Study of Transient Neutronic-Thermalhydraulic Coupling: A Case Study Using a Backbone Interface

Y. Liu[1] and T. Xu[2]
Fuel and Fuel Channel Safety Branch, Canadian Nuclear Laboratories, Chalk River, Ontario, Canada K0J 1J0
Thermalhydraulics Branch, Canadian Nuclear Laboratories, Chalk River, Ontario, Canada K0J 1J0

yu.liu@cnl.ca; tao.xu@cnl.ca

Abstract

Multi-scale, multi-physics computer simulations are increasingly utilized to enhance the predictive capabilities of transient behaviour in nuclear reactor operation to improve performance and safety. As a result, multiple computer simulations from different disciplines must exchange data during run-times to address these inter-dependencies. The Backbone, currently under development at the Canadian Nuclear Laboratories (CNL), is a hybrid code coupling framework using both Common Object Request Broker Architecture (CORBA) and Message Passing Interface (MPI) systems. Recently, major advancements have been achieved by synchronizing the coupling codes through the Time Step Controller (TSC). This paper presents progress made in enhancing performance of code coupling between the thermal-hydraulics code CATHENA with the neutron transport code 3D-Solver via the Backbone by comparing the simulation results to experimental data as well as the CPU time of the simulations. In addition, the strategies evaluated include a simple heuristic method and two more advanced third order controllers (the PID and General Third Order TSC). Both the PID and General Third Order TSC have demonstrated a performance gain of at least 100 times greater than non-TSC coupling.

Keywords: Code Coupling, Thermal-hydraulics, Neutron Transport, CORBA, MPI

1. Introduction

With the increasing availability of affordable high performance computing resources, coupling multiple physical phenomena together in an integrated computational framework becomes an increasingly attractive prospect to enhance the predictive capability of nuclear reactor behaviour. This requires capturing many physical interactions between complex phenomena through rigorous coupling at the time step level [1]. In addition, multi-scale multi-physics computer simulations are increasingly demanded to enhance the capabilities of transient prediction in nuclear reactor performance and safety analyses, multiple computer simulations of different disciplines must also exchange data during run-time to address these inter-dependencies. To match these simulation demands, a new framework called “Backbone” is being developed at the Canadian Nuclear Laboratories (CNL) to couple a suite of reactor performance and safety analysis codes.

Although many tools or platforms (e.g., MOOSE [2], LIME [3], SALOME [4]) have been developed for code coupling, the capabilities and intended applications of each code differ. First, Backbone is a parallel and distributed system for enhanced predictions of nuclear reactor behaviour through multi-physics code coupling. The coupled codes exchange pertinent data at each time step during execution, while each code involved still solves its own problem independently. Second, the Backbone is a completely code independent framework, and all the data is transferred through a data stream, which is transparent to the Backbone. The Backbone framework is based on a hybrid MPI [5] and CORBA [6] design. So, this design has the capability of offering the advantages of both parallel and distributed communication paradigms. In code coupling, different codes can be distributed to different nodes in the system and each code can also be processed in parallel. Moreover, unlike MOOSE and LIME, the Backbone's data is mesh-independent, partially because data transfer tends to occur at temporal-spatial boundaries where meshes do not overlap.

Co-ordination and synchronization of individual simulation time steps of codes coupled through the Backbone are both essential and complex. On the other hand, it is quite valuable to perform such simulations in a reasonable time frame, which makes an efficient Time Step Controller (TSC) highly desirable in code coupling. A previous example of the importance of good adaptive TSC used for nuclear reactor kinematics is described by Cheon [7], yielding a significant performance improvement over fixed methods, while retaining acceptable margins of error. As a result, the authors have developed a centralized TSC, offering a few different step selection strategies. A number of classes of TSC are outlined by Soderlind [8], and are used as the basis for our configurable PID and General Third Order TSC, which are utilized in the evaluation of this paper.

