Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Clean Energy—Article

Computational Tools for the Integrated Design of Advanced Nuclear Reactors

Nicholas W. Touran, John Gilleland*, Graham T. Malmgren, Charles Whitmer, William H. Gates III

TerraPower, LLC, Bellevue, WA 98005, USA

ARTICLE INFO

Article history:
Received 16 March 2017
Revised 7 July 2017
Accepted 12 July 2017
Available online 3 August 2017

Keywords:
Simulation
Nuclear energy
Electricity generation
Advanced reactor
Traveling wave reactor

ABSTRACT

Advanced nuclear reactors offer safe, clean, and reliable energy at the global scale. The development of such devices relies heavily upon computational models, from the pre-conceptual stages through detailed design, licensing, and operation. An integrated reactor modeling framework that enables seamless communication, coupling, automation, and continuous development brings significant new capabilities and efficiencies to the practice of reactor design. In such a system, key performance metrics (e.g., optimal fuel management, peak cladding temperature in design-basis accidents, levelized cost of electricity) can be explicitly linked to design inputs (e.g., assembly duct thickness, tolerances), enabling an exceptional level of design consistency. Coupled with high-performance computing, thousands of integrated cases can be executed simultaneously to analyze the full system, perform complete sensitivity studies, and efficiently and robustly evaluate various design tradeoffs. TerraPower has developed such a tool—the Advanced Reactor Modeling Interface (ARMI) code system—and has deployed it to support the TerraPower Traveling Wave Reactor design and other innovative energy products currently under development. The ARMI code system employs pre-existing tools with strong pedigrees alongside many new physics and data management modules necessary for innovative design. Verification and validation against previous and new physical measurements, which remain an essential element of any sound design, are being carried out. This paper summarizes the integrated core engineering tools and practices in production at TerraPower.

1. Introduction

Computing environments provide all engineering design efforts with vast virtual laboratories in which ideas can mature into sophisticated, well-optimized systems. Leveraging such virtual facilities has been and remains particularly essential to progress in the nuclear industry due to the associated complex physics and high experimental costs. Indeed, modern computing and nuclear technology were born together, as the first digital computer's first task was to solve a set of partial differential equations modeling deuterium-tritium behavior for Edward Teller in 1945 [1].

The first nuclear reactors were designed with analytic methods informed by experiments. Soon, radiation transport problems were being solved numerically on early computers to supplement shielding and core design activities. In 1949, submarine reactor designers at Knoll's Atomic Power Laboratory obtained the first known computer solution of the neutron diffusion equation on an IBM 604 [2]. Increasingly powerful computers, as well as the advent of the Fortran programming language on the IBM 704 in 1957, enabled more sophisticated physics to be treated numerically. Concurrent advances in storage technology enabled the inclusion of extended nuclear data libraries, a key input to reactor simulations representing reaction probabilities. To this day, however, new reactor design concepts are generally simulated physically, often by zero-power critical arrangements of fuel assemblies in experimental core mock-up facilities.

From the 1970s through the 1990s, reactor simulation software became increasingly precise for isolated physics problems, covering the neutronics, heat transfer, fuel performance, transient analysis, and mechanical fields, among others. These codes were largely

* Corresponding author.
E-mail address: johng@terrapower.com

http://dx.doi.org/10.1016/J.ENG.2017.04.016

developed independently from one another, with individual subject matter experts being responsible for understanding their range of applicability and operation. In many cases, design teams created standard interface files to transfer data, allowing full system treatment in a somewhat manual, though effective fashion. Many commercial tools now perform some essential physics coupling (e.g., neutronics and thermal/hydraulics in water-cooled reactors), but much modeling focus in the advanced reactor regime has been on increasingly high-fidelity physics models, taking advantage of modern computer architectures to minimize approximations. Today's ongoing high-fidelity and multiphysics simulation efforts are a scientific endeavor aimed at reducing uncertainties, enhancing understanding, and enabling predictive capabilities. However, they are not meant to be directly usable by the industry yet.

