ANL/MCS-TM-345

Coupled Physics Environment (CouPE) library – Design, implementation and release

Mathematics and Computer Science Division

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

Availability of This Report
This report is available, at no cost, at http://www.osti.gov/bridge. It is also available on paper to the U.S. Department of Energy and its contractors, for a processing fee, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone (865) 576-8401
fax (865) 576-5728
reports@adonis.osti.gov

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 UChicago Argonne, LLC, nor any of their employees or officers, 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 document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

ANL/MCS-TM-345

Coupled Physics Environment (CouPE) library – Design, implementation and release

Vijay S. Mahadevan
Mathematics and Computer Science Division,
Argonne National Laboratory

September 30, 2014

SUMMARY

Over several years, high fidelity, validated mono-physics solvers with proven scalability on peta-scale architectures have been developed independently. Based on a unified component-based architecture, these existing codes can be coupled with a unified mesh-data backplane and a flexible coupling-strategy-based driver suite to produce a viable tool for analysts. In this report, we present details on the design decisions and developments on CouPE, an acronym that stands for Coupled Physics Environment that orchestrates a coupled physics solver through the interfaces exposed by MOAB array-based unstructured mesh, both of which are part of SIGMA (Scalable Interfaces for Geometry and Mesh-Based Applications) toolkit.

The SIGMA toolkit contains libraries that enable scalable geometry and unstructured mesh creation and handling in a memory and computationally efficient implementation. The CouPE version being prepared for a full open-source release along with updated documentation will contain several useful examples that will enable users to start developing their applications natively using the native MOAB mesh and couple their models to existing physics applications to analyze and solve real world problems of interest.

An integrated multi-physics simulation capability for the design and analysis of current and future nuclear reactor models is also being investigated as part of the NEAMS RPL, to tightly couple neutron transport, thermal-hydraulics and structural mechanics physics under the SHARP framework. This report summarizes the efforts that have been invested in CouPE to bring together several existing physics applications namely PROTEUS (neutron transport code), Nek5000 (computational fluid-dynamics code) and Diablo (structural mechanics code). The goal of the SHARP framework is to perform fully resolved coupled physics analysis of a reactor on heterogeneous geometry, in order to reduce the overall numerical uncertainty while leveraging available computational resources.

The design of CouPE along with motivations that led to implementation choices are also discussed. The first release of the library will be different from the current version of the code that integrates the components in SHARP and explanation on the need for forking the source base will also be provided. Enhancements in the functionality and improved user guides will be available as part of the release. CouPE v0.1 is scheduled for an open-source release in December 2014 along with SIGMA v1.1 components that provide support for language-agnostic mesh loading, traversal and query interfaces along with scalable solution transfer of fields between different physics codes. The coupling methodology and software interfaces of the library are presented, along with verification studies on two representative fast sodium-cooled reactor demonstration problems to prove the usability of the CouPE library.

TABLE OF CONTENTS

Summary i

Table of Contents ii

List of Figures iii

List of Tables iii

1 Introduction 4

2 Background – Numerical methods 6

2.1 Explicit coupling strategy 6
2.2 Implicit coupling strategy 8

3 Design, components and implementation of CouPE 11

3.1 Design of CouPE 12
3.1.1 MOAB Interfaces for physics applications (use case: SHARP) 13
3.1.2 Lightweight physics wrappers in CouPE 14
3.2 Configuration and build system 15
3.2.1 Build System 16
3.3 CouPE v0.1.0 Release and notes 16

4 Some demonstration problems 17

4.1 Numerical error measurement 17
4.2 Initial conditions and transient specification 18
4.3 Simplified hexagonal fuel assembly 18
4.3.1 Solution transfer verification 20
4.3.2 Coupled pseudo-transient verification 21
4.4 Realistic benchmark problem: XX09 assembly 23
4.4.1 Coupled solution results 24

5 Conclusions 25

Acknowledgments 26

REFERENCES 26

LIST OF FIGURES

Figure 1. SIGMA toolkit 5
Figure 2. Two lower order OS coupling strategies. 7
(a) Simultaneous OS coupling, and (b) Staggered OS coupling. 7
Figure 3. Higher order, converged, iterative split coupling strategy. 9
Figure 4. Depiction of the CouPE solvers interacting with MOAB data backplane driving the standalone or coupled physics calculations. 11
Figure 5. SAHEX problem geometry and sample mesh. 19
Figure 6. SS solution distributions. (a) Power profile (W) and (b) temperatures (K). 19
Figure 7. Power solution transferred from neutronics mesh to thermal-hydraulics. 21
(a) Coarse resolution (top) and (b) fine resolution (bottom). 21
Figure 8. k_{eff} transient profile as a function of feedback and temporal resolution. 22
Figure 9. Transient evolution of coupled field profiles at the beginning, during and at the end of the perturbation. (a) Power distribution (W) and (b) temperature (K). 23
Figure 10. Temperature profile in the axial center of the assembly 24
Figure 11. Shift in power distribution 25

LIST OF TABLES

Table 1. k_{eff} spatial convergence study 20

1 Introduction

High-fidelity computer simulations of multi-physics problems require solving large systems of complex, coupled, nonlinear, stiff equations. Many examples of nonlinear multi-physics phenomena occur in a spectrum of scientific fields, raising the need to develop and validate accurate and stable numerical modeling and solution procedures. Such robust coupling methods are often used in radiation hydrodynamics, nuclear reactor analysis, fluid–structure interaction and climate model problems because of the need to accurately resolve the fine-scale effects in physics evolution [1].

Traditional solution techniques for coupled multi-physics phenomena have often relied on Operator-Split (OS) coupling strategies, which can introduce several sources of numerical errors in the solution fields as a result of inconsistent spatio-temporal treatment of the nonlinear terms. It is imperative to verify the accuracy preservation in these methods for problems of interest since effective resolution of the disparate characteristic physical scales is non-trivial. In this report, we introduce an integrated multi-physics coupling capability through a flexible software library that enables a multi-resolution hierarchy that is designed to ultimately span the full range of length and time scales present in relevant nuclear reactor design and safety analyses.

In order to produce a flexible multi-physics simulation capability, two obvious approaches can be pursued. In one approach, pieces of existing single-physics codes can be assembled into an overall coupled simulation code with appropriate interfaces to communicate between the components. This is generally referred to as a 'bottom-up' framework approach (MCT [2], SHARP [3,4]). The other approach is to use an integrated, coupled-physics modeling framework, with new code pieces for each relevant physics area developed inside that framework. This is sometimes referred to as a 'top-down' framework approach (DUNE [5], MOOSE [6], KARMA [7], among several others). The former approach takes advantage of the fact that several man-years invested in these existing verified and validated individual physics modeling codes are reusable, but producing a multi-physics capability will then require some intrusive modifications to enable appropriate software interfaces. The top-down framework approach avoids such intrusive implementations by providing unified physics interface guidelines that simplify the software overhead but at the substantial cost of re-writing all the necessary physics models from scratch, including their verification and validation (V&V) suites.

In the current report, we introduce the motivations and design of the Coupled Physics Environment (CouPE) library that provides robust parallel infrastructure to tie different applications tightly together based on a unified interface to unstructured mesh databases and description of field variables on entities defined on this mesh data-structure. CouPE utilizes the standard mesh interface definitions provided by ITAPS and implemented by the array-based data-structures in MOAB [8] for this purpose. MOAB in combination with several other libraries and tools namely CGM [9], MeshKit [10], RGG provide the capability to generate, handle and manipulate geometry and mesh representations. These tools were consolidated under one umbrella of tools named SIGMA (Scalable Interfaces for Geometry and Mesh-Based Applications) [11] developed at Argonne National Laboratory; the latest release of all the tools under SIGMA v1.0 is available as of July 25, 2014.

SIGMA components: The Common Geometry Module (CGM) [9] provides functions for constructing, modifying, and querying geometric models in solid model-based and other formats. While CGM can evaluate geometry from several underlying geometry engines, this work relies mostly

on ACIS, with an Open Cascade–based version also supported. Finite-element mesh and mesh-related data are stored in the SIGMA component: Mesh-Oriented datABase (MOAB) [8]. MOAB provides query, construction, and modification of finite-element meshes, plus polygons and polyhedra. Various options are available for writing and visualizing the final meshes produced by meshing algorithms. MOAB uses an HDF5-based file format, which can be visualized by using a ParaView plugin that is implemented by the MOAB library. The Visit visualization tool can also be configured and built with MOAB to provide a similar import capability. The MeshKit library [10] contains several mesh-generation algorithms that utilize CGM and MOAB descriptions of geometry and mesh to discretize complex geometric domains. Nuclear reactor specific geometries can be easily meshed through a GUI tool from Kitware built on top of Reactor Geometry Generator written in MeshKit whose overall goal is to develop a complete package for generating reactor core models.

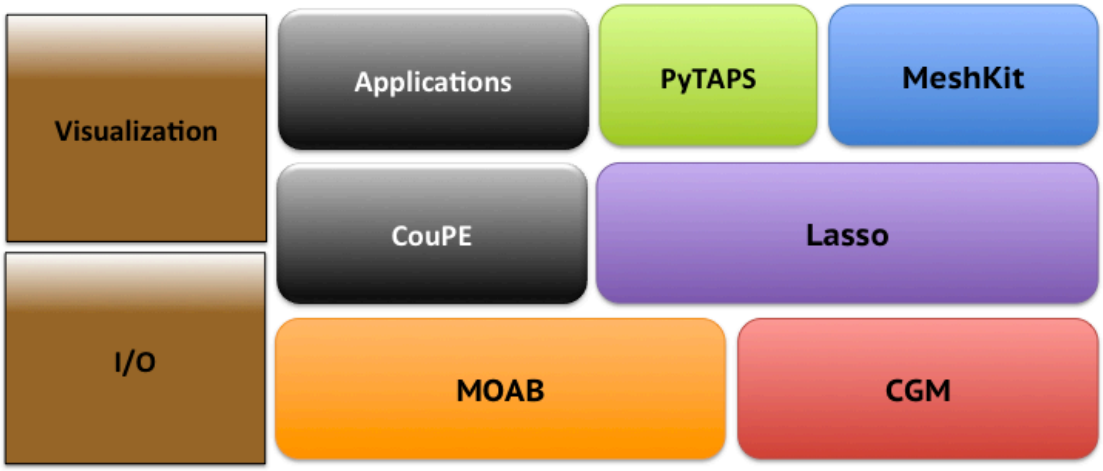

Figure 1. SIGMA toolkit

The implementation of a verified multi-physics solver code also imposes a number of requirements on the overall design aspects of the underlying libraries and the interface requirements for the framework. Hence we need flexible interfaces and robust solver options that encompass variations in a hierarchy of coupling algorithms affecting the frequency and degree of coupling between the physics. Since the choice of a coupling method is both physics and problem-specific, the reactor analysis framework in SHARP utilizes CouPE to expose a spectrum of numerical techniques to tackle the relevant scales in physical phenomena often seen in nuclear reactor design. The necessary background on the methods and implementation details in CouPE are provided in the following sections. The recent solution to single and multi-assembly demonstration problems in SHARP have also been published in papers [12, 13, 14] and reports [15, 16].