First, the architecture and design strategies of Backbone and its TSC are described, and then attention is focused on studying the transient neutronic-thermalhydraulic coupling between the Canadian Algorithm for Thermalhydraulic Network Analysis (CATHENA) [9] and the neutron transport code 3D-Solver via the Backbone. In this study, the authors evaluate three numerical strategies of controlling time step selection to synchronize CATHENA with 3D-Solver by comparing the simulation results to the experimental data as well as the CPU time of the simulations under different TSCs. In addition, the strategies evaluated include a simple heuristic method and two more advanced third order methods. Note that this coupling solution is sufficiently flexible to be applied to nuclear simulation codes and other non-nuclear scientific software requiring multi-physics coupling.

2. Backbone

2.1 General Architecture

As mentioned above, the Backbone design is based on a combination of CORBA and MPI. First of all, the Backbone treats the coupled codes as blackboxes, while in any of the previously mentioned CORBA/MPI work, knowledge of the internal structure of the code and access to the source code is required in order to perform the decomposition of the program. Second, the

Backbone is based on an integration of CORBA and MPI technologies, but the codes being coupled are not CORBA objects. These codes are either pure MPI-based codes, which are the case for many scientific codes, or they may use an MPI-based wrapper in order for the code to communicate with the Backbone using MPI. Since MPI's main strength is to enable parallelism and CORBA is designed for distributed programming, the goal was to create a server that can run parallel programs in CORBA's client-server application.

An MPI process management interface was used for this implementation. The interface allows a CORBA application to spawn one or more multiple parallel MPI processes. This approach has a number of advantages over a pure CORBA-based application. One such advantage is that managing the scope of the data becomes a much simpler task than would otherwise be the case with a pure CORBA application, since MPI has a direct Fortran language implementation. Another advantage of a CORBA-MPI application using an MPI process management interface is that in the case where no coupling is needed, the code can be executed in stand-alone mode without using the CORBA middleware.

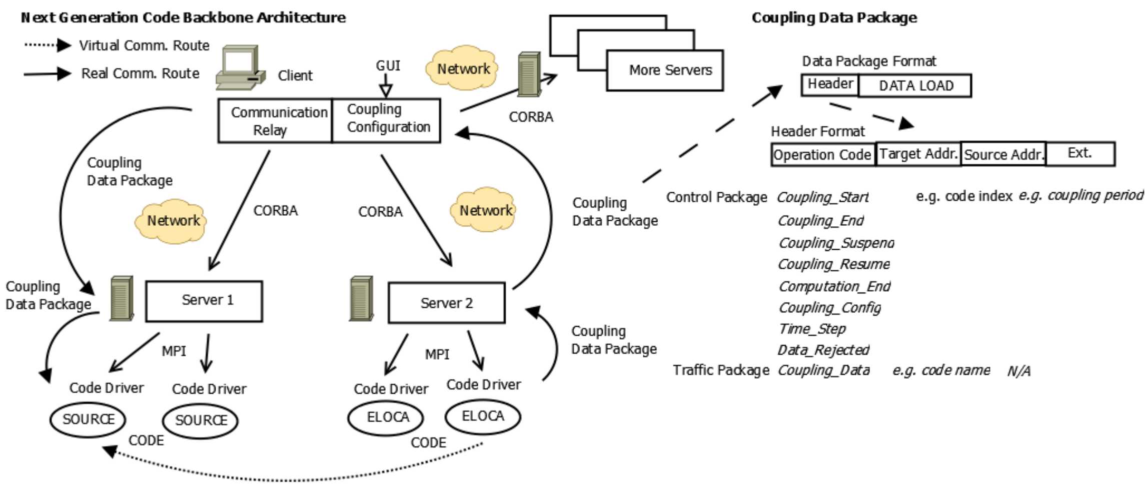

Figure 1 The Infrastructure of the Backbone.

The detailed infrastructure and communication protocol of the Backbone is described in Figure 1. The codes are spawned by the Backbone as separate processes, and then their coupling are coordinated by the TSC. During the progress of code coupling, Backbone checks if it needs to reset data to the start of the time step, handles that if necessary, then if it has permission to run the next time step (or it is just a mere reset), it executes its calculation for the current time step. The code then checks if it needs to issue a reset to other codes (i.e., due to unacceptable error, failure to converge or similar problems), does so if necessary and discarding its data for that particular time-step. Otherwise, it transfers its data and waits for the next message from the Backbone, which will either be a reset of the step has been declared for a shorter time step, or a request to execute another step.

