SHARP Assembly-Scale Multiphysics Demonstration Simulations

Mathematics and Computation Division
& Nuclear Engineering 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.

SHARP Assembly-Scale Multiphysics Demonstration Simulations

prepared by

Timothy J. Tautges, Paul Fischer, Iulian Grindeanu, Rajeev Jain, Vijay Mahadevan, Aleksandr Obabko

Mathematics and Computer Science Division, Argonne National Laboratory

Micheal A. Smith, Elia Merzari

Nuclear Engineering Division, Argonne National Laboratory

Robert M. Ferencz

Methods Development Group

Lawrence Livermore National Laboratory

March 31, 2013

Contents

Executive Summary	iv
1 Introduction	1
2 Background	2
2.1 MOAB and MBCoupler	2
2.2 Nek5000	3
2.3 PROTEUS	3
2.4 Diablo	5
3 Constructing the SHARP Multiphysics code	5
3.1 Approach	5
3.2 PROTEUS-MOAB Integration	7
3.3 Nek-MOAB Integration	8
3.4 MBCoupler Enhancements	9
3.5 COUPE : A New Coupled Driver Code	10
4 Test Problems	11
4.1 Simplified Single Assembly	12
4.2 Complete Single Assembly - XX09	13
4.3 Simplified Multiple Assembly Test Problem	14
5 Simulation Results	16
5.1 SAHEX1 Test Results	16
5.1.1 Nek5000 SAHEX1 Results	16
5.1.2 PROTEUS SAHEX1 Results	16
5.1.3 Coupled Nek5000/PROTEUS SAHEX1 Results	18
5.2 XX09 Results	25
5.2.1 Nek5000 Uncoupled Results for XX09 Assembly	25
5.2.2 PROTEUS Uncoupled Results for XX09 Assembly	25
5.2.3 XX09 Coupled Results	28
5.2.4 XX09 Diablo Results	34
5.3 Preliminary results for Multiple Assemblies	36
6 Future Plans	37
7 Conclusions	37
Acknowledgments	39

A	RGG Input for XX09 Assembly	40
B	Sample coupled physics driver using COUPE	41
C	Coupled Nek5000/UNIC Model Output for SAHEX1	42
	References	55

Executive Summary

The NEAMS Reactor Product Line effort aims to develop an integrated multiphysics simulation capability for the design and analysis of future generations of nuclear power plants. The Reactor Product Line code suite’s multi-resolution hierarchy is being designed to ultimately span the full range of length and time scales present in relevant reactor design and safety analyses, as well as scale from desktop to petaflop computing platforms. In this report, we describe our efforts to integrate thermal/hydraulics, neutronics, and structural mechanics modeling codes to perform coupled analysis of a representative fast sodium-cooled reactor fuel assembly.

Over the past five years, the Reactor Product Line effort has developed high-fidelity single-physics codes for neutron transport modeling, in the PROTEUS code, and Large Eddy Simulation-based thermal/fluid modeling in the Nek5000 code. Both these codes have been exercised on over 100,000 processors of the IBM Blue Gene/P. The Diablo code has been used to perform structural mechanics and thermo-mechanical modeling. MOAB, the Reactor Geometry Generator (RGG), and MeshKit have been developed to generate and manipulate mesh and mesh-based data, in both serial and parallel environments. These tools together form a strong basis on which to build a multi-physics modeling capability. The goal of developing such a tool is to perform multi-physics neutronics, thermal/fluid, and structural mechanics modeling of the components inside a reactor core. While the focus of this report is on modeling a fast sodium-cooled reactor, it is also the goal that this simulation tool be useful for most reactor types.

Here we report on the integration of PROTEUS and Nek5000 into the NEAMS framework. Both codes have been modified to read and write data from and to the MOAB library, in both serial and parallel environments. We focused on broad support for as many original physics code options as possible, rather than focusing only on near-term needs. Both codes were demonstrated through this interface on a relatively simple, seven-pin hexagonal fuel assembly. To support coupled analysis, the new multi-physics driver library COUPE was developed; this library provides flexible support for both loose and tight coupling, and can interface flexibly with codes in multiple programming languages. To couple results between the separate meshes used in each physics, the driver, PROTEUS, and Nek5000 all interface with the MBCoupler solution transfer tool provided with MOAB. Together, these tools form the basis of a new, coupled neutronics and thermal/fluid modeling code, that works in both serial and parallel. In a related effort, a driver in the Diablo code was used to access Nek5000-computed results written to a MOAB database, as well as the temperature data used to drive structural mechanics computations.

An important goal of this effort is to perform coupled modeling of the EBR-II Shutdown Heat Removal test. In the present work we report on some steps toward that goal:

- Fully coupled steady state runs were completed for a simplified 7 pin assembly and single EBR-II assembly (the instrumented XX09 assembly).
- Pseudo steady-state (transient where global power is considered constant) loss of heat sink transient was completed on both a simplified 7 pin assembly and a single EBR-II assembly (the instrumented XX09 assembly).
- A full loss of heat sink transient was completed for a 7 pin assembly.
- Further capabilities related to homogenization, and necessary for full core simulations, were also developed.

1 Introduction

The NEAMS Reactor Product Line aims to develop an integrated multi-physics simulation capability for the design and analysis of future generations of nuclear power plants. The Reactor Product Line multiresolution hierarchy is being designed to ultimately span the full range of length and time scales present in relevant reactor design and safety analyses, as well as scale from desktop to petaflop computing platforms. In this report, we describe our efforts to integrate thermal/hydraulics, neutronics, and fuel performance modeling codes to perform coupled analysis of a representative fast sodium-cooled reactor fuel assembly.

There are two obvious approaches to assembling a coupled-physics simulation capability for analyzing reactor core systems. In one approach, pieces of existing single-physics codes are assembled into an overall coupled simulation code. This has been referred to as a “small-f” or “bottom-up” framework approach [1, 2]. 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 “large-F” or “top-down” framework approach, with several implementations available [3, 4]. One advantage to the former approach is that it clearly takes advantage of the several man-years already invested in existing physics modeling codes, while a possible disadvantage is that it may preclude fully-coupled solution methods that are the strength of “large-F” frameworks. The overall approach being pursued in the Reactor IPSC effort is to develop and demonstrate a small-f framework for performing coupled multiphysics analysis of reactor core systems. This system takes advantage of single-physics codes also sponsored by the overall NEAMS program.

In this report, we describe the assembly of the SHARP Integrated Performance and Safety Code suite and its use in multiphysics modeling of a sodium fast reactor (SFR) fuel assembly both in steady-state and transient mode.

The long-term goal is to simulate the Shutdown Heat Removal Tests (SHRT) [5, 6] that demonstrated the passive safety features of the EBR-II Experimental Breeder Reactor. The specific test from which this problem is derived is SHRT17, which is a loss of flow (LoF) with SCRAM experiment. This test configuration was selected for the multiphysics demonstration because of the availability of temperature and flow validation data available from the experiment. Another advantage of that data set is the highly coupled nature of the transient itself.

As an intermediate step, this report focuses on recent efforts to demonstrate key capabilities that will be instrumental for performing such simulation, particularly the following:

- simulation of an initial steady-state at the rating conditions,
- simulation of a pseudo steady-state transient at constant power,
- simulation of a transient with varying power, and
- simulation of a steady state with multiple assemblies, part of which are homogenized.

The remainder of this report is organized as follows. In Section 2, we give background on the single-physics codes used, and the infrastructure used to integrate them into a multi-physics capability. Section 3 describes the code modifications and new developments necessary to perform this integration. In Section 4 we describe the set of problems used to test and demonstrate this multiphysics capability:

- SAHEX1 is a simple 7-pin rod bundle with surrounding duct,

- XX09 is an SFR fuel assembly test conducted in EBR-II, and
- SAHEX1+6 is an extension of SAHEX1 with six surrounding homogenized assemblies.

Results of coupled simulations are described in Section 5; future plans for coupled simulation development are given in Section 6 and conclusions in Section 7.

2 Background

In the present section the different components of SHARP (MOAB and MBcoupler, Nek5000, PROTEUS and Diablo) are described.

2.1 MOAB and MBCoupler

One of the critical aspects in assembling a multiphysics modeling code is mapping the results from one physics domain to another. In the small-f framework approach used in the Reactor Product Line, we use a common mesh library for this purpose. We use the Mesh-Oriented database (MOAB) as a "data backplane" to link physics through their spatial domains, and MOAB's MBCoupler package for transferring physics results between those domains.

MOAB is a library for query and modification of structured and unstructured mesh and field data associated with the mesh [7, 8]. MOAB can represent all entities typically found in the finite element zoo, as well as polygons, polyhedra, and structured meshes. MOAB provides parallel functionality for resolving entity sharing and ghosting between processors, with sharing and ghosting information available as annotations on the local mesh [9]. MOAB's parallel I/O is based on the parallel HDF5 library, and has been demonstrated on processor counts up to 16k on the IBM BG/P. A partitioning tool has been implemented by interfacing with the Zoltan partitioning library implementation [10], with visualization provided by a Paraview [11] plugin. MOAB can read meshes generated by the CUBIT mesh generation toolkit [12], and can represent the various mesh types used in this effort.

MOAB's data model consists of four fundamental data types:

- **Entity:** A basic entity in the discrete model, e.g. vertex, quadrilateral, tetrahedron.
- **Entity Set:** Arbitrary collection of entities and other sets.
- **Interface:** The database object or instance from an implementation point of view.
- **Tag:** A piece of data annotated on objects of any of the other three data types.

This data model, while simple, can represent all the data necessary to run the coupled simulation. In particular, tags can be used to store both fine-grained solution data on individual vertices and elements, and coarse-grained annotation of sets to identify them as boundary conditions, material types, or processor partitions. Two particular groupings are common to this effort and are described here. Material sets, also referred to as "element blocks", group elements by material definition. In MOAB, these are represented as entity sets, tagged with a "MATERIAL SET" tag whose value stores a user-assigned id number. Neumann sets, also referred to as "sidesets", store groups of lower-dimensional entities (in a 3D mesh, Neumann sets contain mesh faces, for example); similarly, these groups are marked with "NEUMANN SET" tag whose value is the user-assigned id number.

Other tags used to store field and other data for PROTEUS and Nek are described in Sections 3.2 and 3.3, respectively.

MBCoupler, a MOAB-based tool for solution transfer [13], has been demonstrated on up to 4,000 processors. This tool allows the source and target meshes to be distributed across processors in whichever way is best for the physics associated with each mesh. Target-to-source mesh point location is performed in parallel, with bounding box-based acceleration used to determine possible source mesh processors containing every point and with kdtree decomposition used locally on each processor. This tool can transfer solutions using both linear finite element and piecewise-constant shape functions. As described in Section 3.4, it has recently been extended to incorporate spectral element shape functions as well. In this effort, MBCoupler is used to map the results computed by one physics module to boundary conditions on the mesh used by the next physics module.

2.2 Nek5000

The Nek5000 computational fluid dynamics (CFD) simulations are based on the spectral element method developed by Patera [14]. Nek5000 supports two formulations for spatial and temporal discretization of the Navier-Stokes equations. The first is the $P_N - P_{N-2}$ method [15, 16, 17], in which the velocity/pressure spaces are based on tensor-product polynomials of degree N and $N-2$, respectively, in the reference element $\hat{\Omega} := [-1, 1]^d$, for $d = 2$ or 3 . The computational domain consists of the union of E elements Ω^e , each of which is parametrically mapped from $\hat{\Omega}$ to yield a body-fitted mesh. The second is the low-Mach number formulation due to Tomboulides and Orszag [18, 19], which uses consistent order- N approximation spaces for both the velocity and pressure. The low-Mach number formulation is also valid in the zero-Mach (incompressible) limit [20]. Both formulations yield a decoupled set of elliptic problems to be solved at each timestep. In $d=3$ space dimensions, one has three Helmholtz solves of the form $(\beta I + \nu \Delta t A) \underline{u}_i^n = \underline{f}_i^n$, $i = 1, \dots, d$, and a pressure Poisson solve of the form $A \underline{p}^n = \underline{g}^n$ at each timestep, t^n , $n = 1, \dots$. Here, A is the symmetric positive definite Laplace operator, and β is an order-unity coefficient coming from a third order backward-difference approximation to the temporal derivative. (For the $P_N - P_{N-2}$ method, the Laplace operator A is replaced by a spectrally equivalent matrix arising from the unsteady Stokes equations [17, 21].) For marginally resolved LES cases, we find that the higher-order pressure approximation of the $P_N - P_N$ formulation tends to yield improved skin-friction estimates, and this is consequently the formulation considered here.

The conjugate heat transfer problems presented here are typical of nuclear engineering applications. The following boundary conditions are generally applied:

- the inlet surface has uniform prescribed velocity and fixed temperature,
- standard outflow boundary conditions are specified at the outlet surface, and
- the wall surfaces have velocity non-slip and insulated temperature boundary conditions.

2.3 PROTEUS

As part of the NEAMS activity in DOE, the PROTEUS (formerly UNIC) high-fidelity deterministic neutron transport code [2, 22, 23] was developed, providing a common framework for reactor neutronics tool development. The application scope targeted for PROTEUS ranged from the homogenized assembly approaches prevalent in current reactor analysis methodologies to explicit geometry approaches, and time dependent transport calculations coupled to thermal-hydraulics and

structural mechanics calculations for reactor accident simulations. The creation of a single solver that can perform all of these calculations and be competitive with the wide range of analysis tools already in use is somewhat formidable. To date, research under the UNIC umbrella has focused on a PN based finite element solver (PN2ND), an SN-based finite-element solver (SN2ND), a method of characteristics (MOC)-based finite-element solver (MOCFE), and a nodal diffusion theory solver (NODAL).

The SN2ND solver of UNIC is the only one with a proven capability of running calculations like XX09 which was based upon the research results of the past few years and thus the need for the other solvers within UNIC is eliminated. To highlight the transition from a scoping tool with multiple different solver options to a production tool with an explicit focus on meeting the immediate and long term analysis needs of NEAMS, we renamed the toolset PROTEUS. In PROTEUS, we have only two research tools: the continued production development step of SN2ND, called PROTEUS-SN, and the additional development of a 2D-1D MOC solver (PROTEUS-MOC) based upon a semi-structured finite element geometry. PROTEUS-SN is the only tool capable of handling the EBR-II calculations at this point.

Development on PROTEUS-SN started in FY2007, and it is still the only solver proven capable of using large scale parallel machines to solve large problems. The PROTEUS-SN solver is based upon the even-parity transport equation, which has known limitations for fine detailed reactor geometries like those targeted by NEAMS. The early version of the PROTEUS-SN solver was rewritten in FY2008 and early FY2009 and subsequently demonstrated good scalability and performance at 100,000 processors of BlueGene/P [23]. The achievements of PROTEUS-SN were made possible by partitioning the space-angle system of equations over the available processors and utilizing established iterative solution techniques from the neutron transport community combined with the parallel algorithms in the PETSc toolbox [24]. Follow-on improvements were made in FY2009 and FY2010 with respect to performance in order to gain a better understanding of the capabilities and limitations of the even-parity based solver.

While the solver was found to be acceptable for a large range of problem sizes, the implementation was still a scoping study capability and could not handle many of the desired aspects of NEAMS problems. In FY2010 through FY2012, efforts were made to completely rebuild PROTEUS-SN to include:

- parallelism in energy,
- a spatial multigrid concept,
- SN vector spaces to accommodate the adjoint and time dependent functionalities needed for NEAMS work, and
- elementwise cross section evaluations.

Unfortunately, the FY2011 version is not as robust or as reliable as the well-optimized FY2010 version, and thus we relied upon the FY2009 version for the XX09 assembly study in this document.

The elementwise cross section feature is the more important one for this work, since the FY2009 version is significantly restricted in its modeling accuracy. As an example, the geometry of the XX09 assembly consists of just a few materials (sodium, HT9, fuel) and just a few blocks of element data (13 at a minimum for XX09). In the FY2010 version, the temperature and density of the materials are averaged over the entire "block", and if the part of the mesh defining the fuel is contained in a single block, only a single cross section evaluation is used everywhere in the fuel. While we correctly

incorporate linear interpolation of the cross section data with respect to $\log T$, the temperature is still averaged over all of the elements unless the mesh is broken into additional pieces.

This approach creates a considerable amount of extra effort when it comes to mesh generation, but it was acceptable for most of the scoping studies performed with the tool. In the coupled calculation of this work, we made only marginal efforts to partition the elements into smaller pieces, and thus the accuracy of the solution is rather limited. However this is a proof of principle calculation and we intend to transition to the much improved PROTEUS-SN in future efforts.

2.4 Diablo

The Diablo code being developed at LLNL uses implicit finite element methods for the simulation of multiphysics events over moderate to long time frames [25]. A primary focus is nonlinear structural mechanics and heat transfer. The code provides a venue for parallel computation leveraging discretization technologies developed and user-tested in the legacy uniprocessor codes NIKE3D and TOPAZ3D. Diablo is architected around Fortran 95 data structure objects and a message-passing programming model. The architecture provides flexibility for the addition of other field problems. For example, we have added electromagnetics via the EMSolve library to address electro-thermo-mechanical problems such as resistive Joule heating.

