Nuclear Engineering and Design xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Nuclear Engineering and Design

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

Verification of the HTR code package (HCP) as a comprehensive HTR steady state and transient safety analysis framework

S. Kasselmann[a,*], A. Xhonneux[a], F. Tantillo[a,b], A. Trabadela[a,b], D. Lambertz[a], H.-J. Allelein[a,b]

[a] Forschungszentrum Jülich, 52425 Jülich, Germany
[b] Institute for Reactor Safety and Reactor Technology, RWTH Aachen University, 52064 Aachen, Germany

ARTICLE INFO

Keywords:
HTR
Simulation
Reactor Physics
Code Package
HCP

ABSTRACT

The HTR code package (HCP) allows for the simulation of several safety-related aspects of a High Temperature Reactor core in a highly integrated manner. HCP currently couples the thermo-fluid dynamics and time dependent neutronics code MGT, the spectrum code TRISHA, the burn up code TNT, the fuel management code SHUFLE and the fission product release code STACY. During its development, state-of-the-art programming techniques and standards are applied. The cross sections in HCP are generated by a 0D- or 1D-solver featuring an innovative approach to treat the double heterogeneity of pebble fuel. Also major improvements have been made to optimize the nuclear data library based on ENDF/B-VII. Two advanced fuel management models are provided by the code module SHUFLE that go far beyond the capabilities of existing system codes like VSOP. The source term analysis code module STACY is coupled to HCP providing release rate calculations with a high spatial resolution making use of the nuclide densities provided by TNT. An outstanding new feature of HCP is the possibility to simulate long-term operation scenarios based on OTTO or MEDUL fuel shuffling schemes as well as selected transients in one integrated code package. Even alternating steady state/transient/steady state simulations are possible. So with HCP different fuel strategies and their influence on various kinds of accidents can be examined with one consistent reactor model. This paper provides an overview of the development status of the HCP and reports about selected benchmark results. It is demonstrated that the new system code HCP is capable to replace existing stand-alone codes like VSOP, TINTE/MGT, FRESCO or PANAMA while introducing new features, which so far to our knowledge were not available in the field of HTR safety research.

1. Introduction

The history of gas cooled high-temperature reactor research in Germany is closely related to Forschungszentrum Jülich and its 'Institute for nuclear waste management and reactor safety (IEK-6)'. While HTR research at IEK-6 was formerly focused on the development of pebble bed reactor designs, the key task today is the evaluation and improvement of safety aspects during both normal operation and accident conditions.

A variety of individual computer codes have been developed, validated and optimized to simulate the different aspects of HTR, which allow conducting the mentioned safety analyses. These codes are widely used today and are applied in recent licensing procedures. These codes are covering different aspects of HTR. Among these codes the two system codes VSOP (Rütten et al., 2012) and MGT (Druska et al., 2009; Gerwin, 1987; Gerwin, 1989) are applied to simulate the reactor core/primary loop of HTR. VSOP is used for core design, fuel cycle and reactor operation simulation on a long time scale. This allows for studying safety aspects under normal operating conditions, e.g. establish a core status as input for an accident analysis. For some accident scenarios where neutronics are not involved, the analysis can also be performed with VSOP. This code establishes the basis for each more in-depth safety studies.

MGT is applied to look at the dynamic behavior of HTR on a shorter time scale. All these codes offer a wide range of physics modules including 2D/3D coupled neutronics and fluid dynamics, fuel shuffling, burn up, forced flow and natural convection, gas mixture and gas diffusion as well as graphite corrosion chemistry. The calculation of the power history with correlated burn up and isotope compositions can be used to run decay heat calculation codes such as NAKURE (Rütten and Haas, 2003) on one hand and fission product release and fuel performance codes such as FRESCO-II (Krohn and Finken, 1983) and PANAMA (Verfondern and Nabielek, 1985) on the other hand.

In order to document and conserve the know-how gained during decades in the field of HTR safety studies, to overcome the limitations of the old legacy codes and to exploit the advantages of modern

* Corresponding author.
E-mail address: s.kasselmann@fz-juelich.de (S. Kasselmann).

https://doi.org/10.1016/j.nucengdes.2017.11.044
Received 25 April 2017; Received in revised form 21 November 2017; Accepted 28 November 2017
0029-5493/

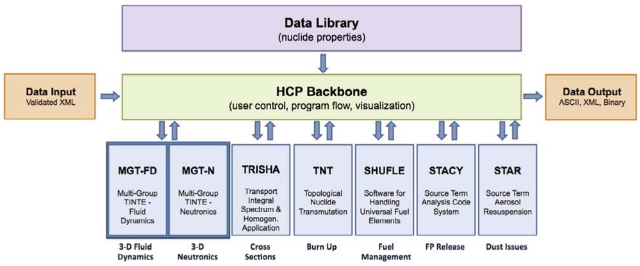

Fig. 1. Architecture of the HTR code package (HCP).

computer clusters, these individual programs developed at Aachen University and Forschungszentrum Jülich were integrated into a consistent code package applying state-of-the-art programming techniques and standards. For some HTR specific issues new modeling approaches are used, which document the knowledge in contemporary fashion. Users can now study safety-related aspects of HTR cores with one integrated code package and one reactor model input.

2. The HTR code package (HCP)

The HTR code package (HCP) couples the related and recently applied physics models in a highly integrated manner. This allows the simulation of steady state and transient operating conditions of a full 3D reactor model including physics aspects such as fission product release calculations for each core zone or dust production and transport simulation which allow for more in-depth safety analysis. The architecture of the final HCP is displayed in Fig. 1.

The HCP features the HCP backbone software (driver) which is a new code written in C++. The backbone performs all I/O operations, the data management and the main program control. Several physics modules are coupled to this backbone. Each of the modules deals with a different physics aspect of an HTR reactor core. The neutronics (MGT-N) as well as the fluid dynamics module (MGT-FD) are derived from the code MGT-3D. As a first step towards the separate modules MGT-N and MGT-FD, the MGT-3D code was split into two distinct modules for neutronics and fluid dynamics. New features like the treatment of block type fuel elements are included (Shi, 2014). The algorithms for the calculation of burn up (TNT) and fuel management (SHUFLE) offer a lot of new features compared to VSOP and have been re-implemented completely in C++. The system code VSOP itself is not part of the HCP as the code basis is not well suited to be ported to such a modern code platform. STACY is a code module replacing legacy codes such as FRESCO-II, FRESCO-I and PANAMA. Its task is to simulate the fuel performance as well as the fission product release from coated particles and fuel elements.

2.1. Nuclear data library generation

One milestone of the HCP project was the usage of one consistent nuclear data library for all physics code modules. Therefore point wise ENDF/B-VII data is first processed with NJOY2012 (Kahler, 2012). The nuclear data library is then generated by a code called LibGen, which fills data from the ENDF-6 data files of NJOY into the HCP data model. To do this a code was developed that fills nuclide data objects from ENDF-6 raw files (see Fig. 2).

The HCP data library contains basic nuclide properties like decay information, reaction channels, microscopic cross sections, spontaneous and independent fission yields for up to three incident neutron energies or scattering matrices for graphite and H_2O.

Special emphasis was put on the correct treatment of the double heterogeneous structure of HTR fuel. From the data library point of view this means that the Nordheim Numerical Method (Nordheim, 1961) dealing with the resonance treatment was used during the NJOY

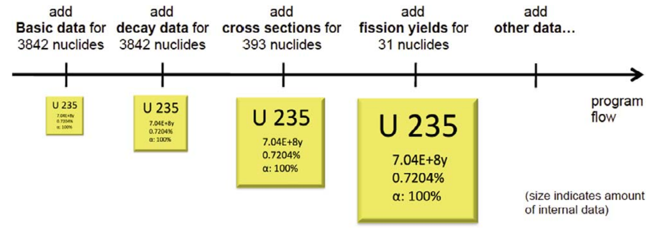

Fig. 2. The nuclear data library generation process.

calculation. As a result a two-dimensional user-defined table depending on fuel temperature and so-called background cross sections is generated and interpolated for actual values during the simulation (here: 5 temperatures, 20 background cross sections according to the MUPO scheme of the former codes). For condensing the point wise cross sections into fine group cross sections by NJOY (so far the 43 energy group MUPO structure (Nyffenegger and Schlösser, 1964), a typical HTR weighting spectrum or a specific design dependent spectrum from another code has to be applied.

2.2. Data model and input/output concept

All input and output data is stored in an object-oriented, hierarchical data model where data is organized in objects reflecting the structure of the physics problems (e.g. *Material, CoatedParticle, FuelElement, Node, Batch, Mesh*). Fig. 3 shows a code excerpt for a typical pebble fuel element. The object-oriented programming approach makes the code more readable and thus increases maintainability and extensibility. HCP also makes use of basic physics objects like *Energy* or *Mass* with corresponding units, so no hidden unit conversions are performed in the source code itself.

Another important aspect of the data model is to keep it up-to-date regarding the applied techniques to preserve the applicability and extensibility for the future. One example is the introduction of smart pointers from the C++11-standard, which helps to reduce the effort for memory management enormously.

As a central part of the HCP, a new data input and output concept was designed and implemented. Reactor models in HCP are defined in XML (*Extensible Markup Language*). This allows for comprehensive input structures matching the object-oriented data model hierarchy. XML validation schemes help to assure the consistency, correctness and completeness of the input even before the actual simulation run. A high level of readability enables fast and robust modeling and can help to reduce maintenance costs.

In addition to the new input concept, a new generic data output concept is being developed. This concept provides a variety of output options for the HCP (see Fig. 4).

The basic idea is that in principle all data can be kept in memory at runtime. Applying user-defined filters can then reduce the amount of data. All data passing the filter is kept so that the user can access it after

```
FuelPebble pebble;

MatCell outerLayer, innerLayer;

outerLayer.setRadius(3.0);
pebble.addMatCell(outerLayer);

innerLayer.setRadius(2.5);
pebble.addMatCell(innerLayer);


Particle coatedParticle;
. . .

ParticleComposition cpComposition;
cpComposition.addFraction(coatedParticle, 1.0);

pebble.addParticleComposition(cpComposition);
```

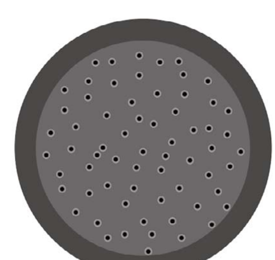

Fig. 3. Data model versus real-world object: A fuel element definition in the HCP source code.