2.2 Time Step Controller

The Backbone is a framework designed to allow the coupling of disparate simulation codes through well defined interfaces, presently consisting of arrays of double-precision floating point numbers. Codes accept sets of numbers, execute, and then return other numbers representing the parts of their state relevant to other codes, which then execute in turn using that information. With the addition of the TSC, the Backbone offers superior control of error, improved robustness (i.e., to dynamically adjust to chaotic or stable patches of simulation) and efficiency (i.e., decelerating at times of rapid change or acceleration over stable periods). The TSC is an essential component of the Backbone, ensuring that coupled codes proceed only with data for their next time-steps, and limiting estimated error of those codes for their various parameters. It also makes configuring time steps for a coupled simulation a simple run-time decision.

Time Step Controller supports a number of strategies, different functions to govern time step selection based on user specified inputs. One is able to employ a simple controller, a H312 PID controller, or a General Third Order controller. The three controllers that are implemented in the Backbone each have distinctive origins and behaviours. The simple controller is a heuristic one; it increases the step size each successful step (a time step accepted by the code based on convergence criteria), and decreases it each unsuccessful step (a time step rejected by the code also based on the same convergence criteria). The size of the increase and the decrease is pre-defined and is assigned a fixed value. This yields a controller that increases the time step size very slowly and smoothly, and decreases at a moderate rate. In comparison, the H312 PID and General Third Order controllers are more complex configurations and are able to adapt fairly aggressively to changes in the numerical system with an optimal time step size. The H312 PID controller is defined by Equation 1, and the General Third Order controller is defined by Equation 2 below.

$$ts_{i+1} = \left(\frac{err_{i-1}}{err_i}\right)^{b_1}\left(\frac{TOL}{err_i}\right)^{b_2}\left(\frac{(err_{i-1})^2}{err_i(err_{i-2})}\right)^{b_3} ts_i \quad (1)$$

$$ts_{i+1} = \left(\frac{(TOL)(\gamma)}{err_i}\right)^{b_1}\left(\frac{TOL(\gamma)}{err_{i-1}}\right)^{b_2}\left(\frac{TOL(\gamma)}{err_{i-2}}\right)^{b_3}\left(\frac{ts_{i-1}}{ts_i}\right)^{a_1}\left(\frac{ts_{i-2}}{ts_{i-1}}\right)^{a_2} ts_i \quad (2)$$

where TOL is error tolerance, γ is the safety factor, usually between 0.8 and 1, ts_i is the size of the i^{th} time step, err_i is the magnitude of error in the i^{th} step (specifically, the greatest magnitude of error for all coupled variables of all codes, and error is usually the convergence error in the code), a_n and b_n are configurable parameters (described in this paper as “step weight” and “error weight”, respectively). Note that the H312 PID TSC is a special case of the Third Order TSC.

3. Transient Neutronic-Thermalhydraulic Coupling

CATHENA is a non-equilibrium, two phase network analysis code. The 3D-Solver is a prototype of a neutron transport code, which solves the time-independent multi-group neutron diffusion equation. CATHENA was developed in Fortran 95 and executes on Windows, while 3D-Solver was developed in Python and executes on Linux. One advantage of the Backbone is its ability to couple codes across platforms, which was the case herein. During execution, CATHENA provided 3D-Solver with an array of values corresponding to coolant density, coolant temperature and average fuel temperatures at each time step for each fuel site. This data was used as input by the 3D-Solver, which returns CATHENA the appropriate neutronic power coefficient corresponding to the thermalhydraulics conditions received. CATHENA, upon receiving the new power coefficient, updates the internal heat generation rate in its fuel model and continues to the next time step with updated fuel power. This transient neutronic-thermalhydraulic coupling is managed by the Backbone and the synchronization interval is coordinated through the TSC of Backbone, which is demonstrated in Figure 2.

Note that the 3D-Solver is a prototype code under development at CNL. It solves equations that are independent of time. However, these calculations are based on the coupling data from CATHENA, such as temperature, density and etc, which are time dependent. Since the coupling data is time dependent for transient problems, 3D-solver's computing results effectively become time dependent as well. The main purpose of this work is to test the performance of Backbone with TSC under "blowdown". In the future, a more realistic coupling (e.g. CATHENA vs. NESTLE-C) would be implemented to do the reactor transient simulation.