In structural analysis of mechanical assemblies, a key functionality is "contact": capturing the interaction between unbonded material interfaces. The LLNL team has broad experience with contact problems and has created state-of-the-art algorithms for their solution (e.g., [26]). Our experience with contact motivates the use of low-order spatial discretizations such as eight-node hexahedra for continuum and four-node quadrilaterals for shells. Appropriate formulations are employed to accommodate nearly incompressible material models such as for metal plasticity and rubber elasticity. Global algorithms include second-order and quasi-steady time integration and a number of approaches for nonlinear iteration: full Newton, modified-Newton, multiple quasi-Newton updates and line search. Linear solvers are utilized from multiple libraries.

3 Constructing the SHARP Multiphysics code

One can construct a multiphysics reactor core modeling code in many ways, and numerous efforts have attempted to enable to do so by providing a stepping-stone for future efforts. What distinguishes the SHARP effort from others is the goal of flexibility in the physics, discretization types, and software options supported by the framework. We begin by describing in more detail the approach we take to build a coupled multiphysics code framework. Then we describe the modifications necessary to connect physics modules to this framework.

The section ends with a description of the drivers constructed to run coupled problems. Although the primary purpose of this section is to report work done to run the test problems discussed in the report, its secondary goal is illustrate how various existing physics codes have been connected to this framework.

3.1 Approach

As stated in Section 1, there are two obvious approaches to assembling a coupled-physics simulation capability for analyzing reactor core systems. Of the two options, we choose to use the "bottom-up" approach for its ability to use existing physics codes and to take advantage of existing infrastructure

capabilities in MOAB and associated tools. Using an existing physics code in this system requires that the system support whichever mesh type(s) the individual physics uses. In practice, this means that the coupled system may be solved on multiple meshes, each of which models part or all of the physical domain of the problem. 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 because of coupling between physics. “Two-way” transfer is required in cases where the physics depend on each other’s solution fields, for example where Physics A computes heat generation based on temperature properties computed by Physics B, which depends in part on heat generation computed by Physics A.

Figure 1 illustrates the bottom-up approach used in this effort. The MOAB library provides a representation of the meshes and solution data associated with each physics. Each physics model can retain its own native representation of the mesh, which gets transferred to and from MOAB’s representation through a Mesh Adaptor (Physics Model A in Figure 1); or, it can use MOAB’s representation directly (Physics Model B). MOAB can be accessed directly by applications implemented in the C++ language (Physics Model C) or from C- and Fortran-based applications through the iMesh interface (Physics Models A and B).

At the most basic level of capability, integration of a given physics code into this system requires reading the mesh and possibly associated data from iMesh/MOAB on the front end and writing solution variables back onto the mesh after their computation. 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 can be quite difficult. This is described in more detail for the Nek5000 and PROTEUS models in Sections 3.3 and 3.2 respectively.

MOAB provides access to various mesh-based services through executable programs or libraries provided with MOAB. Several of these services are central to the overall goals of this effort. For example, MOAB can access mesh files written by the CUBIT [12] and the Reactor Geometry Generator [27] tools, which were used to generate models for this project (see Section 4). The *mbpart* tool performs partitioning of a mesh for parallel solution using the Zoltan library [10]. MOAB includes a plugin for the ParaView visualization tool [11], and can also output results in the *vtk* file format, readable by Visit [28] and other visualization tools.

To facilitate the coupling of solution data between physics modules in the multiphysics codes, we start by assuming each code will obtain its mesh definition by first reading the mesh into MOAB, then by loading that mesh data into the physics module. A single instance of the MOAB library is used throughout the code, with individual meshes used by each physics module distinguished by reading them into a “file set” specific to each physics (see Section 2.1 for a description of MOAB’s data model.) In the case of PROTEUS, MOAB is used to initialize a single layer of ghost or halo elements around the mesh on a given processor, using iMeshP functions to distinguish between locally owned and non-locally owned vertices and elements. The mapping between the mesh representations in MOAB and each physics is maintained in order to allow the application to write solution data back into MOAB. The MOAB-based representation of this data is necessary for both MOAB-based I/O and also because the solution transfer between meshes is implemented to interact directly with MOAB-based data.

Coordination of the overall multiphysics computation is the job of the coupled multiphysics driver. The COUPE (Coupled Physics Environment) driver, developed at Argonne, supports two-way coupling, with physics models and supporting services (i.e., MBCoupler) all interacting with the MOAB library directly in memory. A variety of coupling methods will eventually be supported by this driver, from loosely coupled to tightly coupled methods. The COUPE driver is described in

more detail in Section 3.5.

Moreover, the Diablo code itself supports one-way coupling of Nek5000 temperature results to the structural mechanics module. Diablo uses a MOAB interface to obtain temperature data from the Nek5000 solution in order to compute displacements.

Figure 1: Depiction of the SHARP framework, with MOAB as data backplane and COUPE driving the standalone or coupled physics calculations.

3.2 PROTEUS-MOAB Integration

Four primary types of information are necessary to initialize PROTEUS from MOAB:

Vertices: Mesh vertices, including physical coordinates

Elements: Mesh elements, element type (triangle, quad, brick) and its mesh vertex connectivity

Element Block: Organization of elements according to their material or element block assignment

Sides: Identification of the domain surface Neumann-type boundary conditions, represented as element sides (each side defined by a global element id and a local side number).

Elements and vertices are numbered globally and locally, such that they can be uniquely associated with degrees of freedom in the solution vectors. Mapping arrays must be provided that associate all vertices and elements with a global index. The list of vertex positions is stored in a single multidimensional array while the elements are grouped into blocks depending upon the element type and material designation. In parallel, standard domain decomposition is used where each process is assigned a subset of the vertices and elements on which it computes the solution. A single layer of ghost or halo elements is also required. In parallel, the vertices and elements are grouped into locally owned and ghosted sets, where the combined information is termed "visible." All elements in a block are either owned or ghosted. The set of owned elements (and thus owned

blocks) appear first on a given processor followed by the set of ghosted elements (and thus ghosted blocks). Similarly, the owned vertices appear first followed by the ghosted vertices.

PROTEUS is written primarily in Fortran90 and uses data structures contained in Fortran modules to store the mesh and solution data. Focusing on the mesh alone, the `NTmesh` (Neutron Transport mesh) Fortran module stores the processor assigned piece of the domain using the local ordering. `NTmesh` includes the vertex position information, element type and connectivity, and any domain boundary conditions applied to the locally visible elements. In parallel calculations, an `NTmesh` object is stored within the `PNTmesh` (Parallel Neutron Transport mesh) data structure which is defined in the `PNTmesh` module. `PNTmesh` stores the global index information of both the vertices and elements visible on the processor. For the connection to MOAB, an additional data structure was added that includes vectors storing the MOAB vertex and element handles. The ordering of these arrays is consistent with that of PROTEUS in order to facilitate transfer of PROTEUS-computed fields back to MOAB. As for the connectivity of each element, the element setup in PROTEUS is either identical to MOAB's or translated internally to PROTEUS at run time when the elementwise trial function evaluations are called. Note that not all element types in PROTEUS are supported by the MOAB interface at this time.

The solution data from PROTEUS is stored in MOAB in the form of tags on vertices. Table 1 shows the tag names, data types, sizes, and corresponding variables in PROTEUS for the input and output variables. Inside PROTEUS the element properties of temperature and density are kept in the data structure `Input_MeshBlockProperties` in the array `Evaluation`, dimensioned by field name (f1/f2 in Table 1) and block.

As input, temperature and density are updated at the beginning of each PROTEUS solution step where both are volumetrically averaged over a given block of elements, as discussed in Section 2.3 With regard to output, an elementwise power distribution is computed by integrating the flux solution over space, angle, and energy, where energy (group) dependent conversion factors are used to convert fission and capture (in place) reaction rates into power. This computation is performed in the `src/SN2ND/SN2ND_MOAB_ExportData.F90` file in the PROTEUS source and converted to an estimated vertex based quantity for use in MOAB for this initial coupling work.

3.3 Nek-MOAB Integration

Nek5000 relies on information from MOAB similar to that needed by PROTEUS: vertices, elements, element blocks or materials, and boundary conditions or "sides". Nek5000 models only two materials, fluid and solid. Nek5000 requires that all fluid elements be numbered before all solid elements, both locally on each processor and in the global numbering. Nek5000 requires 27-node hexahedral elements, on which it computes a N^{th} -order spectral mesh.

Nek5000 already had the ability to read and write meshes and results from and to MOAB, as part of previous years' NEAMS-sponsored work. For this effort, the Nek-MOAB interface was enhanced in several significant ways:

- The interface was broadened to treat both solid and fluid elements; previously the Nek-MOAB interface had only treated fluid elements.
- The input language and processing for text-based input was expanded to allow user specification of materials and boundary conditions by element block or sideset in the input. This also enabled Nek5000 to initialize from subsets of mesh files. Previously, the code was compiled with hard-set numbers of fluid and total elements, with all fluid elements having to be

Table 1: Tags used for input and output of solution data between Nek5000 and MOAB.

Description	Tag name	Assigned to (V=vertices, E=elements)	Data type (I=Int, D=Dbf)	Size (#values)	Variable PROTEUS (module%variable)/ Nek5000 (common:variable)
Power	VPOWER	V	D	1	none ¹ / BQCB:BQ
Power	POWER	E	D	n_{tot}^2	none ¹ / BQCB:BQ
Temperature	VTEMP	V	D	1	Input_MeshBlockProperties%Evaluation(f1,b) / VPTSOL:T
Temperature	TEMP	E	D	n_{tot}^2	Input_MeshBlockProperties%Evaluation(f1,b) / VPTSOL:T
Density	VDENSITY	V	D	1	Input_MeshBlockProperties%Evaluation(f2,b) / VPTSOL:VTRANS
Density	DENSITY	E	D	n_{tot}^2	Input_MeshBlockProperties%Evaluation(f2,b) / VPTSOL:VTRANS
Spectral vertex positions	SEM_X, SEM_Y, SEM_Z	E E	D D	n_{tot}^2 n_{tot}^2	none / GXYZ: XM1, YM1, ZM1 none / GXYZ: XM1, YM1, ZM1
Velocity	VX, VY, VZ	E	D	n_{tot}^2	none / GXYZ: XM1, YM1, ZM1
Pressure	PRESS	E	D	n_{tot}^2	none / CBM2: PR
Spectral order	SEM_DIMS	FS ³	I	3	none / DIMN: NX, NY, NZ

Notes:

1. In PROTEUS, power values are computed purely for output to MOAB, and are discarded afterwards.
2. $n_{tot} = n_x \times n_y \times n_z = n^{th}$ -order spectral mesh in each element
3. FS = file set, the set into which entities read from a file are placed.

read from the mesh file before all solid elements. This provided the flexibility needed to read meshes with the complexity of the XX09 SFR assembly.

- Subroutines were written to support both input and output of various types of solution data between Nek5000 and MOAB. This was necessary to support true two-way coupled analysis. Data was transferred in two varieties, one type at corner vertices of mesh elements, and another stored on mesh elements, in arrays corresponding to the spectral vertices in each element.
- Global ids are read with the elements from MOAB, however those ids do not have the ordering requirements imposed by Nek5000. Therefore, a new algorithm was implemented to compute a global numbering that places all fluid elements before all solid elements in the numbering.

Table 1 shows the solution data and tags used to import/export solution data between Nek5000 and MOAB. The corner vertex-defined fields are used with linear finite element shape functions in the coupler, while the element-based arrays are used with a true spectral element shape function. Spectral element support was added to MBCoupler as part of this effort, but that part of the code remains untested for the problems described in this report.

3.4 MBCoupler Enhancements

Nek5000 is based on a high order spectral element method, where the order of spatial discretization varies usually from 5 to 15. The solution data is stored/computed for each element at the Gauss-Lobatto points which includes corner vertex nodes. The MBCoupler tool was extended to be able to couple solutions for this type of element without giving up the higher order spatial accuracy. Field function evaluation and inverse computation of parametric positions for a spectral element were implemented outside Nek5000, to enable efficient coupling with other physics (PROTEUS), while keeping solution codes relatively independent.

Field solution transfer is possible in both directions, from a source spectral mesh to a target linear element mesh, and vice-versa. This capability has been utilized to enable two-way coupling with the COUPE driver.

3.5 COUPE : A New Coupled Driver Code

COUPE stands for the Coupled Physics Environment library and is part of the SHARP framework. COUPE enables scalable and extensible coupling of different physics components that are nonlinearly coupled to each other. The SHARP multiphysics coupled code for reactor analysis problems employs validated and verified efficient monophysics codes with MPI architecture to achieve the loose coupling with an operator split (OS) methodology [29]. Such "divide and conquer" methods provide flexibility in the usage of standard industrial codes and avoid replicating man-years of development and testing by following the bottom-up approach described in Section 3.1.

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

1. Make use of existing libraries and physics codes in order to minimize development time, and to base the framework on already well-verified and validated monophysics codes and libraries;
2. To 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.
3. To 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.
4. To be able to 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 solve all of the physics components under a unified framework in order to exchange the solution from one physics to another and converge the coupled physics solution fields to user-specified tolerances 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 also provides the necessary tools to quickly implement any of the popular variations of an operator split coupled solver (Marchuk, Strang, Yanenko among others) or a more rigorous matrix-free inexact-Newton solver with a Jacobian-free Newton-Krylov (JFNK) technique [30]. At the time of writing the report, the Marchuk splitting with Picard iteration over the physics components has been implemented as part of the library; the other types of coupling strategies will be written and tested later this year.

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 monophysics 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 [24]. The current design of COUPE was put forth in order to satisfy the need for a loosely coupled 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 an uniform interface. The driver is simple, transparent and extensible and can be referred to 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.

The two primary objects exposed by the COUPE library are:

Coupe

The coupled physics solver implementation; currently, two subtypes of first order splitting exist and can be invoked via command-line arguments.

Physics

The physics object that interfaces to each underlying monophysics code, with wrapper functions for creating, initializing, setting up, solving and destroying data structures relevant to the wrapped code.

Similar to the PETSc toolkit library, COUPE was 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 the solver, and in advanced usage, can even dynamically change the type of the physics being coupled. This is made possible by completely abstracting out behavior of the core object until runtime, even though the internals of these objects are fully available to the driver if necessary. Hence, the core implementation of a physics object is hidden while the driver utilizes only the methods exposed in the public interface. 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 reused in a variety of different coupling methods (e.g., loose versus tight coupling).

A sample driver implementation for the PROTEUS-Nek two-way coupling is provided in Appendix B. The code shows how a mesh loaded externally in the driver can be referenced inside each of the individual codes, namely PROTEUS, Nek5000, or Diablo, and then associated with the coupled physics object COUPE. With the corresponding `iMesh_Instance` and `File_Set` references, the COUPE object can call the `MBCoupler` to interpolate and project the solution defined by one physics (on the source mesh) to the corresponding vertex points of the other physics (the target mesh).

The coupled fields are iterated until convergence to user-required accuracy for any problem of interest. Preliminary results obtained from this coupled physics driver dependent on the COUPE library are shown in Section 5.

4 Test Problems

AssyGen is a part of the Reactor Geometry (and mesh) Generator (RGG) [27, 31] in MeshKit [32], and has been used to create the assembly geometry and meshing script for the reactor assemblies described below. Support for creating frustum or conical pincells and creating pincells/ducts with multiple Z-heights was added to this tool. The automation provided by **AssyGen** greatly improved the total human and computational time required to generate such models. We created several test problems.

4.1 Simplified Single Assembly

A simple hexagonal assembly (SAHEX1) and a simple rectangular assembly were defined. Results for coupled simulation of SAHEX1 are presented in Section 5.1.3. Meshes with varying resolution are used for PROTEUS (2694 hex8 elements) and Nek5000 (1506 hex27 elements). The model consists of fuel pins, cladding, control rod, steel can and sodium. This model is carefully chosen to demonstrate the proof of concept for solving the XX09 assembly used in EBR II SHRT-17 (described later in this section). The SAHEX1 geometry and meshes are shown in Figure 2.

Figure 2: Test problem meshes: (i) SAHEX1 PROTEUS mesh, (ii) SAHEX1 Nek5000 mesh, (iii) 8 partitions of SAHEX1, and (iv) Simple rectangular assembly mesh.

Partitions are created using the `mbpart` utility in MOAB. For PROTEUS we define five materials fuel, clad, control rod, moderator and duct. Density of all materials is chosen to be 1.0. Boundary conditions are applied on the top/bottom and outermost surface of the model. More about the PROTEUS setup and boundary condition specification is given in Section 2.3. Nek5000 uses a coarser mesh (based on total number of hexes) than the PROTEUS mesh. Conjugate heat transfer is setup with input heat generation provided by PROTEUS simulation. The outermost walls are considered insulated, temperature and velocity boundary conditions are applied to the inlet and outlet, and side surfaces between fluid and clad are set as wall boundary conditions. More about the Nek5000 setup and boundary condition specification is given in Section 2.2.

4.2 Complete Single Assembly - XX09

Detailed specifications for the XX09 assembly are described in a previous report [6]. In order to generate the XX09 assembly, two **AssyGen** input files were constructed, uassembly.inp and lshield.inp. The first input file creates the upper assembly and the second creates the lower shield or inlet region of the assembly. These files are included in Appendix A of this report. Using these input files, **AssyGen** creates four CUBIT mesh scripts. These scripts are modified to create fillets around the edges of the hexagon ducts. The loft described in the lshield.inp file is a box with circular top and hexagonal bottom. Meshing around the region of the loft poses additional challenges; appropriate Z-intervals are chosen to successfully sweep the loft, and the hexagonal source surface of the sweep has all the pins that need to be swept to the circular target surface. The region around the loft is swept radially. Figure 3 (ii) shows the Z section with the loft region where hexagonal geometry transitions to circular geometry.