In this report, we detail the background, design, implementation and verification efforts for the coupled multi-physics library CouPE, which has been utilized successfully for several coupled analysis of reactor core systems using the SHARP framework. The organization of the report is as follows: In §2, the background on the numerical schemes motivating the solver choices in CouPE are described followed by §3 detailing the design and components that make CouPE a flexible options-based library to test both loosely coupled and tightly converged coupling strategies under one roof. Then we show several demonstrations with SHARP in §4 through verification studies performed on a fully heterogeneous assembly-scale problem in order to gauge efficiency and accuracy metrics that give the necessary foundation for simulating a large-scale, realistic nuclear engineering benchmark validation problem. We shall then conclude and end the report with enumeration of the list of design

and feature changes that are needed to address some critical issues encountered and learnt during the coupling verification in FY2014.

2 Background – Numerical methods

For illustration, let the nonlinear vector-valued function representing a coupled partial differential equation (PDE) system be written in a general form as

$$F(y) = N(Y)Y + LY - b = 0 \tag{2.1}$$

where **Y** is the solution vector that is dependent on both space and time, respectively, and **F**: $\mathbf{R}^n \rightarrow \mathbf{R}^n$, where **F** is the nonlinear operator representing the coupled system and *n* is the total number of unknowns. For ease of comprehension, we can write **F** as in the second equality of Eq. (2.1), where **N** is a purely nonlinear operator, **L** a purely linear operator and **b** is the load vector. It helps to represent **Y** as a vector comprised the solution vector for each of the *M* physics components involved, i.e. $\mathbf{Y}=[y_1, y_2, \ldots y_M]^T$. A similar definition holds for **F(Y)**, and its *m*th component is the nonlinear residual stemming from the *m*th physics component and may depend effectively on all other fields, $\mathbf{F_m(Y)}=\mathbf{F_m}(y_1, y_2, \ldots, y_M)$.

Almost all physical phenomena can be represented by the general nonlinear equation shown in Eq. (2.1). For instance, in nuclear engineering applications whether there are several physical phenomena that are nonlinearly dependent on each other's solution fields namely neutronics (particle transport), thermal-hydraulics (fluid and heat transport), structural deformation (material stress propagation). These physical equations often either have the linear operator **L** and/or the nonlinear operator **N** defined when considered alone. But when treated as a coupled PDE together with other physics equations, an additional coupled nonlinear operator $\mathbf{N}^{i,j}$ becomes important in determining the most effective solver to compute the solution of the larger system of equations. The remainder of this section will provide details used to resolve these nonlinear terms efficiently under one unified framework (CouPE).

2.1 Explicit coupling strategy

In the past few decades, high-fidelity modeling of nonlinear multi-physics problems has been subdivided into several distinct domains of physics and solved individually as mono-disciplinary blocks with specialized codes, without rigorous coupling between the different physics using OS. With the advent of parallel virtual machines (PVM) and the message passing interfaces (MPI) in the 1990s, the OS coupling of several existing specialized single physics codes has become the main multi-physics paradigm in reactor analysis. This kind of modeling is based on coupling several existing specialized mono-disciplinary codes using a 'black-box' strategy, where the input of one code is the output of other, thereby producing solutions that are weakly coupled. Schematics of such models are shown in Fig. (1), where the system of PDEs arising from the spatial and temporal discretization of physical models is decomposed into simpler sub-problems. Each physics component is solved by an independent, specialized single-physics code and the data between codes are exchanged through message passing paradigms. Often, this strategy is non-iterative, and the nonlinearities due to the coupling between the physics components are not resolved over a time step, reducing the overall accuracy in the time-stepping procedure to first order $O(\Delta t)$, even though high-order time integration might have been used for the individual physics components [17, 18]. Note that this explicit linearization of the problem in the OS strategy does not resolve the nonlinearities between the

different physics. Yet, these isolated physical models in reality describe physical phenomena that are tightly intertwined and rely heavily on the solution field of each other.

For illustration, consider the nonlinear, coupled physics system shown in Eq. (2.1). In the OS loose coupling strategy, the nonlinear operator is linearized as follows through an explicit treatment:

$$F\left(Y^{l+1}\right) = N\left(Y^{l}\right)Y^{l+1} + LY^{l+1} - b = 0 \tag{2.2}$$

Hence the updated coupled solution is obtained by solving the system of equations:

$$\left(N\left(Y^{l}\right) + L\right)Y^{l+1} = b \tag{2.3}$$

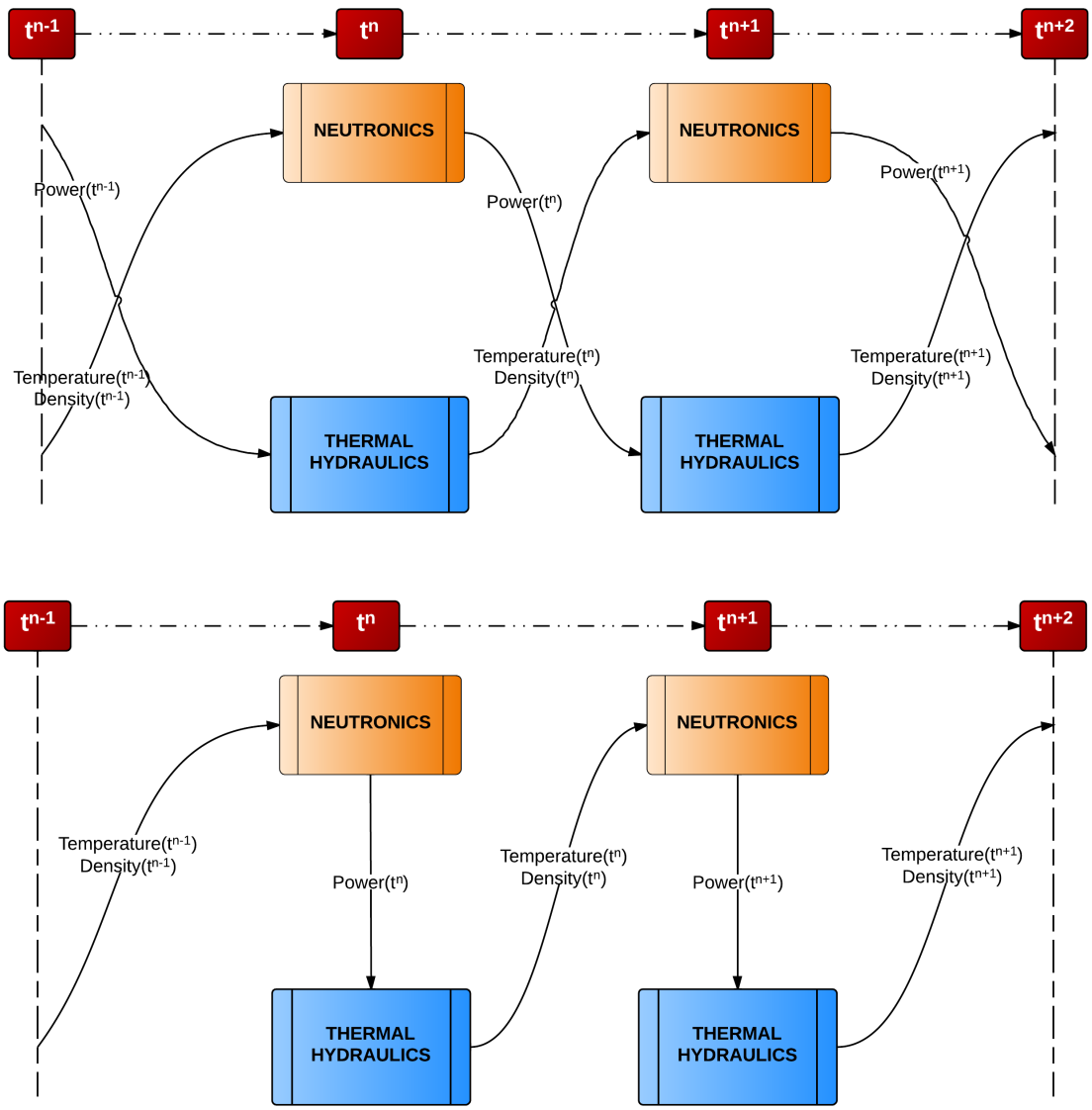

Figure 2. Two lower order OS coupling strategies.
(a) Simultaneous OS coupling, and (b) Staggered OS coupling.

Although OS allows parts of the problem to be treated implicitly and others explicitly, the lack of iterations in the conventional strategy degrades the solution accuracy in time to first order, and the explicit linearization imposes a conditional stability limit for the time-step selection. The direct implication of using smaller time steps to achieve a reasonable accuracy is that the computations need greater CPU time and resources. Despite these drawbacks, this is still one of the major coupling paradigms used today for solving nonlinear multi-physics systems.

The attractive feature of such a coupling strategy is that the legacy of many man-years of mono disciplinary code development and V&V is preserved. It is of prime importance to analyze the coupling strategies that can produce highly accurate solutions even in the complex scenarios usually encountered in multi-physics applications. As mentioned earlier, nuclear reactor analysis is a good example of a highly nonlinear, coupled, stiff problem, and the nonlinearities at the heart of reactor design, analysis and safety calculations provide a good state-space to test robust, high-fidelity numerical methods for multi-physics problems. Physical phenomena such as these found in reactor accidents, involve rapidly varying transients yielding stiff systems of differential equations that are characterized by solutions having fast varying modes together with slower varying modes, requiring time integrators that can handle such disparate time scales. Stiff problems necessitate the use of implicit time discretization for stability reasons, indicating that non-iterative OS coupling could prove disadvantageous in terms of efficiency (cost for obtaining a certain accuracy in the solution).

Current examples of multi-physics coupling in the field of nuclear reactor analysis involve the following pairs of deterministic neutronics/thermal-hydraulics codes: NURESIM based on SALOME [19], PARCS/TRACE [20] and NESTLE/RELAP [21] and other variations with stochastic neutronics methods with MCNP/Star-CCM [22, 23]. More recently, research on using OpenFOAM for performing fine-scale modeling of PWR cores [24] has also been explored with moderate success. Several advanced OS strategies exist that can yield up to the second-order accurate solutions in time, but they are complicated to implement generally in the context of legacy codes and hence are not typically employed. For more details regarding these higher order OS schemes, we refer the reader to [1, 25-27].