In 2006, TerraPower set out to develop sustainable, scalable, and low-carbon energy in the Traveling Wave Reactor (TWR) program, featuring a fourth-generation, sodium-cooled, metal-fueled reactor unique in that it uses a once-through deep-burn fuel cycle to achieve many fast reactor capabilities (natural safety, reduced waste, reduction and eventual elimination of enrichment, and high fuel and thermal efficiency) without requiring reprocessing [3]. High-fidelity (but decoupled) physics models demonstrated the fundamental feasibility of the TWR design. As the organization grew, new software was developed and procured to support the evolving reactor design. In June 2009, development of the Advanced Reactor Modeling Interface (ARMI) code system began with the intent of incorporating new and existing physics modeling tools with data management and automation routines into a consistent reactor design toolbox. The relatively clean slate and initially small team provided the impetus for applying modern design patterns and programming practices to the challenge of highly efficient, scalable, and integrated reactor design. As the framework and data management developed, subject experts focused on creating new physics modules or adapters to high-quality external physics solvers. As the tools were integrated, each member of each team could seamlessly run the entire system analysis, from specifying the pin dimensions and tolerances to computing the system cost and peak cladding temperature during design-basis transients. Detailed and meaningful design, innovation, and sensitivity studies could be done with ease. Such a system has allowed TerraPower to develop its designs with aggressive timescales and small, agile teams.

2. Architecture

The ARMI framework comprises two key entities: the *reactor model* and the *interface stack*. The reactor model is responsible for maintaining a self-consistent state representing the full physical description of the reactor, including the geometry, material properties, power, temperatures, and so forth. The interface stack is responsible for performing various physics modeling activities using and updating the reactor model. This object-oriented architecture allows the coupling of reactor physics and analysis tools while promoting the decoupling of the source code itself, which is ideal, given the breadth of advanced nuclear design tools and the specialization of engineers with expertise in operating them. The diverse specialization of the ARMI contributors (who typically have advanced degrees in engineering) suggests the use of a high-level programming language for data management that can be learned relatively quickly (i.e., Python[†] for the majority of the code).

2.1. Reactor model

The reactor model in the ARMI code system is a composite pattern [4] mirroring physical components of a nuclear reactor core, as depicted in Fig. 1. The *reactor* object comprises a number of *assemblies*, which are made of subsections known as *blocks*, and which in turn contain shaped *components* such as fuel pins, cladding, and coolant. Each component is linked to a *material* with temperature- and composition-dependent properties. Data management in the ARMI code system simply involves reading and writing the state to the reactor model via the defined model interface. Standardizing this eponymous reactor model interface is key to the multiphysics interoperability of the ARMI code system, as any data passed through it lose their code-specific character and become globally accessible.

An application programming interface (API) provides users and programmers with intricate access to the entire reactor model. Weighted averages of any state variable over any arbitrary selection of the reactor are immediately available, and all-encompassing state changes (e.g., changing coolant density) can be programmed in a single statement.

A consistent and programmatically accessible material properties database is essential for integrated reactor design. Material objects have been created with various correlations embedded for thermo-mechanical properties: mass density, thermal expansion, heat capacity, viscosity, thermal conductivity, Young's modulus, yield strength, and so forth. Physics modules that in the past may have queried for *sodium* properties at a region's current temperature now only query for *coolant* properties. Thus, with a single line change to the input, a module that had been computing the sodium density coefficient of reactivity instantly computes the fluoride salt density coefficient of reactivity. A feature to automatically evaluate the ranges of the correlations and print them to a report for offline use has also proven valuable. In addition, the intricate link to the materials library allows for automatic and consistent thermal expansion from cold, as-manufactured dimensions to hot, full-power dimensions, or anywhere in between.

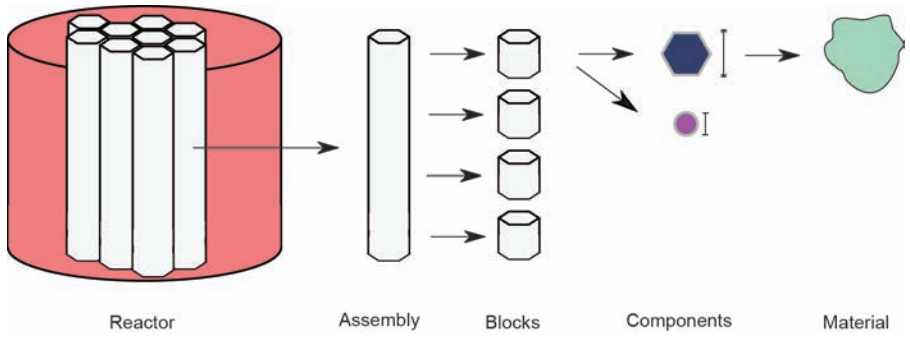

Fig. 1. The ARMI reactor model composite pattern in hexagonal geometry.

[†] A widely used programming language created by Guido van Rossum in 1991.