Figure 3: XZ-plane section views of the entire assembly from inlet to outlet. Material coloring - pink:stainless steel, yellow: sodium, blue: clad, light blue: fuel, white: bond sodium and red: fission gas. (i) Cylindrical inlet transitions to form a cone from left to right, (ii) interface of inlet sodium and pointed conical lower plug (also shows the fuel), (iii) top of fuel surrounded by bond sodium and filled on top with bond sodium and fission gas, and(iv) Upper plug and outlet sodium.

The mesh size of the model is governed by the smaller radii of the frustum (0.5mm). The total Z-height of the model is 1337.36 mm and the total number of elements in the model are dictated by the number of intervals in different Z-subdivisions. The model is created in millimeters and then scaled to centimeters to match the different units specified in the boundary conditions.

The Nek5000 mesh used in the simulation consists of 137537 hex27 elements, whereas the

PROTEUS mesh contains 505815 hex8 elements. The PROTEUS mesh has four times more elements in the fuel region than the Nek5000 mesh. The meshing process starts at the XY plane; this surface is the key to meshing both the upper assembly and lower shield. Mesh sizes and schemes for circular pincell and surrounding hex cell surfaces are chosen to form quadrilateral elements with an aspect ratio close to 1.0. In the overall 3D model, the ratio of maximum to minimum aspect ratio of the elements used in this model is 1320 because of the large axial height compared to small characteristic size in the cross section. Such high values of aspect ratio are found in the inlet region, where all the fuel region mesh is swept towards the small circular region, giving rise to long slender elements.

For PROTEUS, 13 material and 5 boundary conditions are created for PROTEUS. The model is divided into four regions based on the axial height: inlet, active core, plenum, and outlet. Boundary conditions for the top, bottom, and outermost surfaces are applied. More about PROTEUS setup and boundary condition specification is given in Section 2.3. For Nek5000, 6 materials and 13 boundary conditions are created. Wall boundary conditions are applied on all side surfaces of the geometry. Velocity and temperature boundary conditions are applied at the bottom surface of the model, and outlet and temperature boundary conditions are applied at the top surface of the model. Note that in the simplified loss of heat sink transients for both SAHEX1 and XX09 the temperature is specified as a function of time:

$$T(t) = T_0 \quad (1)$$

until $t = t_0$ and then as:

$$T(t) = T_0 + (T_1 - T_0) \tanh \frac{t - t_0}{\delta t} \quad (2)$$

The time t_0 is therefore the beginning of the transient, T_0 is the initial temperature and δt is the duration of the transient at the inlet. T_1 is the final temperature at the inlet. $T_1 - T_0$ has been set 120 K in the present work.

The mesh is partitioned to 1024 parts using `mbpart`. Figure 4 shows the mesh partition, and the top view of the mesh with 59 fuel pins and 2 flowmeters.

4.3 Simplified Multiple Assembly Test Problem

In order to perform a full core simulation with PROTEUS and Nek5000, some form of homogenization is necessary as solving the full core heterogeneously is in fact likely to result in excessive computational requirements.

An important step toward a full-core simulation is therefore to test homogenization capabilities in both Nek5000 and PROTEUS. The chosen test case is an extension of the SAHEX1 model described in this section. It adds six homogenized neighboring assemblies to the full heterogeneous SAHEX1 model.

The PROTEUS mesh for this model is shown in Fig. 5. In the PROTEUS model, reflective boundary conditions are assumed at the boundaries of the domain. In the Nek5000 model the boundary is assumed adiabatic.

In the homogenized assemblies material properties and cross sections are homogenized in PROTEUS. In Nek5000, in each homogenized assembly two one-dimensional equations are solved for the temperature T in time t and axial coordinate x :

Figure 4: XX09 assembly: (i) inlet region partitioned mesh, (ii) isometric view of 1024 partitions, (iii) top view of the outlet partition, and (iv) top view of mesh showing 61 pins.

Figure 5: SAHEX1 multiple assembly (SAHEX1+6), PROTEUS mesh

$$\rho^f c_p^f \frac{\partial}{\partial t} T^f = -h l^f (T^f - T^s) + \ddot{q}(x) \quad (3)$$

$$\rho^s c_p^s \frac{\partial}{\partial t} T^s = h l^s (T^f - T^s) - v^s \rho^s c_p^s \frac{\partial}{\partial x} T^s \quad (4)$$

where the superscript s refers to sodium and the superscript f refers to fuel. This very simple model removes the thermal inertia of the ducts and the cladding; these can be added later.

The variable h is the heat transfer coefficient to be determined by appropriate correlations. The variable v^s is the bulk velocity of sodium in the assembly. Here q is the volumetric heat, and is obtained from integration of PROTEUS results. The variables l^f and l^s are appropriate heat transfer length-scales.

Both temperatures T_s and T_f are stored, but only the temperature of sodium is passed to PROTEUS which then recomputes the properties of the homogenized regions.

5 Simulation Results

In this section results from the set of demonstration simulations are discussed.

5.1 SAHEX1 Test Results

In this section the results pertinent to the simplified single assembly are discussed. At first the uncoupled results are discussed, then the results of the coupled simulations are presented (Section 5.1.3).

5.1.1 Nek5000 SAHEX1 Results

Most development testing has been performed using the MOAB-coupled conjugate heat transfer problems in the Nek5000 repository. As an additional test, we ran the SAHEX1 problem described in the previous section (Section 4).

For the coupled simulation the material properties have then been updated to their dimensional values using appropriate correlations. The results of this new run have been used as an initial condition of the coupled problem. Figure 6 shows preliminary results for the temperature distribution with polynomial order $N = 2$.

5.1.2 PROTEUS SAHEX1 Results

Because of the size of the XX09 benchmark, the SAHEX1 problem was chosen as the initial proof of principle test to ensure a reliable and accurate solution in the full benchmark. The decoupled neutronics solution (and the initial condition of the coupled calculation) uses average core temperatures, densities, and compositions from the XX09. Similar to the XX09 problem, reflected boundary conditions are used in the radial direction, and both upper and lower surfaces are assigned vacuum boundary conditions. We also note that the control rod was filled with sodium.

The neutronics solution without feedback (i.e., initial condition) is trivial and has a cosine power shape as displayed in the upper picture of Figure 7, where the lower picture shows the geometry and the middle picture the fuel pins. Since all the elements are in a single block, the existing

Figure 6: Steady state temperature distribution for the test configuration of the SAHEX1 problem. Temperatures in [K]

Figure 7: Power profile for the SAHEX1 problem.

scheme imposes an average temperature for all fuel pins which merely adjusts the temperature/-density to the proper values derived from the volume of the fuel and selected coolant flowrate (and properties). Given an accurate solution in that configuration, we can improve the feedback model in the neutronics calculation to allow an axial profile of temperature and densities.

The peak of the cosine shape of the power distribution shifts slightly towards the inlet of the core, while the peak temperatures are observable near the outlet. The final test phase of the SAHEX1 problem allows each fuel pin to have its own radial and axial feedback assignment, at which point the coolant mixing between the fuel pins should cause notable patterns in the power distribution. On the SAHEX1 geometry, such patterns are meaningful, and any errors with interpolation is immediately visible, whereas in the large and complicated XX09 geometry catching mistakes is harder.

5.1.3 Coupled Nek5000/PROTEUS SAHEX1 Results

As part of the coupled simulations effort, several SAHEX1 cases have been run to test and expand the coupling capabilities of SHARP. In particular, the following key capabilities needed to simulate a full core transient have been tested:

- simulation of an initial steady-state at the rating conditions,
- simulation of a pseudo steady-state transient at constant power, and
- simulation of a transient with varying power.

Three SAHEX1 cases have therefore been run:

- **SAHEX1-a**, an initial steady-state transient,
- **SAHEX1-b**, a pseudo-steady state loss of heat sink transient where the inlet temperature is modified at a given time following a function of the type described in 2.2, and
- **SAHEX1-c**, a full loss of heat sink transient (the difference with the previous case is that the total power is not considered constant).

In the following subsections, the procedures to obtain the initial steady-state and pseudo steady-state fields are described. Also described is the newly added capability in PROTEUS to compute the power transients using the quasi-static approximation and the point-kinetics equations.

Initial Steady-State and Pseudo Steady-State Transient In this section the initial steady-state (ISS) and the pseudo steady-state (PSS) loss-of-heat-sink transient are presented. The initial condition in Nek has been fixed to assume constant temperature within the pins and within the fluid shown in Figure 8. The initial power distribution projected on the Nek5000 mesh, is shown in Figure 9.

Figure 8: Volume rendering of initial condition in the Nek5000 PSS [K].

After a fixed number of time steps in Nek5000, the power distribution is updated through a PROTEUS iteration. At a given time t_0 , COUPE switches the transient to a simplified loss-of-heat-sink PSS by updating the inlet boundary conditions in Nek5000 using an arctan function. The inlet temperature goes from 600 K to 720 K during the transient.

For both the ISS and PSS, the power distributions are normalized by the same value. This corresponds to a localized transient, or a transient where it is assumed that the injection/subtraction of reactivity is compensated by some external action of the operator. A more realistic transient is described in the following section.

Figure 9: Volume rendering of the power distribution at the beginning of the PSS [W].

The end result is independent of the number of time steps in each Nek5000 iteration. The path toward the end state is, however, dependent upon the number of time-steps per iteration. In fact with a lower number of Nek5000 time steps per iteration the power-shape distribution and neutronic multiplication factor k_{eff} will be updated more frequently, influencing the overall evolution of the transient.

Several ISS+PSS transients have been performed to test for different feedbacks and sensitivity to the number of time steps per iteration. The cases are summarized in Table 2. Note that cases 3 and 4 differ in the feedback effects included. In case 4, the density data is not used in PROTEUS and only the temperature field is updated.

Table 2: ISS+PSS SAHEX1 cases

Case	Number of Nek5000 t.s./it	Temperature Coupling	Density Coupling
1	250	Y	Y
2	500	Y	Y
3	1000	Y	Y
4	1000	Y	N

Figure 11 shows the change in k_{eff} as a function of time. Figure 10 shows the difference in power shape between the beginning of the transient and the end of the transient. Note the sensitivity to the number of time steps per iteration (the number listed for each case in the legend represent the number of Nek5000 time steps per iteration). The flow time of the sodium through the assembly is 0.89 s, the total time length of a single Nek5000 iteration should be significantly smaller than the total flow through time. The path followed by the k_{eff} rapidly converges as the number of Nek5000 time steps is halved and this condition is met. Note that feedback based on both temperature and density are necessary.

Let us now examine in detail Case 1 from Table 2. Figure 12 shows the evolution of the temperature distribution in the axial plane $z = L/2$, where L is the length of the domain in the axial direction. The figure progresses in time from left to right.

In conclusion, an initial steady-state and a pseudo steady-state capability have been tested for SAHEX1. The sensitivity tests led to important conclusions on:

1. the number of time step per iteration necessary for accuracy, and
2. the importance of the inclusion of all different types of feedback.

Power Transient The PROTEUS-SN code was updated to include a classic quasi-static time-dependent modeling capability. This capability most closely represents point kinetics, where the kinetics parameters are updated at each time step, rather than conventional point kinetics where the parameters change only via empirical expressions (i.e., linear change with respect to temperature). The time-dependent modeling accuracy of this scheme is known to be poor for most reactor systems because it fails to adequately account for the spatial dependence of the precursor concentrations.

We considered using the improved quasi-static time dependent scheme, the second most commonly used scheme (after point kinetics), but it requires the inclusion of a non-homogeneous solver, which we did not have sufficient time to include. With time we will either replace the existing time-dependent scheme (termed adiabatic) with the improved quasistatic formulation or a high-order time-dependent formulation.

The quasi-static capability allows a full transient to be run with varying power. Using this capability, we reran the simulations performed in the previous section and allow the global power

Figure 10: Volume rendering of difference in power between initial condition and final condition for Case 1 (Table 2).

Figure 11: ISS+PSS transients.

Figure 12: ISS+PSS Case 1 (Table 2), temperature on the middle cross section [K].

to change. The number of Nek5000 iterations per global iteration required to correctly represent the transient dropped by an order of magnitude, making the transient much more computationally expensive (i.e., requires a higher number of PROTEUS solves).

The power as a function of the global iteration is plotted in Figure 13, where one can see the beginning the transient after approximately 3000 iterations.

Figure 13: Full quasi-static transient for SAHEX1, total power as a function of global iteration.

The changes in the profile can be seen in Figure 14. The time advances from left to right, with a decrease of the maximum power at latter times.

Figure 14: Full quasi-static transient for SAHEX1, power in the fuel pins as it progresses in time [W].

The corresponding evolution of the temperature profiles can be seen in Figure 15. Note that the significant change in power profile observed in Figure 14 corresponds to only a minor penetration of the high temperature front within the domain. This indicates a very fast response of the system.

Figure 15: Full quasi-static transient for SAHEX1, temperature profiles [K].

5.2 XX09 Results

In this section the results of the XX09 simulations are discussed. At first the uncoupled results for Nek5000 and PROTEUS are presented. The coupled Nek5000/PROTEUS simulations are then discussed (Section 5.2.3). In the last subsection the results of the Diablo simulations are presented.

5.2.1 Nek5000 Uncoupled Results for XX09 Assembly

We computed a test configuration of the XX09 problem with Nek5000 coupled to MOAB (“one-way coupling”). For simplicity of the test, we originally used a nondimensional setup with fixed physical properties and uniform unit heat generation distribution in fuel elements of the subassembly geometry (similar to Section 5.1.1). As a second step, a fully dimensional simulation was performed, and the results were used as initial conditions for the coupled problem.

The results used as the initial conditions are shown for the temperature and velocity magnitude distribution in Figure 16 and Figure 17, respectively.

5.2.2 PROTEUS Uncoupled Results for XX09 Assembly

Several calculations were performed without the Nek5000 coupling, in order to verify the setup of the neutronics calculation. The Legendre-Tchebychev product cubature was used for all these calculations because it allows the radial and axial directions to be varied independently. The minimal angular cubature usable on the XX09 domain is L1-T2 with 12 angular directions (2 axial and 6 radial), while we used L3-T5 for most of the results which contains 48 angular directions (4 axial and 12 radial). For energy, standard 4 group and 9 group structures were used which were expected to be insufficient to produce an accurate solution, but sufficient to demonstrate the coupling capability.

In fact, we consider the XX09 calculation to be a proof of principle calculation, and thus an average assembly material composition from the EBR-II core operating at the time of the experiment was used for the calculations. Despite the low angular resolution and poor energy resolutions, an eigenvalue of 1.27 was obtained which is realistic for the XX09 assembly under these conditions.

Figure 18 shows the power (lower two parts) and the peak energy flux distribution (upper part) for the XX09 assembly, where reflected boundary conditions were applied radially and vac-

Figure 16: Initial condition for the coupled transient: temperature distribution for the test configuration of the XX09 problem [K].

Figure 17: Initial conditions of the coupled transient: velocity magnitude for the test configuration of the XX09 problem [cm/s].

uum boundary conditions were applied axially. The furthest end of the geometry is the upper plenum region, while the closest most clearly displays the cone shaped inlet below the core. The dominant gradient is seen in the axial direction corresponding to the only path for leakage. Although not viewable on this figure, the XX09 geometry includes nonphysical reflected boundary conditions. While a plane geometry with edges would likely lead to more realistic boundary conditions for neutronics, the physical geometry specifies rounded edges. They alter the solution and cause difficulties with the neutronics solver.

Figure 18: Power (lower two parts) and the peak energy flux distribution (upper part) for the XX09 assembly.

Overall, the uncoupled neutronics solutions using SN2ND produced the expected results given that they were designed to minimize run time rather than to maximize accuracy. The SN2ND flux solution is not currently exported to MOAB, only the power solution is currently exported (Figure 18).

5.2.3 XX09 Coupled Results

In this section the results of the coupled Nek5000-PROTEUS simulation are reported. The simulation is identical in structure to that for SAHEX1 (Section 5.1.3). COUPE is used to drive the coupled PROTEUS/Nek5000 simulation. Each iteration includes 5000 Nek5000 time steps, corresponding to 0.03 s in total time.

The relative error during the first 20 iterations of the ISS calculation are shown in Figure 19, which shows the asymptotic regime of linear decrease in the global error as expected by the implemented Picard iteration.

Figure 19: Initial steady-state, relative error in PROTEUS and Nek5000.

After an initial steady state is reached: a pseudo state transient is performed following the same pattern as in SAHEX1. The k_{eff} as a function of the number of global iterations in the PSS is shown in Figure 20. The increased temperature produces a reduction in reactivity.

Figure 20: k_{eff} as a function of global iteration.

In the initial stage of the transient, a front with increased temperature slowly advances through the assembly as shown in Figure 21.

In the second stage of the transient, the temperature increases in the fuel region (Figure 22, Figure 23) producing the steepest change in reactivity.

Figure 21: Temperature [K] in the PSS, sequence from the left to right (initial stages of the transient)

Figure 22: Temperature [K] in the PSS, sequence from left to right. Axial location in the middle of the assembly.

Figure 23: Temperature [K] in the PSS, sequence from left to right. Middle-section side view.