2.2 Implicit coupling strategy

An alternative to explicit or loosely coupled OS strategies is to converge the nonlinearities between the physics at every time level to obtain a tightly coupled solution that is consistent with the nonlinear system of PDEs. This preserves the higher order temporal accuracy of specialized schemes that can be used to resolve the disparate temporal scales in the different physics. Even though the cost/(time step) can be larger than that of an OS time step, we stress that the stability of the higher order discretization scheme can be maintained by using this procedure, unlike the explicit linearization (OS) method where the solution is only conditionally stable.

To devise such a tightly coupled solution procedure, one needs to apply a nonlinear iterative scheme in order to solve the coupled physics and converge the dependent terms to within user-specified tolerances. Two techniques for nonlinear system of equations are mentioned next: the Picard iteration technique (or variations of nonlinear-Richardson) and the well-known variants of Newton's method via Jacobian-free Newton–Krylov (JFNK) approximations [28]. In this report, we will concentrate on the former, because it allows effective reuse of existing physics components directly and avoids solving large monolithic systems by solving linearized sub-problems in a scalable fashion, thereby reducing the overall computational complexity of the simulation. The results shown in this

report utilize successively iterated OS schemes that are fully converged via Picard iterations so that the nonlinearities are resolved at every time step thereby yielding unconditional stability and recovering high-order accuracy. The linear rate of convergence is accelerated with the Steffensen method and with an explicit second-order predictor [18, 29].

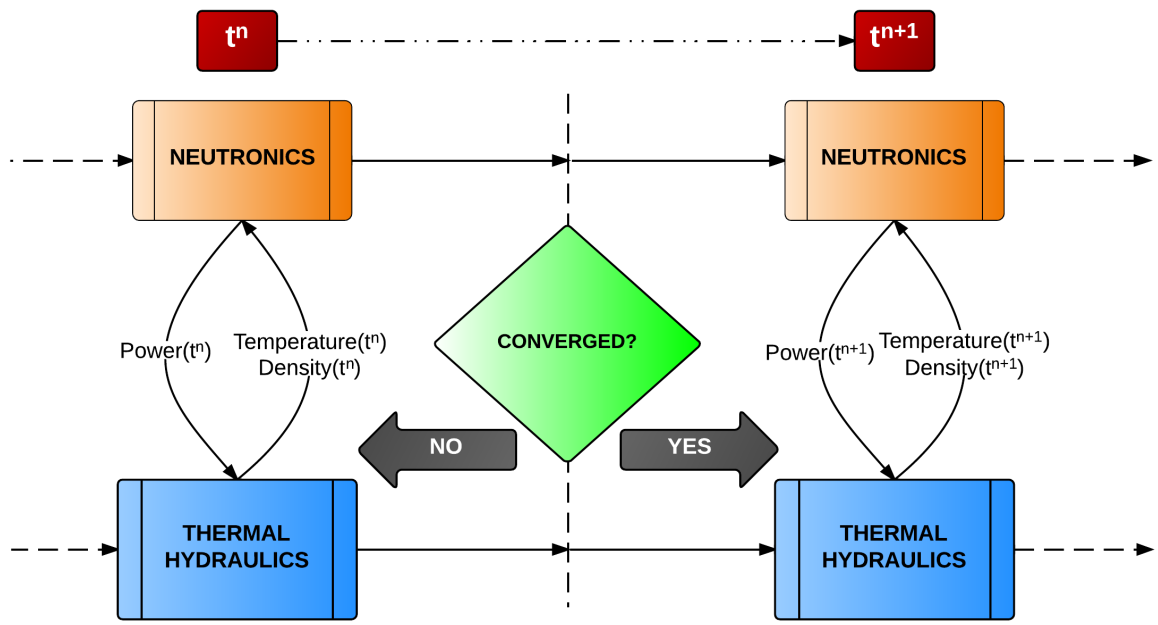

Figure 3. Higher order, converged, iterative split coupling strategy.

The Picard iteration technique is a well-analyzed nonlinear method that can be used to converge the nonlinearities over the different physics when an OS coupling technique is employed to couple multiple physics codes. Picard iterations can restore the convergence order of a higher order scheme and eliminate the loss of accuracy due to the crude explicit linearization in a loosely coupled strategy. The schematic for such a method is shown in Fig. 2. This essentially involves iterating over the solution obtained by successively solving Eq. (2.3). The advantage of such a coupling scheme is that it is non-intrusive and can easily use existing an framework of codes to obtain a tightly coupled solution, an approach rightly suited for SHARP. But the primary disadvantage of using such a strategy to restore the accuracy is the increase in computational cost due to linear convergence of Picard iterations. In order to overcome this issue, some form of nonlinear acceleration technique is necessary to make this scheme efficient and feasible [18]. Previous research using Aitken's iterated Δ^2 technique (or more popularly Steffensen method) suggests that usage of such acceleration schemes can be advantageous yielding super-linear convergence, and efforts to apply Wynn-Epsilon method [29] and other schemes should be pursued as future extensions.

The family of Newton-like methods is another appropriate choice for solving the global nonlinear problems. Unlike other methods, the Newton methods require the computation of the Jacobian matrix for the nonlinear system, which can be very expensive if calculated numerically or analytically for every coupling block. Unless the problem under consideration involves straight-forward nonlinear expansions (typically closed form), analytical Jacobians cannot be explicitly found and hence pose a severe restriction on the evaluation of the nonlinear operator. To circumvent this overhead, a modified Newton iteration called the Inexact Newton can be utilized where the calculation of the Jacobi can be done only once in several *m* steps (lagged), if the variation of the

Jacobian with respect to solution field is weak. Such a scheme does however lose the quadratic convergence properties of the full-Newton scheme. When comparing accelerated Picard iterations, the gain in using inexact-Newton scheme is strictly problem dependent since the spectral radius of the Jacobian operator determines the optimal method to be used [30].

With a reformulation of the iterative scheme shown in Eq. (2.3), we can obtain the standard nonlinear update for the solution as

$$J(Y^l)\delta Y^l = -F(Y^l) \tag{2.4}$$

where $J(Y^l)$ is the Jacobian evaluated with the solution Y^l and $F(Y^l)$ is the residual of the lth iterate.

Depending on the physics components that are treated implicitly, we can then obtain a global system of nonlinear equations that have a blocked structure amenable for iterative solvers. For instance, if we performed an ordering of the physics components that are coupled only to the ones before, we end up with a discrete coupled system of equations to be solved with the following form:

$$J(y) = \begin{bmatrix} N^{1,1}(y_1) + L^1 & 0 & \cdots & 0 \\ N^{2,1}(y_1) & N^{2,2}(y_2) + L^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ N^{P,1}(y_1) & \cdots & N^{P,P-1}(y_{P-1}) & N^{P,P}(y_P) + L^P \end{bmatrix} \tag{2.5}$$

where $i, j \in [1, P]$ with P being the total number of coupled physics components, $\mathbf{N}^{i,j}$ represent the nonlinear block corresponding to physics i coupled to solution from physics j and $\mathbf{y}_i$ is the i-th physics component solution. Since the diagonal nonlinear blocks $\mathbf{N}^{i,i}(\mathbf{y}_i)\forall i \in [1, P]$ require only the inversion of the single physics operator, this fixed point iteration procedure can be continued to generate a sequence of solutions that converge to the true coupled physics fields.

The family of Picard and Newton type nonlinear iterations for the global coupled system of equations can however be implemented with a single scheme by utilizing a solver that only looks at Eq. (2.4) without imposition on whether the Jacobian is exact. This leads to a global nonlinear Richardson type scheme, which can yield Picard or Newton when the Jacobian is essentially, ***I*** (Identity operator) or exact Jacobian respectively. Eq. (2.5) produces a nonlinear Richardson iteration procedure with a Jacobian matrix that resembles Block–Gauss–Siedel (BGS) coupling between the physics and hence can be solved effectively by solving sub-problems using existing mono-physics codes. Note that in this general framework, several different block structured matrices (Block-Jacobi or RBGS) can be used to capture stronger physics coupling between the components, instead of BGS scheme shown in Eq. (2.5).

CouPE implements such preconditioned nonlinear solvers at its core to provide both loosely coupled and tightly convergent schemes in a unified framework thereby enabling comparisons and verification studies to be performed in order to quantify numerical errors better. This is made possible by design decisions to provide an array of coupling strategies in CouPE, controllable with runtime command-line options (no recompilation) to control the solver methodology.

3 Design, components and implementation of CouPE

One can construct a multi-physics reactor core modeling code in many ways, and numerous efforts have attempted to do so by providing a stepping-stone for future efforts [1]. What distinguishes the CouPE from others is the goal of flexibility in the physics model and implementations, underlying discretization types, multi-fidelity resolution and flexible software options supported by the library. We begin by describing the components in CouPE and elaborate on the steps needed to couple a new physics module to existing components.

Using an existing physics code in this system requires that the system support whichever mesh type(s) the individual physics natively uses. The physics models can retain their own native representation of the discrete mesh, which gets transferred to and from MOAB's representation through a mesh adaptor or alternately, it can use MOAB internal representation directly through the language interoperable interfaces.

In practice, the coupled system may be solved on multiple physics meshes, each of which models part or the entire physical domain of the problem but resolving relevant spatial scales pertaining to a single physics. In order to perform efficient coupled calculations, the results must be transferred from the physics/mesh on which they are generated (source) to the physics/mesh for which they provide initial or boundary conditions (target) due to nonlinearity introduced by physics coupling. 'Two-way' transfer is required in cases where the physics depend on each other's solution fields, for example in reactor analysis where neutronics computes total fission heat generation based on material properties that are temperature dependent, which are computed by thermal-hydraulics component using the heat generation source term computed by neutronics.