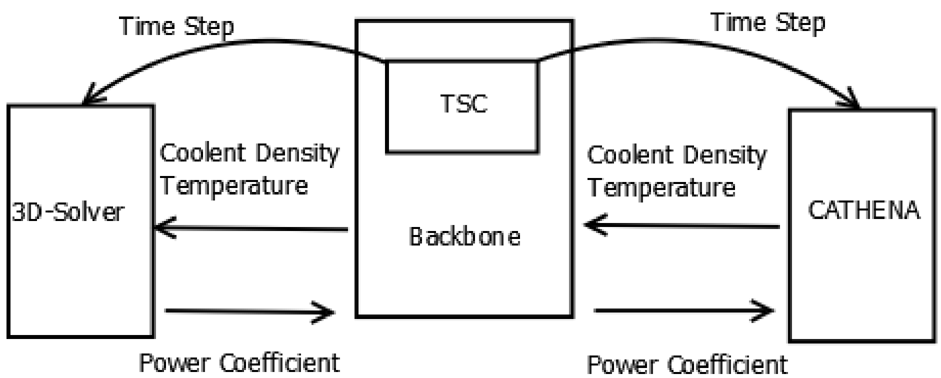

Figure 2 Transient Neutronic-Thermalhydraulic Coupling through the Backbone.

The fast depressurization of the pressure vessel or piping systems containing subcooled or saturated liquids is of large interest for safety analyses. A "blow-down" process might originate by a structural failure or by an operational opening of safety valves to prevent damage to the plant [10]. Thus, the show case of coupling in this paper is to simulate the Edwards Blowdown Problem, which was performed at the Atomic Weapons Research Establishment, Foulness, U.K.

The test facility consisted of a straight pipe 4.096 m in length and 73 mm internal diameter. First, the pipe was mounted horizontally and supported in such a manner as to permit small axial movements, e.g., thermal expansion and end thrust. Second, one end of the pipe was closed and connected to a load cell and a concrete abutment to allow measurement of end thrust. The other end of the pipe was sealed by a toughened glass disc that could be ruptured by the impact of a steel-tipped lead pellet from a compressed air gun to initiate the test. Third, the pipe was connected to a remotely operated valve to a reservoir of de-mineralized water and a pump that was used to establish the initial fluid conditions in the pipe. The remotely operated valve enabled the blow-down pipe to be isolated from the reservoir just prior to rupturing the glass disc. In addition, the pipes were electrically heated by sheathed band heaters, formed to cover about 70% of the pipe circumference. The heaters were spaced about 13 mm apart and had a total heat rate of 700 watts. Specially shaped heaters were used at the two void fraction measurement locations and for the rupture disc-mounting. The heaters were connected in ten groups, each supplied with separate voltage controls so that reasonably uniform axial temperature conditions could be maintained. To further reduce heat losses, the pipes were insulated with a 25.4 mm thick layer of asbestos [11]. More details of the Edwards Blowdown Experiment described by Mallory in [11] and are shown in Figure 3.

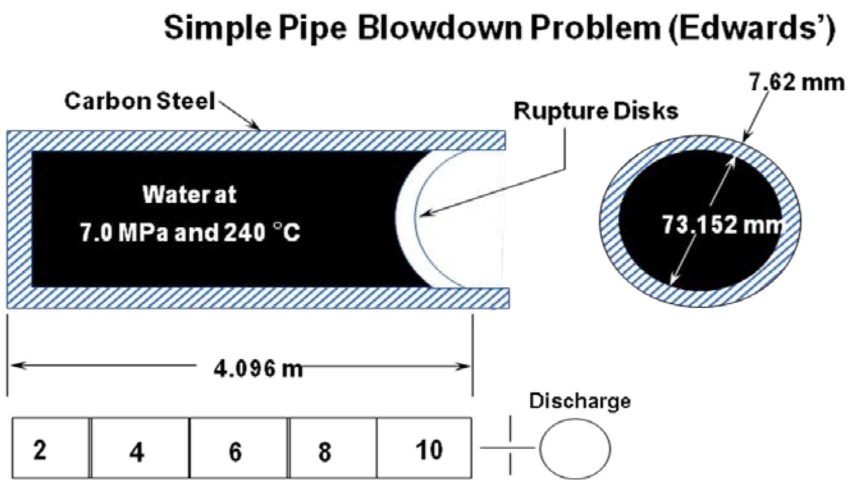

Figure 3 Illustration of Broken Location and Size of Edwards Blowdown Problem [11].