5.2.4 XX09 Diablo Results

The thermal results obtained in the Section 5.2.3 were loaded into Diablo, and some preliminary results were obtained. The finite element mesh used for structural mechanics is identical to that used for neutronics in PROTEUS. Although not optimal for structural response, the mesh is functional for this initial demonstration. We applied kinematic boundary conditions at the base of the assembly sufficient to eliminate any rigid-body motions. Specifically, all axial (Z) displacements were set to zero. For a small number of nodes at the base of the inner and outer duct the transverse displacements (X or Y) were also set to zero. Handbook properties for stainless steel were utilized for the duct and cladding. Material properties for uranium metal were used for the fuel elements. The sodium regions of the model were retained, but the elastic modulus was set 10^6 times smaller than the steel to preclude it from providing any constraint against motion of the structural components.

Figure 24 shows a plot of axial displacement contours with lateral displacements scaled by a factor 25 to illustrate the warping of the duct. The total axial deformation was checked with a hand calculation.

Figure 24: Duct transverse displacements enhanced by 25X

Figure 25 shows the assembly with most of the duct removed to reveal the fuel assembly. It illustrates contours of the temperature field on the fuel pins and cladding as imposed by using data from a MOAB database written by Nek5000.

Figure 26 shows the XX09 assembly (with most of duct removed) with contours of temperature field on fuel pins and cladding, with lateral displacements scaled 100 times. The pins spread away from each other as they are typically hotter on the side facing the center of the bundle. Collectively they show a bowing to one side, which is consistent with the temperature field being asymmetric in the XX09 assembly.

Figure 27 shows the assembly with most of duct removed; contours of temperature field on fuel pins and cladding with lateral displacements are scaled by a factor of 10^2 . In this case, slightly different from Figure 26, the fuel Young's modulus was lowered 10^3 to prevent the fuel-cladding

Figure 25: Pin temperatures as used for thermal boundary conditions in Diablo

Figure 26: Pin lateral displacements enhanced by a factor 10^2 .

interaction from being overestimated using the merged neutronics mesh. The sodium layer appears to provide adequate relief, and displacements are thus not significantly different.

Figure 27: Pin displacement enhanced by a factor 10^2 , Special case with reduced Young's modulus for the fuel.

Figure 28 shows the assembly with most of duct removed; with contours of effective stress field on fuel pins and cladding, with lateral displacements scaled by a factor 10^6 . The soft fuel case on the right shows the expected effect of lowering the fuel pin stress.

Figure 28: Stress distributions for two values of the stress modulus.

5.3 Preliminary results for Multiple Assemblies

To fulfill the objective of being able to simulate a full core as part of the coupled simulation effort, a homogenized assembly capability is being developed within each individual physics code. At this

stage only the standalone capabilities of the individual physics modules have been demonstrated. A full coupled simulation has not yet been performed because of time constraints.

Figure 29 shows preliminary results for the temperature distribution obtained with Nek5000 for the SAHEX1+6 problem (Section 4.3). In the heterogeneous regions the local temperature (either fluid or solid) is plotted. In the homogenized regions the temperature corresponds to the fluid temperature.

The methodology used to compute the temperature in the homogenized regions is described in Section 4.3. Properties and coefficients for the homogenized model have been determined using appropriate correlations.

6 Future Plans

This report describes a work in progress building a coupled neutronics, thermal/fluids, fuel performance, and structural mechanics modeling capability for modeling reactor cores. Work in the immediate future is planned to extend the demonstration of a coupled neutronics/thermal-fluids simulation to the full EBR-II core. A loss-of-heat-sink transient will be first simulated with full coupling, including the structural mechanics code Diablo. In such a simulation the XX09 fuel assembly and a few other assemblies will be represented with their full geometric complexity while most assemblies in the core will be homogenized.

Next the full SHRT-17 test will be simulated. Since the coupled code has been demonstrated on a smaller problem, this should be a relatively straightforward task, though computationally expensive.

From there, we will proceed along multiple paths. Specific tasks include further implementation and verification of the MOAB mesh interface so that it can contain the mesh information, and the MOAB tag interface, so that it can hold the parallel vectors of data.

Along with expanding the vertical capability in terms of new physics models, the work may also be expanded horizontally, that is, by adding additional modules for physics already covered by existing modules. The URANS capability currently under development in Nek5000 is of particular interest. This capability will help reduce the computational requirements to achieve reasonable accuracy.

In the longer term, we see this code framework as a good basis for studying interesting questions about the numerical aspects of coupling. Specifically, we envision exploring the tradeoffs between various loose and tight coupling methods, in terms of accuracy, computational cost, and code complexity.

The ultimate goal of this coupling simulation effort is to compute the end result of the EBR-II series of tests: the safe shutdown of a fast reactor as a result of the combined feedback of neutronics, thermal/fluid and structural mechanics processes.

7 Conclusions

The NEAMS Reactor Product Line effort aims to develop an integrated multiphysics simulation capability for the design and analysis of future generations of nuclear power plants. Ultimately, the Reactor Product Line multiresolution hierarchy is being designed to span the full range of length and time scales present in relevant reactor design and safety analyses, as well as scale from

Figure 29: Preliminary Nek5000 steady-state temperature distribution for the test configuration of the SAHEX1+6 problem.

desktop to petaflop computing platforms. In this report, we described our efforts to integrate thermal/hydraulics, neutronics, and fuel performance modeling codes to perform coupled analysis of a representative fast sodium-cooled reactor fuel assembly.

We reported on the integration of PROTEUS and Nek5000 into the SHARP framework. Both codes have been modified to read and write data from and to the MOAB library, in both serial and parallel environments. We focused on broad support for as many original physics code options as possible, rather than focusing only on near-term needs. Both codes were demonstrated through this interface on a relatively simple seven-pin hexagonal fuel assembly. In order to support coupled analysis, the new multi-physics driver library COUPE was developed; this library provides flexible support for both loose and tight coupling and can interface flexibly with codes in multiple programming languages. In order to couple results between separate meshes used in each physics, the COUPE driver, PROTEUS, and Nek5000 all interface with the MBCoupler solution transfer tool provided with MOAB. Together, these tools form the basis of a new, coupled neutronics and thermal/fluid modeling code that works in both serial and parallel. In a related effort, a driver in the Diablo code was used to access Nek5000-computed results through the MBCoupler solution transfer tool, and used to drive structural mechanics computations.

Another important goal of this effort is to perform coupled modeling of a series of test problems, including a simplified loss-of-heat-sink transient in an EBR-II fuel assembly (labeled XX09 in the EBR-II Shutdown Heat Removal test). Several high-fidelity meshes were generated for this model, in the range of 130-800k hexahedral elements.

Several single-physics and coupled calculations were performed on this model. The objective was to demonstrate reasonable behavior during preliminary tests and in this sense they were successful. Due to lack of validation data for coupled simulation data, this is a necessary intermediate step.

In particular, the following capabilities have been demonstrated:

- simulation of an initial steady-state at the rating conditions,
- simulation of a pseudo steady-state transient at constant power,
- simulation of a transient with varying power, and
- simulation of a steady state with multiple assemblies, part of which are homogenized.

All of these capabilities are important milestones on the path to a full core simulation.

Acknowledgments

This work was completed as part of the SHARP reactor performance and safety code suite development project, which is funded under the auspices of the Nuclear Energy Advanced Modeling and Simulation (NEAMS) program of the U.S. Department of Energy Office of Nuclear Energy, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

A RGG Input for XX09 Assembly

1. Upper Assembly AssyGen Input File.

```

#####
! AssyGen Input File: Upper Assembly: XX09 assembly used in EBR II STRT-17 assembly
! All dimensions are in mm; Note: The final model is scaled to cm
! Location of origin is "Bottom of Fuel"
#####
GeometryType Hexagonal
#####
Materials 6 Sodium NA StSteel SS Fuel F Clad C BondNA G FissionGas FG
#####
Duct 4 0.0 0.0 -7.07 -3.76 46.41 48.44 56.13 58.17 NA SS NA SS ! Lower Plug
Duct 4 0.0 0.0 -3.76 0.0 46.41 48.44 56.13 58.17 NA SS NA SS ! Lower Plug
Duct 4 0.0 0.0 0.0 342.9 46.41 48.44 56.13 58.17 NA SS NA SS ! Fuel
Duct 4 0.0 0.0 342.90 355.6 46.41 48.44 56.13 58.17 NA SS NA SS ! Bond Sodium
Duct 4 0.0 0.0 355.6 590.43 46.41 48.44 56.13 58.17 NA SS NA SS ! Fission Gas Plenum
Duct 4 0.0 0.0 590.43 593.09 46.41 48.44 56.13 58.17 NA SS NA SS ! Upper Plug
Duct 4 0.0 0.0 593.09 816.66 46.41 48.44 56.13 58.17 NA SS NA SS ! Upper Assembly
#####
Pincells 2 5.67
Mark_2 P 7
Frustum 3 0.0 0.0 -7.07 -3.76 0.37 1.65 0.43 1.91 0.5 2.21 C C C ! Lower Plug
Cylinder 3 0.0 0.0 -3.76 0.0 1.65 1.91 2.21 C C C ! Lower Plug
Cylinder 3 0.0 0.0 0.0 342.9 1.65 1.91 2.21 F G C ! Fuel
Cylinder 3 0.0 0.0 342.90 355.6 1.65 1.91 2.21 G G C ! Bond Sodium
Cylinder 3 0.0 0.0 355.60 590.43 1.65 1.91 2.21 FG FG C ! Fission Gas Plenum
Cylinder 3 0.0 0.0 590.43 593.09 1.65 1.91 2.21 C C C ! Upper Plug
CellMaterial -7.07 -3.76 NA &
-3.76 0.0 NA &
0.0 342.9 NA &
342.90 355.6 NA &
355.6 590.43 NA &
590.43 593.09 NA
###Creating 3 cylinders when one needed; this simplifies sweeping.#####
Flowmeter F 7
Frustum 3 0.0 0.0 -7.07 -3.76 0.37 1.65 0.43 1.91 0.5 2.21 C C C ! Lower Plug
Cylinder 3 0.0 0.0 -3.76 0.0 1.65 1.91 2.21 G G C ! Lower Plug
Cylinder 3 0.0 0.0 0.0 342.9 1.65 1.91 2.21 G G C ! Fuel
Cylinder 3 0.0 0.0 342.0 355.6 1.65 1.91 2.21 G G C ! Bond Sodium
Cylinder 3 0.0 0.0 355.6 590.43 1.65 1.91 2.21 FG FG C ! Fission Gas Plenum
Cylinder 3 0.0 0.0 590.43 593.09 1.65 1.91 2.21 C C C ! Upper Plug
CellMaterial -7.07 -3.76 NA &
-3.76 0.0 NA &
0.0 342.9 NA &
342.90 355.6 NA &
355.60 590.43 NA &
590.43 593.09 NA
#####
Assembly 5
P P P P P
P P P P P P
P P P P P P P
P P P P P P P P
P P P P P P P P P
P P P P P P P P P
P P P P P P
F P P P P P
F P P P P P
#####
Center
End

```

2. Lower Shield AssyGen Input File.

```

#####
! AssyGen Input File: Lower Shield: XX09 assembly used in EBR II STRT-17 assembly

```

```

! All dimensions are in mm; Note: The final model is scaled to cm
! Position this assembly such that upper assembly and this lower shield form the complete assembly.
!#####
Materials 2 Sodium NA StSteel SS
!#####
Duct 4 0.0 0.0 -520.7 -276.1 46.41 48.44 56.13 58.17 SS SS NA SS !Lower Assembly
Duct 4 0.0 0.0 -276.1 -42.75 46.41 48.44 56.13 58.17 SS SS NA SS !Lower Assembly
Duct 4 0.0 0.0 -42.75 -35.75 46.41 48.44 56.13 58.17 NA SS NA SS !Loft for meshing
Duct 4 0.0 0.0 -35.75 -7.07 46.41 48.44 56.13 58.17 NA SS NA SS !Lower Assembly
Pincells 1 5.67
Diffuser D 2
Frustum 1 0.0 0.0 -276.1 -42.75 7.94 21.425 NA ! length 233.35
Cylinder 1 0.0 0.0 -520.7 -276.1 7.94 NA ! length 244.6
Assembly 1
D
!#####
Center
End

```

B Sample coupled physics driver using COUPE

Listing 1: UNIC-Nek driver

```

1  #include "coupe.h"
2
3  // Usage: ./driver <mesh_filename_unic> <mesh_filename_nek>
4  int main(int argc, char **argv)
5  {
6      PetscErrorCode ierr;
7      iMesh_Instance mesh;
8      Physics unic, nek;
9      Coupe coupe;
10
11     // instantiate an iMesh object, create file-sets, load the meshes and ↵
12     // partition
13     iMesh_newMesh("", &mesh, &ierr, 0);CHKERRQ(ierr);
14     ....
15
16     // Create the COUPE object that holds the mapping and collection to all ↵
17     // physics
18     ierr = CoupeCreate(PETSC_COMM_WORLD, 2, &coupe);CHKERRQ(ierr);
19     ierr = CoupeSetFromOptions(coupe);CHKERRQ(ierr);
20
21     // Create the UNIC physics wrapper object
22     ierr = PhysicsCreateUNIC(PETSC_COMM_WORLD, &unic);CHKERRQ(ierr);
23     ierr = PhysicsSetFromOptions(unic);CHKERRQ(ierr);
24
25     // Set the iMesh reference to the physics so that the MOAB mesh can be used
26     // UNIC_ID]);CHKERRQ(ierr);
27     ierr = CoupeSetPhysics(coupe, UNIC_ID, &unic);CHKERRQ(ierr);
28
29     // Create the NEK physics wrapper object
30     ierr = PhysicsCreateNEK(PETSC_COMM_WORLD, &nek);CHKERRQ(ierr);
31     ierr = PhysicsSetFromOptions(nek);CHKERRQ(ierr);
32
33     // Set the iMesh reference to the physics so that the MOAB mesh can be used

```

```

33   ierr = PhysicsSetMeshReference(nek, &mesh, &phandle[NEK_ID], &fsets[NEK_ID←
    ]);CHKERRQ(ierr);
34   ierr = CoupeSetPhysics(coupe, NEK_ID, &nek);CHKERRQ(ierr);
35
36   // SetUp routines
37   ierr = CoupeSetUp(coupe);CHKERRQ(ierr);
38
39   // Solve routines
40   ierr = CoupeSolve(coupe);CHKERRQ(ierr);
41
42   // Output routines
43   ierr = CoupeOutputData(coupe);CHKERRQ(ierr);
44
45   // Logging and routines providing internal details
46   ierr = CoupeView(coupe, 0);CHKERRQ(ierr);
47
48   // Finalization routines
49   ierr = PhysicsDestroy(&unic);CHKERRQ(ierr);
50   ierr = PhysicsDestroy(&nek);CHKERRQ(ierr);
51   ierr = CoupeDestroy(&coupe);CHKERRQ(ierr);
52 }

```

C Coupled Nek5000/UNIC Model Output for SAHEX1

```

[COUPELOOSE] Creating the necessary data structures.
[COUPELOOSE] Setting options from command line.
[PHY-UNIC] Creating the necessary data structures.
[PHY-UNIC] Getting all command line/configuration options.
[PHY-UNIC] Setting the iMesh_Instance reference.
[PHY-NEK] Creating the necessary data structures.
[PHY-NEK] The session name is sahex1.
[PHY-NEK] Getting all command line/configuration options.
[PHY-NEK] Setting the iMesh_Instance reference.
[COUPELOOSE] Setting up the necessary structures before coupled solve.
[PHY-UNIC] Setting up necessary data.
[PHY-NEK] Setting up necessary data.
.....
.....000000000000.....000.....00.....00000.....000.....00.....00000000000.....
.....000000000000.....00000.....00.....00000000.....00000.....00.....000000000000.....
.....000.....00.....00.....00.....000.....00.....00.....00.....00.....00.....
.....000.....00.....00.....00.....000.....00.....00.....00.....00.....00.....
.....000000000000.....00.....00.....000.....000.....00.....00.....00.....00.....
.....000000000000.....00.....00.....00.....000.....00.....00.....00.....00.....
.....000.....00.....00.....00.....000.....00.....00.....00.....00.....00.....
.....000.....00.....00.....00.....000.....00.....00.....00.....00.....00.....
.....000000000000.....00.....0000.....000000000000.....00.....0000.....000000000000.....
.....000000000000.....00.....000.....000000000000.....00.....000.....000000000000.....
.....
.....Argonne National Laboratory Neutronics Package (UNIC).....
.....A steady-state solver of the multigroup neutron transport equation.....
.....Geometry inputs include Cartesian 1-D, 2-D, 3-D unstructured FE meshes.....
.....
.....Primary code authors   Current Institution   Contact Email.....
.....-----
.....Dinesh Kaushik         KAUST/ANL         Dinesh.Kaushik@KAUST.EDU.SA.....
.....Yunzhao Li             Argonne National Lab   yli@anl.gov.....
.....Abel Marin-Lafleche    Argonne National Lab   amarinlafleche@anl.gov.....
.....Cristian Rabiti        Idaho National Lab     masmith@anl.gov.....
.....Micheal A. Smith       Argonne National Lab   masmith@anl.gov.....
.....Allan Wollaber         Los Alamos National Lab.....
.....
.....Component Library      Code contributors and Responsible Parties.....
.....-----
.....PN2ND                 Micheal A. Smith, Dinesh Kaushik.....
.....SN2ND                 Micheal A. Smith & Allan Wollaber, Dinesh Kaushik.....
.....MOCFE                 Micheal A. Smith, Abel Marin-Lafleche, Cristian Rabiti.....
.....NODAL                 Micheal A. Smith, Yunzhao Li, Elmer Lewis.....
.....
[SN2ND]...Successfully imported the UNIC input (all errors reported above).....
[SN2ND]...The 10000000 - spatial vertices - are chopped into 1 processor communication segments.....
[SN2ND]...The 6 - angular moments - are chopped into 1 processor communication segments.....
[SN2ND]...The 4 - energy groups - are chopped into 1 processor communication segments.....
[SN2ND]...Successfully imported the ASSIGNMENT input (all errors reported above).....