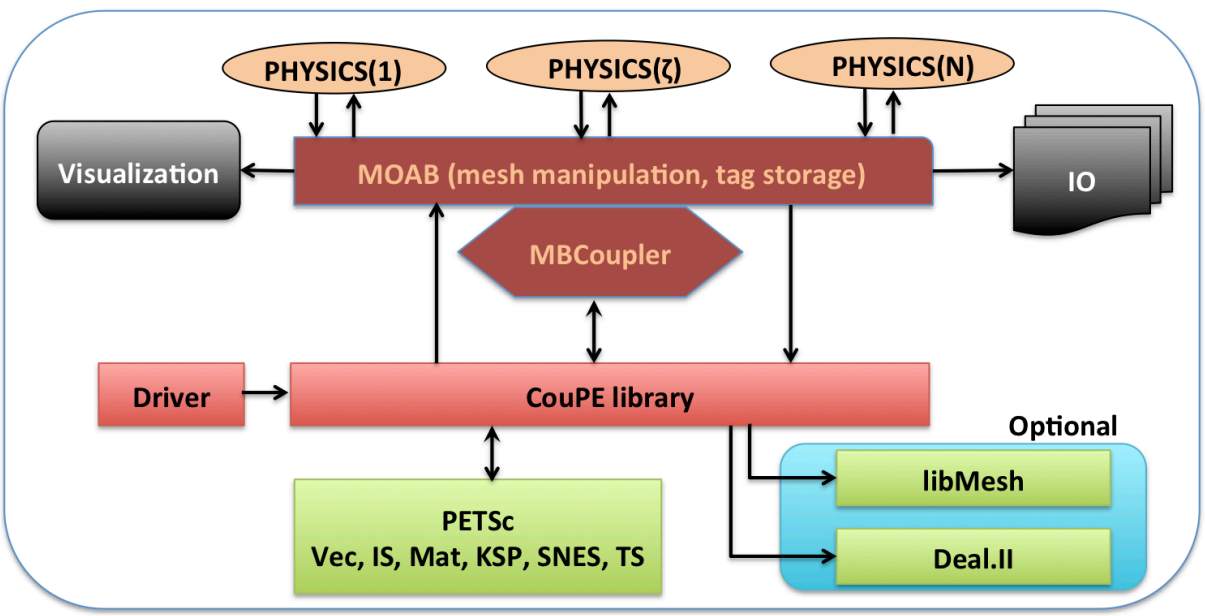

Figure 4. Depiction of the CouPE solvers interacting with MOAB data backplane driving the standalone or coupled physics calculations.

Fig. (4) illustrates the schematic of the SHARP framework used in this paper. MOAB provides a representation of the meshes, and MBCoupler (a MOAB based tool for parallel solution data

transfer [31, 32]) to interpolate (or via L_2 projection) each dependent physics component solution from the source to the target, with appropriate conservation prescriptions [33]. The CouPE library is responsible for implementing multi-physics coupling methods to consistently and accurately couple the different components, in order to solve the nonlinear reactor-physics problem. The combination of these tools provides the basis for the CouPE component-based framework.

3.1 Design of CouPE

CouPE aims to enable scalable and extensible coupling of different physics components that are nonlinearly dependent on each other. The SHARP multi-physics framework for solving reactor analysis problems employs validated and verified efficient mono-physics codes with MPI-level parallelism to implement several coupling strategies including OS and tightly coupled, iterated methods. The current design of CouPE is intended to satisfy the need for a flexible coupling tool even when the physical phenomena are strongly coupled to each other.

The aims in designing the CouPE code library included the following:

1. Component-based design, plugin architecture, scalable interfaces to couple codes across scales and physics,
2. When possible, leverage existing mono-physics codes in order to minimize development time, and to base the framework on already well-verified and validated codes and libraries;
3. Language agnostic to integrate physics codes and enabling scientists to focus on new science, explore and prototype new model formulations;
4. Interfaces to define and map the association of fields with respect to physical scales, units, context of the solution coming to and from a physics code, definition of a field that gets transferred and transformed across meshes and/or models, mapping of data in terms of model parameters (e.g., in multiphysics more about mesh-based entity associations and in multi-scale, more about sub-grid scale correlation callback associations);
5. Components to abstract out scalable solution data transfer and transformations with control given to physics codes i.e., optional reverse callbacks to physics via registration semantics to query, interpolate, and transform data depending on the context;
6. Enable a flexible and accurate data exchange framework between codes in a mesh, numerics, and physics aware fashion i.e., maintain consistency, accuracy and conservation of key fields.
7. Provide flexible data containers and physics objects in order to facilitate and simplify the evaluation of the non-linear residuals representing the fully discrete PDE for different physics components.
8. Use different kinds of multi-physics coupling strategies within the same architecture with minimal changes in the driver; enable runtime object polymorphism.

CouPE aims to integrate all the physics components under a unified framework in order to exchange the solution from one physics to another (interfaces to MBCoupler) and converge the coupled physics solution fields to user-specified tolerances (typically 10^{-4} –10^{-8} [30]) without sacrificing numerical stability or accuracy. CouPE provides the necessary components and layers to wrap existing physics codes or write a complete description of a physics problem from scratch in order to solve phenomena of interest, that is, to enable both bottom-up and top-down approaches. The library provides the necessary tools to quickly implement any of the popular variations of an OS

coupled solver (Marchuk, Strang and Yanenko among others) or a more rigorous matrix-free inexact-Newton solver with a JFNK technique [28].

Similar to the PETSc library [34], CouPE is designed to allow the user to specify command-line arguments in order to control the dynamic behavior of the coupled solver. The parameter specifications include the input for individual physics components, input mesh parameters, and type of solver along with ability to enable dynamically choosing the fidelity of the physics being coupled. This is made possible by abstracting out behavior of the core object until runtime. The advantage of such a method is that the implementation of the coupled physics driver and the accompanying physics components need to be compiled, linked and verified only once and then can be re-used in a variety of different coupling methods (e.g. loose versus tight coupling). These coupled fields can then be successively iterated until convergence to user-required accuracy, for any problem of interest.

For stationary, coupled nonlinear problems, the primary source of error stems from the exchange of physics solutions that reside in different spatial discretizations and resolutions. CouPE utilizes the iMesh interfaces and more specifically, its implementation by MOAB along with the spatial coupling tool MBCoupler to enable seamless integration of the mono-physics codes. This is made possible by exposing a minimal interface to be implemented by the physics wrappers, whose design follows the software paradigms of PETSc. The current design of CouPE was put forth in order to satisfy the need for a flexible software framework in order to solve strongly coupled physics modules. The implementation of coupled methods is usually nontrivial and CouPE can ease the difficulties and reduce the development time by providing a template to solve a collection of nonlinearly coupled physics objects via a uniform interface. The driver is simple, transparent and extensible and can be referred, as a glass-box solver rather than a black-box solver since it provides access to all the internal details of the physics are the corresponding internal mesh structures and the ability to supply and override the behavior at runtime.

There are two primary tasks that are necessary in order to seamlessly solve the physics modules coupled to each other with CouPE. This involves

1) Utilizing ITAPS (iMesh/iMeshP) interfaces to translate and represent the native mesh data-structures in terms of MOAB constructs (entities, tags, sets),
2) Identify and implement thin, lightweight wrappers for individual application codes such that clearly defined phases of execution are defined i.e., initialize, finalize, setup, solve etc.

In the SHARP framework, we have currently implemented MOAB interfaces to 3 different physics components that are relevant to fast reactor physics analysis. In order to add a new physics component to the framework, we require integration and ability to read the mesh and possibly associated data from iMesh/MOAB formats, along with implementation to propagate solution variables back onto the mesh after their computation via tags defined either on discrete vertices or elements. Because of the various storage formats used in physics modules, and the parallel domain-decomposed environment in which these calculations are usually run, this integration process requires involvement from both MOAB and physics module developers.

3.1.1 MOAB Interfaces for physics applications (use case: SHARP)

Interfaces to the MOAB mesh database have been written to handle PROTEUS [35] meshes that describe detailed geometries with multiple blocks (regions) with appropriate specification hooks for

temperature and density dependent material cross-section evaluation and interpolation. This interface is essential to capture the nonlinear feedback effect from thermal-hydraulics. Inherent ability to use a deformed mesh with appropriate recalculation of the density changes within materials (thereby affecting cross-sections) have also been implemented to enable direct coupling to a deformation code such as Diablo.

Using the standard iMesh based interfaces to MOAB, several different mesh formats can be natively used with Nek5000 [36, 37] along with the extended ability to couple with other physics components in the SHARP framework. When running Nek5000 in the fully coupled mode, the fluid/solid temperature along with their corresponding densities are stored in MOAB tags to be used either by the structural mechanic or neutronics components, propagated by the unified data-transfer mechanisms detailed earlier. The MOAB interface also enables Nek5000 to perform check-pointing and restart capabilities independent of the number of processors, thereby enabling the user an opportunity to investigate the validity of the coupled solution before proceeding further for critical problems.

In structural analysis of mechanical assemblies as implemented in Diablo [38], we created several API routines that can seamlessly handle use cases to load up a mesh file along with temperature/deformation data stored as tags in parallel and to create a native Diablo mesh representation in-memory. The reverse use case is also supported in order to abstract out the details of material definitions and contact functionality from MOAB. The experiences of the Diablo team, with contact, motivate the use of low-order spatial discretizations such as eight-node hexahedra for continuum and four-node quadrilaterals for shells. Hence the meshes have a similar resolution to PROTEUS.

3.1.2 Lightweight physics wrappers in CouPE

A physics application code needs to expose few fundamental routines as a library API in order to treat it as a component that can be utilized with CouPE. These impositions are placed more as guidelines to divide the distinct phases of execution during the driver runs. For illustration, let us consider the interface strategy to an arbitrary physics code. CouPE expects certain guarantees from the component that needs to be coupled in order to seamlessly abstract out the details regarding the global nonlinear solver and solution transfer between different physics components.

The details on the minimal set of API needed for the wrappers are provided here. These interfaces generally provide callbacks to routines that load a mesh, translate a mesh (from native to MOAB and vice-versa), to add and retrieve a field on the mesh (defined as tags), to compute residual and preconditioner operators (for advanced coupling schemes), and to write out the mesh data (in native MOAB H5M format) to disk among various other utility functionalities. Several public interface routines are exposed as part of this interface as follows:

CouPE_Physics_Initialize	PHY_SetOptions() – set filenames and other options PHY_Initialize() – initialize physics data structures PHY_InitializeTimeStep() – initialize a time Step
CouPE_Physics_PreStep	PHY_InitializeTimeStep() – initialize and prepare for transient calculation, form initial guess

CouPE_Physics_PreSolve	PHY_Import_Fields() – read necessary coupled physics data transferred from other physics (stored in tags)
CouPE_Physics_Solve	PHY_Solve() – invoke the mono-physics solver with the linearization to coupled operator based on projected solution
CouPE_Physics_PostSolve	PHY_Export_Fields() – export the computed coupled field data onto the computational mesh as MOAB tag data
CouPE_Physics_PostStep	PHY_FinalizeTimeStep() – Finish time step, advance fields, write check-pointing data
CouPE_Physics_Finalize	PHY_Finalize() – Close output files, write summary statistics PHY_Destroy() – Destroy the relevant physics data-structures

From the above description of the calling sequences, it is clear that a new physics module can easily be added to interface with CouPE once the mesh representation is provided with MOAB for in-memory query. This enables abstractions that make CouPE agnostic to the underlying data-structures in the individual physics modules and hence provides a unified interface to drive both the solution transfer mechanisms and coupled physics solver methods seamlessly.

CouPE provides higher levels of abstraction to solving coupled multi-physics problems by exposing the kernels necessary to describe a linearized mono-physics solver completely. These information when combined across multiple physics components provide the complete description of the nature of the coupled problem that can be utilized in the global nonlinear solvers implemented in the library with user controlled options to control the method and convergence.

