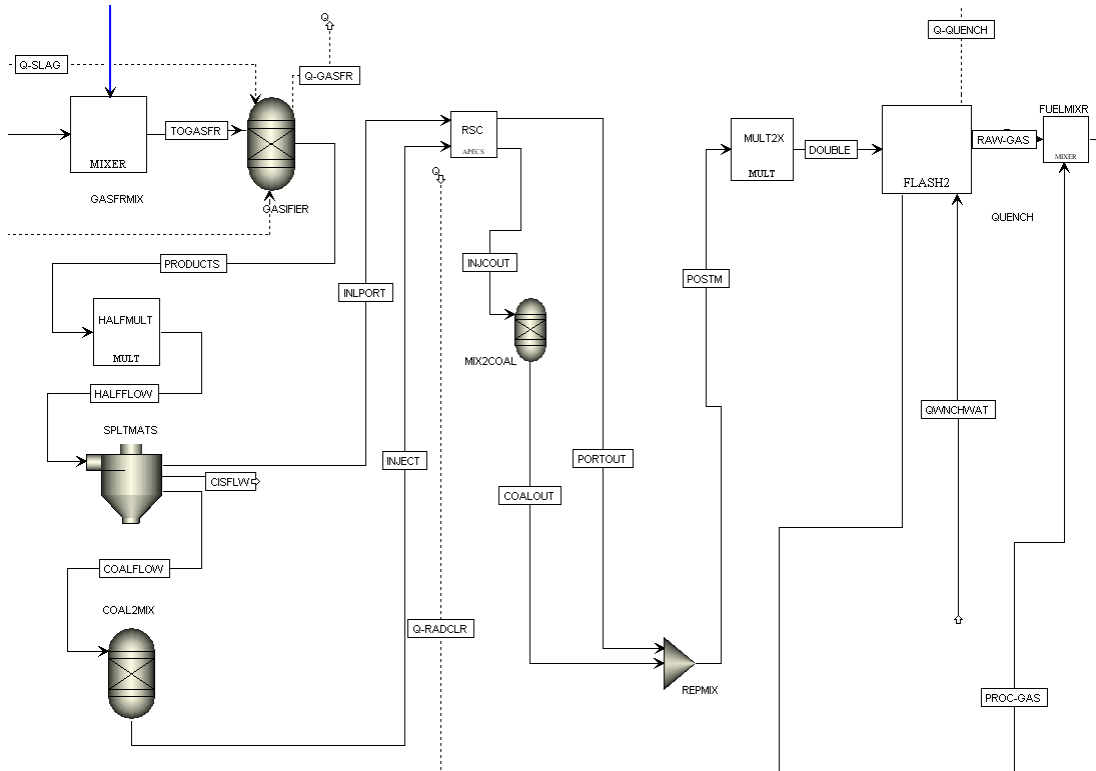


Figure 24: Preparation of the Flowsheet for Instantiation of the RSC ROM.

Since the flowsheet was constructed on a dual-train basis, and since the RSC ROM block represents a portion of a single RSC train, the PRODUCTS stream exhausting from the GASIFIER block must first be halved. This is accomplished in the block HALFMULT (Mult block). The flow exhausting from the RSC block must likewise be multiplied by 2 in block MULT2X (Mult block) before the exhaust stream can be hooked back up to the rest of the cycle.

The flowsheet provided by NETL (in IGCC005_GEWCO2.zip) has 3 substreams:

- an NC (non-conventional) substream, consisting of coal, ash, and slag.
- a CISOLID (conventional inert solids) substream, consisting of C (carbon) and S (sulfur).
- a MIXED substream consisting of 20 species: H₂O, AR, CO₂, O₂, N₂, O₂S, CH₄, CO, COS, H₂, H₂S, NH₃, SO₂, H₂SO₄, SO₃, C, S, S₂, S₆, and S₈ (where the species are

displayed with the names used by Aspen Plus). The species SIO2 was added to the species listing in the flowsheet in order to serve as a surrogate for ash/slag. Note that C and S are in both the CISOLID and the MIXED substreams.

It is a general requirement that all NC or CISOLID substreams must be converted into a MIXED stream before said stream reaches the FLUENT® (or FLUENT®-derived ROM) block. This is due in part to limitations within the prevailing CAPE-OPEN standard, as well as to limitations in the implementation of the standard and solid substreams in Aspen Plus. In like manner, the streams exhausting from the FLUENT® (or FLUENT®-derived ROM) block are mandated to be MIXED streams only, and must be converted to NC or CISOLID streams, if such streams are desired downstream of the FLUENT® block.

In block SPLTMATS (a SSplit block), the MIXED substream is given to stream INLPORT, the CISOLID substream is allocated to stream CISFLW, and the NC substream (which is 100% slag at this juncture) is given to the stream COALFLOW. The NC slag is passed through an RYIELD reactor (non-stoichiometric reactor based on known yield distribution) called COAL2MIX, and the slag is turned into a MIXED substream, consisting wholly of SIO2 (i.e., surrogate for ash). However, the actual substream composition would not matter because only the total mass flow rate and the temperature are passed on to the FLUENT® Lagrangian particle injections through the DPM physical model ports. The resulting converted MIXED substream is exhausted into stream INJECT and routed to the RSC block.

At the physical model outlet port (SLAG-OUTLET), a MIXED substream consisting of SIO2 (as surrogate for slag) is passed to the connecting material stream INJCOUT. Again an RYIELD reactor (MIX2COAL) is used to convert the MIXED substream component (SIO2) into the NC component slag, and the resulting NC substream is given to the material stream COALOUT. As mentioned previously, in the present flowsheet the NC slag contains 88.34% ash and 11.66% carbon. The MIXED substream in material stream PORTOUT is mixed with the NC slag in stream COALOUT in a mixer block. The mass flow rate of the resulting stream (POSTM) is then doubled, and the mixture is reconnected to the remainder of the dual-train cycle.

The various ports and stream connections for the FLUENT®-derived ROM block, all of which are associated with a single domain (Domain-1), are itemized as follows:

Table 3: FLUENT® Block Ports and Connecting Streams

FLUENT® ROM Computational Domain Port	Inlet or Outlet Port	Description of Port	Type of Port	Corresponding Aspen Plus® Stream Name
INLET	inlet	inlet from gasifier exhaust	inlet port	INLPORT
INJECTION-0	inlet	slag DPM injection	DPP	INJECT
OUTLET	outlet	outlet for gas flow	outlet port	PORTOUT
SLAG-OUTLET	outlet	outlet for DPM injections	DPP	INJCOUT
DPP =	discrete phase port			

The CAPE-OPEN variables for the DPM physical model outlet ports (SLAG-OUTLET), as defined and stored within the FLUENT® case file, along with current (or default) values (SI units), are given below:

```
(CAPE-OPEN/dpm/slag-outlet/flow-rate 67.4)
(CAPE-OPEN/dpm/slag-outlet/temperature 868)
(CAPE-OPEN/dpm/slag-outlet/mass-fracs (0 0 0 0 0 0 0 0 0 0 0 0 0 1 0))
```

The series of mass fraction values are those corresponding to the sequence of 14 species listed in the FLUENT® case under the Materials → Species panel (see the left-hand column of Table 4). Other than the mass fractions, only the mass flow rate and temperature are required to complete the definition of the FLUENT® block physical model outlet port.

The species listed in both the FLUENT® case and the Aspen Plus® flowsheet, as well as their respective mappings to each other, are provided in Table 4.

Table 4: Species Listing and Mapping

	Species in the Materials Listing in FLUENT®	FLUENT® Species Exposed to the PME	Flowsheet Species
(1)	h2o	h2o	H2O
(2)	ar	ar	AR
(3)	co2	co2	CO2
(4)	o2	o2	O2
(5)	ch4	ch4	CH4
(6)	co	co	CO
(7)	cos	cos	COS
(8)	h2	h2	H2
(9)	h2s	h2s	H2S
(10)	nh3	nh3	NH3
(11)	csolid	csolid	C
(12)	ssolid	ssolid	S
(13)	sio2	sio2	SIO2
(14)	n2	n2	N2
(15)	---	Zero Flow Rate Species No. 1	H2SO4
(16)	---	Zero Flow Rate Species No. 2	S2
(17)	---	Zero Flow Rate Species No. 3	S6
(18)	---	Zero Flow Rate Species No. 4	S8
(19)	---	Zero Flow Rate Species No. 5	O2S
(20)	---	Zero Flow Rate Species No. 6	SO2
(21)	---	Zero Flow Rate Species No. 7	SO3

The CAPE-OPEN variables for the DPM physical model port is set in the “DEFINE_DPM_BC(bc_outlet, p, t, f, f_normal, dim)” UDF (in file pmsink.c) through the lines:

```
sprintf(slag_mass_fracs, "(0 0 0 0 0 0 0 0 0 0 0 0 0 %f 0)", x_ash);
RP_Set_String("CAPE-OPEN/dpm/slag-outlet/mass-fracs", slag_mass_fracs);
```

```
RP_Set_Real("CAPE-OPEN/dpm/slag-outlet/flow-rate", part_outlet_mass_flux);
RP_Set_Real("CAPE-OPEN/dpm/slag-outlet/temperature", part_outlet_temperature);
```

It should be noted that the above UDF for the “bc_outlet” has not been properly parallelized, which means that the FLUENT® case can only be run on a single processor within the ROM Builder environment. The CAPE-OPEN parameters set through the APECS Wizard are provided in Table 5.

Table 5: CAPE-OPEN Parameters As Implemented Through APECS Wizard

Parameters	Type	Description	Access Mode	Lower Bound	Upper Bound	Units	Default Value
inlet-flow-rate	real	mass flow rate of gas in inlet stream	read-only	1	200	kg/s	67.4
pm-energy-const	real	porous medium energy sink constant	read-only	-1E+06	-1	W/m ³	-190,000
energy-sink-pm	real	total energy sink associated with porous medium	read-only	1	1E+09	W	6.1E+07
heat-flux-ext	real	surface heat flux through external walls	read-only	1	1E+09	W	3.1E+07
wall-absorption	real	total effective wall absorption	read-only	1	1E+10	W	9.19E+07
calculate-user-journal-activate	integer	activates the user journal file	read-write	0	1	---	1
calculate-user-journal-name	option	name of user journal file	read-only	---	---	---	prep_resid.jou
maximum-iterations	integer	maximum number of FLUENT® iterations	read-write	1	10,000	---	2,000
model-views	option	views for post-processing	read-write	---	---	---	FLUENT2D

The calculate-user-journal-activate parameter is actually a Boolean, but a bug in Aspen Plus® has precluded such Booleans from being manipulated in a calculator block. Therefore, it is more convenient to treat the parameter as an integer switch (0=FALSE, 1=TRUE).

Five of the CAPE-OPEN parameter variables are set in the “DEFINE_ADJUST(mass_flow, d)” UDF (in file pmsink.c) through the lines:

```
RP_Set_Real("CAPE-OPEN/param/inlet-flow-rate", flow_inlet);
RP_Set_Real("CAPE-OPEN/param/pm-energy-const", svalue);
RP_Set_Real("CAPE-OPEN/param/energy-sink-pm", energy_sink);
RP_Set_Real("CAPE-OPEN/param/heat-flux-ext", heat_flux);
RP_Set_Real("CAPE-OPEN/param/wall-absorption", total_wall_absorp);
```

The wall absorption parameter (“wall-absorption”) is the total RSC absorption. It is the sum of the heat flux to all of the wall surfaces (“heat-flux-ext”) plus the energy absorbed in the platen porous medium section (“energy-sink-pm”). The energy absorbed in the platen porous medium section is

equal to the volumetric extraction rate (“pm-energy-const” in W/m³) specified as a sink term in the porous medium panel, multiplied by the volume of the porous medium section (provided as a constant in the UDF).

5.2 Computational ROM Results

5.2.1 ROM Construction

The 2-D, axisymmetric CFD case was used in the ROM Builder. It was run on a Windows PC with the following software:

- ANSYS EKM® V1.2
- FLUENT® V6.3.26
- ANSYS APECS V1.7.0.019
- COFluent.6.3.26 Wrapper V1.7.0.015

The FLUENT® case files were prepared and made CAPE-OPEN and APECS compatible by sequentially performing the steps outlined in the APECS User’s Manual (Ref. 10):

- Execute FLUENT® CAPE-OPEN Configuration Wizard
- Execute APECS Model Wizard
- Upload files to the ANSYS Engineering Knowledge Manager (EKM®) database
- Launch the APECS User Interface to build the ROM

The files uploaded to the EKM® database consist of the following:

axisym-dpm-2dbase-15540.cas
axisym-dpm-2dbase-15540.dat
dpm_Qprop_bitum.c
libudf/ntx86/2d/libudf.dll
pmsink.c
prep_resid.jou

The APECS software allows for a number of journal files, which are helpful if providing increased control to the run sequence. ANSYS has prepared a user journal file activated via the CAPE-OPEN parameters:

- calculate-user-journal-name
- calculate-user-journal-activate

For the present task, this user journal file, called “prep_resid.jou,” was used to help run a minimum number of iterations by:

- (a) lowering one or more of the residual tolerances
- (b) running a specified number of iterations
- (c) and then putting the residual tolerances back where they were originally

The APECS calculate command is executed after the journal file commands are completed. If the residuals are less than the stipulated thresholds, then the case is deemed converged; otherwise, it begins to calculate the number of iterations dictated by APECS. While any need for this journal file

may be obviated by judicious selection of the residual thresholds, it is sometimes difficult to tell when a case is converged merely by looking at the residuals. (Certain monitors, coupled with residuals are usually much more helpful.) Furthermore, when performing a co-simulation, it is difficult for the user to know what the ideal convergence residual levels should be, unless the user has monitored the residuals for a large number of cases. Since such monitors do not seem to be presently available when the CFD block is executed remotely over a LAN, and since it is difficult to determine the proper convergence levels for the residuals, a good compromise is to utilize a user-journal file which forces a user-specified minimum number of iterations.

```
; prep_resid.jou
;
;
epsilon    do      h2o    ar      co2    o2      ch4    co      cos    h2      h2s
nh3
;
/solve/monitors/residual/convergence-criteria 1.e-09 2.e-04 8.e-05 1.e-06 5.e-04
1.e-03 8.0e-06 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08
6.e-08
/solve/iterate 1000
/solve/monitors/residual/convergence-criteria 8.e-04 2.e-04 8.e-05 1.e-06 5.e-04
1.e-03 8.0e-06 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08 6.e-08
6.e-08
```

To build the ROM, the APECS User Interface was launched (by entering the command “apecs”). In the “Configure ROM” panel, “General” tab, variables were selected with which to do the PCA transformation:

- Axial Velocity
- Radial Velocity
- Velocity Magnitude
- Static Temperature
- Mass Fraction of CO
- Mass Fraction of CO2
- Mass Fraction of H2
- Mass Fraction of H2O

It should be noted that selection of the species mass fractions was regarded only as a “check,” since the species fields are uniform for all operating conditions.