4. Evaluation Methodology

The authors evaluate the performance of the transient neutronic-thermalhydraulic coupling in TSC mode and non-TSC mode. In the TSC mode, three different TSCs are used to synchronize the coupling, including the simple TSC, the H312 PID TSC, and the General Third Order TSC. Also, in the non-TSC mode, CATHENA (the master code) drives the 3D-Solver (the slave code) to achieve the synchronization, in which the time step is determined by the CATHENA and forces 3D-Solver to follow it.

As mentioned in Section 3, the Edwards Blowdown test is used as the test case, and both the short-term and long-term cases have been evaluated in this work. The simulation results are compared to the experimental results to validate the correctness of the simulations. Moreover, the simulation performance has been evaluated through comparing the simulation time elapsed on the same computers involved (a Windows and a Linux based workstation) to test the efficiency of each method. The authors use the experimental data from the Edwards blowdown test from the experiment performed at the Atomic Weapons Research Establishment for the United Kingdom Atomic Energy Authority. To be consistent to this experiment, the following conditions were set for the simulation using the CATHENA. In addition, the experiment performed at AWRE was based on a fixed temperature as below. In other words, no neutronic effects are involved in the Edwards blowdown experiment. The code coupling between thermalhydraulic and neutron diffusion code in this work is only used for the initial evaluation of the Backbone. In order to match this experiment, the authors have to use the same fixed temperature in the CATHENA simulation, although 3D-Solver uses the coupling data from

CATHENA and returns the power coefficient to CATHENA through the Backbone. A future experiment and simulation at CNL will consider involving the neutronic effects to use the variable wall temperature in time.

- Horizontal Test Section Diameter = 0.073152 m,
- Length = 4.096 m,
- Pressure = 7 MPa, and
- Temperature = 242.7°C.

The principle TSC setting at the Backbone are shown below, including the setting for the simple, the PID, and the General Third Order TSC. These settings are valid for both CATHENA and 3D-Solver. For the simulation without TSC involvement, the same initial, maximum and minimum time step setting are put into the CATHENA input file. As a result, CATHENA will calculate its own time step using its existing time step controller and then drives 3-D-Solver for synchronization through an interface to the Backbone.

- Initial Time Step: 0.00001 s
- Maximum Time Step: 0.01 s
- Minimum Time Step: 0.000001 s
- Error Tolerance (the PID and third order controller): 0.00005
- Step-up Ratio (the simple controller): 1.001
- Step-down Radio (the simple controller): 0.1

5. Test Results

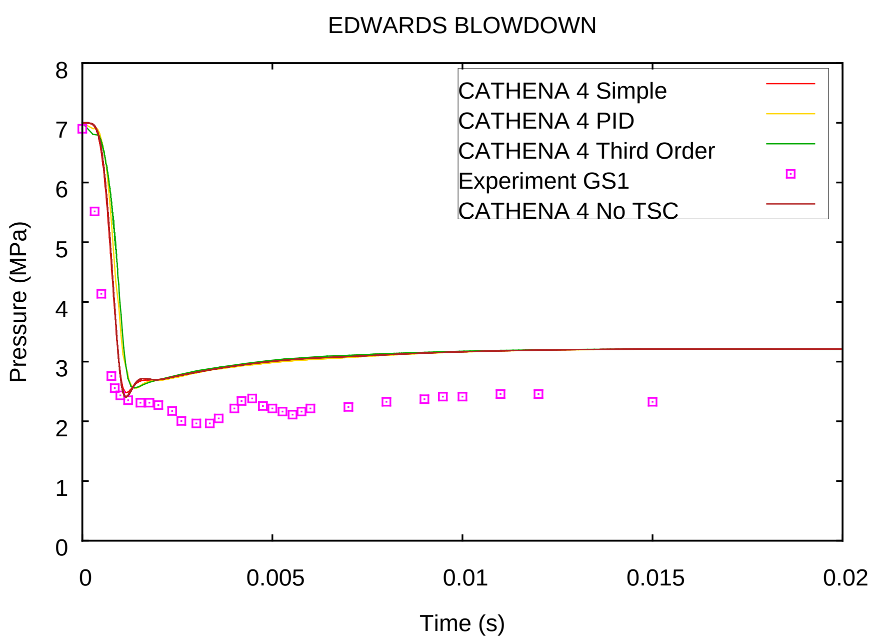

Figure 4 The Comparison of Simulated and Experimental Pipe Pressures under Edwards Blowdown Program in the Short-Term Case.

In this section, the simulation data and its analyses are presented. Figure 4 shows the pipe pressure comparison among different simulations and the experiment under Edwards Blowdown Program in the short-term case, while Figure 5 is the same type of comparison in the long-term case. Also, these simulations consist of both TSC modes with three different controllers and non-TSC mode. From Figure 4 and Figure 5, they clearly indicate that all the simulations capture the correct behaviour of the Edwards Blowdown Problem against the experimental data. When the break happens, the pressure tends to drop sharply in a very short time, and then subsequently stabilizes. In other words, the coupling simulations that used both TSC and non-TSC modes have demonstrated to be consistent and agree very well with experimental measurements. However, it is still hard to say which simulation result is better than the rest based on the following two figures.

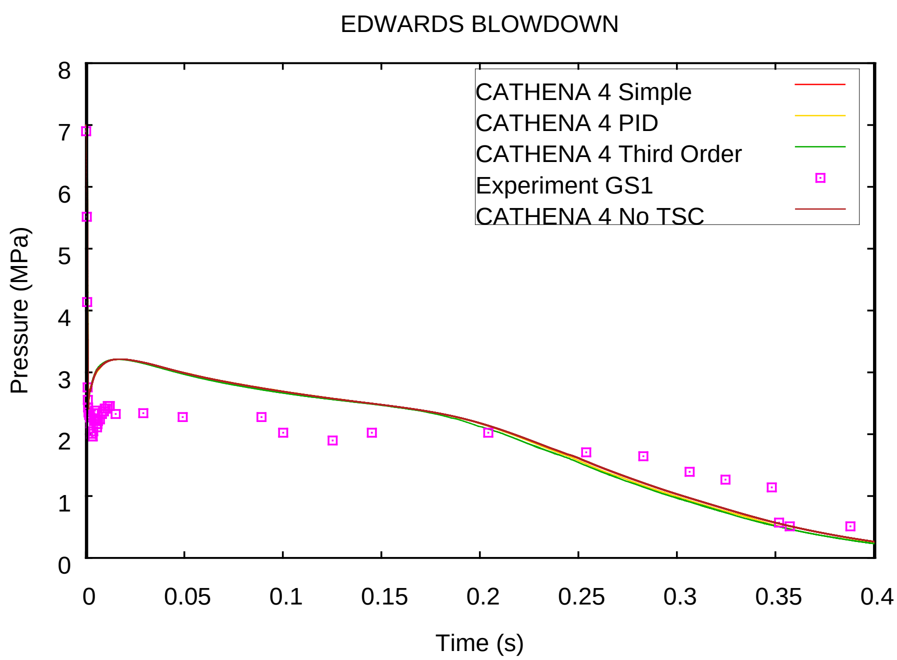

Figure 5 The Comparison of Simulated and Experimental Pipe Pressures under Edwards Blowdown Program in the Long-Term Case.

Table 1 shows the elapsed wall-clock time (CPU time) under different synchronization modes of code coupling. The authors observe the TSC mode significantly performs better than non-TSC mode. Even the simple TSC improves performance by 97%. In addition, the simulation using CATHENA's own time step took 400,001 time steps, while the simulation with the Simple TSC only took 2,863 time steps. The PID and Third Order consumes only 2,365 and 2,555 time steps respectively to obtain the similar results shown in Table 1, which implies that the algorithms employed in TSC are more efficient to predict the larger time steps and keep the numerical scheme stable during the run-time. The reason is that the Backbone TSC has the capability of optimizing the time step to the maximum accepted value based on the feedback from the code. The simple TSC is an off-line algorithm, providing fixed step-up and step-down ratio. Consequently, its performance is not as good as the adaptive TSCs (the PID and Third Order save additional 70% CPU time), which could dynamically adjust the step-up and step-down ratios according to the code feedback in the run-time. The difference between the PID and General Third Order TSC is pretty minor so that both of them could offer a great performance boost for the code coupling.

Table 1 The Comparison of Simulation Time and Number of Time Steps under Different Synchronization Modes of Codes Coupling.

	Non-TSC	Simple TSC	PID TSC	Third Order TSC
Simulation Time [s]	9,686	288	74	82
# of Time Steps	400,001	2,863	2,365	2,555

6. Conclusion

Computer simulation plays a major role in nuclear industry as well as addressing a variety of more general scientific and data analysis problems. Unlike other code coupling suites, Backbone has been designed to have the capability of coupling existing industry codes with minimum or no modification to the original source code, which may be written in different programming languages and executed on different operating systems.

In this paper, a pilot study is presented that evaluated three TSCs to synchronize the thermal-hydraulics code CATHENA with the neutron diffusion code 3D-Solver by comparing the simulation results to experimental data as well as the simulation wall-clock time. The test results indicate the PID and the General Third Order TSC greatly increase the performance while not greatly impact the accuracy of the simulation results. Both the PID and Third Order TSC have demonstrated a performance gain of at least 100 times greater than non-TSC. In the future, CNL will continue working on the Backbone toward releasing this distribution software. In addition, the neutronic efforts will be involved in the Edward blowdown experiment and the simulation in a longer transient time for more realistic applications.

7. Acknowledgement

The authors would like to thank Dr. Markus Piro and Dr. Marat Seydaliev at CNL for helpful discussions.

8. Reference

[1] K. Ivanov and M. Avramova, Progress and challenges in the development and qualification of multi-level multi-physics coupled methodologies for reactor analysis, in the Proc. of the International Congress on Advances in Nuclear Power Plants, 2008

[2] D. Gaston, C. Newman, G. Hansen and D. Lebrun-Grandie, MOOSE: A parallel computational framework for coupled systems of nonlinear equations, Nuclear Engineering and Design, 239(10) (2009) 1768-1778

[3] R. Schmidt, N. Belcourt, R. Hooper and R. Pawlowski, An Introduction to LIME 1.0 and its Use in Coupling Codes for Multiphysics Simulations, Sandia Report, SAND2011-8524, 2011

[4] A. Baudron, N. Crouzet, C. Doderlein, A. Geay, J. Lautard, E. Richebois and P. Sireta, "Unstructured 3D MINOS/FLICA4 coupling in SALOME application to JHR transient analysis", International Conference on the Physics of Reactors, 2008

[5] MPI: A Message-Passing Interface Standard Version 2.2, Message Passing Interface Forum, High Performance Computing Center, 2009

[6] Object Management Group: http://www.omg.org

[7] Cheon Bo Shim, Yeon Sang Jung, Joo IL Yoon, and Han Gyu Joo, Application of Backward Differentiation Formula to Spatial Reactor Kinetics Calculation with Adaptive Time Step Control, Nuclear Engineering And Technology, Vol. 43, No. 6, 2011 December

[8] G. Soderlind, Digital Filters in Adaptive Time-Stepping, ACM Transactions on Mathematical Software, Vol. V, No. N, 2005 October

[9] N. U. Aydemir, Recent Developments in CATHENA 4: A Fully-Implicit Network Thermalhydraulics Code, in the Proceedings of NURETH-15, Pisa, Italy, May 12-17, 2013.

[10] H. Staedtke, Gasdynamic Aspects of Two-Phase Flow: Hyperbolicity, Wave Propagation Phenomena and Related Numerical Methods, Wiley-VCH, 1st Edition, 2006 October

[11] J.P. Mallory, T.V. Tran and T.E. Macdonald, Validation of CATHENA MOD3.5b/Rev0 for the Power Pulse/Reactor Trip Phase of Large LOCA and LOCA/LOECI Edwards Blowdown and Christensens Power Void Tests, AECL Report RC-1690-3, 1998 December