3.2 Configuration and build system

CouPE uses the Git version control system and is hosted on Bitbucket website. Git is a relatively new distributed revision control and source code management system. The flexibility and the seamless workflow that Git provides is making it a widely adopted version control system for software development and all the SIGMA tools utilize the best practices put forth to enable collaborative code developement. The use of pull requests through the Bitbucket interface will also bring in contributions from users willing to be involved with the development of CouPE and examples based on the library. This system also encourages peer review on commits and requests thereby combining some SQA process with CI to ensure the robustness and efficiency of the code. Several combinations of builds with different compilers including the GNU, PGI, Intel and Clang have been added to the suite of builds that are run nightly.

One of the complex parts of the CouPE library, due to the sheer nature of its dependencies is the configuration and build system. Efforts to simplify the overall build system has been undertaken as a parallel effort to writing the solver pieces to ensure seamless integration of new physics components to both SHARP and CouPE. Some details on this build system are provided next.

3.2.1 Build System

The build system of Due to the number of required and optional dependencies (MOAB, HDF5, Zoltan, Metis, Parmetis, NetCDF, Zlib, PETSc, Blas/Lapack, Blacs), the packaging and build system for applications using CouPE can be quite complex. There are several concurrent efforts to ameliorate the difficulties encountered and to make it flexible for the user to configure based on –download-PACKAGE and –with-PACKAGE primitives. There are two main paths a user could take:

1. Autotools

2. CMake

All SIGMA tools from their inception have used autotools build system and it continues to be well supported. Alternatively, the CMake-based build system allows a user to build several components in the SIGMA tools: CGM, MOAB, Lasso and MeshKit using CMake. There are options provided to download and install these dependencies. Part of this effort is being implemented in collaboration with Kitware using the SuperBuild system.

3.3 CouPE v0.1.0 Release and notes

The CouPE library is being rebuilt based on some lessons learnt during the process of integrating several mono-physics codes in the SHARP framework. These motivations will enable better flexibility in the interfaces to couple only specific fields between physics components, to better describe field variables based on the nature of their discretization in order to maintain consistency during spatial projections and to better leverage the new implementations in PETSc solvers (v3.5 vs v3.1).

The next release of SIGMA (v1.1) that is expected to occur in December 2014 will contain the first release of CouPE v0.1 along with the associated libraries including MOAB and MeshKit with updated documentation for each. Some details on different aspects of manuals, documentation and code repositories are provided below.

1. **SIGMA Website:** Contains the latest news and developments about CGM, MOAB, Lasso, MeshKit and CouPE. It contains links to other sources of documentation mentioned in this section. Introduction, download and build instructions for all the SIGMA components is also available here.

2. **Documentation:** CouPE sources and docs are being modified to conform to the SIGMA *doxygen* documentation guidelines along with generation of user and developers guide to be showcased in the reference pages. The CouPE documentation was originally designed to follow the same format as that of PETSc but in a recent design decision, the function references and hierarchy of data structures are being generated with the help of Doxygen tool. These artifacts will be updated nightly, once that process is enabled with continuous integration support with all of the library components.

3. **Repository:** All SIGMA code, including CouPE is hosted on Bitbucket, there are build instructions along with information on branches and other developer related information.

4. **Continuous Integration:** The SIGMA Buildbot system [39] is currently setup to perform both nightly and continuous branch and integration tests for all of the component libraries. The tests are run for different configuration options, combinations of dependencies, variations in compiler types, versions, MPI implementations and machine architectures. This system is also being modified to include testing of various aspects of the CouPE library before its first release.

Next we will look at some applications that were developed with the CouPE library and show some verification studies performed in the context of coupled physics analysis.

4 Some demonstration problems

Verification of single-physics codes is a daunting task, and implementing the process for a complex multi-physics simulation requires quantifying errors at many stages. The spatial projection errors that occur when the solution field is transferred from one physics to another dominates the coupling errors along with the treatment of the nonlinear terms. In this paper, we present two demonstration problems that will help determine the accuracy of the solvers. First, a simplified hexagonal assembly (SAHEX) is taken and we test the data transfer mechanism for optimal accuracy and consistency as the source and target meshes are refined. Then, a realistic assembly that was employed in the EBR II reactor (the instrumented XX09 assembly) is solved for various SS and quasi-static transient problems, in order to investigate the advantage of using tightly coupled solvers to numerically resolve the relevant scales in the problem. The boundary conditions for the problems described are summarized in §2.

4.1 Numerical error measurement

At the most basic level of capability, integration of a given physics code into this system requires reading the mesh along with the associated data and writing solution fields back onto the mesh after their computation. This enables the interfaces in CouPE to transparently interface with the physics solvers and to pull/push the necessary data to drive the global nonlinear solver to convergence. The convergence criterion of the global nonlinear iteration between the physics is evaluated by measuring the error in the coupled solutions in each physics component. If both the absolute and the relative tolerances are below user-specified values, the iteration is set to have converged.

The convergence criteria for the solvers are determined based on the actual L_2 global error of the numerical solution $\mathbf{Y}_{\text{num}}$ computed on the corresponding physics mesh using the following definition:

$$\|error\|_2 = \left\|Y_{ref} - Y_{num}\right\|_2 = \sqrt[2]{\frac{1}{\Omega}\int\left[Y_{ref} - Y_{num}\right]^2 d\Omega} \qquad (4.1)$$

where Ω is the spatial domain and $\mathbf{Y}_{\text{ref}}$ is the reference solution computed on a refined/resolved spatio-temporal grid. Other error measures in L_1 and L_∞ norms can also be used.

4.2 Initial conditions and transient specification

The neutronics solution without any coupled feedback is essentially a linear generalized eigenvalue problem that computes the fundamental eigenvalue, eigenmode pair. The neutronics solver assumes constant material-dependent temperature/density values to initiate computation of cross sections, which are nominal values, obtained with the initial conditions (ICs) from thermal-hydaulic solver. Owing to the nonlinear nature of the coupled physics, the hydraulics solver assumes an axially cosine shape as the heat source to compute the thermodynamically equilibrium solution to a user specified tolerance.

Each problem was tested in a two-stage process:

1) compute coupled initial SS solutions at rated conditions,
2) perform a quasi-static transient where the power, temperature and density evolve based on the change in total reactivity.

The type of transient examined in the paper is a simplified loss-of-heat-sink, where the temperature of the fluid at the inlet boundary is specified as a function of time and the evolution in the coupled fields is computed. This simulates an accident scenario when the heat exchangers fail to remove excess heat from the coolant, thereby increasing the inlet temperatures steadily, causing feedback effects from different sources to interact nonlinearly between the physics.

$$T(t) = \begin{cases} T_0, & t < t_0 \\ T_0 + \alpha T_0 tanh\left(\frac{t-t_0}{\delta t}\right), & t \geq t_0 \end{cases} \tag{4.2}$$

where t_0 is the transient initiation time, T_0 is the initial converged temperature solution, δt is the duration of the transient at the inlet and α is the damping parameter to control the magnitude of the perturbation (typically 0.2).

4.3 Simplified hexagonal fuel assembly

The SAHEX problem geometry was designed to an outer duct wall that encloses six fuel pins and a control pin in the center. Meshes with varying resolution were used for neutronics and thermal-hydraulics solvers to resolve the spatial scales in the model and the geometry consisting of fuel pins, cladding, control rod, steel can and sodium coolant. This model is carefully chosen to create a verification test case for solving realistic assemblies to be addressed later. The SAHEX geometry and a representative resolved physics mesh is shown in Fig. (5).

The input data for the neutronics solver were generated a priori using the MC^2 library (associated with PROTEUS solver) to obtain parameterized 9-group cross sections as a function of temperature and density. Since this is an isolated assembly model, a vacuum boundary (non-reentrant) condition is applied on the top and bottom and reflective boundary condition on all other outer surfaces of the neutronics model. The velocity and temperature boundary conditions are applied at the bottom surface of the model (Dirichlet for the fluid, Neumann for the solid) and outlet boundary conditions are applied at the top surface. In terms of the mesh resolution, the hydraulics solver uses a comparatively coarser mesh than neutronics, consistent with the use of quadratic elements and nature of spectral discretization.

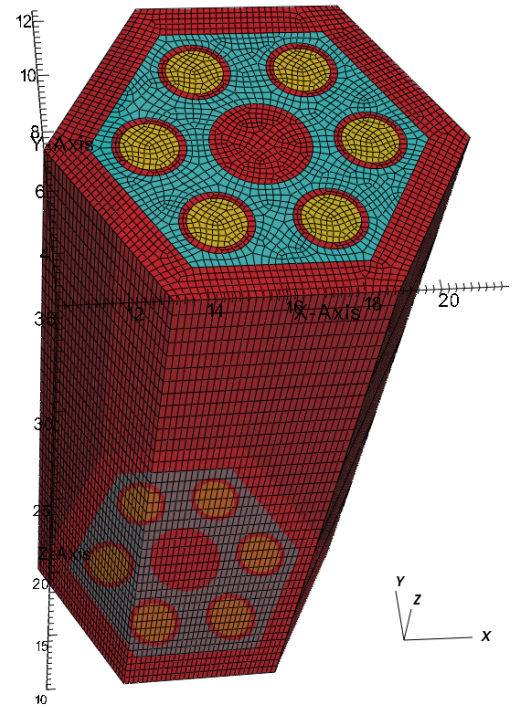

Figure 5. SAHEX problem geometry and sample mesh.

In the following section, the verification studies for spatio-temporal convergence performed on coupled physics simulations for the simplified single assembly problem will be discussed. Following the procedure detailed in §4.1, the SS solution fields for the individual physics were computed. The profile of the integral power based on the angular flux computed from solving the Boltzmann neutron transport equation and the temperature profile from thermal-hydraulic solver are shown in Fig. (6). The decoupled profiles are physically meaningful and provide a good initial guess for coupled physics solver to obtain an SS solution. It is observed the peak of the cosine shape of the power distribution shifts towards the inlet of the core due to lower material density at the top of the assembly, while the peak temperatures are observable near the outlet since the coolant temperature is monotonically increasing.