```



```

timer accuracy: 1.2159348E-06 sec

read .rea file

mapping elements to processors
RANK 0 IEG 1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88
89 90 91 92 93 94 95 96
97 98 99 100 101 102 103 104
105 106 107 108 109 110 111 112
113 114 115 116 117 118 119 120
121 122 123 124 125 126 127 128
129 130 131 132 133 134 135 136
137 138 139 140 141 142 143 144
145 146 147 148 149 150 151 152
153 154 155 156 157 158 159 160
161 162 163 164 165 166 167 168
169 170 171 172 173 174 175 176
177 178 179 180 181 182 183 184
185 186 187 188 189 190 191 192
193 194 195 196 197 198 199 200
201 202 203 204 205 206 207 208
209 210 211 212 213 214 215 216
217 218 219 220 221 222 223 224
225 226 227 228 229 230 231 232
233 234 235 236 237 238 239 240
241 242 243 244 245 246 247 248
249 250 251 252 253 254 255 256
257 258 259 260 261 262 263 264
265 266 267 268 269 270 271 272
273 274 275 276 277 278 279 280
281 282 283 284 285 286 287 288
289 290 291 292 293 294 295 296
297 298 299 300 301 302 303 304
305 306 307 308 309 310 311 312
313 314 315 316 317 318 319 320
321 322 323 324 325 326 327 328
329 330 331 332 333 334 335 336
337 338 339 340 341 342 343 344
345 346 347 348 349 350 351 352
353 354 355 356 357 358 359 360
361 362 363 364 365 366 367 368
369 370 371 372 373 374 375 376
377 378 379 380 381 382 383 384

```

385	386	387	388	389	390	391	392
393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408
409	410	411	412	413	414	415	416
417	418	419	420	421	422	423	424
425	426	427	428	429	430	431	432
433	434	435	436	437	438	439	440
441	442	443	444	445	446	447	448
449	450	451	452	453	454	455	456
457	458	459	460	461	462	463	464
465	466	467	468	469	470	471	472
473	474	475	476	477	478	479	480
481	482	483	484	485	486	487	488
489	490	491	492	493	494	495	496
497	498	499	500	501	502	503	504
505	506	507	508	509	510	511	512
513	514	515	516	517	518	519	520
521	522	523	524	525	526	527	528
529	530	531	532	533	534	535	536
537	538	539	540	541	542	543	544
545	546	547	548	549	550	551	552
553	554	555	556	557	558	559	560
561	562	563	564	565	566	567	568
569	570	571	572	573	574	575	576
577	578	579	580	581	582	583	584
585	586	587	588	589	590	591	592
593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608
609	610	611	612	613	614	615	616
617	618	619	620	621	622	623	624
625	626	627	628	629	630	631	632
633	634	635	636	637	638	639	640
641	642	643	644	645	646	647	648
649	650	651	652	653	654	655	656
657	658	659	660	661	662	663	664
665	666	667	668	669	670	671	672
673	674	675	676	677	678	679	680
681	682	683	684	685	686	687	688
689	690	691	692	693	694	695	696
697	698	699	700	701	702	703	704
705	706	707	708	709	710	711	712
713	714	715	716	717	718	719	720
721	722	723	724	725	726	727	728
729	730	731	732	733	734	735	736
737	738	739	740	741	742	743	744
745	746	747	748	749	750	751	752
753	754	755	756	757	758	759	760
761	762	763	764	765	766	767	768
769	770	771	772	773	774	775	776
777	778	779	780	781	782	783	784
785	786	787	788	789	790	791	792
793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808
809	810	811	812	813	814	815	816
817	818	819	820	821	822	823	824
825	826	827	828	829	830	831	832
833	834	835	836	837	838	839	840
841	842	843	844	845	846	847	848
849	850	851	852	853	854	855	856
857	858	859	860	861	862	863	864
865	866	867	868	869	870	871	872
873	874	875	876	877	878	879	880
881	882	883	884	885	886	887	888
889	890	891	892	893	894	895	896
897	898	899	900	901	902	903	904
905	906	907	908	909	910	911	912
913	914	915	916	917	918	919	920
921	922	923	924	925	926	927	928
gs_setup: 0 unique labels shared							
gs_setup: 0 unique labels shared							
gs_setup: 0 unique labels shared							
929	930	931	932	933	934	935	936
937	938	939	940	941	942	943	944
945	946	947	948	949	950	951	952
953	954	955	956	957	958	959	960
961	962	963	964	965	966	967	968
969	970	971	972	973	974	975	976
977	978	979	980	981	982	983	984
985	986	987	988	989	990	991	992
993	994	995	996	997	998	999	1000
1001	1002	1003	1004	1005	1006	1007	1008
1009	1010	1011	1012	1013	1014	1015	1016
1017	1018	1019	1020	1021	1022	1023	1024
1025	1026	1027	1028	1029	1030	1031	1032
1033	1034	1035	1036	1037	1038	1039	1040
1041	1042	1043	1044	1045	1046	1047	1048
1049	1050	1051	1052	1053	1054	1055	1056
1057	1058	1059	1060	1061	1062	1063	1064
1065	1066	1067	1068	1069	1070	1071	1072
1073	1074	1075	1076	1077	1078	1079	1080
1081	1082	1083	1084	1085	1086	1087	1088
1089	1090	1091	1092	1093	1094	1095	1096

```

1097 1098 1099 1100 1101 1102 1103 1104
1105 1106 1107 1108 1109 1110 1111 1112
1113 1114 1115 1116 1117 1118 1119 1120
1121 1122 1123 1124 1125 1126 1127 1128
1129 1130 1131 1132 1133 1134 1135 1136
1137 1138 1139 1140 1141 1142 1143 1144
1145 1146 1147 1148 1149 1150 1151 1152
1153 1154 1155 1156 1157 1158 1159 1160
1161 1162 1163 1164 1165 1166 1167 1168
1169 1170 1171 1172 1173 1174 1175 1176
1177 1178 1179 1180 1181 1182 1183 1184
1185 1186 1187 1188 1189 1190 1191 1192
1193 1194 1195 1196 1197 1198 1199 1200
1201 1202 1203 1204 1205 1206 1207 1208
1209 1210 1211 1212 1213 1214 1215 1216
1217 1218 1219 1220 1221 1222 1223 1224
1225 1226 1227 1228 1229 1230 1231 1232
1233 1234 1235 1236 1237 1238 1239 1240
1241 1242 1243 1244 1245 1246 1247 1248
1249 1250 1251 1252 1253 1254 1255 1256
1257 1258 1259 1260 1261 1262 1263 1264
1265 1266 1267 1268 1269 1270 1271 1272
1273 1274 1275 1276 1277 1278 1279 1280
1281 1282 1283 1284 1285 1286 1287 1288
1289 1290 1291 1292 1293 1294 1295 1296
1297 1298 1299 1300 1301 1302 1303 1304
1305 1306 1307 1308 1309 1310 1311 1312
1313 1314 1315 1316 1317 1318 1319 1320
1321 1322 1323 1324 1325 1326 1327 1328
1329 1330 1331 1332 1333 1334 1335 1336
1337 1338 1339 1340 1341 1342 1343 1344
1345 1346 1347
element load imbalance: 0 1347 1347
done :: mapping elements to processors

Setid, numids = 1 144
Setid, numids = 2 0
Setid, numids = 3 0
Setid, numids = 4 114
Setid, numids = 4 114
Setid, numids = 5 335
0 objects found
done :: read .rea file 0.58955 sec

setup mesh topology
Right-handed check complete for 1347 elements. OK.
setvert3d: 4 20191 30967 20191 20191
call usrsetvert
done :: usrsetvert

setups time 9.5301E-02 seconds 0 4 20191 342
setvert3d: 4 39885 50661 39885 39885
call usrsetvert
done :: usrsetvert

setups time 3.9099E-01 seconds 1 4 39885 1347
10 max multiplicity
done :: setup mesh topology

call usrdat
done :: usrdat

generate geomerty data
vol_t,vol_v: 433.01270189216297 152.10996621122300
done :: generate geomerty data

call usrdat2
done :: usrdat2

regenerate geomerty data 1
vol_t,vol_v: 433.01270189216297 152.10996621122300
done :: regenerate geomerty data 1

verify mesh topology
10.226497308103744 21.773502691896255 Xrange
2.4019237886466822 12.401923788646689 Yrange
10.000000000000000 15.000000000000000 Zrange
done :: verify mesh topology

103 Parameters from file:/home/tautges/code/sharp/src/tests/sahex1/sahex1_nek.rea
1 1.00000 p1 DENSITY
2 0.50000 p2 VISCOS
7 1.00000 p7 RHOCp
8 0.50000 p8 CONDUCT
11 2 2 p11 NSTEPS
12 .05000 p12 DT
15 100 0. p15 IOSTEP
16 1.00000 p16 PSSOLVER
18 0.250000E-01 p18 GRID
19 -1.00000 p19 INTYPE
20 10.0000 p20 NORDER
21 1.00000E-06 p21 DIVERGENCE

```

```

22      1.000000E-08      p22 HELMHOLTZ
24      1.000000E-05      p24 TOLREL
25      1.000000E-09      p25 TOLABS
26      0.50000          p26 COURANT/NTAU
27      2.00000          p27 TORDER
28      0.                p28 TORDER: mesh velocity (0: p28=p27)
30      1.00000          p30 > 0 ==> properties set in uservp()
43      1.00000          p43 0=semg/1=schwarz
66      4.00000          p66 output : <0=ascii, else binary
67      4.00000          p67 restart: <0=ascii, else binary
99      3.00000          p99 dealiasing: <0--> off/3--> old/4--> new
102     5.00000          p102 Dump out divergence at each time step
103     .01              p103 weight of stabilizing filter (.01)

IFTRAN   = T
IFFLOW   = T
IFHEAT   = T
IFSPLIT  = T
IFLOMACH = F
IFUSERVP = T
IFUSERMV = F
IFSTRS   = F
IFCHAR   = F
IFCYCLIC = F
IFAXIS   = F
IFMVD    = F
IFMELT   = F
IFMODEL  = F
IFKEPS   = F
IFMOAB   = T
IFNEKNEK = F
IFSYNC   = T

IFVCOR   = T
IFINTQ   = F
IFCWUZ   = F
IFSWALL  = F
IFGEOM   = F
IFSURT   = F
IFWCNO   = F

IFTMSH for field      1      = F
IFADVC for field      1      = T
IFNONL for field      1      = F

IFTMSH for field      2      = T
IFADVC for field      2      = T
IFNONL for field      2      = F

Dealiasing enabled, lxd=      6

Estimated eigenvalues
EIGAA = 0.22155205712862044
EIGGA = 4430.6538742190933
EIGAE = 0.11565942657526591
EIGAS = 4.35034802784222893E-002
EIGGE = 4430.6538742190933
EIGGS = 2.0000000000000000

verify mesh topology
10.226497308103744      21.773502691896255      Xrange
2.4019237886466822      12.401923788646689      Yrange
10.000000000000000      15.000000000000000      Zrange
done :: verify mesh topology

E-solver strategy: 0 itr
setup h1 coarse grid, nx_crs=      2
call usrsetvert
done :: usrsetvert

Proc 0: box min/max, tree depth = (10.2265,2.40192,10), (21.7735,12.4019,15), 23
Proc 0: box min/max, tree depth = (10.2265,2.40192,10), (21.7735,12.4019,15), 22
[COUPEL00SE] ** Starting the coupled physics solve **
[PHY-UNIC] *** SOLVING THE PHYSICS ***
done :: setup h1 coarse grid      1.1665511131286621      sec

call usrdat3
done :: usrdat3

set initial conditions
nekuic (1) for ifld      1
nekuic (1) for ifld      2
call nekuic for ifld      2
call nekuic for vel
xyz min      10.226      2.4019      10.000
uvwpt min -0.37737E-19      -1691.5      -0.37737E-19      -0.46038E-19      -0.45711E-19
xyz max      21.774      12.402      15.000
uvwpt max 0.97093E-19      -509.13      0.97093E-19      0.88792E-19      0.13771E-18
done :: set initial conditions

call userchk
schfile:/home/tautges/code/sharp/src/tests/sahex1/sahex1_nek.sch

```

```

call outfld: ifpsco: F

0 0.0000E+00 Write checkpoint:
0 0 OPEN: sahex1_nek0.f00001

0 0.0000E+00 done :: Write checkpoint
file size = 2.7 MB
avg data-throughput = 40.8MB/s
io-nodes = 1

0 0.00000E+00 1.00000E+00 1.00000E+00 tmax
Wrote MOAB H5M file for step 0
done :: userchk

gridpoints unique/tot: 12210 21888
dofs: 10989 12210

Initialization successfully completed 13.858 sec
[SN2ND]...Neutron transport cross section data was imported and distributed.....
[SN2ND]...Obtaining the preconditioner matrices for the second order discrete ordinates method.....
[SN2ND]...Primary control variables used for execution job size..... 1
[SN2ND]...Debug print setting for setup..... 0
[SN2ND]...Debug print setting for A formation..... 0
[SN2ND]...Debug print setting for iteration history..... 0
[SN2ND]...Global number of energy groups..... 4
[SN2ND]...Global number of spatial mesh points..... 6720
[SN2ND]... Corresponding number of finite elements..... 5310
[SN2ND]... Number of spatial dimensions..... 3
[SN2ND]...Cubature to use for the angular domain..... LEG-TCHEBY
[SN2ND]...Global angular expansion order..... 2
[SN2ND]... Number of even-parity directions..... 6
[SN2ND]... Expansion order of the scattering kernel..... 0
[SN2ND]... Even-parity spherical harmonics..... 1
[SN2ND]... Odd-parity spherical harmonics..... 0
[SN2ND]...Global space-angle degrees of freedom..... 40320
[SN2ND]...Source type information.....
[SN2ND]... Is a fission source present..... Yes
[SN2ND]... Is a fixed source present..... No
[SN2ND]... Is a up/down scattering present..... Yes
[SN2ND]...Outer iteration control variables.....
[SN2ND]... Iterative improvement scaling factor..... 3.330E-01
[SN2ND]... Use Tchebychev acceleration..... Yes
[SN2ND]... Targeted error on the eigenvalue..... 1.000E-06
[SN2ND]... Targeted error on the fission source..... 5.000E-06
[SN2ND]... Maximum number of iterations..... 2
[SN2ND]... Maximum upscatter iterations..... 10
[SN2ND]...Scattering iteration control variables.....
[SN2ND]... Use synthetic diffusion acceleration..... Yes
[SN2ND]... Maximum number of scattering iterations..... 1000
[SN2ND]... Targeted error on the flux solution..... 5.000E-07
[SN2ND]... Maximum number of Krylov iterations..... 5000
[SN2ND]... Use p-Multigrid preconditioning..... No
[SN2ND]... p-Multigrid preconditioning "order"..... 0
[SN2ND]...Break down of total job memory usage (MB)..... PEAK TOTAL
[SN2ND]... Memory usage for spatial mesh..... 0.444241 0.444241
[SN2ND]... Memory usage for spatial matrices..... 14.726120 14.726120
[SN2ND]... Vector memory usage..... 0.179707 0.179707
[SN2ND]... Memory usage for A matrices..... 18.016975 18.016975
[SN2ND]... Memory usage for PC matrices..... 1.435547 1.435547
[SN2ND]... Memory usage for scattering source..... 1.296387 1.296387
[SN2ND]... Memory usage for fission source..... 0.747070 0.747070
[SN2ND]... Scratch memory for power iteration..... 2.460937 2.460937
[SN2ND]... Total tracked memory usage.(overhead?)..... 39.306984 39.306984
[SN2ND]... Total reported memory usage..... 2.460937 2.460937
[SN2ND]...Solving the preceding Sn system of equations.....
=SN2ND Fission Source Iteration
=SN2ND Seconds|Itr|Up| Eigenvalue | Error | Fis Err| Flux Err| Dom | Slope| Error |Active| Dom |Scat| CG |Max/G|Group|Assoc.Error| Total | Max/G |Group| Max/A |
=SN2ND 55.9|001|00| 2.78134E-02| 9.8E-01| 9.8E-01|T 2.0E+01| 0.000| 0.000| 4.2E-01| T 0| 0.984| 4| 30| 7| 4| 7.269E+04| 798| 60| 1| 10|
=SN2ND 91.0|002|00| 5.40475E-02| 9.4E-01| 1.0E+00|T 3.1E+01| 0.199| 0.000| 7.7E-02| T 0| 0.026| 7| 48| 14| 4| 2.693E+03| 1290| 53| 1| 9|
=SN2ND 90.6|003|00| 6.58053E-02| 2.2E-01| 2.5E-01|T 3.4E-01| 0.892| 0.000| 2.1E-02| T 0| 0.396| 8| 46| 14| 4| 1.247E+02| 1292| 60| 1| 10|
[SN2ND]...Reached specified iteration limit..... 2
[SN2ND]...!!!!FAILURE!!! Could not converge within the specified criteria.....
[SN2ND]...Sorry, but I must stop
[COUPE] Interpolating solution for physics: nek; FIELD = VPOWER
point location: wanted 12915 got 12915 locally, 0 remote, missing 0
[COUPE] Interpolation done. Printing first 5 values.
3.67414242 3.77450593 4.39777620 5.12140998 5.26070712
[PHY-NEK] *** SOLVING THE PHYSICS ***
[SN2ND]...Final eigenvalue..... 0.06580526
[SN2ND]...Final error on the eigenvalue..... 0.21754448
[SN2ND]...Final iterative error on the flux solution..... 0.34085233
[SN2ND]...Total number of fission source iterations..... 3
[SN2ND]...Total number of upscatter iterations..... 0
[SN2ND]...Total number of scattering source iterations..... 19
[SN2ND]...Within Grp Flx auto adjustment ranges: 1.000E+00 to 1.000E+00 and iteration count of..... 124
[SN2ND]...Preconditioner auto adjustment ranges: 1.000E+00 to 1.000E+00 and iteration count of..... 3380
[SN2ND]...Total number of synthetic diffusion iterations..... 221
[SN2ND]...Total computational time required for real solve (seconds)..... 237.709991

Starting time loop ...

DT/DT CFL/DTFS/DTINIT 0.220E-04 0.220E-04 0.000E+00 0.500E-01

```

```

Step      1, t= 2.2011011E-05, DT= 2.2011011E-05, C= 0.500 0.0000E+00 0.0000E+00
      Solving for heat
      Temperature/Passive scalar solution
      1.000000000000000002E-008 p22      1      2
New CG1-tolerance (RINIT*epsm) = 7.84969262457332813E-014 3.24099544834337811E-026
1 1 Helmholtz TEMP F: 7.8497E-01 1.0000E-08 5.0000E-02 4.5432E+04
1 2 Helmholtz TEMP F: 3.3138E-04 1.0000E-08 5.0000E-02 4.5432E+04
1 3 Helmholtz TEMP F: 2.3310E-07 1.0000E-08 5.0000E-02 4.5432E+04
1 4 Helmholtz TEMP F: 1.6367E-10 1.0000E-08 5.0000E-02 4.5432E+04
1 Hmholtz TEMP: 3 1.6367E-10 7.8497E-01 1.0000E-08
1 2.2011E-05 3.4836E+00 Heat done
      Solving for fluid
      1.000000000000000002E-008 p22      1      1
New CG1-tolerance (RINIT*epsm) = 3.06820412066330581E-005 9.9999999999999955E-007
New CG1-tolerance (Neumann) = 3.06820412066330568E-004
New CG1-tolerance (RINIT*epsm) = 3.06820412066330581E-005 9.9999999999999955E-007
New CG1-tolerance (Neumann) = 3.06820412066330568E-004
1 1.00000E-06 6.93041E+06 1.91715E+07 3.61496E-01 1 Divergence
2 1.00000E-06 4.06539E+06 1.91715E+07 2.12054E-01 1 Divergence
3 1.00000E-06 2.79430E+06 1.91715E+07 1.45753E-01 1 Divergence
4 1.00000E-06 1.92550E+06 1.91715E+07 1.00436E-01 1 Divergence
5 1.00000E-06 1.20566E+06 1.91715E+07 6.28882E-02 1 Divergence
6 1.00000E-06 7.49819E+05 1.91715E+07 3.91111E-02 1 Divergence
7 1.00000E-06 5.07407E+05 1.91715E+07 2.64668E-02 1 Divergence
8 1.00000E-06 3.48603E+05 1.91715E+07 1.81834E-02 1 Divergence
9 1.00000E-06 2.37815E+05 1.91715E+07 1.24046E-02 1 Divergence
10 1.00000E-06 1.79984E+05 1.91715E+07 9.38811E-03 1 Divergence
11 1.00000E-06 1.32672E+05 1.91715E+07 6.92027E-03 1 Divergence
12 1.00000E-06 9.81657E+04 1.91715E+07 5.12040E-03 1 Divergence
13 1.00000E-06 7.71050E+04 1.91715E+07 4.02186E-03 1 Divergence
14 1.00000E-06 6.04029E+04 1.91715E+07 3.15066E-03 1 Divergence
15 1.00000E-06 4.63794E+04 1.91715E+07 2.41919E-03 1 Divergence
16 1.00000E-06 3.65093E+04 1.91715E+07 1.90436E-03 1 Divergence
17 1.00000E-06 2.90788E+04 1.91715E+07 1.51677E-03 1 Divergence
18 1.00000E-06 2.20165E+04 1.91715E+07 1.14840E-03 1 Divergence
19 1.00000E-06 1.60267E+04 1.91715E+07 8.35967E-04 1 Divergence
20 1.00000E-06 1.17832E+04 1.91715E+07 6.14621E-04 1 Divergence
21 1.00000E-06 8.55814E+03 1.91715E+07 4.46399E-04 1 Divergence
22 1.00000E-06 6.36066E+03 1.91715E+07 3.31777E-04 1 Divergence
23 1.00000E-06 4.92076E+03 1.91715E+07 2.56671E-04 1 Divergence
24 1.00000E-06 3.95214E+03 1.91715E+07 2.06147E-04 1 Divergence
25 1.00000E-06 3.22892E+03 1.91715E+07 1.68423E-04 1 Divergence
26 1.00000E-06 2.73266E+03 1.91715E+07 1.42538E-04 1 Divergence
27 1.00000E-06 2.34053E+03 1.91715E+07 1.22084E-04 1 Divergence
28 1.00000E-06 1.93707E+03 1.91715E+07 1.01039E-04 1 Divergence
29 1.00000E-06 1.65538E+03 1.91715E+07 8.63459E-05 1 Divergence
30 1.00000E-06 1.43947E+03 1.91715E+07 7.50839E-05 1 Divergence
31 1.00000E-06 1.23106E+03 1.91715E+07 6.42131E-05 1 Divergence
32 1.00000E-06 1.07654E+03 1.91715E+07 5.61532E-05 1 Divergence
33 1.00000E-06 9.55489E+02 1.91715E+07 4.98390E-05 1 Divergence
34 1.00000E-06 8.50275E+02 1.91715E+07 4.43510E-05 1 Divergence
35 1.00000E-06 7.42629E+02 1.91715E+07 3.87361E-05 1 Divergence
36 1.00000E-06 6.32187E+02 1.91715E+07 3.29754E-05 1 Divergence
37 1.00000E-06 5.40327E+02 1.91715E+07 2.81839E-05 1 Divergence
38 1.00000E-06 4.54492E+02 1.91715E+07 2.37066E-05 1 Divergence
39 1.00000E-06 3.83271E+02 1.91715E+07 1.99917E-05 1 Divergence
40 1.00000E-06 3.20619E+02 1.91715E+07 1.67237E-05 1 Divergence
41 1.00000E-06 2.91034E+02 1.91715E+07 1.51806E-05 1 Divergence
42 1.00000E-06 2.59913E+02 1.91715E+07 1.35572E-05 1 Divergence
43 1.00000E-06 2.28002E+02 1.91715E+07 1.18927E-05 1 Divergence
44 1.00000E-06 2.00688E+02 1.91715E+07 1.04680E-05 1 Divergence
45 1.00000E-06 1.74901E+02 1.91715E+07 9.12296E-06 1 Divergence
46 1.00000E-06 1.53262E+02 1.91715E+07 7.99424E-06 1 Divergence
47 1.00000E-06 1.33849E+02 1.91715E+07 6.98168E-06 1 Divergence
48 1.00000E-06 1.16340E+02 1.91715E+07 6.06839E-06 1 Divergence
49 1.00000E-06 1.01332E+02 1.91715E+07 5.28555E-06 1 Divergence
50 1.00000E-06 8.74766E+01 1.91715E+07 4.56285E-06 1 Divergence
51 1.00000E-06 7.50804E+01 1.91715E+07 3.91625E-06 1 Divergence
52 1.00000E-06 6.35830E+01 1.91715E+07 3.31654E-06 1 Divergence
53 1.00000E-06 5.42876E+01 1.91715E+07 2.83168E-06 1 Divergence
54 1.00000E-06 4.69653E+01 1.91715E+07 2.44975E-06 1 Divergence
55 1.00000E-06 3.90392E+01 1.91715E+07 2.03631E-06 1 Divergence
56 1.00000E-06 3.22886E+01 1.91715E+07 1.68420E-06 1 Divergence
57 1.00000E-06 2.68745E+01 1.91715E+07 1.40179E-06 1 Divergence
58 1.00000E-06 2.16350E+01 1.91715E+07 1.12850E-06 1 Divergence
59 1.00000E-06 1.77610E+01 1.91715E+07 9.26426E-07 1 Divergence
60 1.00000E-06 1.44543E+01 1.91715E+07 7.53947E-07 1 Divergence
61 1.00000E-06 1.17434E+01 1.91715E+07 6.12542E-07 1 Divergence
62 1.00000E-06 9.62884E+00 1.91715E+07 5.02248E-07 1 Divergence
63 1.00000E-06 7.87910E+00 1.91715E+07 4.10980E-07 1 Divergence
64 1.00000E-06 6.33246E+00 1.91715E+07 3.30306E-07 1 Divergence
65 1.00000E-06 5.00884E+00 1.91715E+07 2.61265E-07 1 Divergence
66 1.00000E-06 4.17120E+00 1.91715E+07 2.17573E-07 1 Divergence
67 1.00000E-06 3.59839E+00 1.91715E+07 1.87695E-07 1 Divergence
68 1.00000E-06 3.05369E+00 1.91715E+07 1.59283E-07 1 Divergence
69 1.00000E-06 2.49465E+00 1.91715E+07 1.30123E-07 1 Divergence
70 1.00000E-06 2.02021E+00 1.91715E+07 1.05376E-07 1 Divergence
71 1.00000E-06 1.71783E+00 1.91715E+07 8.96035E-08 1 Divergence
72 1.00000E-06 1.45498E+00 1.91715E+07 7.58931E-08 1 Divergence
73 1.00000E-06 1.21138E+00 1.91715E+07 6.31863E-08 1 Divergence
74 1.00000E-06 1.02964E+00 1.91715E+07 5.37067E-08 1 Divergence
75 1.00000E-06 9.04389E-01 1.91715E+07 4.71736E-08 1 Divergence

```

```

76 1.00000E-06 7.88936E-01 1.91715E+07 4.11515E-08 1 Divergence
77 1.00000E-06 6.59088E-01 1.91715E+07 3.43785E-08 1 Divergence
78 1.00000E-06 5.52811E-01 1.91715E+07 2.88350E-08 1 Divergence
79 1.00000E-06 4.66534E-01 1.91715E+07 2.43348E-08 1 Divergence
80 1.00000E-06 3.76241E-01 1.91715E+07 1.96250E-08 1 Divergence
81 1.00000E-06 3.31679E-01 1.91715E+07 1.73006E-08 1 Divergence
82 1.00000E-06 2.95232E-01 1.91715E+07 1.53995E-08 1 Divergence
83 1.00000E-06 2.60507E-01 1.91715E+07 1.35883E-08 1 Divergence
84 1.00000E-06 2.31394E-01 1.91715E+07 1.20697E-08 1 Divergence
85 1.00000E-06 2.04836E-01 1.91715E+07 1.06844E-08 1 Divergence
86 1.00000E-06 1.75947E-01 1.91715E+07 9.17755E-09 1 Divergence
87 1.00000E-06 1.51079E-01 1.91715E+07 7.88041E-09 1 Divergence
88 1.00000E-06 1.27583E-01 1.91715E+07 6.65482E-09 1 Divergence
89 1.00000E-06 1.05714E-01 1.91715E+07 5.51413E-09 1 Divergence
90 1.00000E-06 8.95579E-02 1.91715E+07 4.67141E-09 1 Divergence
91 1.00000E-06 7.65445E-02 1.91715E+07 3.99262E-09 1 Divergence
92 1.00000E-06 6.43628E-02 1.91715E+07 3.35722E-09 1 Divergence
93 1.00000E-06 5.39301E-02 1.91715E+07 2.81304E-09 1 Divergence
94 1.00000E-06 4.49241E-02 1.91715E+07 2.34328E-09 1 Divergence
95 1.00000E-06 3.72420E-02 1.91715E+07 1.94257E-09 1 Divergence
96 1.00000E-06 3.07376E-02 1.91715E+07 1.60330E-09 1 Divergence
97 1.00000E-06 2.56838E-02 1.91715E+07 1.33447E-09 1 Divergence
98 1.00000E-06 2.14250E-02 1.91715E+07 1.11754E-09 1 Divergence
99 1.00000E-06 1.79753E-02 1.91715E+07 9.37607E-10 1 Divergence
100 1.00000E-06 1.48530E-02 1.91715E+07 7.74745E-10 1 Divergence
101 1.00000E-06 1.19109E-02 1.91715E+07 6.21282E-10 1 Divergence
102 1.00000E-06 9.57692E-03 1.91715E+07 4.99540E-10 1 Divergence
103 1.00000E-06 7.86293E-03 1.91715E+07 4.10136E-10 1 Divergence
104 1.00000E-06 6.46300E-03 1.91715E+07 3.37115E-10 1 Divergence
105 1.00000E-06 5.42238E-03 1.91715E+07 2.82836E-10 1 Divergence
106 1.00000E-06 4.65597E-03 1.91715E+07 2.42859E-10 1 Divergence
107 1.00000E-06 3.92986E-03 1.91715E+07 2.04984E-10 1 Divergence
108 1.00000E-06 3.32699E-03 1.91715E+07 1.73538E-10 1 Divergence
109 1.00000E-06 2.82284E-03 1.91715E+07 1.47241E-10 1 Divergence
110 1.00000E-06 2.42377E-03 1.91715E+07 1.26426E-10 1 Divergence
111 1.00000E-06 2.05951E-03 1.91715E+07 1.07426E-10 1 Divergence
112 1.00000E-06 1.76597E-03 1.91715E+07 9.21143E-11 1 Divergence
113 1.00000E-06 1.52293E-03 1.91715E+07 7.94374E-11 1 Divergence
114 1.00000E-06 1.33473E-03 1.91715E+07 6.96208E-11 1 Divergence
115 1.00000E-06 1.14342E-03 1.91715E+07 5.96417E-11 1 Divergence
116 1.00000E-06 9.74555E-04 1.91715E+07 5.08336E-11 1 Divergence
117 1.00000E-06 8.44042E-04 1.91715E+07 4.40259E-11 1 Divergence
118 1.00000E-06 7.41486E-04 1.91715E+07 3.86765E-11 1 Divergence
119 1.00000E-06 6.50177E-04 1.91715E+07 3.39137E-11 1 Divergence
120 1.00000E-06 5.65097E-04 1.91715E+07 2.94759E-11 1 Divergence
121 1.00000E-06 5.07657E-04 1.91715E+07 2.64798E-11 1 Divergence
122 1.00000E-06 4.56609E-04 1.91715E+07 2.38171E-11 1 Divergence
123 1.00000E-06 4.06189E-04 1.91715E+07 2.11871E-11 1 Divergence
124 1.00000E-06 3.62352E-04 1.91715E+07 1.89005E-11 1 Divergence
125 1.00000E-06 3.12868E-04 1.91715E+07 1.63195E-11 1 Divergence
126 1.00000E-06 2.71401E-04 1.91715E+07 1.41565E-11 1 Divergence
127 1.00000E-06 2.34608E-04 1.91715E+07 1.22373E-11 1 Divergence
128 1.00000E-06 2.02426E-04 1.91715E+07 1.05587E-11 1 Divergence
129 1.00000E-06 1.73591E-04 1.91715E+07 9.05466E-12 1 Divergence
130 1.00000E-06 1.48361E-04 1.91715E+07 7.73864E-12 1 Divergence
131 1.00000E-06 1.27652E-04 1.91715E+07 6.65844E-12 1 Divergence
132 1.00000E-06 1.10381E-04 1.91715E+07 5.75758E-12 1 Divergence
133 1.00000E-06 9.53990E-05 1.91715E+07 4.97609E-12 1 Divergence
134 1.00000E-06 8.19524E-05 1.91715E+07 4.27470E-12 1 Divergence
135 1.00000E-06 6.86745E-05 1.91715E+07 3.58211E-12 1 Divergence
136 1.00000E-06 5.75693E-05 1.91715E+07 3.00286E-12 1 Divergence
137 1.00000E-06 4.82731E-05 1.91715E+07 2.51796E-12 1 Divergence
138 1.00000E-06 4.02077E-05 1.91715E+07 2.09727E-12 1 Divergence
139 1.00000E-06 3.35534E-05 1.91715E+07 1.75017E-12 1 Divergence
140 1.00000E-06 2.73196E-05 1.91715E+07 1.42501E-12 1 Divergence
141 1.00000E-06 2.21809E-05 1.91715E+07 1.15697E-12 1 Divergence
142 1.00000E-06 1.86095E-05 1.91715E+07 9.70684E-13 1 Divergence
143 1.00000E-06 1.55168E-05 1.91715E+07 8.09368E-13 1 Divergence
144 1.00000E-06 1.29808E-05 1.91715E+07 6.77088E-13 1 Divergence
145 1.00000E-06 1.04994E-05 1.91715E+07 5.47655E-13 1 Divergence
146 1.00000E-06 8.84851E-06 1.91715E+07 4.61545E-13 1 Divergence
147 1.00000E-06 7.56557E-06 1.91715E+07 3.94626E-13 1 Divergence
148 1.00000E-06 6.47184E-06 1.91715E+07 3.37576E-13 1 Divergence
149 1.00000E-06 5.29157E-06 1.91715E+07 2.76012E-13 1 Divergence
150 1.00000E-06 4.26276E-06 1.91715E+07 2.22349E-13 1 Divergence
151 1.00000E-06 3.56767E-06 1.91715E+07 1.86092E-13 1 Divergence
152 1.00000E-06 2.99323E-06 1.91715E+07 1.56129E-13 1 Divergence
153 1.00000E-06 2.47461E-06 1.91715E+07 1.29078E-13 1 Divergence
154 1.00000E-06 2.04709E-06 1.91715E+07 1.06778E-13 1 Divergence
155 1.00000E-06 1.72175E-06 1.91715E+07 8.98078E-14 1 Divergence
156 1.00000E-06 1.49760E-06 1.91715E+07 7.81162E-14 1 Divergence
157 1.00000E-06 1.30401E-06 1.91715E+07 6.80179E-14 1 Divergence
158 1.00000E-06 1.13350E-06 1.91715E+07 5.91245E-14 1 Divergence
159 1.00000E-06 9.99033E-07 1.91715E+07 5.21103E-14 1 Divergence
1 PRES gmes: 159 9.9903E-07 1.0000E-06 1.9171E+07 4.1639E+00 1.2036E+01 F
1.000000000000000002E-008 p22 1 1
1 1 Helmholtz VELX F: 5.5558E+06 1.0000E-08 5.0000E-01 4.5432E+04
1 2 Helmholtz VELX F: 1.9411E+04 1.0000E-08 5.0000E-01 4.5432E+04
1 3 Helmholtz VELX F: 1.0356E+02 1.0000E-08 5.0000E-01 4.5432E+04
1 4 Helmholtz VELX F: 1.0193E+00 1.0000E-08 5.0000E-01 4.5432E+04
1 5 Helmholtz VELX F: 1.2669E-02 1.0000E-08 5.0000E-01 4.5432E+04
1 6 Helmholtz VELX F: 1.1085E-04 1.0000E-08 5.0000E-01 4.5432E+04

```

```

1 7 Helmholtz VELX F: 9.8258E-07 1.0000E-08 5.0000E-01 4.5432E+04
1 8 Helmholtz VELX F: 9.4710E-09 1.0000E-08 5.0000E-01 4.5432E+04
1 Hmholtz VELX: 7 9.4710E-09 5.5558E+06 1.0000E-08
1.000000000000000000000002E-008 p22 1 1
1 1 Helmholtz VELY F: 4.7168E+07 1.0000E-08 5.0000E-01 4.5432E+04
1 2 Helmholtz VELY F: 1.1216E+05 1.0000E-08 5.0000E-01 4.5432E+04
1 3 Helmholtz VELY F: 6.2477E+02 1.0000E-08 5.0000E-01 4.5432E+04
1 4 Helmholtz VELY F: 6.0792E+00 1.0000E-08 5.0000E-01 4.5432E+04
1 5 Helmholtz VELY F: 7.9255E-02 1.0000E-08 5.0000E-01 4.5432E+04
1 6 Helmholtz VELY F: 6.2011E-04 1.0000E-08 5.0000E-01 4.5432E+04
1 7 Helmholtz VELY F: 5.4986E-06 1.0000E-08 5.0000E-01 4.5432E+04
1 8 Helmholtz VELY F: 5.1768E-08 1.0000E-08 5.0000E-01 4.5432E+04
1 9 Helmholtz VELY F: 5.2419E-10 1.0000E-08 5.0000E-01 4.5432E+04
1 Hmholtz VELY: 8 5.2419E-10 4.7168E+07 1.0000E-08
1.000000000000000000000002E-008 p22 1 1
1 1 Helmholtz VELZ F: 3.9457E+00 1.0000E-08 5.0000E-01 4.5432E+04
1 2 Helmholtz VELZ F: 6.1881E-03 1.0000E-08 5.0000E-01 4.5432E+04
1 3 Helmholtz VELZ F: 3.3957E-05 1.0000E-08 5.0000E-01 4.5432E+04
1 4 Helmholtz VELZ F: 4.1984E-07 1.0000E-08 5.0000E-01 4.5432E+04
1 5 Helmholtz VELZ F: 4.6021E-09 1.0000E-08 5.0000E-01 4.5432E+04

```

```

[COUPE] Interpolating solution for physics: unic; FIELD = VTEMP
point location: wanted 6720 got 6720 locally, 0 remote, missing 0
[COUPE] Interpolation done. Printing first 5 values.
0.00001840 -0.00000223 0.00001902 0.00001864 0.00001840

```

```

[COUPE] Interpolating solution for physics: unic; FIELD = VDENSITY
[COUPE] Interpolation done. Printing first 5 values.
1.00000000 1.00000000 1.00000000 1.00000000 1.00000000

```

```

[PHY-UNIC] *** SOLVING THE PHYSICS ***
[COUPE] Interpolating solution for physics: nek; FIELD = VPOWER
[COUPE] Interpolation done. Printing first 5 values.
7.08264832 7.37698660 7.19233983 7.30203135 7.40920624

```

```

[PHY-NEK] *** SOLVING THE PHYSICS ***
1 Hmholtz VELZ: 4 4.6021E-09 3.9457E+00 1.0000E-08
L1/L2 DIV(V) : 3.5197E-01 2.0883E+01
L1/L2 QTL : 0.0000E+00 0.0000E+00
L1/L2 DIV(V)-QTL: 3.5197E-01 2.0883E+01
WARNING: DIV(V)-QTL too large!
1 2.2011E-05 1.3573E+01 Fluid done

```

```

filt amp 0.0000 0.0000 0.0000 0.0100
filt trn 1.0000 1.0000 1.0000 0.9900
Step 2, t= 4.4022022E-05, DT= 2.2011011E-05, C= 0.500 1.7598E+01 1.7598E+01

```

```

Solving for heat
1.000000000000000000000002E-008 p22 2 2
2 Hmholtz TEMP: 3 7.2871E-11 1.1774E+00 1.0000E-08
2 4.4022E-05 3.4218E+00 Heat done

```

```

Solving for fluid
1.000000000000000000000002E-008 p22 2 1
2 PRES gmres: 165 9.0303E-07 1.0000E-06 2.8677E+07 4.3064E+00 1.2356E+01 F
1.000000000000000000000002E-008 p22 2 1
2 Hmholtz VELX: 7 6.6553E-11 9.4090E+05 1.0000E-08
1.000000000000000000000002E-008 p22 2 1
2 Hmholtz VELY: 7 1.4747E-10 2.6970E+06 1.0000E-08
1.000000000000000000000002E-008 p22 2 1
2 Hmholtz VELZ: 7 2.3743E-10 3.1232E+06 1.0000E-08
L1/L2 DIV(V) : 6.0543E-01 1.3960E+02
L1/L2 QTL : 0.0000E+00 0.0000E+00
L1/L2 DIV(V)-QTL: 6.0543E-01 1.3960E+02
WARNING: DIV(V)-QTL too large!
2 4.4022E-05 1.3591E+01 Fluid done

```

```

call outfld: ifpsco: F

```

```

2 4.4022E-05 Write checkpoint:
0 2 OPEN: sahex1_nek0.f00002
2 4.4022E-05 done :: Write checkpoint
file size = 2.7 MB
avg data-throughput = 39.9MB/s
io-nodes = 1

```

```

end of time-step loop

```

```

[SN2ND]...Neutron transport cross section data was imported and distributed.....
[SN2ND]...Restarting the eigenvalue/eigenvector solver with the updated input.....
[SN2ND]...Obtaining the preconditioner matrices for the second order discrete ordinates method.....
=SN2ND Fission Source Iteration | b-Ax | Tchebychev | Within Group Flux | Preconditioner CG |
=SN2ND Seconds|Itr|Up| Eigenvalue | Error | Fis Err| Flux Err| Dom | Slope| Error |Active| Dom |Scat| CG |Max/G|Group|Assoc.Error| Total | Max/G |Group| Max/A |
=SN2ND 108.3|001|00| 1.36840E-02| 7.9E-01| 8.0E-01|T 3.3E-01|*****| 0.000| 3.2E+00| T 0| 0.798| 4| 66| 10| 4| 5.457E+02| 1754| 53| 1| 10|
=SN2ND 292.3|002|00| 5.07535E-03| 6.3E-01| 6.3E-01|T 3.4E+01| 0.236| 0.000| 4.7E-02| T 0| 0.162| 8| 186| 35| 4| 1.199E+01| 4278| 66| 2| 12|
=SN2ND 248.1|003|00| 5.07458E-03| 1.5E-04| 7.4E-03|T 1.0E-02| 0.005| 0.000| 4.9E-04| T 1| 0.004| 8| 155| 32| 4| 9.475E-02| 3646| 63| 1| 12|
[SN2ND]...Reached specified iteration limit..... 2
[SN2ND]...!!!!FAILURE!!!! Could not converge within the specified criteria.....
[SN2ND]...Sorry, but I must stop
[SN2ND]...Final eigenvalue..... 0.00507458
[SN2ND]...Final error on the eigenvalue..... 0.00015132
[SN2ND]...Final iterative error on the flux solution..... 0.01007397
[SN2ND]...Total number of fission source iterations..... 3
[SN2ND]...Total number of upscatter iterations..... 0
[SN2ND]...Total number of scattering source iterations..... 20
[SN2ND]...Within Grp Flx auto adjustment ranges: 1.000E+00 to 1.000E+00 and iteration count of..... 407
[SN2ND]...Preconditioner auto adjustment ranges: 1.000E+00 to 1.000E+00 and iteration count of..... 9678
[SN2ND]...Total number of synthetic diffusion iterations..... 293

```


[SN2ND]...Total computational time required for real solve (seconds)..... 648.979980

Starting time loop ...

```
Solving for heat
Temperature/Passive scalar solution
1.000000000000000002E-008 p22 1 2
1 1 Helmholtz TEMP F: 7.8495E-01 1.0000E-08 5.0000E-02 4.5432E+04
1 2 Helmholtz TEMP F: 3.3138E-04 1.0000E-08 5.0000E-02 4.5432E+04
1 3 Helmholtz TEMP F: 2.3309E-07 1.0000E-08 5.0000E-02 4.5432E+04
1 4 Helmholtz TEMP F: 1.6365E-10 1.0000E-08 5.0000E-02 4.5432E+04
1 Hmholtz TEMP: 3 1.6365E-10 7.8495E-01 1.0000E-08
1 6.6033E-05 3.3281E+00 Heat done
Solving for fluid
1.000000000000000002E-008 p22 1 1
New CG1-tolerance (RINIT*epsm) = 1.53141937300573490E-005 9.9999999999999955E-007
New CG1-tolerance (Neumann) = 1.53141937300573503E-004
New CG1-tolerance (RINIT*epsm) = 1.53141937300573490E-005 9.9999999999999955E-007
New CG1-tolerance (Neumann) = 1.53141937300573503E-004
1 1.00000E-06 3.72779E+06 9.56031E+06 3.89923E-01 1 Divergence
2 1.00000E-06 2.33798E+06 9.56031E+06 2.44550E-01 1 Divergence
3 1.00000E-06 1.69458E+06 9.56031E+06 1.77252E-01 1 Divergence
4 1.00000E-06 1.22587E+06 9.56031E+06 1.28225E-01 1 Divergence
5 1.00000E-06 8.26777E+05 9.56031E+06 8.64802E-02 1 Divergence
6 1.00000E-06 5.71005E+05 9.56031E+06 5.97266E-02 1 Divergence
7 1.00000E-06 4.42873E+05 9.56031E+06 4.63242E-02 1 Divergence
8 1.00000E-06 3.17595E+05 9.56031E+06 3.32201E-02 1 Divergence
9 1.00000E-06 2.31748E+05 9.56031E+06 2.42407E-02 1 Divergence
10 1.00000E-06 1.64637E+05 9.56031E+06 1.72209E-02 1 Divergence
11 1.00000E-06 1.17014E+05 9.56031E+06 1.22396E-02 1 Divergence
12 1.00000E-06 8.56999E+04 9.56031E+06 8.96414E-03 1 Divergence
13 1.00000E-06 6.63706E+04 9.56031E+06 6.94230E-03 1 Divergence
14 1.00000E-06 5.03150E+04 9.56031E+06 5.26291E-03 1 Divergence
15 1.00000E-06 3.78358E+04 9.56031E+06 3.95759E-03 1 Divergence
16 1.00000E-06 2.96656E+04 9.56031E+06 3.10300E-03 1 Divergence
17 1.00000E-06 2.32478E+04 9.56031E+06 2.43170E-03 1 Divergence
18 1.00000E-06 1.78904E+04 9.56031E+06 1.87132E-03 1 Divergence
19 1.00000E-06 1.41704E+04 9.56031E+06 1.48221E-03 1 Divergence
20 1.00000E-06 1.16281E+04 9.56031E+06 1.21628E-03 1 Divergence
21 1.00000E-06 9.22730E+03 9.56031E+06 9.65167E-04 1 Divergence
22 1.00000E-06 7.43710E+03 9.56031E+06 7.77914E-04 1 Divergence
23 1.00000E-06 5.79969E+03 9.56031E+06 6.06642E-04 1 Divergence
24 1.00000E-06 4.54892E+03 9.56031E+06 4.75813E-04 1 Divergence
25 1.00000E-06 3.79442E+03 9.56031E+06 3.96893E-04 1 Divergence
26 1.00000E-06 3.13708E+03 9.56031E+06 3.28136E-04 1 Divergence
27 1.00000E-06 2.53693E+03 9.56031E+06 2.65361E-04 1 Divergence
28 1.00000E-06 2.05168E+03 9.56031E+06 2.14604E-04 1 Divergence
29 1.00000E-06 1.65519E+03 9.56031E+06 1.73132E-04 1 Divergence
30 1.00000E-06 1.38258E+03 9.56031E+06 1.44617E-04 1 Divergence
31 1.00000E-06 1.14752E+03 9.56031E+06 1.20030E-04 1 Divergence
32 1.00000E-06 9.37671E+02 9.56031E+06 9.80795E-05 1 Divergence
33 1.00000E-06 7.63966E+02 9.56031E+06 7.99102E-05 1 Divergence
34 1.00000E-06 6.28980E+02 9.56031E+06 6.57907E-05 1 Divergence
35 1.00000E-06 5.11431E+02 9.56031E+06 5.34953E-05 1 Divergence
36 1.00000E-06 4.23085E+02 9.56031E+06 4.42543E-05 1 Divergence
37 1.00000E-06 3.55332E+02 9.56031E+06 3.71674E-05 1 Divergence
38 1.00000E-06 2.97811E+02 9.56031E+06 3.11507E-05 1 Divergence
39 1.00000E-06 2.56282E+02 9.56031E+06 2.68069E-05 1 Divergence
40 1.00000E-06 2.18843E+02 9.56031E+06 2.28907E-05 1 Divergence
41 1.00000E-06 2.01857E+02 9.56031E+06 2.11141E-05 1 Divergence
42 1.00000E-06 1.84390E+02 9.56031E+06 1.92870E-05 1 Divergence
43 1.00000E-06 1.63967E+02 9.56031E+06 1.71508E-05 1 Divergence
44 1.00000E-06 1.43788E+02 9.56031E+06 1.50401E-05 1 Divergence
45 1.00000E-06 1.25710E+02 9.56031E+06 1.31492E-05 1 Divergence
46 1.00000E-06 1.08587E+02 9.56031E+06 1.13581E-05 1 Divergence
47 1.00000E-06 9.27885E+01 9.56031E+06 9.70560E-06 1 Divergence
48 1.00000E-06 7.93437E+01 9.56031E+06 8.29928E-06 1 Divergence
49 1.00000E-06 6.77840E+01 9.56031E+06 7.09014E-06 1 Divergence
50 1.00000E-06 5.70421E+01 9.56031E+06 5.96655E-06 1 Divergence
51 1.00000E-06 4.76637E+01 9.56031E+06 4.98558E-06 1 Divergence
52 1.00000E-06 4.09408E+01 9.56031E+06 4.28237E-06 1 Divergence
53 1.00000E-06 3.46741E+01 9.56031E+06 3.62689E-06 1 Divergence
54 1.00000E-06 2.92516E+01 9.56031E+06 3.05969E-06 1 Divergence
55 1.00000E-06 2.44449E+01 9.56031E+06 2.55691E-06 1 Divergence
56 1.00000E-06 2.04198E+01 9.56031E+06 2.13590E-06 1 Divergence
57 1.00000E-06 1.73837E+01 9.56031E+06 1.81832E-06 1 Divergence
58 1.00000E-06 1.44116E+01 9.56031E+06 1.50744E-06 1 Divergence
59 1.00000E-06 1.21208E+01 9.56031E+06 1.26783E-06 1 Divergence
60 1.00000E-06 1.04597E+01 9.56031E+06 1.09408E-06 1 Divergence
61 1.00000E-06 8.75185E+00 9.56031E+06 9.15436E-07 1 Divergence
62 1.00000E-06 7.33276E+00 9.56031E+06 7.67000E-07 1 Divergence
63 1.00000E-06 6.15169E+00 9.56031E+06 6.43461E-07 1 Divergence
64 1.00000E-06 5.19249E+00 9.56031E+06 5.43130E-07 1 Divergence
65 1.00000E-06 4.42768E+00 9.56031E+06 4.63131E-07 1 Divergence
66 1.00000E-06 3.79748E+00 9.56031E+06 3.97213E-07 1 Divergence
67 1.00000E-06 3.26487E+00 9.56031E+06 3.41502E-07 1 Divergence
68 1.00000E-06 2.79620E+00 9.56031E+06 2.92480E-07 1 Divergence
69 1.00000E-06 2.33683E+00 9.56031E+06 2.44430E-07 1 Divergence
70 1.00000E-06 1.95997E+00 9.56031E+06 2.05011E-07 1 Divergence
71 1.00000E-06 1.68037E+00 9.56031E+06 1.75765E-07 1 Divergence
72 1.00000E-06 1.43498E+00 9.56031E+06 1.50097E-07 1 Divergence
73 1.00000E-06 1.19760E+00 9.56031E+06 1.25268E-07 1 Divergence
```

```

74 1.00000E-06 9.89498E-01 9.56031E+06 1.03501E-07 1 Divergence
75 1.00000E-06 8.16632E-01 9.56031E+06 8.54189E-08 1 Divergence
76 1.00000E-06 6.66019E-01 9.56031E+06 6.96650E-08 1 Divergence
77 1.00000E-06 5.20365E-01 9.56031E+06 5.44297E-08 1 Divergence
78 1.00000E-06 4.23437E-01 9.56031E+06 4.42912E-08 1 Divergence
79 1.00000E-06 3.55137E-01 9.56031E+06 3.71471E-08 1 Divergence
80 1.00000E-06 2.95675E-01 9.56031E+06 3.09273E-08 1 Divergence
81 1.00000E-06 2.66619E-01 9.56031E+06 2.78881E-08 1 Divergence
82 1.00000E-06 2.39660E-01 9.56031E+06 2.50682E-08 1 Divergence
83 1.00000E-06 2.09175E-01 9.56031E+06 2.18795E-08 1 Divergence
84 1.00000E-06 1.83036E-01 9.56031E+06 1.91454E-08 1 Divergence
85 1.00000E-06 1.58358E-01 9.56031E+06 1.65641E-08 1 Divergence
86 1.00000E-06 1.36890E-01 9.56031E+06 1.43185E-08 1 Divergence
87 1.00000E-06 1.16605E-01 9.56031E+06 1.21968E-08 1 Divergence
88 1.00000E-06 9.65318E-02 9.56031E+06 1.00971E-08 1 Divergence
89 1.00000E-06 7.95673E-02 9.56031E+06 8.32267E-09 1 Divergence
90 1.00000E-06 6.57941E-02 9.56031E+06 6.88200E-09 1 Divergence
91 1.00000E-06 5.56720E-02 9.56031E+06 5.82324E-09 1 Divergence
92 1.00000E-06 4.76256E-02 9.56031E+06 4.98159E-09 1 Divergence
93 1.00000E-06 4.03770E-02 9.56031E+06 4.22340E-09 1 Divergence
94 1.00000E-06 3.42399E-02 9.56031E+06 3.58146E-09 1 Divergence
95 1.00000E-06 2.85749E-02 9.56031E+06 2.98891E-09 1 Divergence
96 1.00000E-06 2.39926E-02 9.56031E+06 2.50961E-09 1 Divergence
97 1.00000E-06 2.01492E-02 9.56031E+06 2.10758E-09 1 Divergence
98 1.00000E-06 1.68094E-02 9.56031E+06 1.75825E-09 1 Divergence
99 1.00000E-06 1.38879E-02 9.56031E+06 1.45266E-09 1 Divergence
100 1.00000E-06 1.15246E-02 9.56031E+06 1.20546E-09 1 Divergence
101 1.00000E-06 9.51222E-03 9.56031E+06 9.94970E-10 1 Divergence
102 1.00000E-06 8.06618E-03 9.56031E+06 8.43715E-10 1 Divergence
103 1.00000E-06 6.82232E-03 9.56031E+06 7.13609E-10 1 Divergence
104 1.00000E-06 5.76496E-03 9.56031E+06 6.03010E-10 1 Divergence
105 1.00000E-06 4.94188E-03 9.56031E+06 5.16916E-10 1 Divergence
106 1.00000E-06 4.23752E-03 9.56031E+06 4.43241E-10 1 Divergence
107 1.00000E-06 3.59862E-03 9.56031E+06 3.76412E-10 1 Divergence
108 1.00000E-06 3.02771E-03 9.56031E+06 3.16695E-10 1 Divergence
109 1.00000E-06 2.54945E-03 9.56031E+06 2.66670E-10 1 Divergence
110 1.00000E-06 2.19158E-03 9.56031E+06 2.29238E-10 1 Divergence
111 1.00000E-06 1.87091E-03 9.56031E+06 1.95695E-10 1 Divergence
112 1.00000E-06 1.57980E-03 9.56031E+06 1.65246E-10 1 Divergence
113 1.00000E-06 1.33207E-03 9.56031E+06 1.39334E-10 1 Divergence
114 1.00000E-06 1.11004E-03 9.56031E+06 1.16109E-10 1 Divergence
115 1.00000E-06 9.26173E-04 9.56031E+06 9.68768E-11 1 Divergence
116 1.00000E-06 7.69503E-04 9.56031E+06 8.04894E-11 1 Divergence
117 1.00000E-06 6.53939E-04 9.56031E+06 6.84014E-11 1 Divergence
118 1.00000E-06 5.66000E-04 9.56031E+06 5.92031E-11 1 Divergence
119 1.00000E-06 4.98374E-04 9.56031E+06 5.21295E-11 1 Divergence
120 1.00000E-06 4.41205E-04 9.56031E+06 4.61496E-11 1 Divergence
121 1.00000E-06 4.03253E-04 9.56031E+06 4.21799E-11 1 Divergence
122 1.00000E-06 3.65336E-04 9.56031E+06 3.82138E-11 1 Divergence
123 1.00000E-06 3.23826E-04 9.56031E+06 3.38719E-11 1 Divergence
124 1.00000E-06 2.87290E-04 9.56031E+06 3.00502E-11 1 Divergence
125 1.00000E-06 2.48924E-04 9.56031E+06 2.60372E-11 1 Divergence
126 1.00000E-06 2.14724E-04 9.56031E+06 2.24599E-11 1 Divergence
127 1.00000E-06 1.85085E-04 9.56031E+06 1.93598E-11 1 Divergence
128 1.00000E-06 1.61780E-04 9.56031E+06 1.69220E-11 1 Divergence
129 1.00000E-06 1.40733E-04 9.56031E+06 1.47205E-11 1 Divergence
130 1.00000E-06 1.21874E-04 9.56031E+06 1.27479E-11 1 Divergence
131 1.00000E-06 1.04207E-04 9.56031E+06 1.08999E-11 1 Divergence
132 1.00000E-06 9.15772E-05 9.56031E+06 9.57889E-12 1 Divergence
133 1.00000E-06 7.95593E-05 9.56031E+06 8.32183E-12 1 Divergence
134 1.00000E-06 6.64855E-05 9.56031E+06 6.95432E-12 1 Divergence
135 1.00000E-06 5.47204E-05 9.56031E+06 5.72371E-12 1 Divergence
136 1.00000E-06 4.44547E-05 9.56031E+06 4.64993E-12 1 Divergence
137 1.00000E-06 3.59687E-05 9.56031E+06 3.76229E-12 1 Divergence
138 1.00000E-06 2.93618E-05 9.56031E+06 3.07122E-12 1 Divergence
139 1.00000E-06 2.42924E-05 9.56031E+06 2.54097E-12 1 Divergence
140 1.00000E-06 2.03927E-05 9.56031E+06 2.13306E-12 1 Divergence
141 1.00000E-06 1.67280E-05 9.56031E+06 1.74973E-12 1 Divergence
142 1.00000E-06 1.40841E-05 9.56031E+06 1.47318E-12 1 Divergence
143 1.00000E-06 1.18287E-05 9.56031E+06 1.23727E-12 1 Divergence
144 1.00000E-06 9.89776E-06 9.56031E+06 1.03530E-12 1 Divergence
145 1.00000E-06 8.08106E-06 9.56031E+06 8.45271E-13 1 Divergence
146 1.00000E-06 6.70568E-06 9.56031E+06 7.01408E-13 1 Divergence
147 1.00000E-06 5.77204E-06 9.56031E+06 6.03751E-13 1 Divergence

Coupe Object: 1 MPI processes
type: loose
number of physics objects=2
Physics Object: 1 MPI processes
type: unic
The default physics implementation details here..
[PHY-UNIC] Dummy skeleton view.
148 1.00000E-06 5.02993E-06 9.56031E+06 5.26126E-13 1 Divergence
149 1.00000E-06 4.16091E-06 9.56031E+06 4.35227E-13 1 Divergence
150 1.00000E-06 3.45018E-06 9.56031E+06 3.60885E-13 1 Divergence
151 1.00000E-06 2.92519E-06 9.56031E+06 3.05972E-13 1 Divergence
152 1.00000E-06 2.47855E-06 9.56031E+06 2.59254E-13 1 Divergence
153 1.00000E-06 2.06596E-06 9.56031E+06 2.16097E-13 1 Divergence
154 1.00000E-06 1.77153E-06 9.56031E+06 1.85301E-13 1 Divergence
155 1.00000E-06 1.53560E-06 9.56031E+06 1.60623E-13 1 Divergence
156 1.00000E-06 1.34384E-06 9.56031E+06 1.40565E-13 1 Divergence
157 1.00000E-06 1.14778E-06 9.56031E+06 1.20057E-13 1 Divergence
158 1.00000E-06 9.68803E-07 9.56031E+06 1.01336E-13 1 Divergence

```

[SN2ND]...	22 SN2ND A*x operations	0.00	609	1.3093	797.3902	76.79
[SN2ND]...	23 SN2ND diagonal block CG solve operation (PETSc)	0.00	567	0.1188	67.3802	6.49
[SN2ND]...	24 SN2ND P-multigrid prolongation	0.00	0	0.0000	0.0000	0.00
[SN2ND]...	25 SN2ND P-multigrid restriction	0.00	0	0.0000	0.0000	0.00
[SN2ND]...	26 SN2ND P-multigrid smoothing	0.00	0	0.0000	0.0000	0.00
[SN2ND]...	27 SN2ND synthetic diffusion solve (PETSc)	0.00	39	0.0605	2.3598	0.231

[SN2ND]...	28 SN2ND DSA P-multigrid prolongation		0.00	0	0.0000	0.0000	0.00
[SN2ND]...	29 SN2ND DSA P-multigrid restriction		0.00	0	0.0000	0.0000	0.00
[SN2ND]...	30 SN2ND DSA P-multigrid smoothing		0.00	0	0.0000	0.0000	0.00
[SN2ND]...	31 SN2ND Communicate the angular flux		0.00	591	0.0000	0.0000	0.00
[SN2ND]...	32 SN2ND global reduce of the PN flux		0.00	111	0.0000	0.0000	0.00
[SN2ND]...	33 SN2ND SN to PN flux conversion		0.00	237	0.0038	0.9099	0.09
[SN2ND].....							

References

- [1] Timothy J. Tautges, Hong-Jun Kim, Alvaro Caceres, and Rajeev Jain. Coupled Multi-Physics simulation frameworks for reactor simulation: A Bottom-Up approach. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011)*, Rio de Janeiro, Brazil, May 2011. American Nuclear Society.
- [2] A. Siegel, T. Tautges, A. Caceres, D. Kaushik, P. Fischer, G. Palmiotti, M. Smith, and J. Ragusa. Software design of SHARP. In *Joint International Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications, M and C + SNA 2007*. American Nuclear Society, April 2007. Conference Proceedings.
- [3] James R Stewart and H.Carter Edwards. A framework approach for developing parallel adaptive multiphysics applications. *Finite Elements in Analysis and Design*, 40(12):1599–1617, July 2004.
- [4] Derek Gaston, Chris Newman, Glen Hansen, and Damien Lebrun-Grandi. MOOSE: a parallel computational framework for coupled systems of nonlinear equations. *Nuclear Engineering and Design*, 239(10):1768–1778, October 2009.
- [5] Adrian Tentner. Benchmark specifications for EBR-II shutdown heat removal test. Technical Report ANL-GenIV-137, Argonne National Laboratory, June 2010.
- [6] Technical report.
- [7] T. J. Tautges, R. Meyers, K. Merkley, C. Stimpson, and C. Ernst. MOAB: a Mesh-Oriented database. SAND2004-1592, Sandia National Laboratories, April 2004. Report.
- [8] Timothy J. Tautges. MOAB wiki. <http://trac.mcs.anl.gov/projects/ITAPS/wiki/MOAB>.
- [9] Tautges, Timothy J., Kraftcheck, Jason, Bertram, Nathan, Sachdeva, Vipin, and Magerlein, John. Mesh interface resolution and ghost exchange in a parallel mesh representation. Shanghai, China, May 2012. IEEE.
- [10] Karen Devine, Erik Boman, Robert Heaphy, Bruce Hendrickson, and Courtenay Vaughan. Zoltan data management services for parallel dynamic applications. *Computing in Science and Engineering*, 4(2):90–97, 2002.
- [11] ParaView - open source scientific visualization. <http://www.paraview.org/>.
- [12] Gregory D. Sjaardema, Timothy J. Tautges, Tammy J. Wilson, Steven J. Owen, Teddy D. Blacker, William J. Bohnhoff, Tony L. Edwards, James R. Hipp, Randall R. Lober, Scott A. Mitchell, and Sandia National Laboratories. CUBIT mesh generation environment volume 1: Users manual. Technical Report SAND-94-1100, Sandia National Laboratories, May 1994, 1994.
- [13] Timothy J. Tautges and Alvaro Caceres. Scalable parallel solution coupling for multiphysics reactor simulation. *Journal of Physics: Conference Series*, 180, 2009.
- [14] A.T. Patera. A spectral element method for fluid dynamics : laminar flow in a channel expansion. 54:468–488, 1984.
- [15] Y. Maday and 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*, pages 71–143. ASME, New York, 1989.
- [16] P.F. Fischer and A.T. Patera. Parallel spectral element of the Stokes problem. 92:380–421, 1991.
- [17] P.F. Fischer. An overlapping Schwarz method for spectral element solution of the incompressible Navier-Stokes equations. 133:84–101, 1997.
- [18] 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.
- [19] A.G. Tomboulides and S.A. Orszag. A quasi-two-dimensional benchmark problem for low Mach number compressible codes. 146:691–706, 1998.
- [20] A.G. Tomboulides, M. Israeli, and G.E. Karniadakis. Efficient removal of boundary-divergence errors in time-splitting methods. *J. Sci. Comput.*, 4:291–308, 1989.
- [21] M.O. Deville, P.F. Fischer, and E.H. Mund. *High-order methods for incompressible fluid flow*. Cambridge University Press, Cambridge, 2002.
- [22] M. A. Smith, C. Rabiti, G. Palmiotti, D. Kaushik, A. Siegel, B. Smith, T. Tautges, and W. S. Yang. UNIC: development of a new reactor physics analysis tool, invited. In *2007 Winter Meeting on International Conference on Making the Renaissance Real*, volume 97, pages 565–566. American Nuclear Society, November 2007. Conference Proceedings.
- [23] Dinesh Kaushik, Micheal Smith, Allan Wollaber, Barry Smith, Andrew Siegel, and Won Sik Yang. Enabling high-fidelity neutron transport simulations on petascale architectures. In *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, page 67:1–12, New York, NY, USA, 2009. ACM.
- [24] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

- [25] I.D. Parsons, J.M. Solberg, R.M. Ferencz, M.A. Havstad, N.E. Hodge, and A.P. Wemhoff. Diablo user manual. UCRL-SM-234927, Lawrence Livermore National Laboratory, September 2007. Report.
- [26] M.A. Puso and T.A. Laursen. A mortar segment-to-segment frictional contact method for large deformations. *Computer Methods In Applied Mechanics and Engineering*, 193(45-47):4891–4913, 2004.
- [27] Timothy J. Tautges and Rajeev Jain. Creating geometry and mesh models for nuclear reactor core geometries using a lattice hierarchy-based approach. *Engineering with Computers*, September 2011.
- [28] VisIt user’s guide. Technical Report UCRL-SM-220449, Lawrence Livermore National Laboratory, October 2005.
- [29] G. I. Marchuk. *On the theory of the splitting-up method*, volume II of *Numerical Solution of Partial Differential Equations*. Academic Press, New York, 1971.
- [30] D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.
- [31] Timothy J. Tautges and Rajeev Jain. Report on FY11 extensions to MeshKit and RGG. Technical Report ANL-11/48, Argonne National Laboratory, September 2011.
- [32] Timothy J. Tautges, Jason Kraftcheck, Porter, Jim, Caceres, Alvaro, Grindeanu, Iulian, Karpeev, Dmitry, Jain, Rajeev, Kim, Hong-Jun, Cai, Shengyong, Jackson, Steve, Hu, Jiangtao, Smith, Brandon, Verma, Chaman, Slattery, Stuart, and Wilson, Paul. MeshKit: a Open-Source library for mesh generation. In *Proceedings, SIAM Conference on Computational Science & Engineering*, Reno, NV, March 2011. SIAM.

**Mathematics and Computation Division &
Nuclear Engineering Division**

Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439

www.anl.gov

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