Under the “Regression ROM” tab, the following options were selected:

- Order of Regression = 1
- Save Every Instance = Yes (required for PCA)
- Mass Balance Tolerance = 0.001

ROM inputs were supplied for the ports INJECTION-0 and INLET, as well as for the CAPE-OPEN parameters. For the inlet ports, the only inlet quantities that varied over a range were the gas and particle flow rates; the relative gas and particle flow rates were such that the gas-particle mass ratio always maintained a constant value of approximately 19.48. The inlet temperature and species mass fractions also remained constant. Therefore, it is as if there is only parameter being varied, with all

other quantities linearly related to it. A condensed version of the variable range table for the ROM construction is provided below:

Table 6: ROM Variable Range

Temperature (K)	1588.71													
Pressure (Pa)	5617158.53													
% Change From Base		-30	-25	-20	-15	-10	-5	Base	5	10	15	20	25	30
Gas Flow Rate (kg/s)		47.180	50.550	53.920	57.290	60.660	64.030	67.400	70.770	74.140	77.510	80.880	84.250	87.620
Species	Mass Frac	Species Flow Rates (kg/s)												
H2O	1.261E-01	5.951	6.376	6.801	7.226	7.651	8.076	8.501	8.926	9.352	9.777	10.202	10.627	11.052
AR	1.555E-02	0.734	0.786	0.839	0.891	0.943	0.996	1.048	1.101	1.153	1.205	1.258	1.310	1.363
CO2	3.258E-01	15.371	16.469	17.567	18.665	19.763	20.861	21.959	23.057	24.155	25.253	26.351	27.449	28.547
O2	0.000E+00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
CH4	7.763E-04	0.037	0.039	0.042	0.044	0.047	0.050	0.052	0.055	0.058	0.060	0.063	0.065	0.068
CO	4.723E-01	22.285	23.877	25.469	27.060	28.652	30.244	31.836	33.428	35.019	36.611	38.203	39.795	41.387
COS	5.566E-04	0.026	0.028	0.030	0.032	0.034	0.036	0.038	0.039	0.041	0.043	0.045	0.047	0.049
H2	3.308E-02	1.561	1.672	1.784	1.895	2.007	2.118	2.230	2.341	2.453	2.564	2.676	2.787	2.899
H2S	1.214E-02	0.573	0.614	0.655	0.696	0.737	0.778	0.818	0.859	0.900	0.941	0.982	1.023	1.064
NH3	1.380E-03	0.065	0.070	0.074	0.079	0.084	0.088	0.093	0.098	0.102	0.107	0.112	0.116	0.121
C(s)	0.000E+00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
S(s)	0.000E+00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
SiO2	0.000E+00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
N2	1.223E-02	0.577	0.618	0.660	0.701	0.742	0.783	0.824	0.866	0.907	0.948	0.989	1.031	1.072
Gas Flow Rate (kg/s)		47.180	50.550	53.920	57.290	60.660	64.030	67.400	70.770	74.140	77.510	80.880	84.250	87.620
Particle Flow Rate (kg/s)		2.422	2.595	2.768	2.941	3.114	3.287	3.460	3.633	3.806	3.979	4.152	4.325	4.498
Gas to Particle Ratio		19.478	19.478	19.478	19.478	19.478	19.478	19.478	19.478	19.478	19.478	19.478	19.478	19.478

The table consists of 13 total simulations – 1 baseline and 12 user-specified simulation experiments, ranging from –30% to +30% of the baseline, at increments of 5%. Thirteen simulations is sufficient for a 1st-order regression, but not for any higher order regressions.

It should be noted that although the mass fraction is uniform at each simulation condition, the ROM Builder requires the user to input species flow rates, rather than mass fractions. The species mass flow rates scale with the overall mass flow rate for each simulation condition, but since, at a given simulation condition, each species concentration does not vary independently of the other species concentrations (i.e., the species mass fractions do not vary), the Design-of-Experiments option cannot be used. Each of the simulations must be input as a user-specified simulation. To individually input each user-specified simulation is obviously more tedious than being able to use the design-of-experiments option, but it is necessary at the present time. It is anticipated that future improvements to the ROM Builder will correct this deficiency. As a result of having to vary the mass fractions, rather than the total flow rate, the ROM Builder is varying 11 inputs. If the user was allowed to input the total mass flow rate, and preserve the mass fractions as non-varying, then only 2 of the inputs would vary – the total mass flow rate of the gas and the total mass flow rate of the particles.

Each of the 13 simulations ran an initial 1,000 iterations (as stipulated by prep_resid.jou), followed by an additional 2,000 potential iterations (as stipulated by the MAXIMUM_ITERATIONS parameter), presuming that the convergence criteria was not achieved, for a total possible maximum number of iterations of 3,000 per simulation. The ROM Builder completed the 13 simulations in approximately 22 hours elapsed time on a single processor of a PC.

A couple of the ROM Builder plot results are shown in Figures 25 and 26, corresponding to the temperature at Port OUTLET and the external wall heat flux, respectively. It should be noted that

only the first 10 out of the 13 runs are plotted (by default) in each Figure. Contour plot comparisons, arbitrarily selected for Simulation #5, are shown in Figures 27 through 30. The legends are automatically plotted for each Figure. The fact that the FLUENT® legend and the ROM legend have almost identical maxima and minima attest to their mutual similarity.

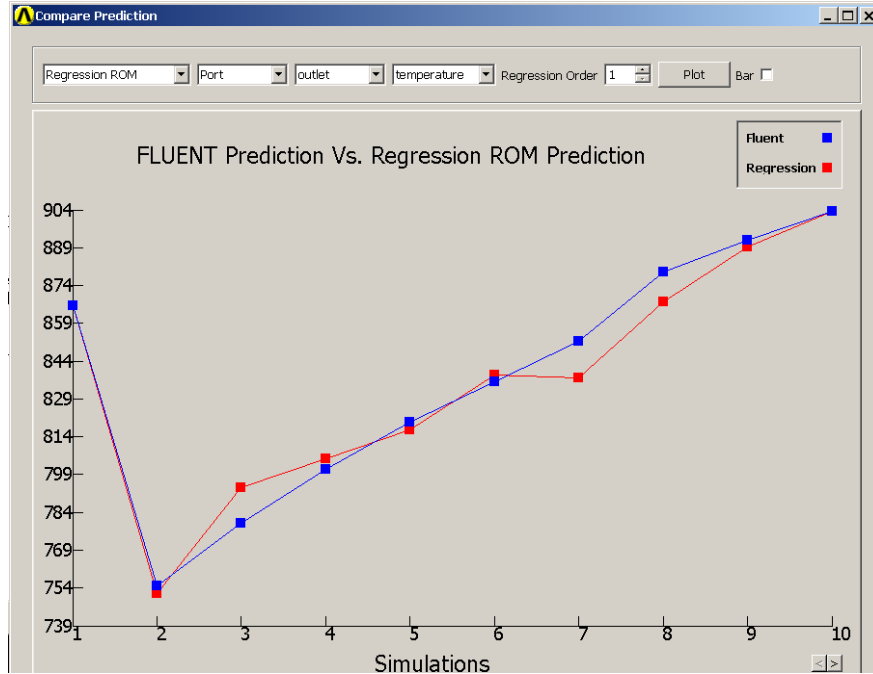


Figure 25: FLUENT® Versus ROM Predictions for Temperature at Port Outlet.

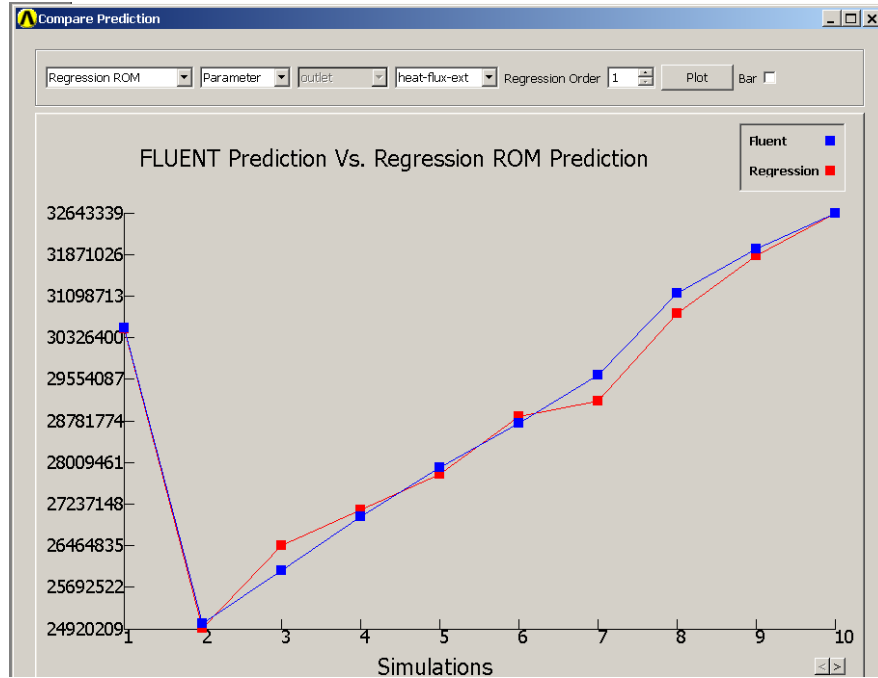


Figure 26: FLUENT® Versus ROM Predictions for External Wall Heat Flux.

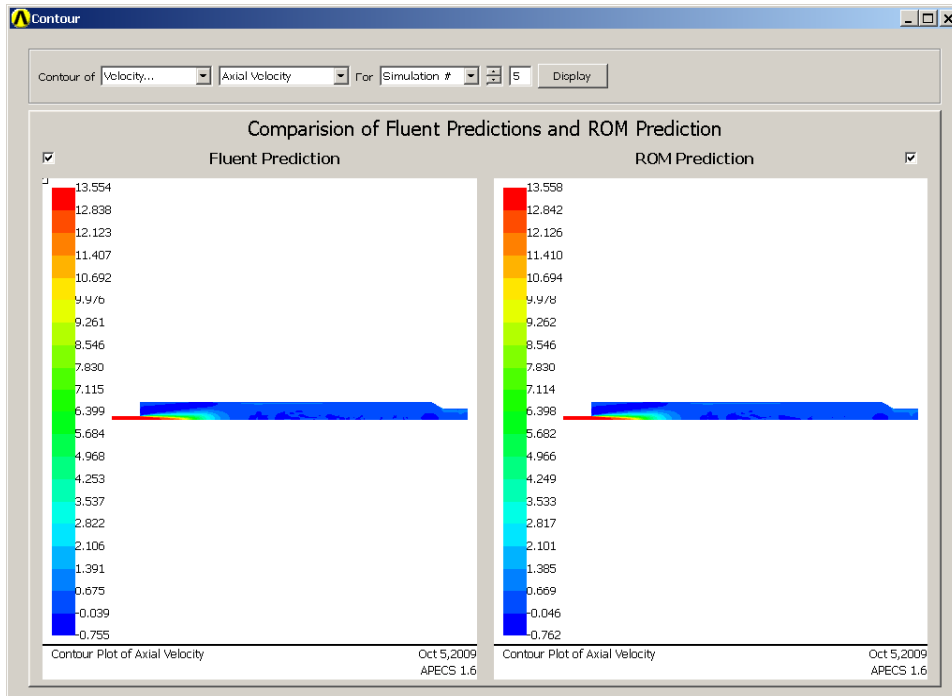


Figure 27: FLUENT® vs. ROM Contour Plots for Axial Velocity for Simulation #5.

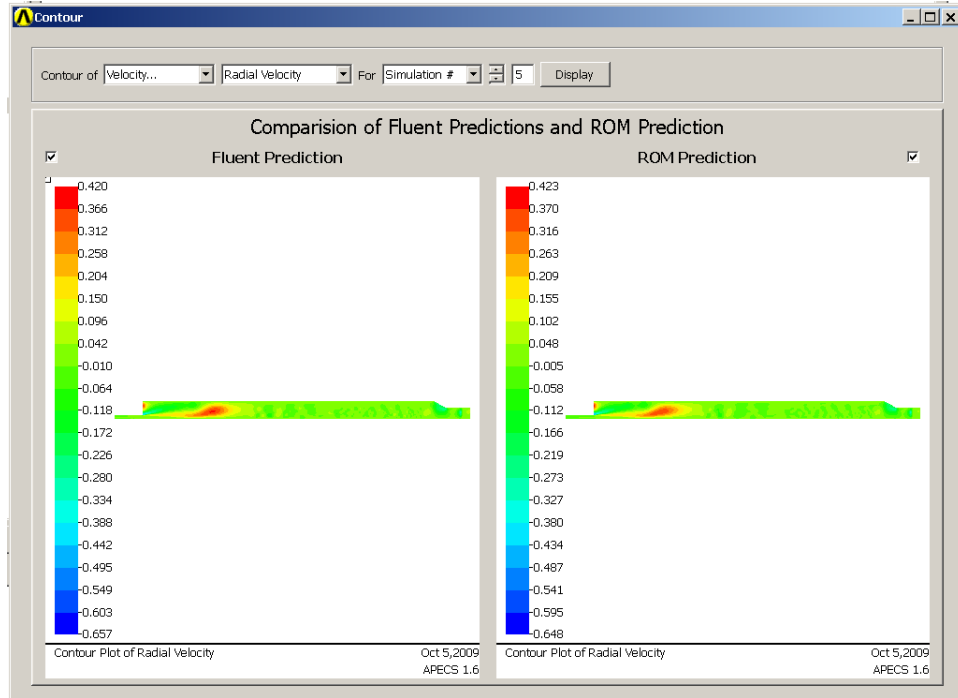


Figure 28: FLUENT® vs. ROM Contour Plots for Radial Velocity for Simulation #5.

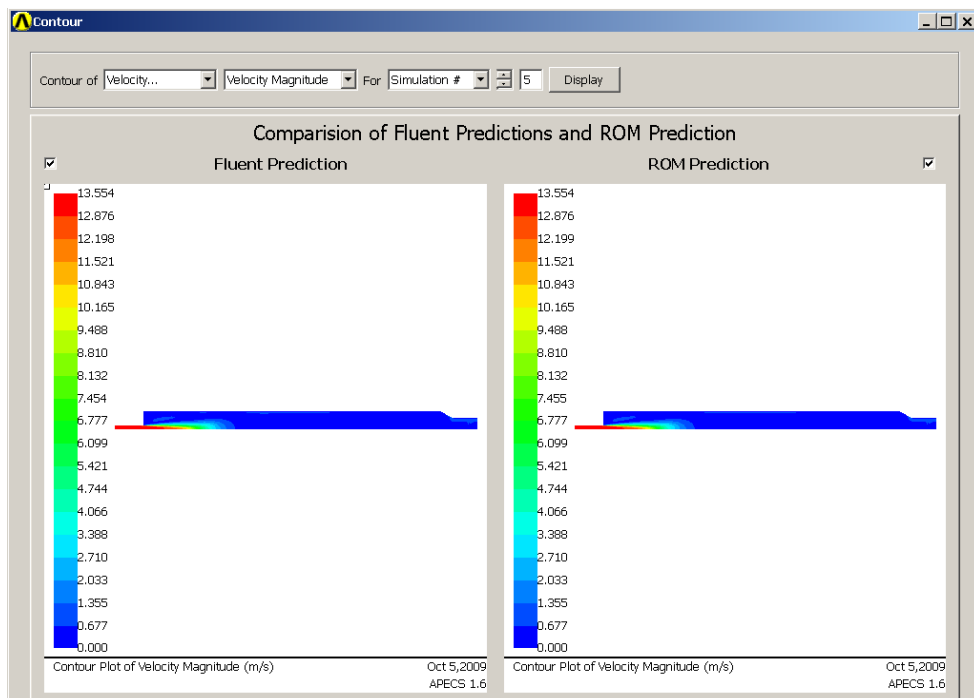


Figure 29: FLUENT® vs. ROM Contour Plots for Velocity Magnitude for Simulation #5.

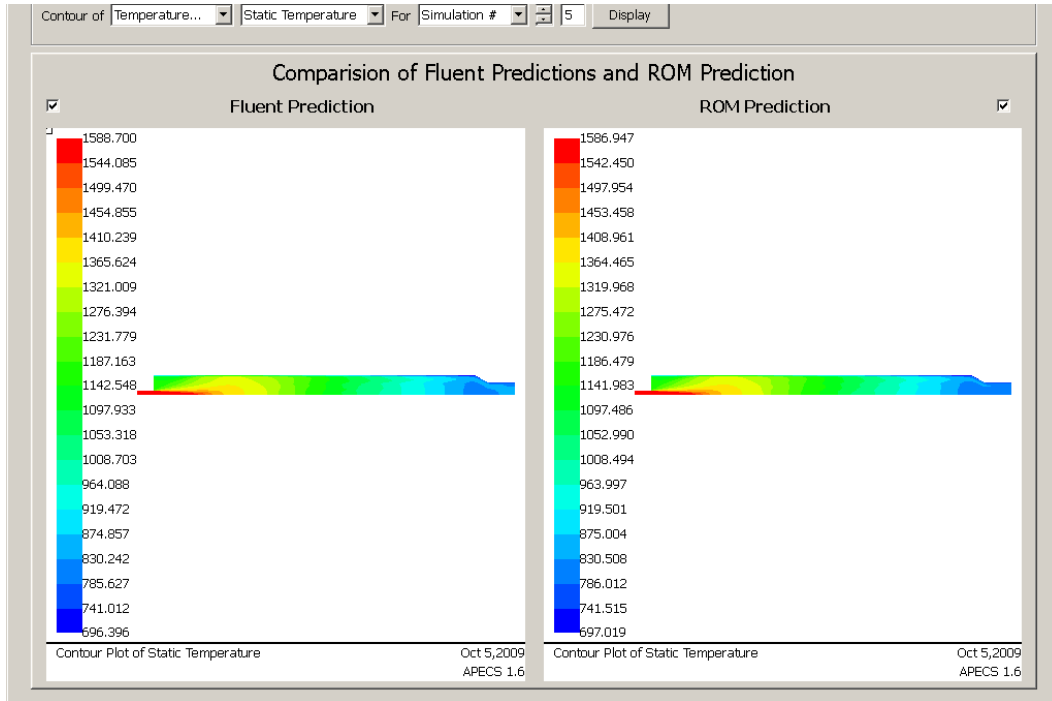


Figure 30: FLUENT® vs. ROM Contour Plots for Temperature for Simulation #5.

All 13 runs were used to make the ROM, as indicated in the following Figure, although none of the runs technically satisfied the simulation convergence criterion or the mass conservation criterion within the 3,000 iterations allocated to each run. However, given the difficulty of the flow field convergence and the use of the “prep_resid.jou” approach, this was not unexpected.

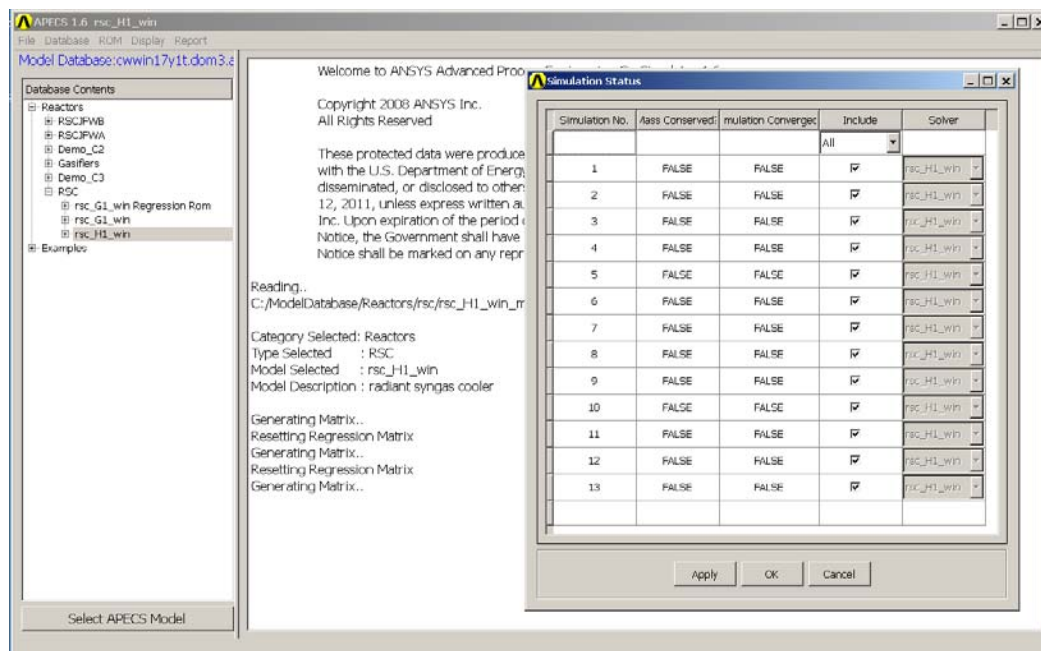


Figure 31: Simulation Status for the ROM Builder Runs.

5.2.2 Integrated Co-Simulation With ROM

The integrated co-simulation was run on a different PC than that of the ROM Builder. The software versions used in the co-simulation included the following:

- ANSYS EKM® V1.2
- Aspen Plus® V7.1 (with Cumulative Patch 1)
- ANSYS APECS V1.7.0.018
- COFluent 12.0.16 Wrapper V1.7.0.004

All of the co-simulation runs were performed with Aspen Plus® as the executive process modelling software. The Aspen Plus® flowsheet was run in the PC Windows environment and the ROM was run in the same environment.

As mentioned previously, the Aspen Plus® process model came from the file: “IGCC005_GEWCO2.zip,” which contained the Aspen Plus® template file “IGCC005.apr,” as well as 10 RDS proprietary .obj files, representing user blocks. The original flowsheet was created and run with Aspen Plus® Version 2004.1. However, it was found, after some experimentation, that the newly instantiated ROMs did not work in either V2004.1 or in V2006.5; the ROMs only worked properly in V7.1. Therefore, all of the co-simulation work was migrated to Aspen Plus® V7.1.

However, it was found that V7.1 presently exhibits the following CAPE-OPEN related problems:

- Aspen Plus® V7.1 does not persist the stream connections to the CAPE-OPEN block until the flowsheet has converged; only then can the user save out a working .apw file. This is problematic in the present task, because the flowsheet does not successfully converge in V7.1; consequently, the reinstatement procedure must be repeated every run. Although the original flowsheet did successfully execute in 2004.1, the 10 RDS proprietary .obj files, representing user blocks, cause the run to stop in V7.1 from internal errors: i.e., “Calculations terminated because of Aspen Plus Internal Errors.” When one gets this error, all of the results are purged and the CFD Viewer cannot be accessed, thus disallowing one from viewing the results of the PCA transformations. Even increasing the number of errors allowed under Setup → Simulation Options does not help. Fortunately, the flowsheet convergence strategy is divided into two main sequences – the first convergence sequence LOOP CONV-7, encompasses the RSC. When LOOP CONV-7 is completed, then LOOP CONV-31 is initiated. There is no additional outer loop that feeds back information back to LOOP CONV-7, so that LOOP CONV-7 is converged only once. This means that for purposes of the present task, the successful convergence of the entire flowsheet is not critical, and calculation of LOOP CONV-31 can be ignored. By inserting a Stop Point (after BLOCK DUMSEP and immediately before LOOP CONV-31 is initiated), the RSC ROM can be instantiated and rigorously tested within the flowsheet, and the CFD Viewer can be used to view the PCA results.
- Aspen Plus® V7.1 is flashing the inlet streams (KODE = 2 (i.e., T-P flash)) in order to facilitate making available additional properties in the inlet streams, so that the CAPE-OPEN unit operation could make use of said properties if necessary. For the present task, this inlet stream flashing has a relatively benign effect because the inlet gas stream is not in a 2-phase regime (but it could potentially be a serious problem if the inlet stream is in the 2-phase regime, such as would occur in a water/steam problem).
- Aspen Plus® V7.1 produces a severe mass imbalance across the APECS block:

```
* WARNING WHILE EXECUTING UNIT OPERATIONS BLOCK: "B1" (MODEL: "CAPE-
OPEN" )
(BALMAS.1)
BLOCK B1 IS NOT IN MASS BALANCE:
MASS INLET FLOW = 0.73653797E+02, MASS OUTLET FLOW = 0.20000000E+71
RELATIVE DIFFERENCE = 0.27154065E+69
CHECK USER SUBROUTINE
```

This mass imbalance is caused by the fact that the material stream outlet ports are erroneously assigned a CISOLID substream flow rate of 0.1E+36, even though no CISOLID enters or leaves the block, and even though the species or component flow rates (e.g., C(CISOLID) and S(CISOLID)) in the substream are identically zero. This bug/problem is partially overcome, at least in terms of preventing downstream imbalances, by putting in print statements (in a calculator block immediately downstream of the APECS block) for the outlet streams for the (a) total CISOLID, (b) C(CISOLID), and (c) S(CISOLID). When the print statements are added, then Aspen Plus® recognizes the discrepancy between the total CISOLID flow rate and the component flow rates and resets the CISOLID flow rate back to zero.

```
** ERROR WHILE EXECUTING CALCULATOR BLOCK: "C-PSTUSR" (ASTREM.2)
TOTAL FLOW SPECIFIED FOR STREAM INJCOUT IS 0.10000E+36.
```

BUT SUM OF COMPONENT FLOWS IS ZERO. TOTAL FLOW SET TO ZERO.

** ERROR WHILE EXECUTING CALCULATOR BLOCK: "C-PSTUSR" (ASTREM.2)
TOTAL FLOW SPECIFIED FOR STREAM PORTOUT IS 0.10000E+36.
BUT SUM OF COMPONENT FLOWS IS ZERO. TOTAL FLOW SET TO ZERO.

The instantiation procedure, in general terms, is given below:

- (1) Instantiate the APECS block and rename it to e.g., "B1" or "RSC"
- (2) Assign the block section = GLOBAL
- (3) Make all of the stream connections in accordance with Table 3.
- (4) Update the appropriate convergence blocks and calculator blocks to accommodate the RSC ROM block:

Convergence → Sequence → GLOBAL: Position Unit Operation RSC between C-PREUSR and C-PSTUSR:

```
SEQUENCE GLOBAL . . . . C-PREUSR B1 C-PSTUSR . . . .
```

```
CALCULATOR C-PSTUSR
```

```
DEFINE QBLOCK BLOCK-VAR BLOCK=B1 VARIABLE=WALL-ABSORPTION  
DEFINE HTFLUX BLOCK-VAR BLOCK=B1 VARIABLE=HEAT-FLUX-EXT  
DEFINE ESNKPM BLOCK-VAR BLOCK=B1 VARIABLE=ENERGY-SINK-PM  
DEFINE INFLOR BLOCK-VAR BLOCK=B1 VARIABLE=INLET-FLOW-RATE  
DEFINE ECONPM BLOCK-VAR BLOCK=B1 VARIABLE=PM-ENERGY-CONST  
EXECUTE AFTER BLOCK RSC
```

- (5) Complete the species mapping, as indicated by the last two columns in Table 4.
- (6) As mentioned previously, one must set a Stop Point after BLOCK DUMSEP in order to stop the calculations before LOOP CONV-31 is initiated, otherwise the RDS user blocks produce the following error: "Calculations terminated because of Aspen Plus Internal Errors." With implementation of the Stop Point, one can use the CFD Viewer and view the PCA results.
- (7) After running the case, one may double click on the block to bring up the APECS Unit Operation panel. One can see the PCA-generated plots by right clicking on the name of the solver and selecting CFD Viewer.

The ROM, with a regression order of 1, was first uploaded into EKM®. Subsequently, it was instantiated on the flowsheet and run with different initial coal flow rates (material stream WET-COAL) to see if the ROM performed in a robust manner over the desired range from -30% to +30% of the baseline values. The flowsheet converged for each initial flow condition without displaying any apparent convergence difficulty, and the number of ROM executions are found to be reasonable.

Table 7: Co-Simulation ROM Robustness Testing

Case	Initial NC Coal Flow Rate (for Stream WET-COAL) (lbm/hr)	Initial NC Coal Flow Rate (for Stream WET-COAL) (kg/s)	Number of ROM Executions Required for Convergence in Flowsheet
-30%	364,074.9	45.873	17
-20%	416,085.6	52.426	16
-10%	468,096.3	58.979	11
Baseline	520,107.0	65.532	12
+10%	572,117.7	72.086	12
+20%	624,128.4	78.639	12
+30%	676,139.1	85.192	13

Each of the above cases converged to approximately the same solution (within the convergence tolerances). The final converged co-simulation results, achieved with the instantiated ROM, compared to target values taken from the original converged flowsheet, are shown in the following Table:

Table 8: Comparison of Co-Simulation Results with Target Values

Case	Target Value from Original Converged Flowsheet	Results from Converged Flowsheet Using ROM	Error
Gas Flow Rate from Gasifier Exhaust (kg/s)	134.8	134.33	-0.35 %
Slag Flow Rate from Gasifier Exhaust (kg/s)	6.92	6.90	-0.35 %
RSC Outlet Temperature (K)	866.5	898.0	+5.15 %
Total RSC Absorption (W)	1.899E+08	1.841E+08	-2.0 %

The co-simulation RSC outlet temperature and absorption deviate from the target values by about +5% and -2%, respectively. The calculated temperature is higher than the target because the total wall absorption is underpredicted. The reason is probably due to the fact that the calibration between the 3-D and 2-D cases was not as good as it should have been. In addition, the porous medium absorption is calculated by simply using the porous medium sink term rather than by some other technique (e.g., enthalpy differences) that may help to provide a more rigorous estimate of the actual heat loss. A contributing factor, as shown in Figures 17 through 19, is that the solution can migrate and the outlet temperature can easily change by ± 10 to ± 20 K. It may require somewhat more care to calibrate a 2-D case with a 3-D case when both cases exhibit such complex (migratory or fluctuating) reverse flow zone patterns. Although the fidelity of the ROM may not be as good as the original FLUENT® case, and the calibration could be improved, the results are nevertheless considered to be acceptable, and constitute a demonstration of a regression/PCA-based ROM. The run time for the co-simulation with the ROM is, of course, negligible relative to that required for an instantiated FLUENT® case.

When the co-simulation was completed (at least down to the Stop Point), the CFD Viewer was accessed, and the plots produced by the PCA transformation were reviewed. The plots are shown in

Figures 32 through 35. (The species plots, though generated, are not noteworthy because they are essentially a constant value.) The plots shown are reminiscent of Figures 27 through 30, and do not contain any unexpected features.

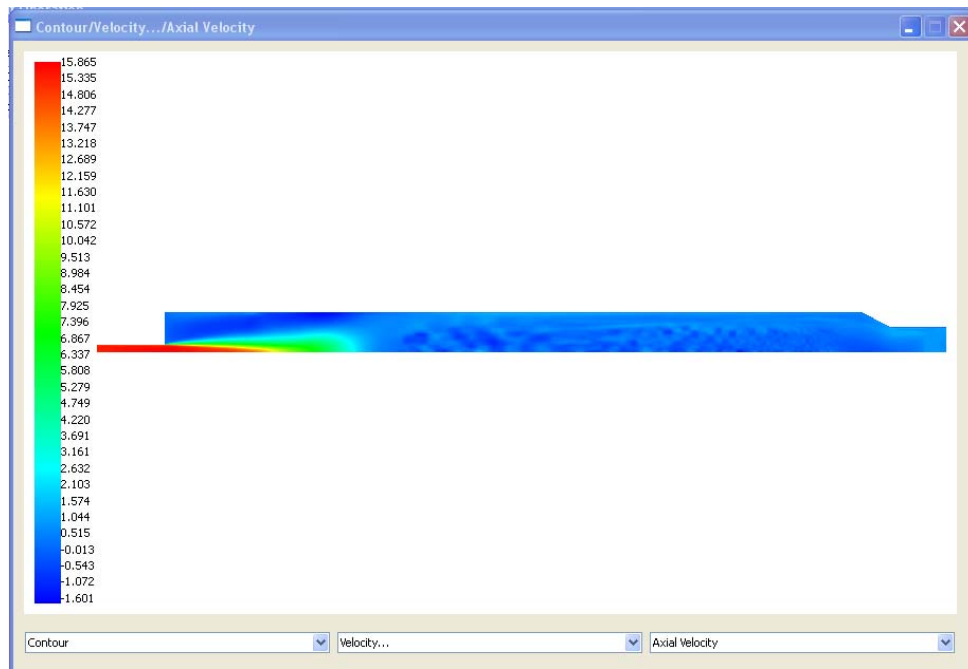


Figure 32: PCA Axial Velocity Results from the Co-Simulation.

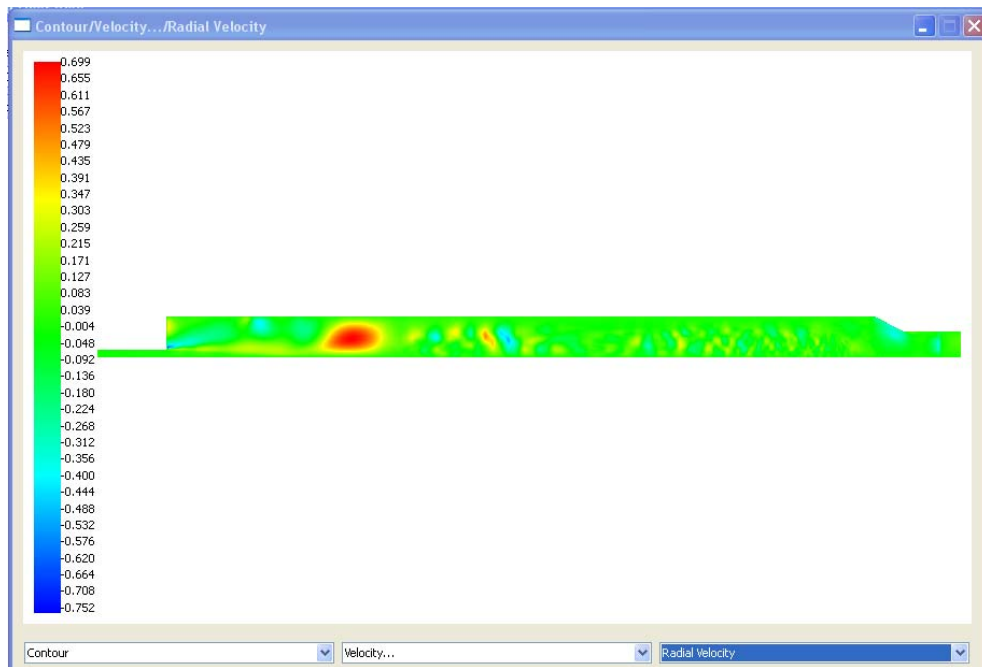


Figure 33: PCA Radial Velocity Results from the Co-Simulation.

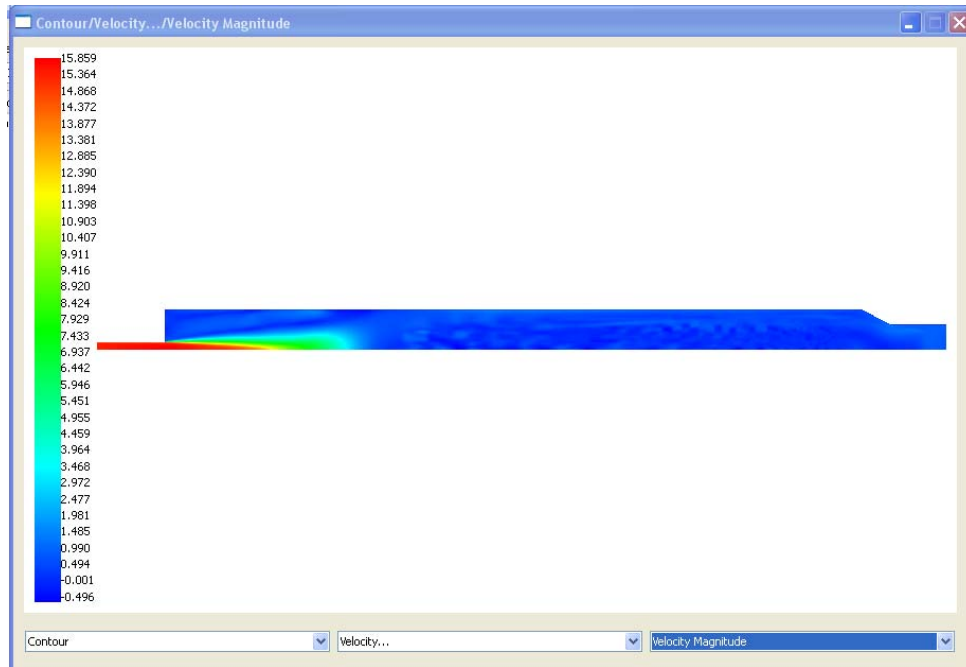


Figure 34: PCA Velocity Magnitude Results from the Co-Simulation.

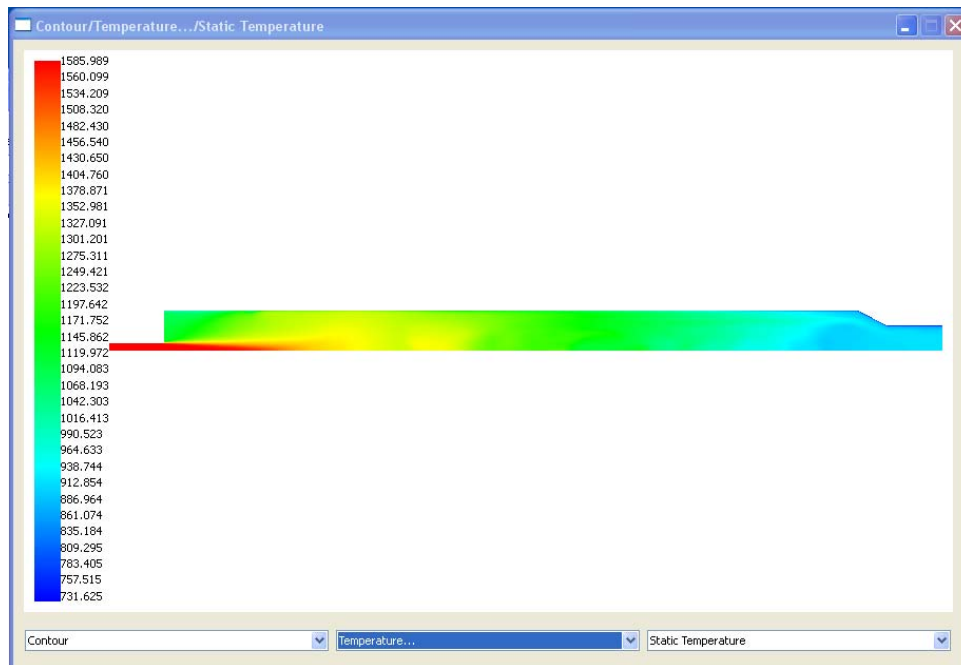


Figure 35: PCA Temperature Results from the Co-Simulation.

6.0 CONCLUSIONS AND RECOMMENDATIONS

The demonstrated results from this APECS project helps to showcase a novel and unique advancement in the field of process modeling and design – the application and use of PCA technology for ROM contour plots. Alstom Power was tasked with the responsibility to (a) develop a ROM for a radiant syngas cooler based on both regression and principal component analysis (PCA), (b) calibrate the ROM based on the equivalent CFD component model, verifying adequate predictive accuracy over the parameter space required for convergence of the FutureGen cycle at the full load condition, and then (c) systematically test the ROM with a simplified version of a FutureGen cycle. This was accomplished.

For a DOE-specified process model, an engineering design for a radiant syngas cooler was completed and a CFD model was constructed in accordance with the design. The CFD model was calibrated by adjusting the wall heat transfer boundary conditions until the desired absorption was achieved. The CFD case was exercised over a range of flow rates. Since the APECS PCA transformation capability is presently only amenable with 2-D cases, a 2-D CFD case was also constructed. A regression ROM was built from multiple runs of the CFD case, and the PCA capability was qualitatively verified from the associated 2-D contour plots. The ROM was instantiated in the DOE flowsheet and exercised for various inlet flow rate perturbations. This report documents the case characteristics and results and discusses the viability and capabilities of the linkage/interface software for regression ROMS with the PCA-generated contour plots. All of the runs were completed successfully.

The CFD runs exposed a recirculation zone on the symmetry axis, with multiple, complex reverse flow zone patterns downstream of the recirculation zone. Such aerodynamic patterns do not exist in an actual Alstom RSC. However, APER GmbH would justifiably not provide the additional engineering design support or information necessary to eradicate such patterns from the computational domain, because of the likelihood that proprietary or sensitive information would be disclosed to the public domain. Consequently, the RSC CFD model provided as a deliverable with this report, is only a gross approximation of a properly operating RSC. Nevertheless, the procedure used here -- to construct a ROM with regression and PCA capability, to instantiate it on a flowsheet, and to run it in a co-simulation -- may be followed for any subsequent syngas cooler design investigated by NETL and its partners.

Substantial progress has been made by ANSYS in facilitating the direct instantiation of FLUENT® on an Aspen Plus® flowsheet, as well in building FLUENT®-derived ROMs. While the APECS-related codes continue to be developed and enhanced, many bugs have been surfaced and eliminated and the software has been made more user-friendly and transparent. The documentation is maturing and is a great help in assisting the user to grasp the work flow.

Some additional conclusions and recommendations are as follows:

- (1) Conclusion: A 2-D surrogate model was necessary in this demonstration task because the PCA technology has not yet been developed for 3-D models.

- [1] Recommendation: Begin to develop 3-D PCA transformations for the APECS tool kit. However, rather than carry the burden of a full 3-D transformation, an alternative or intermediate approach might be to find a way to do 2-D PCA transformations on selected 2-D plotting planes defined by the user within the 3-D domain of the FLUENT® case.
- (2) Conclusion: The 2-D PCA transformation technology developed to date has clearly demonstrated how contour plots can be implemented as part of a regression ROM and accessed within a flowsheet.
- [2-A] Recommendation: Perhaps some interactive capability could be added. For example, the user might desire to (a) change scales on the plot, or (b) enlarge and expand certain areas of the plot to view certain details better, or (b) clip values to emphasize regions of interest (e.g., where reverse flow zones are). Certainly, much of the interactive post-processing features currently in FLUENT® currently might also be put to good advantage in both the ROM Builder and the CFD Viewer (or its equivalent).
- [2-B] Recommendation: Add enhanced post-processing capabilities, such as pathlines, particle tracking, and vector plots.
- (3) Conclusion: The ROM Builder currently requires the user to put in species flow rates. In the present project, the mass fractions were uniform for each simulation, and need not have been varied as variables in the ROM Builder. The use of species flow rates in the ROM Builder required the user to manually (and tediously) input all of the species flow rates for each simulation; it also created additional (unnecessary) independent variables.
- [3] Recommendation: Perhaps allow the user to toggle between the options of (a) putting in the individual species flow rates or (b) inserting the total mass flow rate plus the individual species mass fractions. This will also allow the user to use the “Design of Experiments” methodology rather than having to specify each of the simulations and the corresponding inputs.
- (4) Conclusion: The species mapping must be done by the user after every instantiation. While non-similar names between FLUENT® and Aspen Plus® will require user intervention, some additional work can be done to assist the user with mapping, particularly where mapping is done for 20 species or more.
- [4] Recommendation: Perform a sorting and pre-mapping based on similar names so that much or all of the mapping is already done for the user.
- (5) Conclusion: The Simulation Status panel does not show if one of the FLUENT® runs diverges because of an error (e.g., Error: divergence detected in AMG solver). Only the FLUENT® log file contains that information, if the user searches for the file and any error messages within that file.

- [5] Recommendation: If possible, modify the Simulation Status panel to reflect the exceptions or errors that are thrown when the FLUENT® case is terminated.
- (6) Conclusion: Users frequently use “monitors” to gauge the convergence behaviour of each case, rather than FLUENT® residuals.
- [6] Recommendation: If possible, modify the Simulation Status panel to provide the “monitor” results for each run so that the user can better assess the convergence.

7.0 REFERENCES

1. Zitney, S. E., M. Osawe, L. Collins, E. Ferguson, D. Sloan, W. Fiveland, J. Madsen, “Advanced Process Co-Simulation of the FutureGen Power Plant,” In Proc. of the 31st International Technical Conference on Coal Utilization & Fuel Systems, May 21-25, Clearwater, FL (2006).
2. Sloan, D. G., W. A. Fiveland, S. E. Zitney, and M. Syamlal, "Software Integration for Power Plant Simulations," In Proc. of the 27th International Technical Conference on Coal Utilization & Fuel Systems, March 4-7, Clearwater, FL (2002).
3. Sloan, D. G., W. A. Fiveland, S. E. Zitney, and M. Syamlal, "Power Plant Simulations Using Process Analysis Software Linked to Advanced Modules," In Proc. of the 29th International Technical Conference on Coal Utilization & Fuel Systems, April 18-22, Clearwater, FL (2004).
4. Sloan, D. G., W. A. Fiveland, M. O. Osawe, S. E. Zitney, and M. Syamlal, " Demonstrations of Coupled Cycle Analyses and CFD Simulations over a LAN," In Proc. of the 30th International Technical Conference on Coal Utilization & Fuel Systems, April 17-21, Clearwater, FL (2005).
5. “Project Management Plan, Software Framework for Advanced Power Plant Simulations, Cooperative Agreement No. DE-FC26-05NT42443”, for Ronald W. Breault USDOE-NETL, compiled by John F. Widmann, representing team entities: ANSYS Inc., ALSTOM Power, Aspen Technology Inc., and Carnegie Mellon University (Department of Chemical Engineering), May 6, 2008.
6. http://www.netl.doe.gov/energy-analyses/baseline_studies.html: “Cost and Performance Baseline for Fossil Energy Plants,” DOE/NETL – 2007/1281, Volume 1: Bituminous Coal and Natural Gas to Electricity Final Report (Revision 1, August 2007).
7. Personal Communication with Wolfgang Deeke (Alstom Power energy recovery GmbH), April 24, 2009. The communication references US Patent: Application Number PCT/US2006/041958 (Figures 3 through 6).
8. Personal Communication with Stephen Zitney (NETL), April 9, 2009. The communication provided the James Spenik FLUENT® case and data files (i.e., K9.cas/dat) for the single-stage, downflow, entrained gasifier, as well as other associated files.

9. http://www.shg-schack.com/uploads/media/Syngas_Coolers_for_Gasification_plants.pdf
10. APECS: Advanced Process Engineering Co-Simulation User's Manual, Version 1.6.1, Prepared by ANSYS, Inc., Lebanon, NH (June 2009).

8.0 DELIVERABLES

The following files are delivered with this report:

No.	Files	Description
(1)	IGCC005_rom-prep_v7p1_Rev13.bkp run.dll run.lib save.opt	Aspen Plus® flowsheet, prior to instantiation of ROM
(2)	IGCC005_rom-H1_v7p1_Rev13_+0.his IGCC005_rom-H1_v7p1_Rev13_-10.his IGCC005_rom-H1_v7p1_Rev13_-20.his IGCC005_rom-H1_v7p1_Rev13_-30.his IGCC005_rom-H1_v7p1_Rev13_+10.his IGCC005_rom-H1_v7p1_Rev13_+20.his IGCC005_rom-H1_v7p1_Rev13_+30.his	Aspen Plus® history files
(3)	axisym-dpm-2dbase_15540.cas axisym-dpm-2dbase_15540.dat	FLUENT® 2D case and data files
(4)	pmsink.c dpm_Qprop_bitum.c	UDFs
(5)	axisym-dpm-2dbase_15540.CORBAsolver APECS_grid_connectivity.dat apecs_libudf_temp.jou APECS_node_positions.dat COFluentWrapper.txt	Files created by APECS
(6)	prep_resid.jou	Used to help the user to run a minimum number of iterations.
(7)	Copy of FLUENT_log.trn	Log file of FLUENT® runs during the running of the ROM Builder from the APECS Temp directory.
(8)	rsc_H1_win_complete-13.APECSData rsc_H1_win_complete-13.ioData rsc_H1_win_initial4.APECSData	Initial and final APECSData files for ROM Builder, as well as the final ioData file.
(9)	rsc_H1_win_rom.COInfo rsc_H1_win_rom.CORBASolver	Files created by APECS upon construction of the

	rsc_H1_win_rom.GridConnectivity rsc_H1_win_rom.Mapp rsc_H1_win_rom.Matrix rsc_H1_win_rom.Modes rsc_H1_win_rom.NodeCoordinates report_compare_predictions.txt	Regression ROM
(10)	one16th-61k-dpm-3dbase2_77531.cas one16th-61k-dpm-3dbase2_77531.dat dpm_Qprop_bitum.c	3-D 22.5° sector baseline case (60,996 cells)

**SOFTWARE FRAMEWORK FOR ADVANCED
POWER PLANT SIMULATIONS**

COOPERATIVE AGREEMENT NO DE-FC26-05NT42443

TASK 6 TOPICAL REPORT – DYNAMICS RESEARCH AND PROTOTYPE

PREPARED BY

John F. Widmann
ANSYS, Inc.

September 2010

DOE Cooperative Agreement No. DE-FC26-05NT42443

ANSYS, Inc.
10 Cavendish Court, Lebanon, NH 03766

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ABSTRACT

In Task 6 of the project, APECS was modified to mediate coupled dynamic simulations using Aspen Dynamics. The prototype dynamic simulation capability in APECS used FLUENT to model an equipment item, as is done for steady-state simulations. An event-based quasi-steady state approach was adopted for coupled dynamic simulations, in which FLUENT was run in steady-state mode in response to an event or time-varying input or throughput conditions. A ROM developed from FLUENT predictions was also coupled with Aspen Dynamics to explore the time savings that can be realized when using ROMs instead of CFD models in dynamic co-simulations. Development of the prototype provides insight into the complete requirements of a production version dynamic simulator suitable for broad industrial deployment.

TABLE OF CONTENTS

ABSTRACT.....	ii
1. Introduction.....	1
2. Software Architecture	1
3. Aspen Dynamics Wrapper	2
3.1 ACM Custom Block	3
3.2 Data Transferred	4
4. Results and Discussion	7
5. Future Research	9
Appendix A - Custom Block created in Aspen Custom Modeler.....	10

LIST OF FIGURES

Figure 1. Software Architecture for Dynamic Co-Simulation.....	2
Figure 2. The Aspen Custom Modeler block within the AD Wrapper	4
Figure 3. A simple test case corresponding to a FLUENT model of an adiabatic pipe.....	4
Figure 4. The "Dual Pipe" FLUENT model used for Dynamic Co-simulation testing.....	7

LIST OF TABLES

Table 1. The Aspen Dynamics APIs considered for the AD Wrapper	3
Table 2. The results from the benchmarking runs to investigate computational expense.....	8

1. Introduction

The primary objective of the project is to complete the development of a steady-state simulator for advanced power plants, which will allow the DOE and its contractors to systematically evaluate various power plant concepts. With the interim name of APECS¹, the simulator will link a hierarchy of plant-level and equipment-level models that have varying levels of fidelity and computational speed suitable for either preliminary conceptual design or detailed final design. An important enhancement to the APECS tool will be the creation of computationally efficient reduced-order models (ROM) based on information from high-fidelity computational fluid dynamics (CFD) models. Integration of the steady-state APECS tool with a virtual reality tools will allow process designers to navigate through a 3D virtual representation of a power plant with visualization of simulation results.

A related secondary objective is to develop a prototype dynamic simulator within APECS for advanced power plants and to demonstrate a dynamic simulation that integrates a dynamic simulator and a FLUENT equipment model. This topical report summarizes the research performed toward this second objective.

2. Software Architecture

Development of the prototype Dynamics co-simulator required careful analysis of the differences between Aspen Dynamics and Aspen Plus, and how those difference impact co-simulation. The primary differences that affect APECS co-simulation are (i) the transient solution procedure and (ii) lack of a CAPE-OPEN block in Aspen Dynamics. Because Aspen Dynamics calculates the dynamic response of the system, the FLUENT block should provide a time-varying result consistent with the current timestep. However, this adds significantly to the complexity and computational demands of the co-simulation process. Therefore, it was decided that the prototype would correspond to a quasi-steady FLUENT model in which the transients associated with the equipment item are fast relative to those of the entire process.

Another technical hurdle to the creation of a prototype Dynamic Co-Simulator is the lack of a CAPE-OPEN block in Aspen Dynamics. In fact, the CAPE-OPEN standard itself does not address the needs of a dynamic simulator. To overcome this limitation, an “Aspen Dynamics (AD) Wrapper” was created to communicate directly with APECS. Initially, it was planned to have the AD Wrapper communicate with the APECS controller. However, it was later determined that it would be more efficient to have the AD Wrapper communicate with the

¹ APECS stands for Advanced Process Engineering Co-Simulator, and is the name given to the project software by the DOE.

APECS ROM Builder, which would then communicate with the controller. Figure 1 presents a schematic of the software architecture for the prototype Dynamics Co-Simulator.

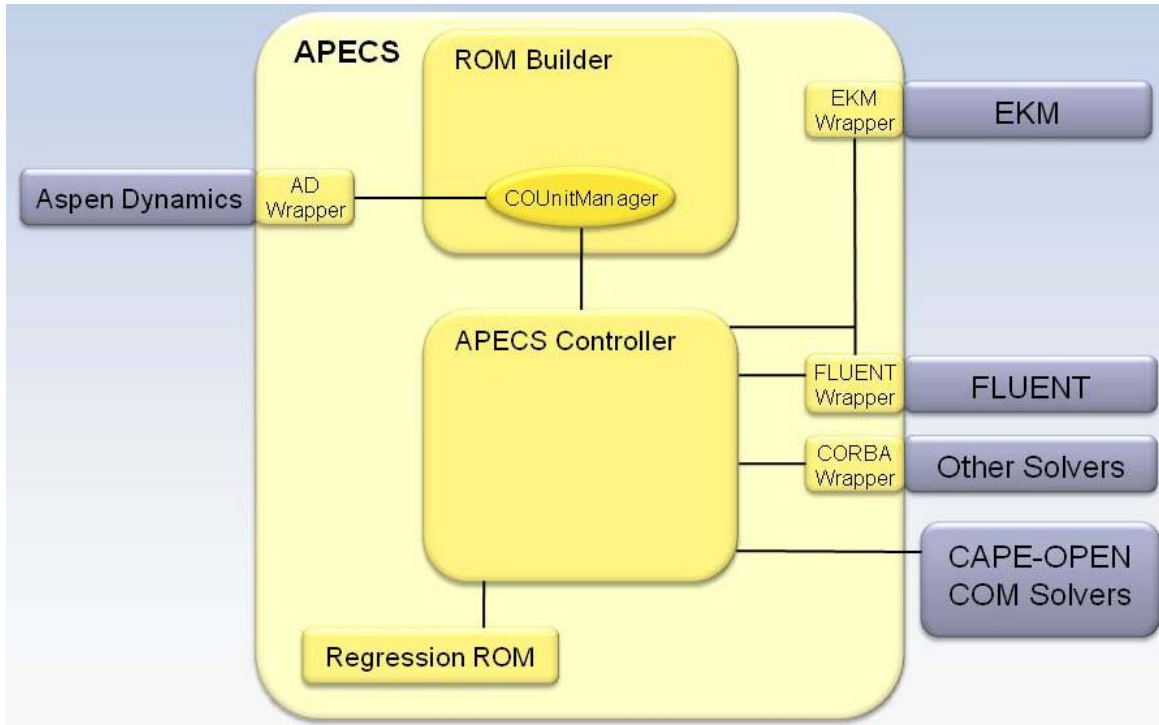


Figure 1. Software Architecture for Dynamic Co-Simulation

The Aspen Dynamics Wrapper, discussed in the next section, exchanges information between Aspen Dynamics and the component of the APECS ROM Builder responsible for running CAPE-OPEN models (including FLUENT). This design has the advantage that the prototype can utilize the APECS Model Database for model storage just like the steady-state co-simulator does. Also, because the APECS Controller is being used, the Dynamics Co-Simulator can run either FLUENT or APECS ROMs.

3. Aspen Dynamics Wrapper

Because the CAPE-OPEN standard does not address dynamic simulators, development of a Dynamic Co-Simulator required a method of exchanging information between Aspen Dynamics and APECS. The AD Wrapper was created for this purpose. Creation of the AD Wrapper involved researching the various methods by which data could be transferred to and from Aspen Dynamics. Also, it was necessary to control when and how data would be transferred. Table 1 summarized the various Application Programming Interfaces (APIs) available for these purposes.

API	Features
Aspen Dynamics SAX	Ability to call external functions at the beginning/end of a timestep Operates at the “flowsheet” level
Aspen Custom Modeler User Blocks	Ability to customize unit operations at the “block” level Custom blocks can call external functions Custom blocks can be used in Aspen Plus or Aspen Dynamics
Aspen Custom Modeler Custom Forms	Ability to create a custom User Interface
Aspen Custom Modeler Events	Ability to trigger events due to certain actions (e.g., when a block is added to the flowsheet, when a block is deleted)

Table 1. The Aspen Dynamics APIs considered for the AD Wrapper

Initially, it was believed that the Simulation Access eXtension (SAX) capability of Aspen Dynamics would be used for this purpose. However, it was determined that the custom blocks available in Aspen Custom Modeler (ACM) provided all of the functionality required for the prototype. Using ACM, a custom “APECS” block was created for the Aspen Dynamics Wrapper. This block, which can be exported from ACM and used in Aspen Plus or Aspen Dynamics, called an external C++ function that communicated with the APECS ROM Builder.

3.1 ACM Custom Block

The functionality of the ACM custom block is shown schematically in Figure 2. Streams from the AD flowsheet are connected to the external ports of the ACM block. Initial attempts to have a variable number of ports failed due to a bug in ACM 2006.5; therefore, it was decided to fix the number of external inlet and outlet ports to ten each. Those ports that were not connected to AD streams were ignored. The first step in passing inputs from AD to APECS was to convert the units to SI units. The stream data was then passed to APECS so the CFD model could be run and results returned. After the results were returned to the AD Wrapper, the units were converted back to those required by Aspen Dynamics. Finally, the enthalpy and molar volume of the streams were calculated by the ACM custom block before the outlet stream data was passed back to Aspen Dynamics.

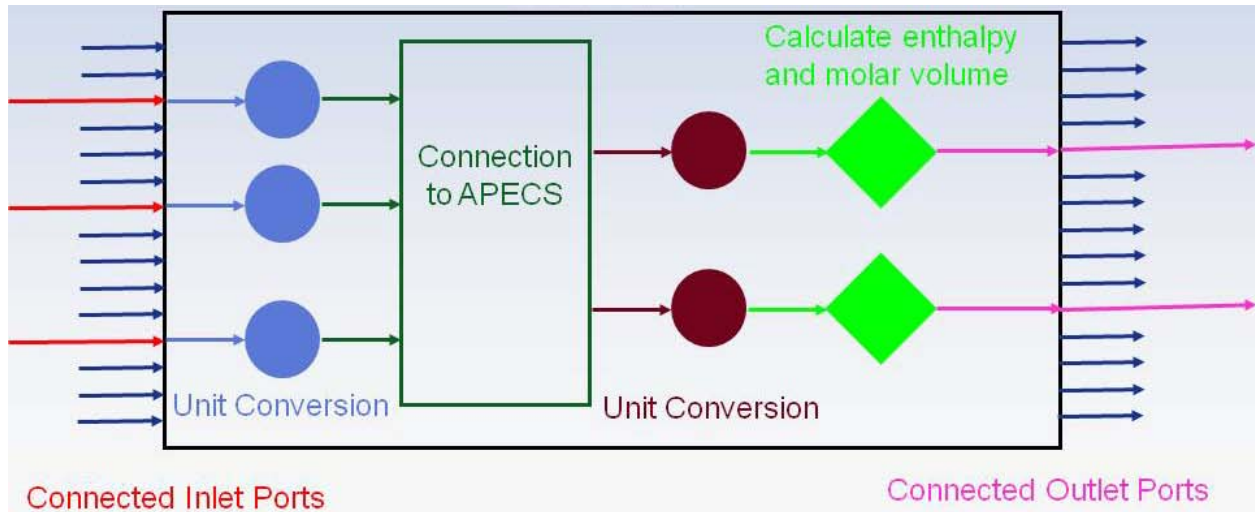


Figure 2. The Aspen Custom Modeler block within the AD Wrapper

Appendix A contains the source code for the ACM custom block used in the AD Wrapper.

3.2 Data Transferred

The data transferred between Aspen Dynamics and APECS corresponds to the information required to select the appropriate FLUENT model and run the simulation. As an example, consider the simple case of a pipe, as shown in Figure 3.

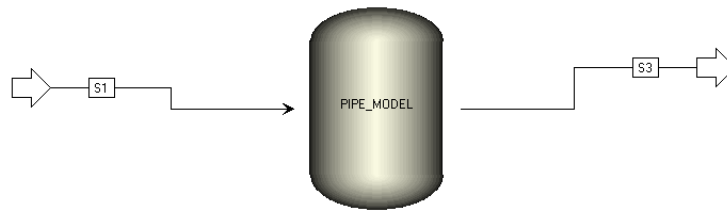


Figure 3. A simple test case corresponding to a FLUENT model of an adiabatic pipe

The inputs passed from the ACM custom block to the external function when running this pipe model are the following.

```

***** INPUTS *****
Block Name: PIPE_MODEL
Number of Inlet Ports: 10
Number of Outlet Ports: 10
Number of CONNECTED Inlet Ports: 1
Number of CONNECTED Outlet Ports: 1

Number of Species: 3
Specie 0 is ETOH

```

Specie 1 is H2O
Specie 2 is MEOH

Inlet Port Name = inlet_port
Inlet Flow Rate = 0.02 kg/s
Inlet Temperature = 313.15 K
Inlet Pressure = 100000.00 Pa
Inlet Mass Fraction [0] = 0.2876
Inlet Mass Fraction [1] = 0.1124
Inlet Mass Fraction [2] = 0.6000

Outlet Port Name = outlet_port

Persistence String 0 is InstanceKey_47
Persistence String 1 is UnusedPersistenceString_1
Persistence String 2 is UnusedPersistenceString_2

The text above is displayed in the Aspen Dynamics console when the external function is called. The persistence strings are used to maintain a connection with the model files (e.g., FLUENT case and data files) that are stored in the EKM Model Database. Similarly, when the outputs are returned, the following text is displayed.

```
***** OUTPUTS *****  
OUTPUTS was called 60 times  
Retrieving the COUnitManager pointer. Block Name: PIPE_MODEL  
Setting values of persistence strings in OUTPUTS now.  
  
Persistence String 0 is now InstanceKey_47  
Persistence String 1 is now UnusedPersistenceString_1  
Persistence String 2 is now UnusedPersistenceString_2  
  
Outlet flow rate for outlet_port is 0.02 kg/s  
  
Outlet temperature for outlet_port is 313.15 K  
  
Outlet Pressure for outlet_port is 99000.00 Pa  
  
Outlet Mass Fractions for outlet_port are:  
Species [0]: 0.28755 (ETOH)  
Species [1]: 0.11246 (H2O)  
Species [2]: 0.59999 (MEOH)  
*****
```

The text above indicates how many times “OUTPUTS” was called. This is an indication that Aspen Dynamics requested that the external function calculate results a certain number of times. When the dynamic simulation is initiated, AD will send the external function a “PRECALL” command. This indicates that the function should initialize and be prepared to calculate. There are then a series of “OUTPUT” calls, each with different input values. The function is expected to return results on “OUTPUT” calls. Finally, when the simulation is complete and the user is closing the flowsheet, AD submits a “POSTCALL” call. The “PRECALL”, “OUTPUT”, and “POSTCALL” calls are analogous to the CAPE-OPEN “Restore”, “Calculate”, and “Terminate” methods, respectively.

When Aspen Dynamics initiates a “PRECALL” command, the AD Wrapper creates a COUnitManager object. This is the part of the APECS ROM Builder responsible for running CAPE-OPEN models when creating ROMs. FLUENT is then started and the case/data files are downloaded from the Model Database and loaded into FLUENT. Because this is a prototype, a User Interface was not created specifically for Aspen Dynamics. Therefore, a method of identifying the correct FLUENT model (or APECS ROM) in the Model Database was required. It was also necessary to provide certain information to the COUnitManger that is normally provided by the APECS User Interface. This was performed by creating several input files and storing them in a directory corresponding to the model name. This directory had the form

```
%AD_INPUT_FILES%/MODEL_NAME
```

Where AD_INPUT_FILES is an environment variable with the name of the parent directory and MODEL_NAME is the name of the block in the Aspen Dynamics flowsheet. In this manner, several APECS blocks could run simultaneously in one flowsheet. Within this directory, the following files were present:

```
InputNames.txt
InputValues.txt
MolecularWeights.txt
OutputNames.txt
Persistent.txt
```

Each of the files contains information required by the COUnitManager. For example, a ROM with 3 inlet ports may have an InputNames.txt file that contains the following:

```
Port:inlet_3:temperature:Real
Port:inlet_3:pressure:Real
Port:inlet_3:specie:ethanol:Real
Port:inlet_3:specie:water:Real
Port:inlet_3:specie:methanol:Real
Port:inlet_2:temperature:Real
Port:inlet_2:pressure:Real
Port:inlet_2:specie:ethanol:Real
Port:inlet_2:specie:water:Real
Port:inlet_2:specie:methanol:Real
Port:inlet_1:temperature:Real
Port:inlet_1:pressure:Real
Port:inlet_1:specie:ethanol:Real
Port:inlet_1:specie:water:Real
Port:inlet_1:specie:methanol:Real
Parameter:enforce-mass-balance:Boolean
```

The above text indicates that each inlet port will have a corresponding temperature, pressure, methanol flow rate, ethanol flow rate, and water flow rate. Furthermore, each of these inputs are Real values. The last line in the file indicates that the ROM also has an input parameter called “enforce-mass-balance” that is a Boolean parameter. Similarly, the other input files contain other information required to run the model.

Upon receiving an "OUTPUTS" call, the AD Wrapper transfers the inlet boundary conditions to FLUENT, converges the FLUENT model, and returns the outputs to Aspen Dynamics. Finally, when the AD Wrapper receives a "POSTCALL" command, it exits FLUENT and releases any allocated memory.

4. Results and Discussion

The prototype Dynamics Co-Simulation was demonstrated using several simple FLUENT models and APECS ROMs created from FLUENT simulations. The first test case was the simple FLUENT pipe model discussed above. This was used to ensure that inputs could be transferred to the outlet correctly. Next, a FLUENT model was created with two inlet ports and two outlet ports. This model was essentially two channels that permit heat transfer but not mass transfer between the streams. This verified that the block was accurately coordinating the connections when multiple streams were connected. Figure 4 shows the block and stream details.

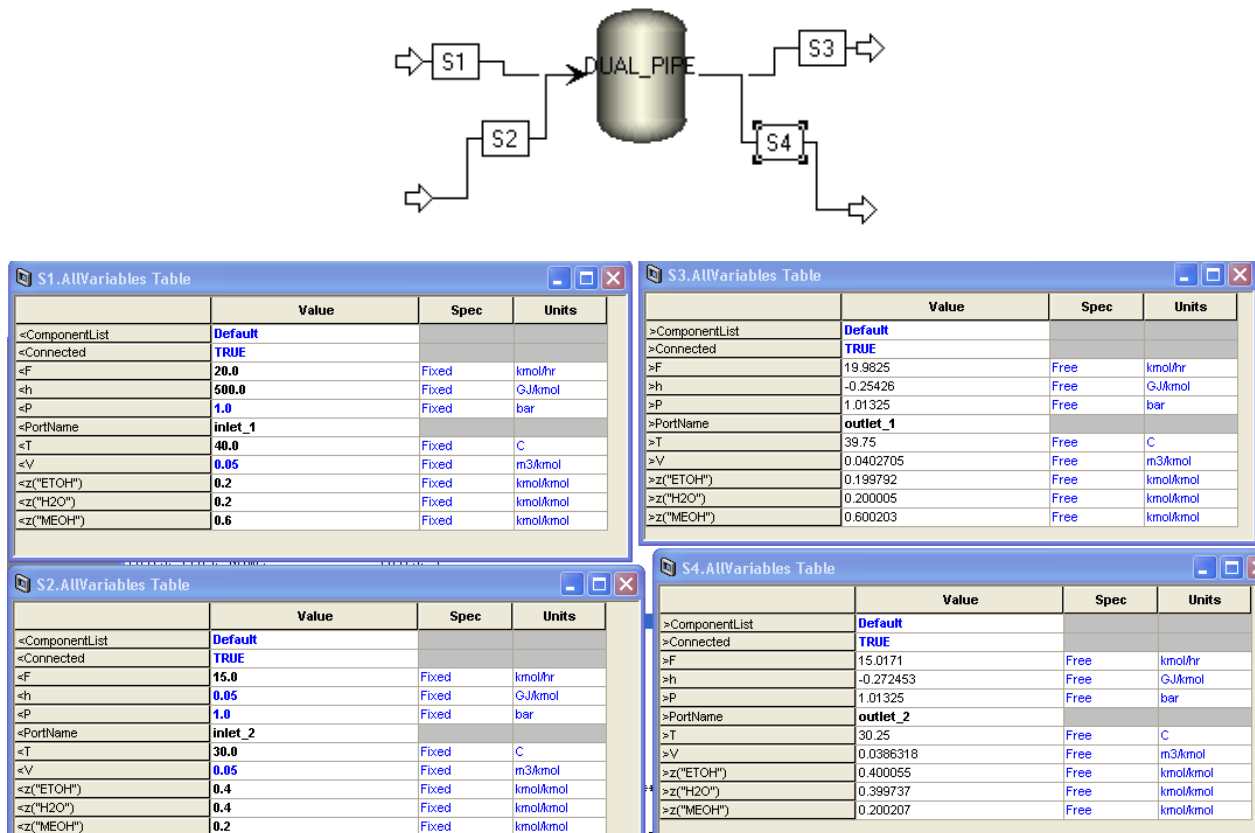


Figure 4. The "Dual Pipe" FLUENT model used for Dynamic Co-simulation testing

Additional simulations were performed to investigate the computational requirements of Dynamic Co-Simulation. The following two APECS models were used for the benchmarking:

Case 1 - The Pipe Model ROM Template (Generic CORBA Solver)
 Fastest APECS block available
 Used for both Steady and Dynamic Co-Simulation in Dynamic Simulator

Case 2 – A 2D FLUENT model with three channels
 1200 Quad Cells
 3 inlets, 3 outlets; No mixing – only heat transfer between streams
 Used for both Steady and Dynamic Co-Simulation in Dynamic Simulator

The results from the benchmarking are shown in Table 2. Each of the APECS models were run within the Dynamic Simulator in both steady and dynamic modes. The actual time (“wall time”) for the simulations is shown in the table. In addition, the number of “OUTPUT” calls required is presented. This corresponds to how many times inputs were passed to APECS requesting updated outputs. Even for the simple models used here, APECS was called many times while the flowsheet converged. This is because the simulator is using an “equation oriented” algorithm instead of the “sequential” algorithm available in Aspen Plus. It is clear that these multiple calls result in the simulations taking significantly longer than they do in Aspen Plus. As a comparison, an Aspen Plus flowsheet containing only the first benchmark case in the table (Pipe Model ROM Template running in “steady” mode) converges on the order of one second. The dynamic simulator using the same model takes approximately 400 times longer. This ROM corresponds to the fastest APECS model available. The other benchmark cases all took substantially longer.

Model	Mode	Inputs	Simulation Time, s	Real Time, s	“Output” Calls
Pipe Model ROM Template	Steady			407	60
Pipe Model ROM Template	Unsteady	Steady	5	7200	1080
2D FLUENT Model	Steady			1096	60
2D FLUENT Model	Unsteady	Ramped (5 seconds)	10	22098	1380

Table 2. The results from the benchmarking runs to investigate computational expense

5. Future Research

The research and prototype demonstration discussed here revealed that the computation expense presents a significant obstacle to the practical application of Dynamic Co-Simulation. One possible solution is to accelerate the flowsheet convergence by calculating the partial derivatives of the outputs with respect to the inputs and passing the Jacobian to the flowsheet. This would eliminate the need for the flowsheet to numerically calculate the derivatives and speed the flowsheet convergence. This would not be possible with FLUENT models; however, the APECS Regression ROM can calculate explicitly the derivatives with little computational overhead. This represents a promising approach to accelerating the convergence of the flowsheet during Dynamic Co-Simulation.

Appendix A - Custom Block created in Aspen Custom Modeler

Declaration

```
Procedure procConnectToAPECS6
  Library:    "AspenDynamicsWrapper";
  Call:      "ConnectToAPECS6";
  // Compatibility: "ACM2004";      Option prevents export to A+ (bug in ACM 2006)
  Implementation: Subroutine "ADWrapper6.cpp";
  Language:   " C++";
  Options:    PROPERTIES, PRECALL, POSTCALL, ALWAYS CALL, CHANGES INPUTS;

  Inputs:    iNrInletPorts          as integer,
             iNrOutletPorts         as integer,
             fInletFlowRate         as flowrate(*),
             fInletTemperature      as temperature(*),
             fInletPressure         as pressure(*),
             fInletMassFractions    as massfraction(*,*),
             stringInletPortNames   as string(*),
             stringOutletPortNames  as string(*),
             stringConnectedInputs  as string(*),
             stringConnectedOutputs as string(*),
             stringBlockName        as string,
             stringPersistenceStrings as string(*),
             iNrPersistenceStrings  as integer;

  Outputs:   fOutletFlowRate        as flowrate(*),
             fOutletTemperature     as temperature(*),
             fOutletPressure        as pressure(*),
             fOutletMassFractions    as massfraction(*,*);

End
```

Implementation

```
Port InletPort
  MassFractions(Componentlist) as hidden  MassFraction (1/size(Componentlist), fixed);
  Temperature                  as hidden  Temperature (300, fixed);
  Pressure                     as hidden  Pressure (101325, fixed);
  FlowRate                    as hidden  FlowRate (1, fixed);
  PortName                    as hidden  stringparameter ("InletPort");
  Connected                    as hidden  stringparameter ("FALSE");

End

Port OutletPort
  MassFractions(componentlist) as hidden  MassFraction (1/size(Componentlist), free);
  Temperature                  as hidden  Temperature (300, free);
  Pressure                     as hidden  Pressure (101325, free);
  FlowRate                    as hidden  FlowRate (1, free);
  PortName                    as hidden  stringparameter ("OutletPort");
  Connected                    as hidden  stringparameter ("FALSE");

End
```

Port APECSMoleFractionPort_SI uses MoleFractionPort_SI

```
PortName as stringparameter ("port");  
End
```

Port InletPortMoles

```
MoleFractions(Componentlist) as hidden MoleFraction (1/size(Componentlist), fixed);  
Temperature as hidden Temperature (310, fixed);  
Pressure as hidden Pressure (101325, fixed);  
FlowRate as hidden FlowRate (1, fixed);  
PortName as hidden stringparameter ("InletPort");  
Connected as hidden stringparameter ("FALSE");  
End
```

Port OutletPortMoles

```
MoleFractions(componentlist) as hidden MoleFraction (1/size(Componentlist), free);  
Temperature as hidden Temperature (300, free);  
Pressure as hidden Pressure (101325, free);  
FlowRate as hidden FlowRate (1, free);  
PortName as hidden stringparameter ("OutletPort");  
Connected as hidden stringparameter ("FALSE");  
End
```

Port APECS_Hidden_External_Port

```
ExportAs as ExportParam("MoleFractionPort");  
PortName as stringparameter ("PortName");  
F as hidden notype (Description:"Molar flow rate"); // in kmol/s  
T as hidden notype (Description:"Temperature"); // in K  
P as hidden notype (Description:"Pressure"); // in N/m2  
H as hidden notype (Description:"Molar enthalpy"); // in J/kmol  
V as hidden notype (Description:"Molar volume"); // in m3/kmol  
Z(componentlist) as hidden notype (Description:"Mole fractions"); // in kmol/kmol  
End
```

Port APECSMoleFractionPort uses MoleFractionPort

```
PortName as stringparameter ("PortName", description: "Name of port that will be visible to user ");  
Connected as stringparameter ("FALSE");  
End
```

Model APECS

```
// This submodel is the connection to the APECS Controller.
```

```
// All units are SI (Fluent) units.
```

```
// name of this block in the flowsheet
```

```
stringBlockName as hidden stringparameter ("BlockName");
```

```
// strings required to persist the block in the Aspen Dynamics file
```

```
iNrPersistenceStrings as hidden integerparameter (3);
```

```
iPersistenceStringIndices as hidden integerset ([1:iNrPersistenceStrings]);
```

```
stringPersistenceStrings(iPersistenceStringIndices) as hidden stringparameter ("PersistenceString");
```

```
// ports
```

```
iNrInletPorts as hidden NrInletPorts (10, description:"The number of inlet ports");
```

```
iNrOutletPorts as hidden NrOutletPorts (10, description:"The number of outlet ports");
```

```
iInletPortIndices as hidden integerset ([1:iNrInletPorts]);
```

```
iOutletPortIndices as hidden integerset ([1:iNrOutletPorts]);
```

```
portInlets(iInletPortIndices) as input InletPort;
```

```

portOutlets(iOutletPortIndices) as output    OutletPort;

for iPort in iInletPortIndices do
  if portInlets(iPort).IsConnected or portInlets(iPort).IsLinked then
    portInlets(iPort).FlowRate      : free;
    portInlets(iPort).Temperature    : free;
    portInlets(iPort).Pressure       : free;
    portInlets(iPort).MassFractions : free;
  else
    portInlets(iPort).FlowRate      : fixed;
    portInlets(iPort).Temperature    : fixed;
    portInlets(iPort).Pressure       : fixed;
    portInlets(iPort).MassFractions : fixed;
  endif
endifor

for iPort in iOutletPortIndices do
  portOutlets(iPort).FlowRate      : free;
  portOutlets(iPort).Temperature    : free;
  portOutlets(iPort).Pressure       : free;
  portOutlets(iPort).MassFractions : free;
endifor

// call procConnectToAPECS6
call (portOutlets.FlowRate, portOutlets.Temperature, portOutlets.Pressure, portOutlets.MassFractions) =
  procConnectToAPECS6 (iNrInletPorts, iNrOutletPorts, portInlets.FlowRate, portInlets.Temperature,
    portInlets.Pressure, portInlets.MassFractions, portInlets.PortName, portOutlets.PortName,
    portInlets.Connected, portOutlets.Connected, stringBlockName, stringPersistenceStrings,
    iNrPersistenceStrings);
End

```

Model InletUnitConversion

```
// Submodel to convert the units from Aspen Plus / Aspen Dynamics to APECS units (not ACM units)
```

```
// ports
```

```
portInletPort      as input    APECSMoleFractionPort;
portOutletPort     as output    OutletPort;
```

```
if portInletPort.IsConnected or portInletPort.IsLinked then
```

```
  portInletPort.F      : free;
  portInletPort.T      : free;
  portInletPort.P      : free;
  portInletPort.h      : free;
  portInletPort.V      : free;
  portInletPort.z      : free;
```

```
else
```

```
  portInletPort.F      : fixed;
  portInletPort.T      : fixed;
  portInletPort.P      : fixed;
  portInletPort.h      : fixed;
  portInletPort.V      : fixed;
  portInletPort.z      : fixed;
```

```
endif
```

```
portOutletPort.FlowRate      : free;
portOutletPort.Temperature    : free;
```

```

portOutletPort.Pressure      : free;
portOutletPort.MassFractions : free;

// convert units
// flow rate
fMolWeightMixture           as hidden  molweight (1.0, free);
arrayMoleFractionsToPass(Componentlist) as hidden  molefraction (1/size(componentlist), free);
arrayMoleFractionsToPass    = portInletPort.z;      // kmol/kmol
call (fMolWeightMixture)    = pMolWeight( arrayMoleFractionsToPass );
portOutletPort.FlowRate     = fMolWeightMixture * portInletPort.F / 3600;

// mass fraction
arrayMolWeights(Componentlist) as hidden  molweight (1.0, free);

call (arrayMolWeights)      = pMolWeights();
for iSpecie in Componentlist do
  portOutletPort.MassFractions(iSpecie) = arrayMolWeights(iSpecie) * portInletPort.F *
arrayMoleFractionsToPass(iSpecie) / portOutletPort.FlowRate / 3600;
endfor
// temperature
portOutletPort.Temperature   = portInletPort.T + 273.15;      // K
// pressure
portOutletPort.Pressure      = portInletPort.P * 1.0e5;      // Pa
// port name
portOutletPort.PortName     : portInletPort.PortName;
// connected
portOutletPort.Connected    : portInletPort.Connected;
End

```

Model OutletUnitConversion

// Submodel to convert from APECS units to Aspen Plus / Aspen Dynamics units (not ACM units)

// ports

```

portInletPort      as input  InletPort;
portOutletPort     as output  OutletPortMoles;

```

// MoleFractionPort Variable Types

// Description	Variable Type	Physical Quantity	Base Units
// Molar flow rate	Flow_mol	MoleFlow	kmol/hr
// Mole fractions	Molefraction(*)	MoleFraction	kmol/kmol
// Temperature	Temperature	Temperature	C
// Pressure	Pressure	Pressure	bar
// Molar volume	Vol_mol	MolarVolume	m3/kmol
// Molar enthalpy	Enth_mol	MolarEnthalpy	GJ/kmol

if portInletPort.IsConnected or portInletPort.IsLinked then

```

portInletPort.FlowRate      : free;      // kg/s
portInletPort.Temperature   : free;      // K
portInletPort.Pressure      : free;      // Pa
portInletPort.MassFractions : free;      // kg/kg

```

else

```

portInletPort.FlowRate      : fixed;     // kg/s
portInletPort.Temperature   : fixed;     // K
portInletPort.Pressure      : fixed;     // Pa
portInletPort.MassFractions : fixed;     // kg/kg

```

endif

```

portOutletPort.FlowRate     : free;
portOutletPort.Temperature   : free;

```

```

portOutletPort.Pressure      :      free;
portOutletPort.MoleFractions :      free;

// convert units
// molar flow rate
arrayMolWeights(Componentlist)      as hidden      molweight (1.0, free);
arrayMolarFlowRate(Componentlist)    as hidden      flowrate (1/size(componentlist), free);

call (arrayMolWeights)      =      pMolWeights();
for iSpecie in Componentlist do
    arrayMolarFlowRate(iSpecie)      =      portInletPort.FlowRate * portInletPort.MassFractions(iSpecie) /
arrayMolWeights(iSpecie) * 3600; // kmol/hr of Species i
endfor
portOutletPort.FlowRate      =      sigma (foreach (iSpecie in Componentlist) arrayMolarFlowRate(iSpecie) );

// mole fraction
for iSpecie in Componentlist do
    portOutletPort.MoleFractions(iSpecie)      =      arrayMolarFlowRate(iSpecie) / portOutletPort.FlowRate;
endfor
// temperature
portOutletPort.Temperature      =      portInletPort.Temperature - 273.15;
// pressure
portOutletPort.Pressure      =      portInletPort.Pressure / 1.0e5;
// port name
portOutletPort.PortName      :      portInletPort.PortName;
// connected?
portOutletPort.Connected      :      portInletPort.Connected;
End

```

Model MolarVolumeAndEnthalpy

// Submodel to calculate the molar volume and enthalpy and add it to the stream.

```

// ports
portInletPort      as input      InletPortMoles;
portOutletPort      as output      APECSMoleFractionPort;

```

```

if portInletPort.IsConnected or portInletPort.IsLinked then

```

```

    portInletPort.FlowRate      :      free;
    portInletPort.Temperature      :      free;
    portInletPort.Pressure      :      free;
    portInletPort.MoleFractions      :      free;

```

```

else

```

```

    portInletPort.FlowRate      :      fixed;
    portInletPort.Temperature      :      fixed;
    portInletPort.Pressure      :      fixed;
    portInletPort.MoleFractions      :      fixed;

```

```

endif

```

```

portOutletPort.F      :      free;
portOutletPort.T      :      free;
portOutletPort.P      :      free;
portOutletPort.Z      :      free;
portOutletPort.V      :      free;
portOutletPort.H      :      free;

```

// calculate molar volume and enthalpy

```

fMolarVolume          as hidden  vol_mol (1.0, free);
fMolarEnthalpy        as hidden  enth_mol (1.0, free);
fFlowRate             as hidden  flowrate (1.0, free);      // kmol/hr
fTemperature          as hidden  temperature (1.0, free);   // C
fPressure             as hidden  pressure (1.0, free);      // bar
fMoleFractions(Componentlist) as hidden  molefraction (0.0, free);
fFlowRate             =         portInletPort.FlowRate;
fTemperature          =         portInletPort.Temperature;
fPressure             =         portInletPort.Pressure;
fMoleFractions        =         portInletPort.MoleFractions;

// variables returned from property submodel
fLiquidMolarFraction  as         molefraction (0.0, free);
fVaporMolarFraction   as         molefraction (0.0, free);
fLiquidMolarEnthalpy  as         enth_mol (0.0, free);
fVaporMolarEnthalpy   as         enth_mol (0.0, free);
fLiquidMolarDensity   as         dens_mol (1.0, free);
fVaporMolarDensity    as         dens_mol (1.0, free);

// Create instance of submodel Props_flash2
getEnthalpyAndVolume as Props_flash2;

// Note: for compatibility with Aspen Plus, PropMode should be "Rigorous" and FlashMode should be
"SmoothEquation".
// These can be specified in "Globals".

// inputs
getEnthalpyAndVolume.T = fTemperature;
getEnthalpyAndVolume.P = fPressure;
getEnthalpyAndVolume.z = fMoleFractions;

// outputs
fLiquidMolarFraction = getEnthalpyAndVolume.lf;
fVaporMolarFraction  = getEnthalpyAndVolume.vf;
fLiquidMolarEnthalpy = getEnthalpyAndVolume.hl;
fVaporMolarEnthalpy  = getEnthalpyAndVolume.hv;
fLiquidMolarDensity  = getEnthalpyAndVolume.Rhol;
fVaporMolarDensity   = getEnthalpyAndVolume.Rhov;

// Calculate the molar volume and molar enthalpy from the variables returned
fMolarDensity as hidden dens_mol (1.0, free); // kmol/m^3
fMolarDensity = fLiquidMolarFraction*fLiquidMolarDensity +
fVaporMolarFraction*fVaporMolarDensity;
fMolarEnthalpy = fLiquidMolarFraction*fLiquidMolarEnthalpy +
fVaporMolarFraction*fVaporMolarEnthalpy;

if (fMolarDensity > 0.0) then
  fMolarVolume = 1/fMolarDensity;
else
  fMolarVolume = -1.0; // temporary - create a warning function in dll
endif

// assign values to outlet port variables
portOutletPort.F = fFlowRate;
portOutletPort.T = fTemperature;
portOutletPort.P = fPressure;

```



```

portOutletPort.Z      =      fMoleFractions;
portOutletPort.V      =      fMolarVolume;
portOutletPort.H      =      fMolarEnthalpy;
End

```

Model APECS_BLOCK

```
// Outer wrapper for "APECS" submodels - submodel to "AD_Wrapper"
```

```
// name of this block in the flowsheet
stringBlockName      as hidden      stringparameter ("BlockName");
```

```
// collection of inlet ports
```

```

iNrInletPorts        as hidden      integerparameter (10);
iInletPortIndices    as hidden      integerset   ([1:iNrInletPorts]);
InputFromPME(iInletPortIndices)    as input      APECSMoleFractionPort;
arrayInletPortNames(iInletPortIndices) as hidden      stringparameter;
arrayInletPortNames      :          InputFromPME.PortName;

```

```
// collection of outlet ports
```

```

iNrOutletPorts       as hidden      integerparameter (10);
iOutletPortIndices   as hidden      integerset   ([1:iNrOutletPorts]);
OutputToPME(iOutletPortIndices)    as output      APECSMoleFractionPort;
arrayOutletPortNames(iOutletPortIndices) as hidden      stringparameter;
arrayOutletPortNames      :          OutputToPME.PortName;

```

```
// Verify that all inlet ports are connected to a stream in the "AD Wrapper" model
```

```
fDummyVariableOne(iInletPortIndices) as hidden      realvariable (free);
```

```
for iPort in iInletPortIndices do
```

```
  fDummyVariableOne(iPort)           : 0.0;
```

```
  if InputFromPME(iPort).IsConnected or InputFromPME(iPort).IsLinked then
```

```
    InputFromPME(iPort).F             : free;
```

```
    InputFromPME(iPort).T             : free;
```

```
    InputFromPME(iPort).P             : free;
```

```
    InputFromPME(iPort).h             : free;
```

```
    InputFromPME(iPort).V             : free;
```

```
    InputFromPME(iPort).z             : free;
```

```
  else
```

```
    InputFromPME(iPort).F             : fixed;
```

```
    InputFromPME(iPort).T             : fixed;
```

```
    InputFromPME(iPort).P             : fixed;
```

```
    InputFromPME(iPort).h             : fixed;
```

```
    InputFromPME(iPort).V             : fixed;
```

```
    InputFromPME(iPort).z             : fixed;
```

```
  endif
```

```
endfor
```

```
// Verify that all outlet ports are connected to a stream in the "AD Wrapper" model
```

```
fDummyVariableTwo(iOutletPortIndices) as hidden      realvariable (free);
```

```
for iPort in iOutletPortIndices do
```

```
  OutputToPME(iPort).F               : free;
```

```
  OutputToPME(iPort).T               : free;
```

```
  OutputToPME(iPort).P               : free;
```

```
  OutputToPME(iPort).h               : free;
```

```
  OutputToPME(iPort).V               : free;
```

```
  OutputToPME(iPort).z               : free;
```

```

    fDummyVariableTwo(iPort)          : 0.0;
endfor

//////////////////////////////////// submodels //////////////////////////////////////

// Note: The order of flow is
//     Aspen Plus / Aspen Dynamics
//     InletUnitConversion
//     APECS
//     OutletUnitConversion
//     CalculateMolarVolumeAndEnthalpy
//     Aspen Plus / Aspen Dynamics

// Declare all submodels here
ConvertUnitsAspenToFluent(iInletPortIndices)  as hidden  InletUnitConversion;
FluentBlock                                as hidden  APECS;
ConvertUnitsFluentToAspen(iOutletPortIndices) as hidden  OutletUnitConversion;
MolarVolumeAndEnthalpy(iOutletPortIndices)   as hidden  MolarVolumeAndEnthalpy;

// ConvertUnitsAspenToFluent - submodel to convert Aspen Plus / Aspen Dynamics units to APECS units
// Note there is one instance of this submodel for each inlet port; ports are passed individually
//
for iPort in iInletPortIndices do
    link InputFromPME(iPort) and ConvertUnitsAspenToFluent(iPort).portInletPort;
    connect ConvertUnitsAspenToFluent(iPort).portOutletPort and FluentBlock.portInlets(iPort);
endfor

// FluentBlock - submodel that connects to APECS Controller
// Note there is only one instance of this submodel; ports are passed as arrays
//
FluentBlock.stringBlockName : stringBlockName;
FluentBlock.iNrInletPorts   : iNrInletPorts;
FluentBlock.iNrOutletPorts  : iNrOutletPorts;

// port names and connectivity must be passed to this submodel explicitly
for iPort in iInletPortIndices do
    FluentBlock.portInlets(iPort).PortName      : arrayInletPortNames(iPort);
    FluentBlock.portInlets(iPort).Connected     : InputFromPME(iPort).Connected;
endfor

// visit each submodel outlet port (inlet ports connected in submodel ConvertUnitsAspenToFluent)
for iPort in iOutletPortIndices do
    connect FluentBlock.portOutlets(iPort) and ConvertUnitsFluentToAspen(iPort).portInletPort;

    // port names and connectivity must be passed to this submodel explicitly
    FluentBlock.portOutlets(iPort).PortName      : arrayOutletPortNames(iPort);
    FluentBlock.portOutlets(iPort).Connected     : OutputToPME(iPort).Connected;
endfor

// ConvertUnitsFluentToAspen - submodel to convert APECS units to Aspen Plus / Aspen Dynamics units
// Note there is one instance of this submodel for each outlet port; ports are passed individually
//
for iPort in iOutletPortIndices do
    // visit each submodel outlet port (inlet ports connected in submodel FluentBlock)
    // connect submodel outlet ports to CalculateMolarVolume
    connect MolarVolumeAndEnthalpy(iPort).portInletPort and ConvertUnitsFluentToAspen(iPort).portOutletPort;
endfor

```

```

endifor

// MolarVolumeAndEnthalpy - submodel to calculate the molar volume and enthalpy and add back to the stream
// Note there is one instance of this submodel for each outlet port; ports are passed individually
for iPort in iOutletPortIndices do
  link MolarVolumeAndEnthalpy(iPort).portOutletPort and OutputToPME(iPort);
endifor

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
End

```

Model AD_Wrapper

```

// Outer wrapper containing fixed number of inlet/outlet ports for "APECS_BLOCK"

```

```

// name of this block in the flowsheet

```

```

Block_Name          as          stringparameter ("BlockName");
stringBlockName     as hidden   stringparameter ("BlockName");
stringBlockName     :          Block_Name;

```

```

// collection of inlet ports (arrays of ports will not export to Aspen Plus)

```

```

iNrInletPorts       as hidden   integerparameter (10);
iInletPortIndices   as hidden   integerset ([1:iNrInletPorts]);
arrayInletPortNames(iInletPortIndices) as hidden stringparameter;

```

```

InputFromPME_1      as input APECSMoleFractionPort;
InputFromPME_2      as input APECSMoleFractionPort;
InputFromPME_3      as input APECSMoleFractionPort;
InputFromPME_4      as input APECSMoleFractionPort;
InputFromPME_5      as input APECSMoleFractionPort;
InputFromPME_6      as input APECSMoleFractionPort;
InputFromPME_7      as input APECSMoleFractionPort;
InputFromPME_8      as input APECSMoleFractionPort;
InputFromPME_9      as input APECSMoleFractionPort;
InputFromPME_10     as input APECSMoleFractionPort;

```

```

if InputFromPME_1.IsConnected then

```

```

  InputFromPME_1.F   : fixed;
  InputFromPME_1.T   : fixed;
  InputFromPME_1.P   : fixed;
  InputFromPME_1.h   : fixed;
  InputFromPME_1.V   : fixed;
  InputFromPME_1.z   : fixed;
  InputFromPME_1.Connected : "TRUE";

```

```

else

```

```

  InputFromPME_1.F   : fixed;
  InputFromPME_1.T   : fixed;
  InputFromPME_1.P   : fixed;
  InputFromPME_1.h   : fixed;
  InputFromPME_1.V   : fixed;
  InputFromPME_1.z   : fixed;
  InputFromPME_1.Connected : "FALSE";

```

```

endif

```

```

if InputFromPME_2.IsConnected then

```

```

  InputFromPME_2.F   : fixed;

```

```

InputFromPME_2.T           : fixed;
InputFromPME_2.P           : fixed;
InputFromPME_2.h           : fixed;
InputFromPME_2.V           : fixed;
InputFromPME_2.z           : fixed;
InputFromPME_2.Connected   : "TRUE";
else
InputFromPME_2.F           : fixed;
InputFromPME_2.T           : fixed;
InputFromPME_2.P           : fixed;
InputFromPME_2.h           : fixed;
InputFromPME_2.V           : fixed;
InputFromPME_2.z           : fixed;
InputFromPME_2.Connected   : "FALSE";
endif

if InputFromPME_3.IsConnected then
InputFromPME_3.F           : fixed;
InputFromPME_3.T           : fixed;
InputFromPME_3.P           : fixed;
InputFromPME_3.h           : fixed;
InputFromPME_3.V           : fixed;
InputFromPME_3.z           : fixed;
InputFromPME_3.Connected   : "TRUE";
else
InputFromPME_3.F           : fixed;
InputFromPME_3.T           : fixed;
InputFromPME_3.P           : fixed;
InputFromPME_3.h           : fixed;
InputFromPME_3.V           : fixed;
InputFromPME_3.z           : fixed;
InputFromPME_3.Connected   : "FALSE";
endif

if InputFromPME_4.IsConnected then
InputFromPME_4.F           : fixed;
InputFromPME_4.T           : fixed;
InputFromPME_4.P           : fixed;
InputFromPME_4.h           : fixed;
InputFromPME_4.V           : fixed;
InputFromPME_4.z           : fixed;
InputFromPME_4.Connected   : "TRUE";
else
InputFromPME_4.F           : fixed;
InputFromPME_4.T           : fixed;
InputFromPME_4.P           : fixed;
InputFromPME_4.h           : fixed;
InputFromPME_4.V           : fixed;
InputFromPME_4.z           : fixed;
InputFromPME_4.Connected   : "FALSE";
endif

if InputFromPME_5.IsConnected then
InputFromPME_5.F           : fixed;
InputFromPME_5.T           : fixed;
InputFromPME_5.P           : fixed;

```

```

InputFromPME_5.h           : fixed;
InputFromPME_5.V           : fixed;
InputFromPME_5.z           : fixed;
InputFromPME_5.Connected   : "TRUE";
else
InputFromPME_5.F           : fixed;
InputFromPME_5.T           : fixed;
InputFromPME_5.P           : fixed;
InputFromPME_5.h           : fixed;
InputFromPME_5.V           : fixed;
InputFromPME_5.z           : fixed;
InputFromPME_5.Connected   : "FALSE";
endif

if InputFromPME_6.IsConnected then
InputFromPME_6.F           : fixed;
InputFromPME_6.T           : fixed;
InputFromPME_6.P           : fixed;
InputFromPME_6.h           : fixed;
InputFromPME_6.V           : fixed;
InputFromPME_6.z           : fixed;
InputFromPME_6.Connected   : "TRUE";
else
InputFromPME_6.F           : fixed;
InputFromPME_6.T           : fixed;
InputFromPME_6.P           : fixed;
InputFromPME_6.h           : fixed;
InputFromPME_6.V           : fixed;
InputFromPME_6.z           : fixed;
InputFromPME_6.Connected   : "FALSE";
endif

if InputFromPME_7.IsConnected then
InputFromPME_7.F           : fixed;
InputFromPME_7.T           : fixed;
InputFromPME_7.P           : fixed;
InputFromPME_7.h           : fixed;
InputFromPME_7.V           : fixed;
InputFromPME_7.z           : fixed;
InputFromPME_7.Connected   : "TRUE";
else
InputFromPME_7.F           : fixed;
InputFromPME_7.T           : fixed;
InputFromPME_7.P           : fixed;
InputFromPME_7.h           : fixed;
InputFromPME_7.V           : fixed;
InputFromPME_7.z           : fixed;
InputFromPME_7.Connected   : "FALSE";
endif

if InputFromPME_8.IsConnected then
InputFromPME_8.F           : fixed;
InputFromPME_8.T           : fixed;
InputFromPME_8.P           : fixed;
InputFromPME_8.h           : fixed;
InputFromPME_8.V           : fixed;

```

```

    InputFromPME_8.z          : fixed;
    InputFromPME_8.Connected  : "TRUE";
else
    InputFromPME_8.F          : fixed;
    InputFromPME_8.T          : fixed;
    InputFromPME_8.P          : fixed;
    InputFromPME_8.h          : fixed;
    InputFromPME_8.V          : fixed;
    InputFromPME_8.z          : fixed;
    InputFromPME_8.Connected  : "FALSE";
endif

if InputFromPME_9.IsConnected then
    InputFromPME_9.F          : fixed;
    InputFromPME_9.T          : fixed;
    InputFromPME_9.P          : fixed;
    InputFromPME_9.h          : fixed;
    InputFromPME_9.V          : fixed;
    InputFromPME_9.z          : fixed;
    InputFromPME_9.Connected  : "TRUE";
else
    InputFromPME_9.F          : fixed;
    InputFromPME_9.T          : fixed;
    InputFromPME_9.P          : fixed;
    InputFromPME_9.h          : fixed;
    InputFromPME_9.V          : fixed;
    InputFromPME_9.z          : fixed;
    InputFromPME_9.Connected  : "FALSE";
endif

if InputFromPME_10.IsConnected then
    InputFromPME_10.F         : fixed;
    InputFromPME_10.T         : fixed;
    InputFromPME_10.P         : fixed;
    InputFromPME_10.h         : fixed;
    InputFromPME_10.V         : fixed;
    InputFromPME_10.z         : fixed;
    InputFromPME_10.Connected : "TRUE";
else
    InputFromPME_10.F         : fixed;
    InputFromPME_10.T         : fixed;
    InputFromPME_10.P         : fixed;
    InputFromPME_10.h         : fixed;
    InputFromPME_10.V         : fixed;
    InputFromPME_10.z         : fixed;
    InputFromPME_10.Connected : "FALSE";
endif

// collection of outlet ports (arrays of ports will not export to Aspen Plus)
iNrOutletPorts          as hidden      integerparameter (10);
iOutletPortIndices      as hidden      integerset ([1:iNrOutletPorts]);
arrayOutletPortNames(iOutletPortIndices) as hidden      stringparameter;

OutputToPME_1          as output      APECSMoleFractionPort;
OutputToPME_2          as output      APECSMoleFractionPort;
OutputToPME_3          as output      APECSMoleFractionPort;

```

```

OutputToPME_4      as output  APECSMoleFractionPort;
OutputToPME_5      as output  APECSMoleFractionPort;
OutputToPME_6      as output  APECSMoleFractionPort;
OutputToPME_7      as output  APECSMoleFractionPort;
OutputToPME_8      as output  APECSMoleFractionPort;
OutputToPME_9      as output  APECSMoleFractionPort;
OutputToPME_10     as output  APECSMoleFractionPort;

```

```

if OutputToPME_1.IsConnected then
  OutputToPME_1.F      : free;
  OutputToPME_1.T      : free;
  OutputToPME_1.P      : free;
  OutputToPME_1.h      : free;
  OutputToPME_1.V      : free;
  OutputToPME_1.z      : free;
  OutputToPME_1.Connected : "TRUE";

```

```

else
  OutputToPME_1.F      : free;
  OutputToPME_1.T      : free;
  OutputToPME_1.P      : free;
  OutputToPME_1.h      : free;
  OutputToPME_1.V      : free;
  OutputToPME_1.z      : free;
  OutputToPME_1.Connected : "FALSE";

```

```

endif

```

```

if OutputToPME_2.IsConnected then
  OutputToPME_2.F      : free;
  OutputToPME_2.T      : free;
  OutputToPME_2.P      : free;
  OutputToPME_2.h      : free;
  OutputToPME_2.V      : free;
  OutputToPME_2.z      : free;
  OutputToPME_2.Connected : "TRUE";

```

```

else
  OutputToPME_2.F      : free;
  OutputToPME_2.T      : free;
  OutputToPME_2.P      : free;
  OutputToPME_2.h      : free;
  OutputToPME_2.V      : free;
  OutputToPME_2.z      : free;
  OutputToPME_2.Connected : "FALSE";

```

```

endif

```

```

if OutputToPME_3.IsConnected then
  OutputToPME_3.F      : free;
  OutputToPME_3.T      : free;
  OutputToPME_3.P      : free;
  OutputToPME_3.h      : free;
  OutputToPME_3.V      : free;
  OutputToPME_3.z      : free;
  OutputToPME_3.Connected : "TRUE";

```

```

else
  OutputToPME_3.F      : free;
  OutputToPME_3.T      : free;
  OutputToPME_3.P      : free;

```

```

OutputToPME_3.h           : free;
OutputToPME_3.V           : free;
OutputToPME_3.z           : free;
OutputToPME_3.Connected  : "FALSE";
endif

if OutputToPME_4.IsConnected then
OutputToPME_4.F           : free;
OutputToPME_4.T           : free;
OutputToPME_4.P           : free;
OutputToPME_4.h           : free;
OutputToPME_4.V           : free;
OutputToPME_4.z           : free;
OutputToPME_4.Connected  : "TRUE";
else
OutputToPME_4.F           : free;
OutputToPME_4.T           : free;
OutputToPME_4.P           : free;
OutputToPME_4.h           : free;
OutputToPME_4.V           : free;
OutputToPME_4.z           : free;
OutputToPME_4.Connected  : "FALSE";
endif

if OutputToPME_5.IsConnected then
OutputToPME_5.F           : free;
OutputToPME_5.T           : free;
OutputToPME_5.P           : free;
OutputToPME_5.h           : free;
OutputToPME_5.V           : free;
OutputToPME_5.z           : free;
OutputToPME_5.Connected  : "TRUE";
else
OutputToPME_5.F           : free;
OutputToPME_5.T           : free;
OutputToPME_5.P           : free;
OutputToPME_5.h           : free;
OutputToPME_5.V           : free;
OutputToPME_5.z           : free;
OutputToPME_5.Connected  : "FALSE";
endif

if OutputToPME_6.IsConnected then
OutputToPME_6.F           : free;
OutputToPME_6.T           : free;
OutputToPME_6.P           : free;
OutputToPME_6.h           : free;
OutputToPME_6.V           : free;
OutputToPME_6.z           : free;
OutputToPME_6.Connected  : "TRUE";
else
OutputToPME_6.F           : free;
OutputToPME_6.T           : free;
OutputToPME_6.P           : free;
OutputToPME_6.h           : free;
OutputToPME_6.V           : free;

```



```

OutputToPME_6.z           : free;
OutputToPME_6.Connected  : "FALSE";
endif

if OutputToPME_7.IsConnected then
OutputToPME_7.F           : free;
OutputToPME_7.T           : free;
OutputToPME_7.P           : free;
OutputToPME_7.h           : free;
OutputToPME_7.V           : free;
OutputToPME_7.z           : free;
OutputToPME_7.Connected  : "TRUE";
else
OutputToPME_7.F           : free;
OutputToPME_7.T           : free;
OutputToPME_7.P           : free;
OutputToPME_7.h           : free;
OutputToPME_7.V           : free;
OutputToPME_7.z           : free;
OutputToPME_7.Connected  : "FALSE";
endif

if OutputToPME_8.IsConnected then
OutputToPME_8.F           : free;
OutputToPME_8.T           : free;
OutputToPME_8.P           : free;
OutputToPME_8.h           : free;
OutputToPME_8.V           : free;
OutputToPME_8.z           : free;
OutputToPME_8.Connected  : "TRUE";
else
OutputToPME_8.F           : free;
OutputToPME_8.T           : free;
OutputToPME_8.P           : free;
OutputToPME_8.h           : free;
OutputToPME_8.V           : free;
OutputToPME_8.z           : free;
OutputToPME_8.Connected  : "FALSE";
endif

if OutputToPME_9.IsConnected then
OutputToPME_9.F           : free;
OutputToPME_9.T           : free;
OutputToPME_9.P           : free;
OutputToPME_9.h           : free;
OutputToPME_9.V           : free;
OutputToPME_9.z           : free;
OutputToPME_9.Connected  : "TRUE";
else
OutputToPME_9.F           : free;
OutputToPME_9.T           : free;
OutputToPME_9.P           : free;
OutputToPME_9.h           : free;
OutputToPME_9.V           : free;
OutputToPME_9.z           : free;
OutputToPME_9.Connected  : "FALSE";
endif

```

```

endif

if OutputToPME_10.IsConnected then
  OutputToPME_10.F      : free;
  OutputToPME_10.T      : free;
  OutputToPME_10.P      : free;
  OutputToPME_10.h      : free;
  OutputToPME_10.V      : free;
  OutputToPME_10.z      : free;
  OutputToPME_10.Connected : "TRUE";
else
  OutputToPME_10.F      : free;
  OutputToPME_10.T      : free;
  OutputToPME_10.P      : free;
  OutputToPME_10.h      : free;
  OutputToPME_10.V      : free;
  OutputToPME_10.z      : free;
  OutputToPME_10.Connected : "FALSE";
endif

// declare an instance of the submodel "APECS_BLOCK"
connectionToAPECS      as hidden      APECS_BLOCK;
connectionToAPECS.stringBlockName : stringBlockName;
connectionToAPECS.iNrInletPorts   : iNrInletPorts;           // passing all ports to submodel
connectionToAPECS.iNrOutletPorts  : iNrOutletPorts;

for iPort in iInletPortIndices do
  if (iPort == 1) then
    link connectionToAPECS.InputFromPME(iPort) and InputFromPME_1;
    connectionToAPECS.InputFromPME(iPort).PortName : InputFromPME_1.PortName;
    connectionToAPECS.InputFromPME(iPort).Connected : InputFromPME_1.Connected;
    connectionToAPECS.InputFromPME(iPort).F        : InputFromPME_1.F;
    connectionToAPECS.InputFromPME(iPort).T        : InputFromPME_1.T;
    connectionToAPECS.InputFromPME(iPort).P        : InputFromPME_1.P;
    connectionToAPECS.InputFromPME(iPort).h        : InputFromPME_1.h;
    connectionToAPECS.InputFromPME(iPort).V        : InputFromPME_1.V;
    connectionToAPECS.InputFromPME(iPort).z        : InputFromPME_1.z;
  endif
  if (iPort == 2) then
    link connectionToAPECS.InputFromPME(iPort) and InputFromPME_2;
    connectionToAPECS.InputFromPME(iPort).PortName : InputFromPME_2.PortName;
    connectionToAPECS.InputFromPME(iPort).Connected : InputFromPME_2.Connected;
    connectionToAPECS.InputFromPME(iPort).F        : InputFromPME_2.F;
    connectionToAPECS.InputFromPME(iPort).T        : InputFromPME_2.T;
    connectionToAPECS.InputFromPME(iPort).P        : InputFromPME_2.P;
    connectionToAPECS.InputFromPME(iPort).h        : InputFromPME_2.h;
    connectionToAPECS.InputFromPME(iPort).V        : InputFromPME_2.V;
    connectionToAPECS.InputFromPME(iPort).z        : InputFromPME_2.z;
  endif
  if (iPort == 3) then
    link connectionToAPECS.InputFromPME(iPort) and InputFromPME_3;
    connectionToAPECS.InputFromPME(iPort).PortName : InputFromPME_3.PortName;
    connectionToAPECS.InputFromPME(iPort).Connected : InputFromPME_3.Connected;
    connectionToAPECS.InputFromPME(iPort).F        : InputFromPME_3.F;
    connectionToAPECS.InputFromPME(iPort).T        : InputFromPME_3.T;
    connectionToAPECS.InputFromPME(iPort).P        : InputFromPME_3.P;
  endif
endfor

```



```

    connectionToAPECS.InputFromPME(iPort).V      : InputFromPME_8.V;
    connectionToAPECS.InputFromPME(iPort).z      : InputFromPME_8.z;
endif
if (iPort == 9) then
    link connectionToAPECS.InputFromPME(iPort) and InputFromPME_9;
    connectionToAPECS.InputFromPME(iPort).PortName : InputFromPME_9.PortName;
    connectionToAPECS.InputFromPME(iPort).Connected : InputFromPME_9.Connected;
    connectionToAPECS.InputFromPME(iPort).F      : InputFromPME_9.F;
    connectionToAPECS.InputFromPME(iPort).T      : InputFromPME_9.T;
    connectionToAPECS.InputFromPME(iPort).P      : InputFromPME_9.P;
    connectionToAPECS.InputFromPME(iPort).h      : InputFromPME_9.h;
    connectionToAPECS.InputFromPME(iPort).V      : InputFromPME_9.V;
    connectionToAPECS.InputFromPME(iPort).z      : InputFromPME_9.z;
endif
if (iPort == 10) then
    link connectionToAPECS.InputFromPME(iPort) and InputFromPME_10;
    connectionToAPECS.InputFromPME(iPort).PortName : InputFromPME_10.PortName;
    connectionToAPECS.InputFromPME(iPort).Connected : InputFromPME_10.Connected;
    connectionToAPECS.InputFromPME(iPort).F      : InputFromPME_10.F;
    connectionToAPECS.InputFromPME(iPort).T      : InputFromPME_10.T;
    connectionToAPECS.InputFromPME(iPort).P      : InputFromPME_10.P;
    connectionToAPECS.InputFromPME(iPort).h      : InputFromPME_10.h;
    connectionToAPECS.InputFromPME(iPort).V      : InputFromPME_10.V;
    connectionToAPECS.InputFromPME(iPort).z      : InputFromPME_10.z;
endif
endif
endfor

for iPort in iOutletPortIndices do
    if (iPort == 1) then
        link connectionToAPECS.OutputToPME(iPort) and OutputToPME_1;
        connectionToAPECS.OutputToPME(iPort).PortName : OutputToPME_1.PortName;
        connectionToAPECS.OutputToPME(iPort).Connected : OutputToPME_1.Connected;
    endif
    if (iPort == 2) then
        link connectionToAPECS.OutputToPME(iPort) and OutputToPME_2;
        connectionToAPECS.OutputToPME(iPort).PortName : OutputToPME_2.PortName;
        connectionToAPECS.OutputToPME(iPort).Connected : OutputToPME_2.Connected;
    endif
    if (iPort == 3) then
        link connectionToAPECS.OutputToPME(iPort) and OutputToPME_3;
        connectionToAPECS.OutputToPME(iPort).PortName : OutputToPME_3.PortName;
        connectionToAPECS.OutputToPME(iPort).Connected : OutputToPME_3.Connected;
    endif
    if (iPort == 4) then
        link connectionToAPECS.OutputToPME(iPort) and OutputToPME_4;
        connectionToAPECS.OutputToPME(iPort).PortName : OutputToPME_4.PortName;
        connectionToAPECS.OutputToPME(iPort).Connected : OutputToPME_4.Connected;
    endif
    if (iPort == 5) then
        link connectionToAPECS.OutputToPME(iPort) and OutputToPME_5;
        connectionToAPECS.OutputToPME(iPort).PortName : OutputToPME_5.PortName;
        connectionToAPECS.OutputToPME(iPort).Connected : OutputToPME_5.Connected;
    endif
    if (iPort == 6) then
        link connectionToAPECS.OutputToPME(iPort) and OutputToPME_6;
        connectionToAPECS.OutputToPME(iPort).PortName : OutputToPME_6.PortName;
    endif
endfor

```

```

    connectionToAPECS.OutputToPME(iPort).Connected      :   OutputToPME_6.Connected;
endif
if (iPort == 7) then
    link connectionToAPECS.OutputToPME(iPort)    and   OutputToPME_7;
    connectionToAPECS.OutputToPME(iPort).PortName :   OutputToPME_7.PortName;
    connectionToAPECS.OutputToPME(iPort).Connected :   OutputToPME_7.Connected;
endif
if (iPort == 8) then
    link connectionToAPECS.OutputToPME(iPort)    and   OutputToPME_8;
    connectionToAPECS.OutputToPME(iPort).PortName :   OutputToPME_8.PortName;
    connectionToAPECS.OutputToPME(iPort).Connected :   OutputToPME_8.Connected;
endif
if (iPort == 9) then
    link connectionToAPECS.OutputToPME(iPort)    and   OutputToPME_9;
    connectionToAPECS.OutputToPME(iPort).PortName :   OutputToPME_9.PortName;
    connectionToAPECS.OutputToPME(iPort).Connected :   OutputToPME_9.Connected;
endif
if (iPort == 10) then
    link connectionToAPECS.OutputToPME(iPort)    and   OutputToPME_10;
    connectionToAPECS.OutputToPME(iPort).PortName :   OutputToPME_10.PortName;
    connectionToAPECS.OutputToPME(iPort).Connected :   OutputToPME_10.Connected;
endif
endifor

```