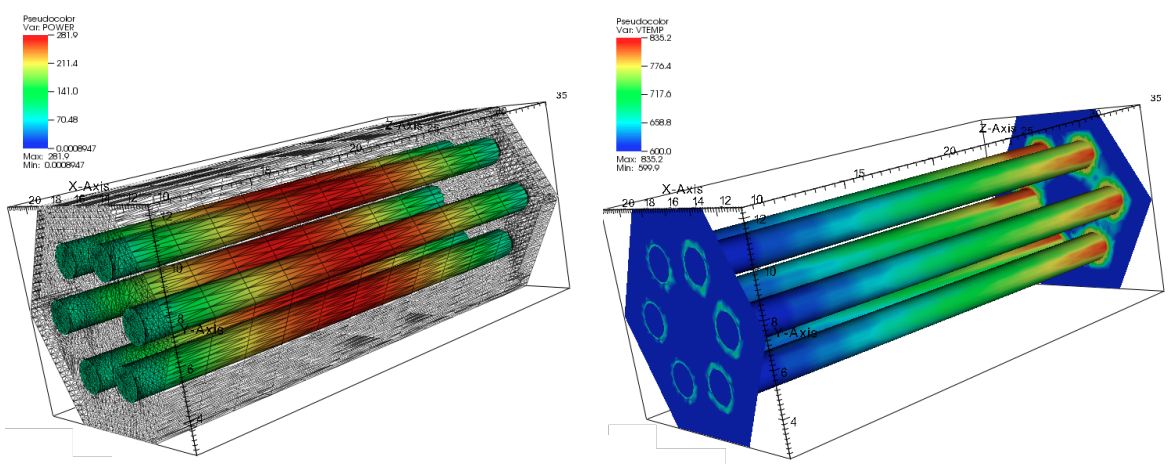

Figure 6. SS solution distributions. (a) Power profile (W) and (b) temperatures (K).

4.3.1 Solution transfer verification

The convergence criteria for the neutronics and hydraulics solver were specified to be $rtol_{NE}$ $=10^{-7}$ and $rtol_{TH}$ $=10^{-4}$, respectively, in order to capture relative variations in the fine-scale scalar flux and temperature profiles. CouPE computes the rate of convergence and determines the stopping criteria for the coupled physics iteration. Every call to the thermal-hydraulic solver is preceded by a conservative solution projection of the power solution field computed in neutronics, which is used to recompute the volumetric power source required in the fluid energy equation.

Nele	Characteristic length	(serial computational time)/(coupled iteration)	k_{eff}	error (%)
7590	0.7144	44.64	0.67387473	0.42
54740	0.3573	63.39	0.67671801	0.114
437920	0.1791	84.27	0.67751902	0.0315
3503380	0.0895	162.93	0.67775429	reference

Table 1. k_{eff} spatial convergence study

It is essential to rigorously verify spatial accuracy constraints in order to conserve the total energy specified for the problem configuration. We performed several successive refinements of the neutronics and hydraulics meshes, and quantified the error convergence in the solution projection and parallel data transfer algorithms. The convergence result of k_{eff} based on a reference mesh solution is tabulated in Tab. (1). From the error convergence of uniform refinements of the mesh, it is evident that the numerical order of convergence of the coupled solver is nearly second order $O(\Delta x^2)$ as expected, since the neutronics discretization is based on linear Lagrange basis functions. It is also imperative to note that the parallel performance of the solvers (on 32 processors) measured using the computational cost per Picard iteration (comprising of the individual physics solves and the two-way solution transfer of dependent fields) increases nearly linearly with the number of degrees-of-freedom.

The projected power solution at the end of the SS iteration from the source neutronics mesh to the target thermal-fluid mesh, after applying conservation prescriptions, is shown in Fig. (7a, 7b) for a coarse and fine spatial resolution. It is evident from the scales and the integral of the profile distribution that the global energy is conserved during field transfers.

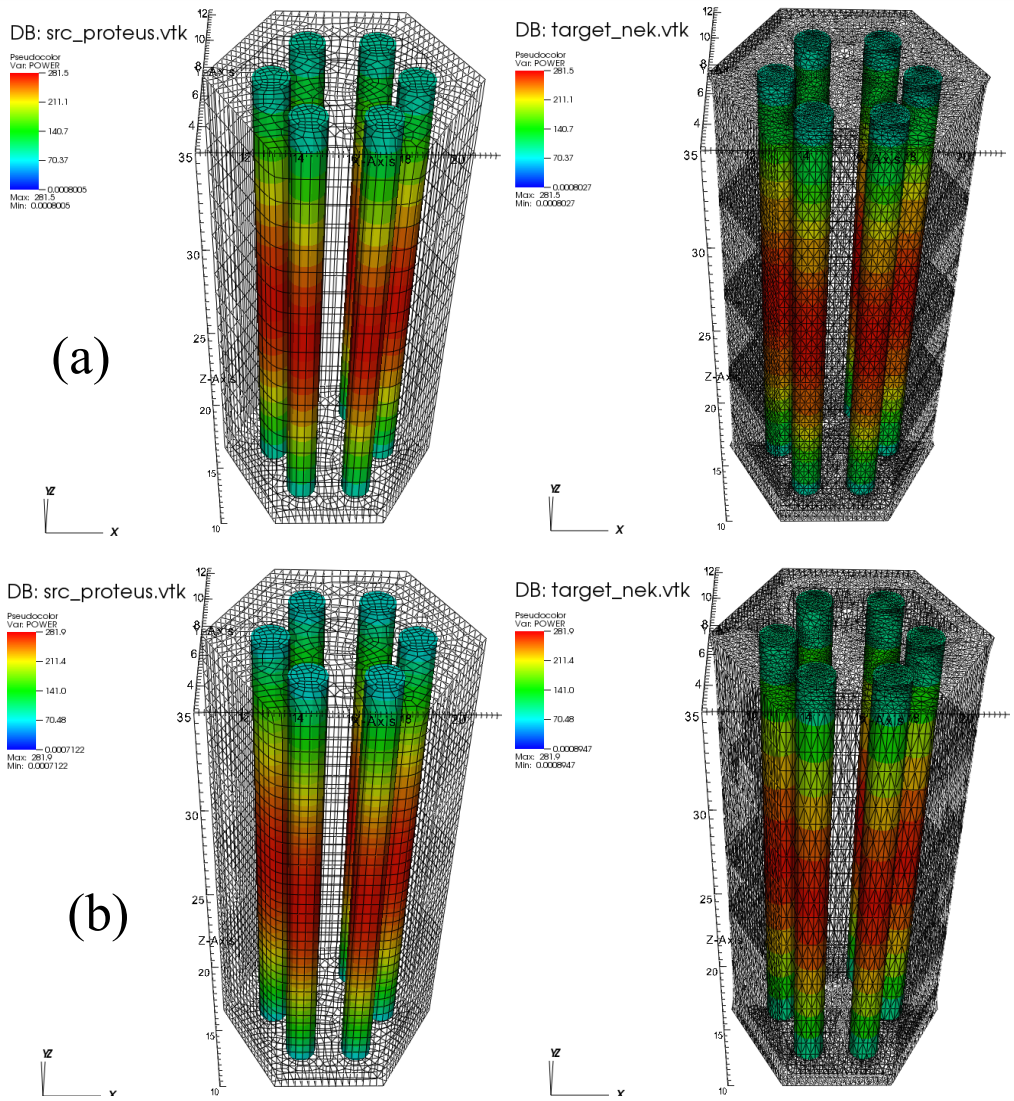

Figure 7. Power solution transferred from neutronics mesh to thermal-hydraulics. (a) Coarse resolution (top) and (b) fine resolution (bottom).

4.3.2 Coupled pseudo-transient verification

Once the ICs are converged, the loss-of-heat-sink simulation is initiated at $t=t_0$, by updating the inlet boundary conditions as shown in Eq. (4.2) to increase inlet boundary temperature from 600 to 720K during the transient. The total power in the assembly is specified as user input and power distributions are normalized accordingly.

In all the cases, the number of sub-cycling steps performed in thermal-hydraulics was specified to resolve the transient change in temperature. Several transients have been performed to test for sensitivity of the coupled field solutions to different feedback effects. Fig. (8) shows the change in k_{eff} as a function of normalized time for different types of coupled feedback effects optionally turned on. As the frequency of coupling is increased, the accuracy of the coupled physics solution improves since the computed criticality converges towards the reference. The flow time of the sodium through the assembly is 0.9 s (characteristic time scale), and the overall time-steps are

reduced consistently to resolve this spatial and temporal scales in successive simulations starting with Δt_{coarse}=0.02 s. The criticality constant k_{eff}, rapidly converges as the number of time steps is halved while approaching the reference solution as O(Δt). Note that feedback based on both temperature and density are necessary to show the complex nonlinear coupling between the neutronics and thermal-hydraulics physics for this test problem since the case where only doppler feedback is considered shows larger sensitivity to the inlet temperature change. In other words, the density and doppler expansion feedback are competing effects as validated from theory and experimental observations.

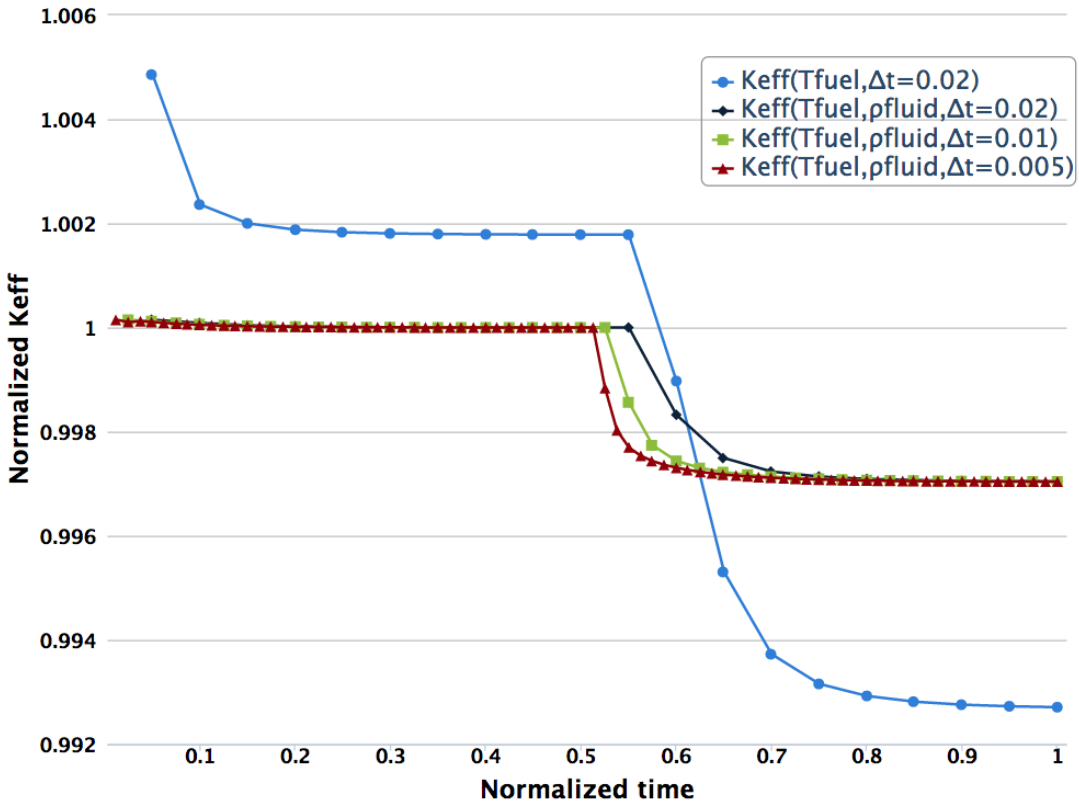

Figure 8. k_{eff} transient profile as a function of feedback and temporal resolution.

The total power decrease in the assembly as the transient progresses can be observed in Fig. (9a). The corresponding evolution of the temperature profiles is shown in Fig. (9b). Note that the significant change in power profile corresponding to only a minor penetration of the high temperature front within the domain indicates a very fast response (high sensitivity) to the boundary condition in the system.

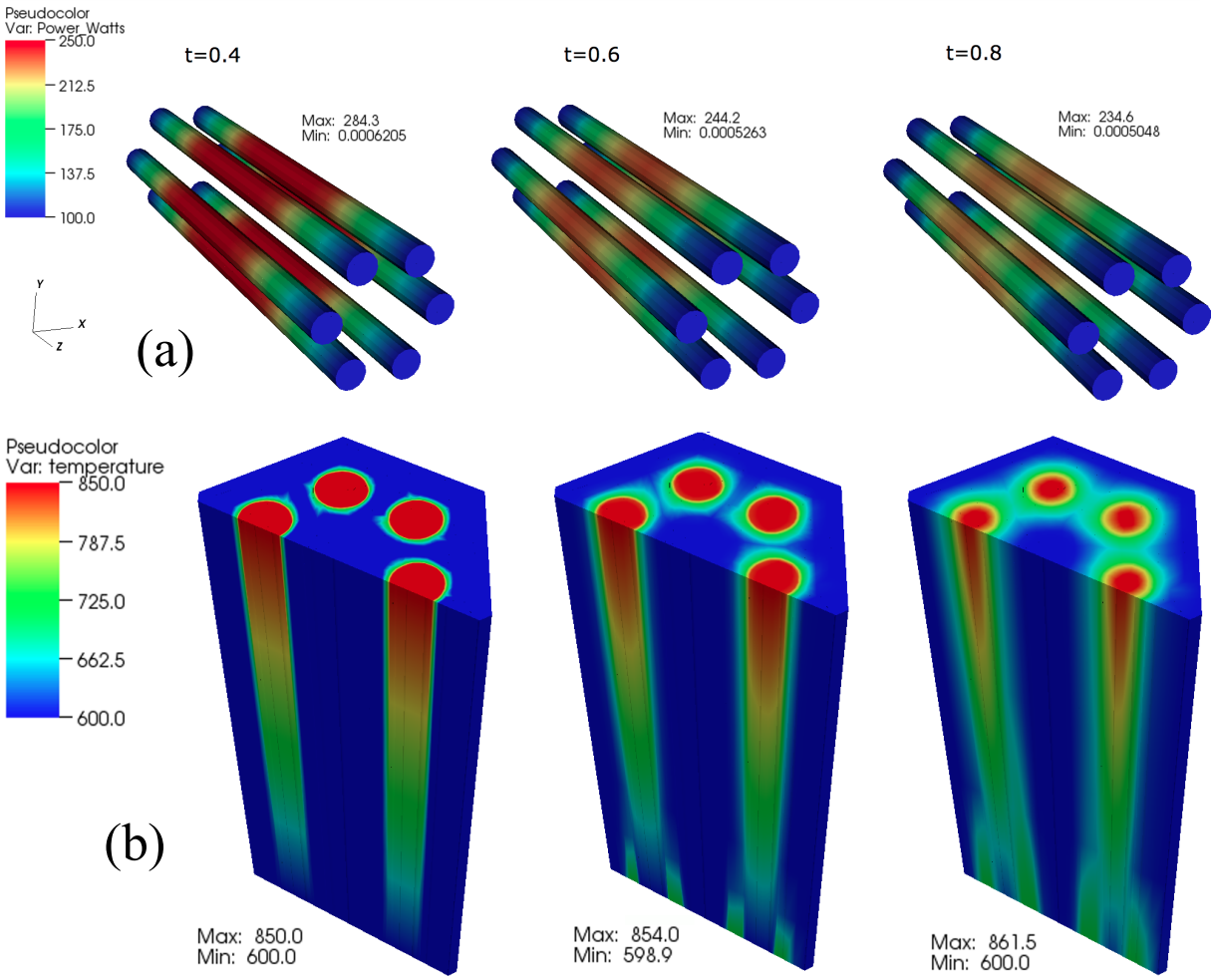

Figure 9. Transient evolution of coupled field profiles at the beginning, during and at the end of the perturbation. (a) Power distribution (W) and (b) temperature (K).

The coupled physics simulation capability with SHARP framework was tested on the SAHEX problem for a loss-of-heat-sink transient and the results obtained have been verified by spatial and temporal solution convergence studies. The sensitivity tests lead to important conclusions on:

i) the time-step size necessary for this transient to maintain accuracy, and
ii) the importance of the inclusion of all types of feedback effects.

4.4 Realistic benchmark problem: XX09 assembly

Detailed specifications for the XX09 assembly can be found in [16,40]. Owing to the nature of the physics discretization (and similar to the SAHEX problem), the neutronics mesh is much more resolved than the corresponding fluid mesh in order to capture the heterogeneity in the geometry. The boundary conditions for the XX09 assembly problem for both neutronic and thermal-hydraulic solvers are similar to SAHEX problem. The ICs are obtained by converging the PSS problem without any perturbations using the procedure described in §4.1. The coupled loss-of-heat transient problem is simulated with this problem to observe spatio-temporal sensitivity and nonlinear feedback effects.

4.4.1 Coupled solution results

The coupled physics results for the XX09 assembly problem uses sub-cycled thermal-hydraulic solves (5000 time steps, with a step size of $\Delta t=6\times10^{-6}$). This step size was chosen to resolve the temporal evolution of the flow and temperature fields to approach an SS solution. A perturbation of the inlet temperature (with α =0.2 in Eq. (4.2) equivalent to change of 120 K) is introduced and simulation is continued to convergence of a final SS conditions. In the initial stage of the transient, a front with increased temperature slowly advances through the assembly. As expected, it was observed that the criticality k_{eff} reduces as the transient proceeds, owing to the negative feedback effect from the increase in fluid temperature flowing through the domain. The doppler temperature increase is evident by observing the change in the profile at the axial center of the assembly as shown in Fig. (10), which also contributes to decrease the sodium density, thereby adding to the negative feedback effects in the assembly as the temperature wave propagates.

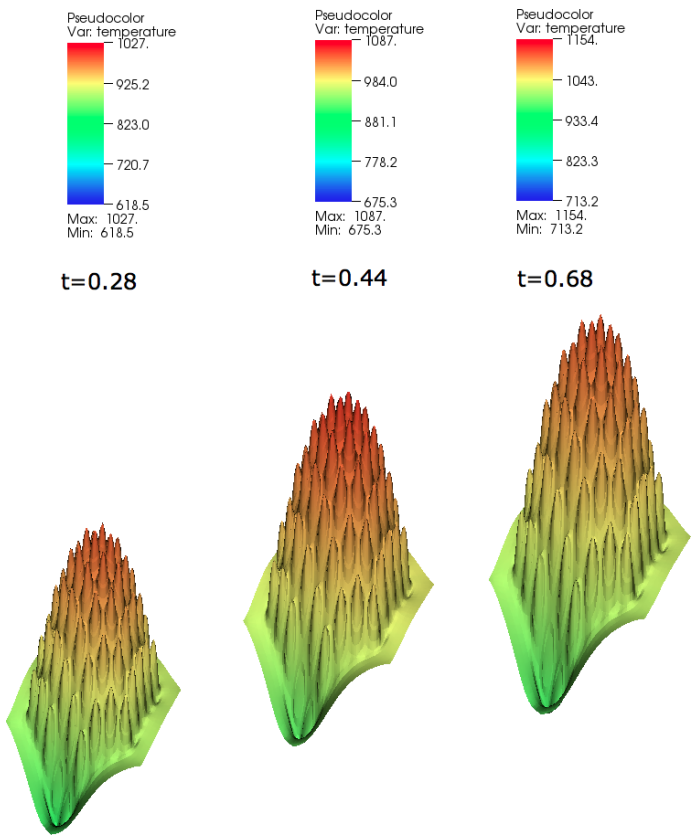

Figure 10. Temperature profile in the axial center of the assembly

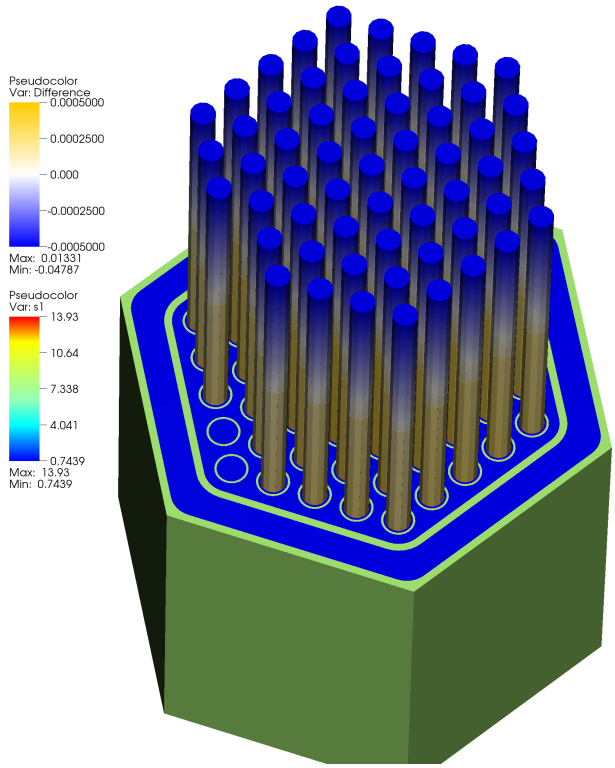

Figure 11. Shift in power distribution

Fig. (11) shows the relative change in power distribution between the final state of the transient and the initial SS result to quantify power sensitivity with respect to inlet temperature. In the second stage of the transient, the temperature increases in the fuel region producing the steepest change in reactivity thereby inducing a direct change in power distribution as shown in Fig. (11). The results obtained from the simulation of XX09 assembly were verified to convergence towards reference solution by successive refinements to resolve the spatio-temporal scales in the physics. A high-resolution video of the loss-of-heat-sink transient for the XX09 assembly problem has been hosted externally [41]. It shows the detailed evolution of the temperature wave propagation along the axial length and the strong nonlinear feedback effects that quickly stabilize the power profile to a final equilibrium.

5 Conclusions

The CouPE library has been designed to seamlessly tackle explicitly heterogeneous reactor assembly and full core geometry configurations and has been demonstrated to be convergent, and accurate in the implementation of the algorithms and the solver. The capability of the framework to flexibly choose different coupling schemes also makes it an important tool to determine the optimal solver method to resolve relevant scales for problems of interest. Using existing physics components expands the scope of SHARP to encompass different physics models with varying fidelity in equations, discretizations and requirements through a uniform abstraction layer. A structural mechanics solver, Diablo [38] is currently being integrated to analyze material deformation in the reactor due to large variations in the core temperature profile. This work has progressed well and the online deformation coupled to neutronics and thermal-hydraulics physics problems is currently being

verified before publication. Further investigations are also necessary to measure the overall parallel (MPI) performance of the tools for a multi-assembly problem on peta-scale architectures.

CouPE, a general coupling library in combination with several SIGMA components, have been demonstrated to integrate key components in SHARP for reactor analysts to perform single and multi-assembly reactor physics calculations in a verifiable manner. Several key improvements to the consistency, accuracy and efficiency of the solver and solution transfer mechanisms are currently being implemented to make it more robust to tackle scales of interest. The initial public release of the library along with detailed documentation, user examples along with tutorials will be performed by December 2014, as part of SIGMA v1.1. Research directions to include discretization support, field descriptions for abstraction data transformation and algorithmic adaptivity in coupling schemes will also be pursued in the coming year.

Acknowledgments

We thank the SIGMA and PETSc group at Argonne, who maintain the libraries required by CouPE. This work was supported in part by the U.S. Department of Energy Office of Nuclear Energy Nuclear Energy Advanced Modeling and Simulation (NEAMS) Program; by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research; and by the U.S. Department of Energy's Scientific Discovery through Advanced Computing program, under Contract DE-AC02-06CH11357.

REFERENCES

1. Keyes DE et al. 2013 Multiphysics simulations: challenges and opportunities. Int. J. High Perform. Comput. Appl. 27, 4–83.
2. Larson J, Jacob R, Ong E. 2005 The Model Coupling Toolkit: a new Fortran90 toolkit for building multiphysics parallel coupled models. Int. J. High Perform. Comput. Appl. 19, 277–292.
3. Siegel A, Tautges T, Caceres A, Kaushik D, Fischer P, Palmiotti G, Smith M, Ragusa J. 2007 Software design of SHARP. In Joint Int. Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications. La Grange Park, IL: American Nuclear Society.
4. Tautges TJ, Kim H, Caceres A, Jain R. 2011 Coupled multi-physics simulation frameworks for reactor simulation: a bottom-up approach. In Int. Conf. on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011), Rio de Janiero, Brazil, 8–12 May 2011. La Grange Park, IL: American Nuclear Society.
5. Bastian P, Blatt M, Engwer C, Dedner A, Kuttanikkad S, Ohlberger M, Sander O. 2006 The distributed and unified numerics environment (DUNE).
6. Gaston D, Newman C, Hansen G, Lebrun-Grandie D. 2009 MOOSE: a parallel computational framework for coupled systems of nonlinear equations. Nucl. Eng. Des., 239, 1768–1778.
7. Mahadevan VS. 2010 High-resolution numerical methods for coupled nonlinear multi-physics simulations with applications in reactor analysis. PhD dissertation, Texas A&M University, College Station, TX, USA.
8. Tautges, T. J., Meyers, R., Merkley, K., Stimpson, C., and Ernst, C. (2004). MOAB: A mesh-oriented database, SAND2004-1592. Sandia National Laboratories, Albuquerque, NM.
9. Tautges, T. J. (2005). CGM: a geometry interface for mesh generation, analysis and other applications. Eng. Comput. 17, 486–490.

10. Jain, Rajeev, Tautges, T. J., (2013). MeshKit. Report of US DOE, Reactor Campaign. ANL/MCS-TM-336.
11. SIGMA (Scalable Interfaces for Geometry and Mesh-Based Applications).
    URL: http://sigma.mcs.anl.gov/
12. Mahadevan VS, Merzari E, Tautges T, Jain R, Obabko A, Smith M, Fischer P., High-resolution coupled physics solvers for analysing fine-scale nuclear reactor design problems, Phil. Trans. R. Soc. A, 2021, 372:20130381, June 2014.
13. V. S. Mahadevan, E. Merzari, T. J. Tautges, "SHARP Coupled Multiphysics toolkit for reactor analysis", proceedings of International Congress on Advances in Nuclear Power Plants (ICAPP), Charlotte, North Carolina, Apr 2014.
14. V. S. Mahadevan, E. Merzari, R. Jain, A. Obabko, M. A. Smith, T. Tautges, P. Fischer, W. D. Pointer, R. Ferencz, "SHARP fuel assembly coupled simulation demonstrations", Proceedings of the ANS Winter Meeting, Washington, DC, Nov 10-14, 2013.
15. E. Merzari, E. Shemon, J.W. Thomas, A. Obabko, R. Jain, V. S. Mahadevan, T. Tautges, J. Solberg, R. Ferencz, R. Whitesides, "Multi-Physics Demonstration Problem with the SHARP Reactor Simulation Toolkit", prepared for the U.S. Department of Energy, Office of Nuclear Energy, ARC Milestone Technical Report, ANL-ARC-284, January 2014.
16. E. Merzari, V. S. Mahadevan, T. J. Tautges, R. Jain, A. Obabko, M. A. Smith, E. Wolters, R. M. Ferencz, "SHARP Multi-Assembly Multiphysics Demonstration Simulations", prepared for the U.S. Department of Energy, Office of Nuclear Energy, Milestone Technical Report, M2MS-13AN06030216, September 2013.
17. Ober CC, Shadid JN. 2004 Studies on the accuracy of time-integration methods for the radiation–diffusion equations. J. Comput. Phys., 195, 743–772.
18. Mahadevan VS. 2006 Nonlinearly consistent schemes for coupled problems in reactor analysis. Master's thesis, Texas A&M University, College Station, TX, USA.
19. Chauliac C, Aragones J-M, Bestion D, Cacuci DG, Crouzet N, Weiss F-P, Zimmermann MA. 2011 NURESIM: a European simulation platform for nuclear reactor safety: multi-scale and multi-physics calculations, sensitivity and uncertainty analysis. Nucl. Eng. Des., 241, 3416–3426.
20. Xu Y, Downar T, Walls R, Ivanov K, Staudenmeier J, March-Lueba J. 2009 Application of TRACE/PARCS to BWR stability analysis. Ann. Nucl. Energy, 36, 317–323.
21. Uspuras E, Kaliatka A, Bubelis E. 2004 Validation of coupled neutronic/thermal-hydraulic code RELAP5-3D for RBMK-1500 reactor analysis application. Ann. Nucl. Energy, 31, 1667–1708.
22. Seker V, Thomas JW, Downar TJ. 2007 Reactor simulation with coupled Monte Carlo and Computational Fluid Dynamics. In Proc. of the Int. Conf. on Emerging Nuclear Energy Systems (ICENES).
23. Griesheimer DP, Gill DF, Lane JW, Aumiller DL. 2008 An integrated thermal hydraulic feedback method for Monte Carlo reactor calculations. In Proceedings of the PHYSOR 2008 Conference.
24. Jareteg K, Vinai P, Demaziere C. 2014 Fine-mesh deterministic modeling of PWR fuel assemblies: proof-of-principle of coupled neutronic/thermal-hydraulic calculations. Ann. Nucl. Energy, 68, 247–256.
25. Lowrie RB. 2004 A comparison of implicit time integration methods for nonlinear relaxation and diffusion. J. Comput. Phys., 196, 566–590.
26. Strang G. 1968 On the construction and comparison of difference schemes. SIAM J. Numer. Anal., 5, 506–517.

27. Marchuk GI. 1971 On the theory of the splitting-up method. In Proc. 2nd Symp. on Numerical Solution of Partial Differential Equations, pp. 469–500. New York, NY: Academic Press.
28. Knoll DA, Chacon L, Margolin LG, Mousseau VA. 2003 On balanced approximations for time integration of multiple time scale systems. J. Comput. Phys., 185, 583–611.
29. Weniger EJ. 2000 Prediction properties of Aitken's iterated 2 process, of Wynn's epsilon algorithm, and of Brezinski's iterated theta algorithm. J. Comput. Appl. Math., 122, 329–356.
30. V. S. Mahadevan, J. C. Ragusa, V. A. Mousseau, "A verification exercise in multi-physics simulations for coupled reactor physics calculations", Progress in Nuclear Energy, 55, 12-32, November 2011.
31. Tautges TJ, Caceres A. 2009 Scalable parallel solution coupling for multiphysics reactor simulation. J. Phys. Conf. Ser. 180, 012017.
32. Tautges TJ, Kraftcheck J, Bertram N, Sachdeva V, Magerlein J. 2012 Mesh interface resolution and ghost exchange in a parallel mesh representation. Shanghai, People's Republic of China: IEEE.
33. Jiao X, Heath MT. 2004 Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. Int. J. Numer. Methods Eng. 61, 2402–2427.
34. Satish Balay, William D. Gropp, Lois Curfman McInnes, Barry F. Smith, "Efficient management of parallelism in object oriented numerical software libraries", Modern Software Tools in Scientific Computing, 163–202, Birkhäuser Press, 1997.
35. M. A. Smith, C. Rabiti, et al, "UNIC: development of a new reactor physics analysis tool", In proceedings of Winter Meeting on International Conference on Making the Renaissance Real, 97, 565–566, American Nuclear Society, Nov. 2007.
36. Y. Maday, A.T. Patera, "Spectral element methods for the Navier-Stokes equations", In A.K. Noor and J.T. Oden, editors, State-of-the-Art Surveys in Computational Mechanics, 71–143, ASME, New York, 1989.
37. A.G. Tomboulides, J.C.Y. Lee, and S.A. Orszag, "Numerical simulation of low Mach number reactive flows", Journal of Scientific Computing, 12, 139–167, June 1997.
38. I.D. Parsons, J.M. Solberg, R.M. Ferencz, M.A. Havstad, N.E. Hodge, A.P. Wemhoff. "Diablo user manual", UCRLSM-234927, Lawrence Livermore National Laboratory, September 2007.
39. SIGMA Buildbot continuous integration engine. URL: http://gnep.mcs.anl.gov:8010/
40. Poloncsik J, Filewicz EC, Kamis GJ, Natoce JT. 1982 The experimental breeder reactor ii (EBRII) instrumented subassemblies, INSAT XX09 and INSAT XX10. In Proceedings, Fast, Thermal and Fusion Reactor Experiments. Salt Lake City, UT: American Nuclear Society.
41. Mahadevan VS, Merzari E, Tautges T, Jain R, Obabko A, Smith M, Fischer P. 2014 A public git repository that hosts all the supporting data for the problems described in the current paper published by R. Soc Int. Proc A. See https://bitbucket.org/vijaysm/rsocpaper2014.

Mathematics and Computer Science Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 240
Argonne, IL 60439

www.anl.gov

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC