HTR Code Package (HCP) General Introduction

Dr.-Ing. André Xhonneux

Joint IAEA-ICTP Workshop on Reactor Physics, Thermal Hydraulics and Plant Design Engineering of Small Modular Reactors

13 April 2025, International Centre for Theoretical Physics (ICTP), Trieste

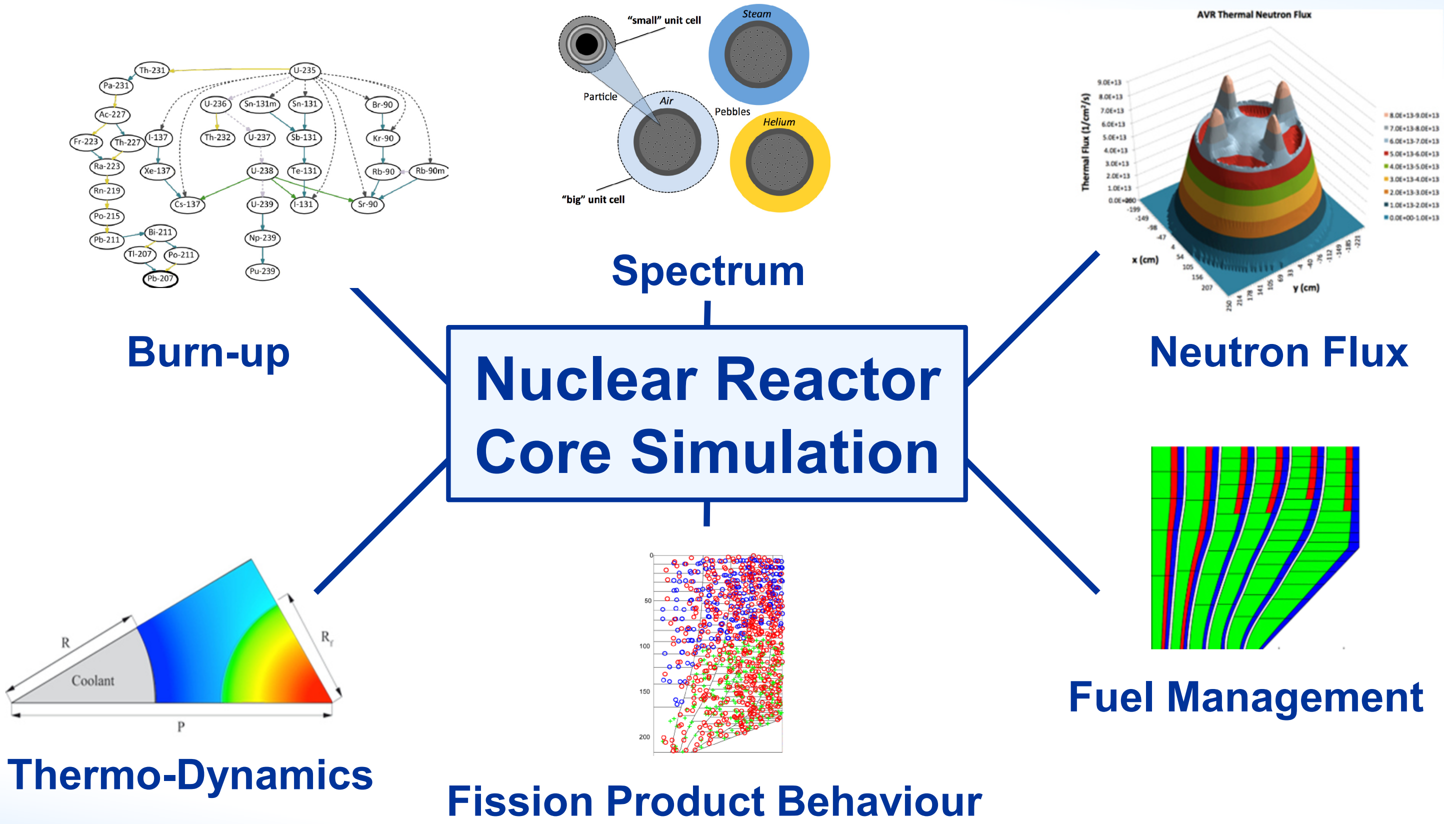
Outline

- Introduction to HTR Code Package (HCP)
- HCP Physics Modules
- HCP Prototype
- Conclusion

Outline

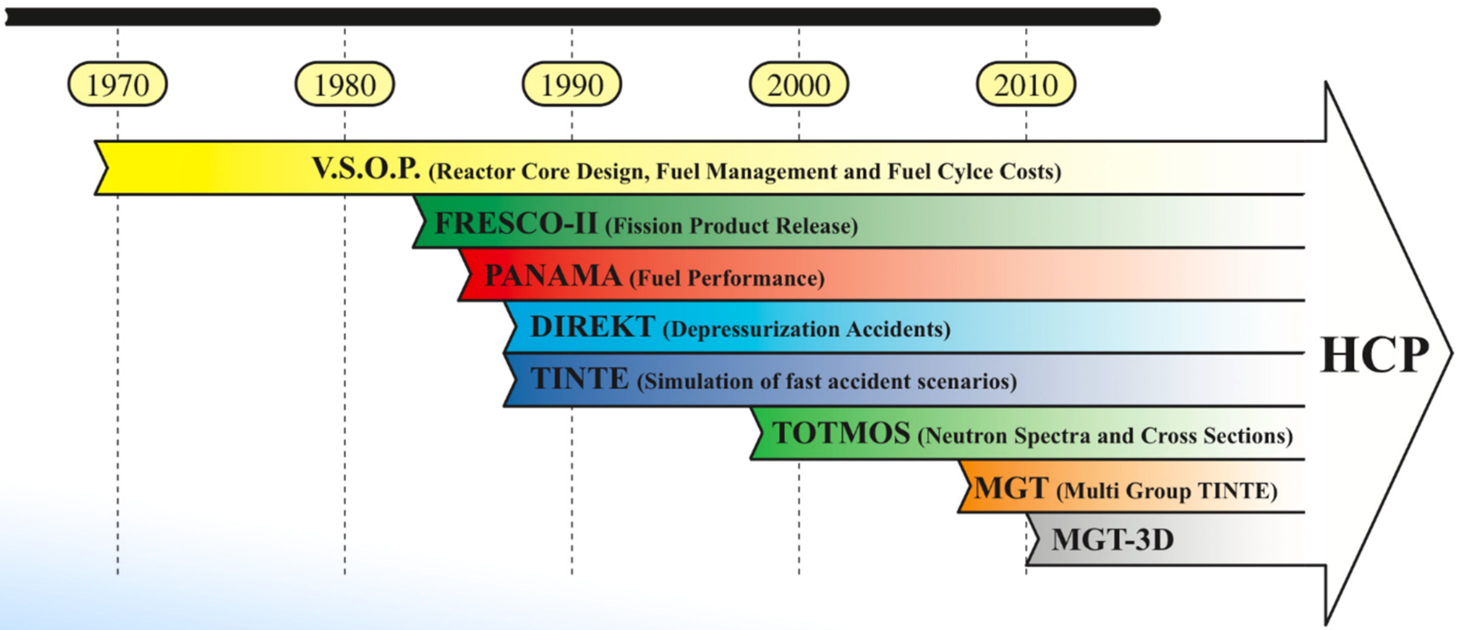
- Introduction to HTR Code Package (HCP)
- HCP Physics Modules
- HCP Prototype
- Conclusion

Aspects of HTR Simulations



HTR Legacy Codes at FZJ

- More than **40 years** of HTR research at FZ Jülich and RWTH Aachen (Germany)
- Recently used HTR codes are worth **thousands of person months**
- Applied for the following reactors / concepts : AVR, THTR, HTR-MODUL, PBMR-400, HTR-PM, HTR-10, …



HCP development

- HCP development was **initiated in 2009 and was ended at FZJ** with the last PhD thesis within this area in 2016

- The **original HCP core development** team had **7 members** (independent development effort for the legacy codes contained!)

- Version developed at FZJ made available to **IAEA member states via IAEA's ONCORE platform**[1]

- HCP currently further developed by **Technical University of Munich (TUM)** and **Becker Technologies**

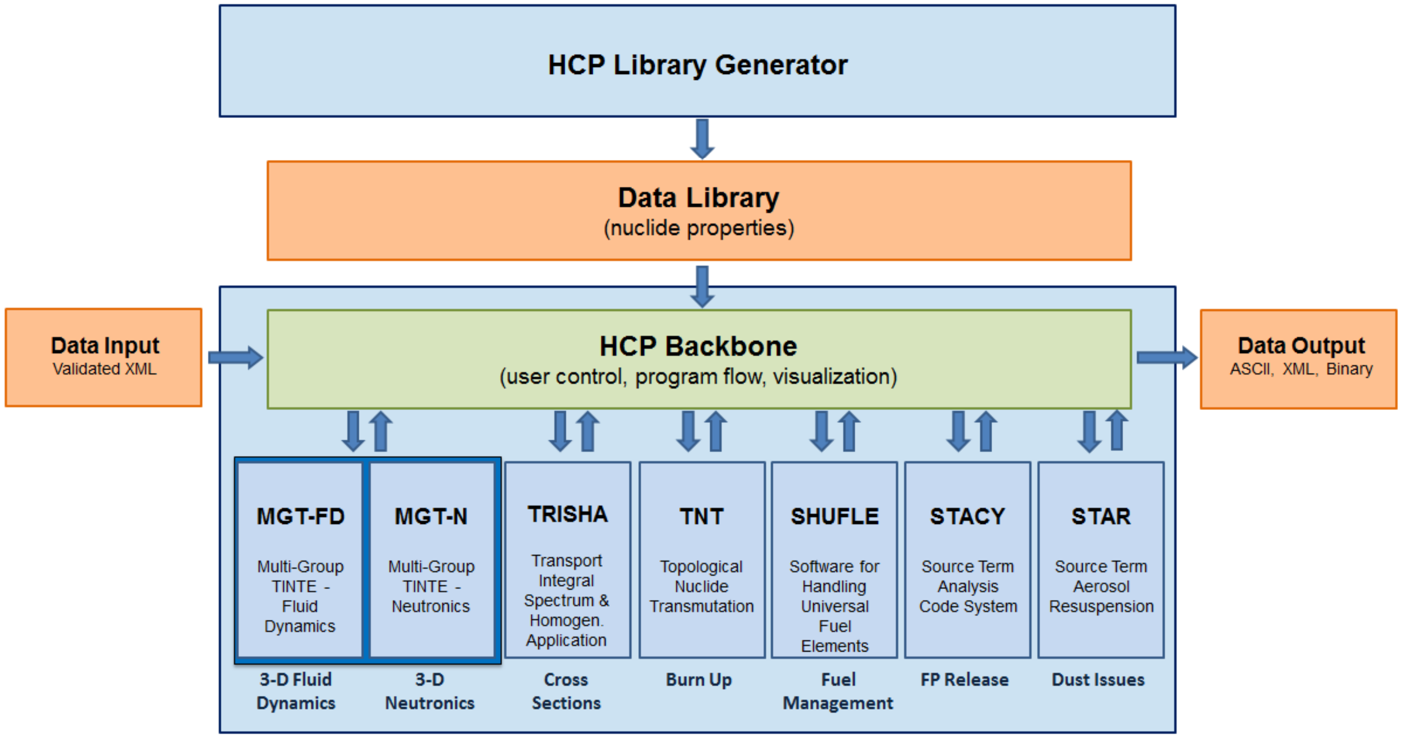
[1]**Open-source Nuclear Codes for Reactor Analysis (ONCORE)**
https://www.iaea.org/topics/nuclear-power-reactors/open-source-nuclear-code-for-reactor-analysis-oncore

Why an HTR Code Package?

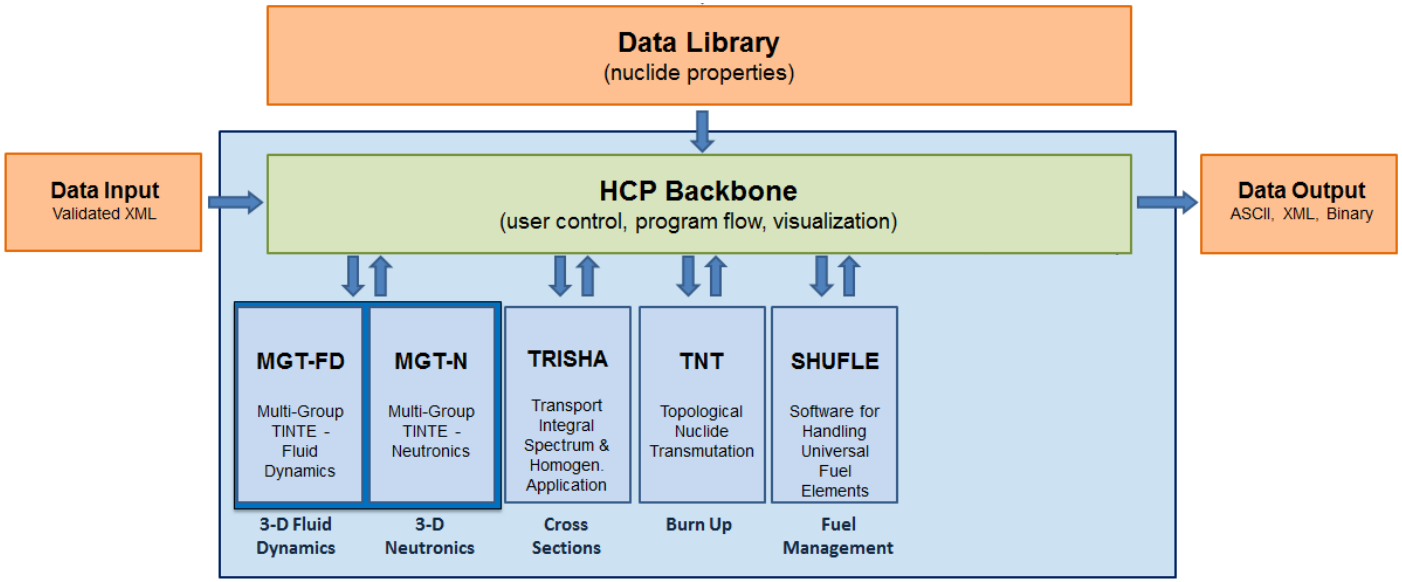
- Overcome **drawbacks and limitations** of individual **legacy codes**
- **Conservation of knowledge** in a contemporary way for future demands
- **Speed up implementation times** for necessary (future) model extensions
- **Reduce software maintenance costs** by avoiding code duplication
- **Increase the numerical stability** by applying recent Fortran coding standards

Design, implementation, verification and validation of software for the simulation of HTR reactor cores applying the modern programming techniques and standards

HTR Code Package (HCP)



HTR Code Package (HCP) Prototype



Available @ Open-source Nuclear Codes for Reactor Analysis (ONCORE)
https://www.iaea.org/topics/nuclear-power-reactors/open-source-nuclear-code-for-reactor-analysis-oncore

Current Code Capabilities

1. **HCP-Proto: Reactor under normal operating conditions**

   Combining neutronics (MGT-N) + Spectrum (TRISHA) + Burnup (TNT) and Fuel management (SHUFLE) simulations in an alternating way

2. **Pure neutronics cases**

3. **Pure fluid dynamics cases**

   Steady-state and transient simulation of experiments as well as reactor models (although actual reactor state has to be provided in the input, result of 1. does not work currently)

4. **In preparation by Becker Technologies:**
   Fuel performance and fission product release simulations

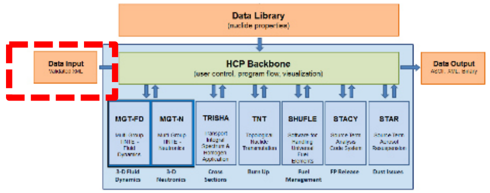
   Spherical fuel element and compacts (based on user defined temperatures and nuclide inventories, no internal coupling to other modules e.g. TNT or MGT-FD yet)

5. …

HCP Input Concept - Example

Definition of a fuel element with homogenized densities:

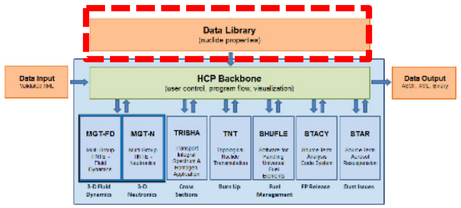
```xml
<FuelElements n="1">
  <FuelElement label="GLE-4" geo="spherical" nMatCells="2" id="1">
    <MatCells n="2">
      <MatCell type="Layer" material="A3-3" fuelZone="false" geoObject="2" id="2">
        <MatCell type="Layer" material="A3-3" fuelZone="true" particleComposition="2" geoObject="1" id="1"/>
      </MatCell>
    </MatCells>
    <NuclideVector n="11">
      <Amount nuclide="U-238"  lib="endfb70" unit="mole"> 3.47053128E-02 </Amount>
      <Amount nuclide="U-235"  lib="endfb70" unit="mole"> 3.13970186E-03 </Amount>
      <Amount nuclide="O-16"   lib="endfb70" unit="mole"> 7.56901007E-02 </Amount>
      <Amount nuclide="B-10"   lib="endfb70" unit="mole"> 1.90584974E-06 </Amount>
      <Amount nuclide="B-11"   lib="endfb70" unit="mole"> 7.71964813E-06 </Amount>
      <Amount nuclide="C-12"   lib="endfb70" unit="mole"> 1.62227385E+01 </Amount>
      <!--- Xenon amount has been added to trigger Xe-Equilibrium Calculation -->
      <Amount nuclide="Xe-135" lib="endfb70" unit="mole"> 1.04756172E-10 </Amount>
      <Amount nuclide="Si-28"  lib="endfb70" unit="mole"> 7.63178454E-02 </Amount>
      <Amount nuclide="Si-29"  lib="endfb70" unit="mole"> 3.86429857E-03 </Amount>
      <Amount nuclide="Si-30"  lib="endfb70" unit="mole"> 2.56516664E-03 </Amount>
      <!--- Helium amount of surrounding volume of pebble has been added -->
      <Amount nuclide="He-4   "  lib="endfb70" unit="mole"> 4.53093055E-03 </Amount>
    </NuclideVector>
  </FuelElement>
</FuelElements>
```

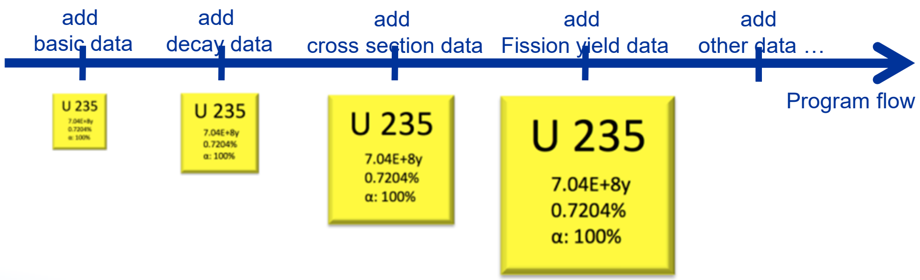


HCP Data Library

One library for all HCP Physics Modules, therefore library contains:

- Basic nuclide data (e.g. Z, A, M, molar mass, ...)
- Decay data (half life, decay modes, branching ratios, Q values, ...)
- Microscopic cross sections (fission, absorption, transport, ...)
- Independent SF and (n,f) yields (for up to 3 energy ranges)
- Resonance data
- Scattering matrices (e.g for graphite, H_2O, D_2O, BeO,...)
- (Diffusion constants for FP release calculations)
- (Input parameter for graphite dust transport calculations)
- ...



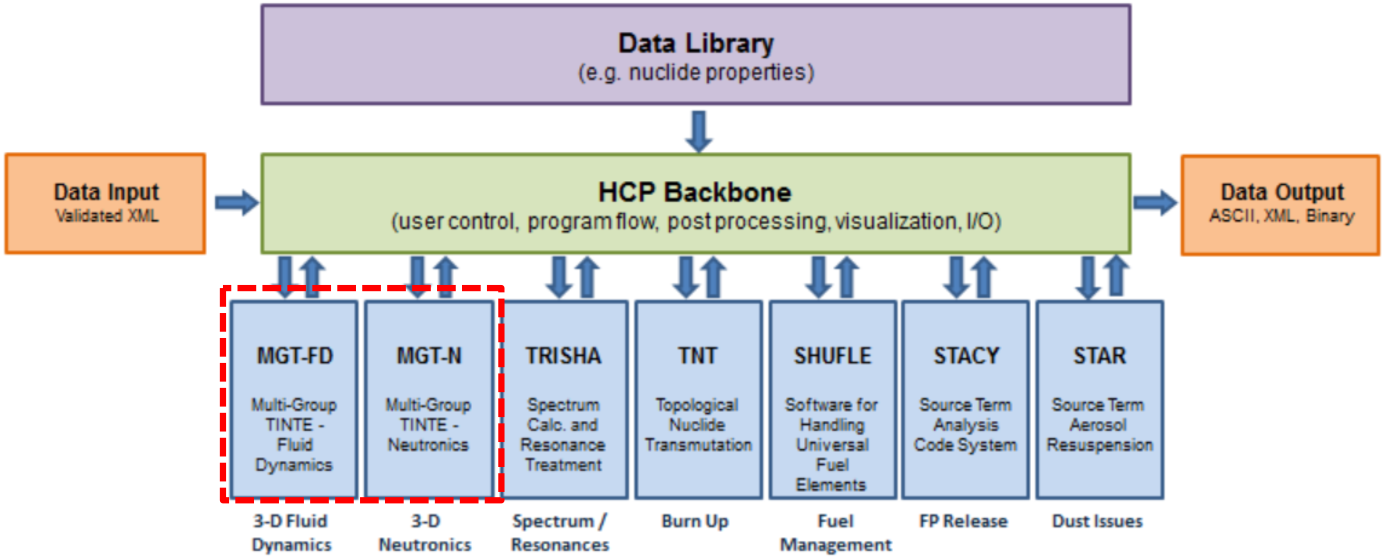


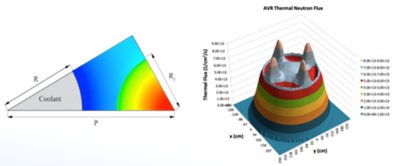
Outline

- Introduction to HTR Code Package (HCP)
- HCP Physics Modules
- HCP Prototype
- Conclusion

HTR-Physics Modules

MGT-FD + MGT-N





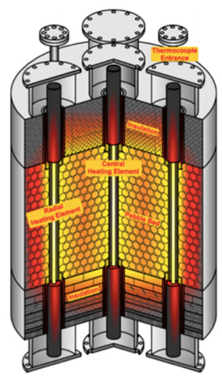
MGT-FD + MGT-N: Reactor Dynamics

- MGT-FD + MGT-N are based on MGT-3D
  (the 3D successors of TINTE)
  (TIme Dependent Neutronics and Temperatures)
  (TINTE = **TI**me dependent **N**eutronics and **TE**mperatures)

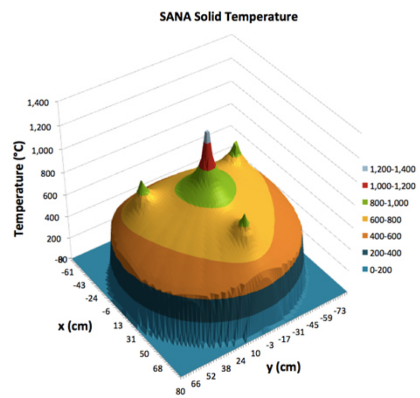
- Simulation of fast accident scenarios of HTR like TWCR, DLOF-C, TECR, water and air ingress and other dynamic processes (e.g. load change)

- Typical use-case: **design basis accidents (DBA)**

Cross-sectional view



HCP Result

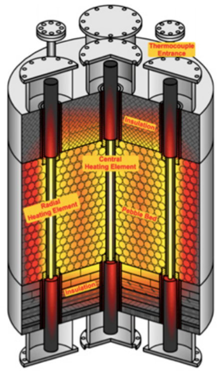


MGT-FD + MGT-N: Reactor Dynamics

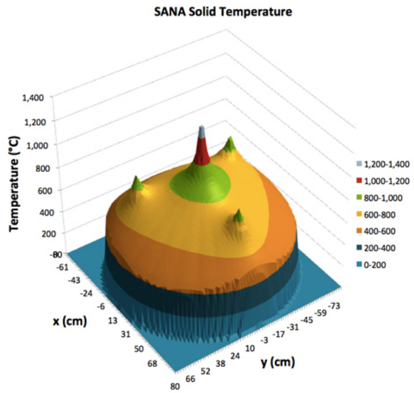
IAEA

- Time dependent neutronics and fluid dynamics for 3D reactor models
- Feedback between neutronics and fluid dynamics
- Neutron diffusion in r/z/phi geometry coupled with thermal fluid dynamics (pseudo x/y/z)
- I/Xe dynamics
- Basic graphite corrosion chemistry
- Local and non-local nuclear heat sources
- Decay heat calculation according to DIN standard (German standard)
- Nuclear and/or non nuclear (external) heat sources
- Heat transport by conductivity, convection and radiation in gaps
- Heterogeneous temperature calculations (temperature distribution within fuel)
- 0-D neutron Spectrum calculation code with up to 43 energy groups
- Gas flow and mixing
- Heat transport

Cross-sectional view

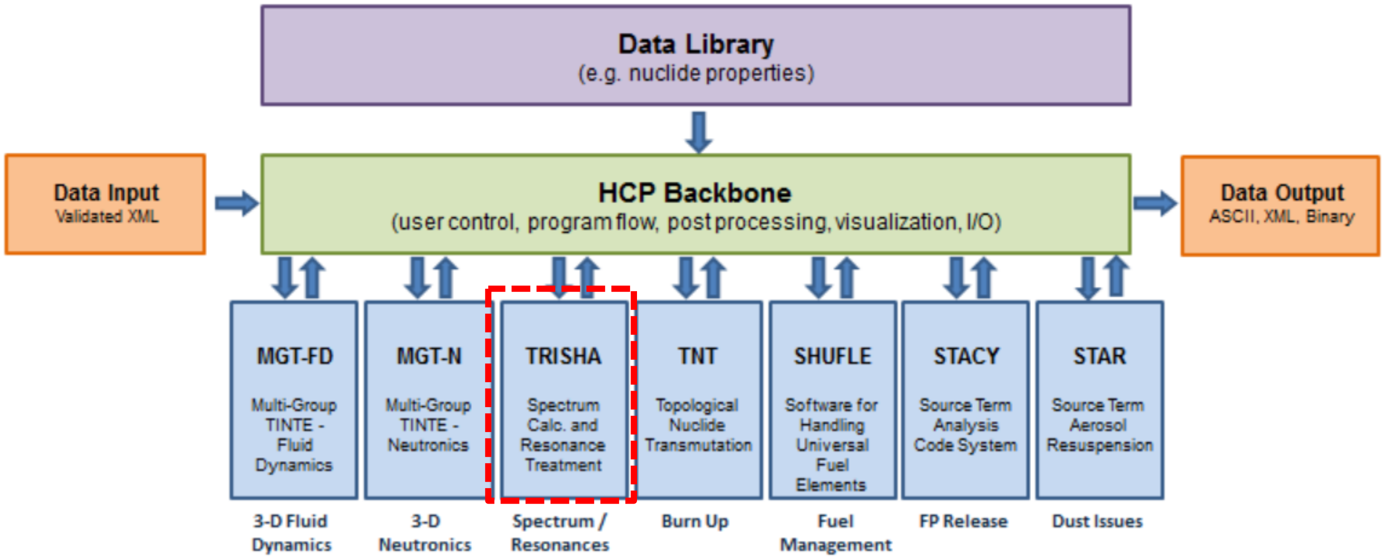


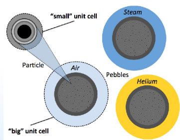
HCP Result



HTR-Physics Modules

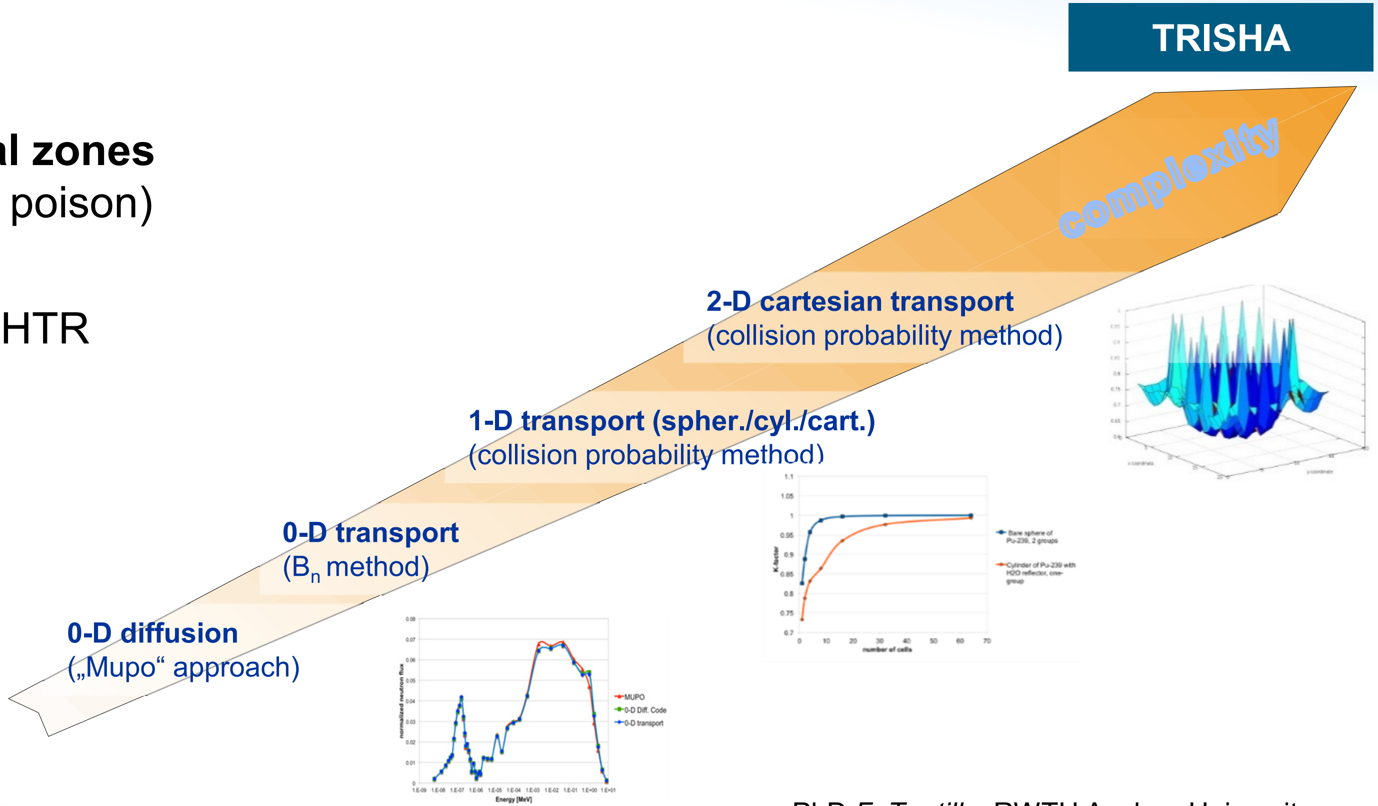
neutron TRansport Integral Spectrum and Homogenisation Application (TRISHA)





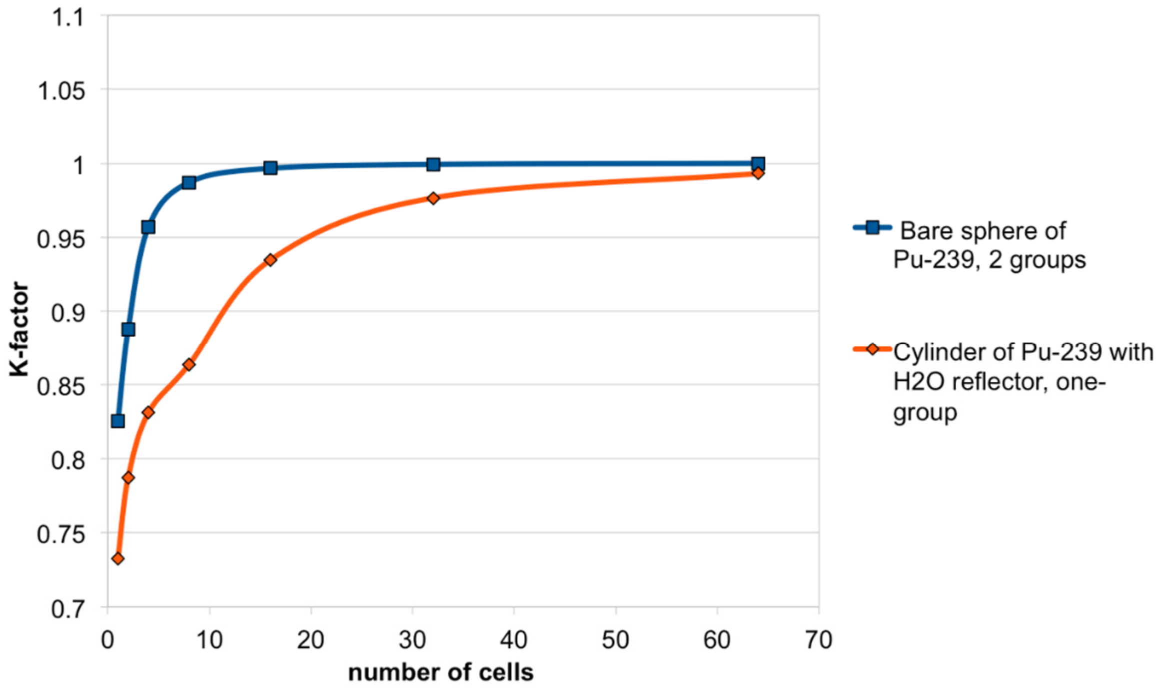
TRISHA - Introduction

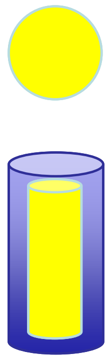
- **Different methods for different core regions**
- **Better treatment of spectral zones** (e.g. reflector/core, burnable poison)
- Applicable to both LWR and HTR



PhD *F. Tantillo*, RWTH Aachen University

TRISHA - 1D transport solvers (k_{eff})

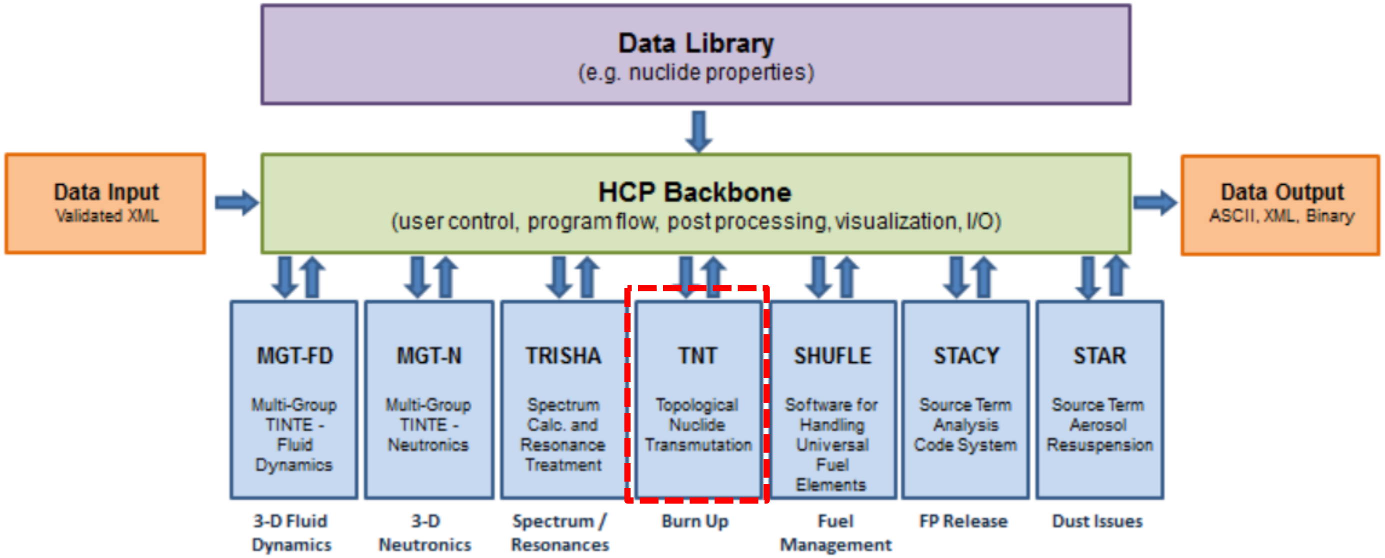


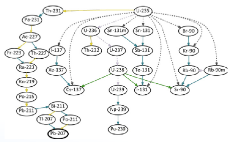


- Analytical criticality benchmarks (data from LANL)
- PhD *F. Tantillo*, RWTH Aachen

HTR-Physics Modules

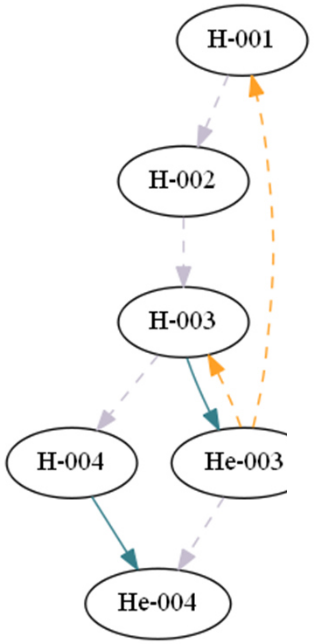
Topological Nuclide Transmutation (TNT)





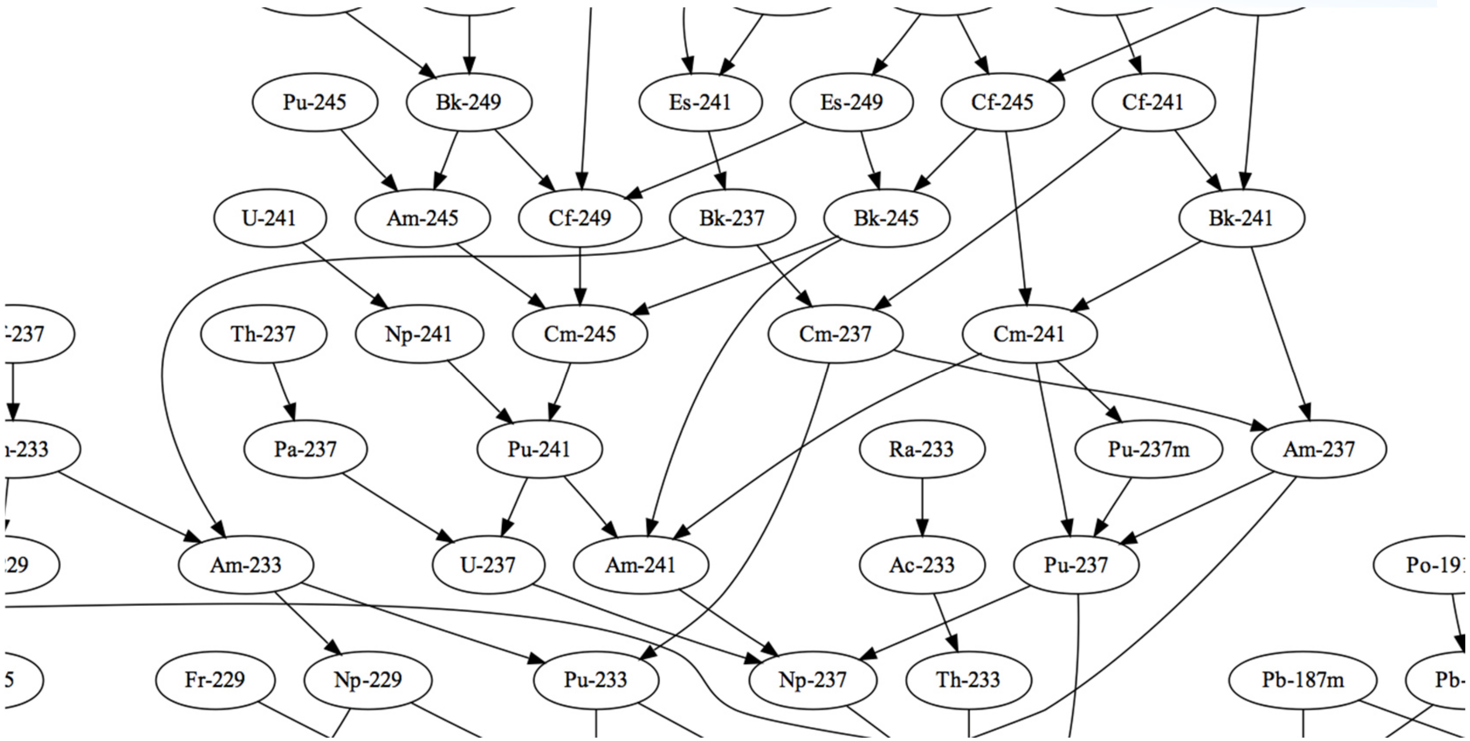
TNT - Introduction

- HCP uses a new developed code called TNT (**T**opological **N**uclide **T**ransformation) for decay and burn-up calculations
- New approach: Apply graph theory (via C++ library boost): nuclide objects are vertices (nodes) and transmutation rates are egdes (arrows) (=> generic programming)
- Advantages:
  - solver exploits the topology information by working along the real nuclide chain (e.g. less iterations needed)
  - code will be much faster compared to a solution with I/O coupling using ASCII files (as done so far in most cases)
  - new faster matrix solver: Padé approximation in addition of matrix exponential method
  - online decay heat calculation is feasible due to parallelization



Usage of Nuclide Graph in TNT

Nuclide graph of ENDF/B VII decay library (3842 nuclides):



Verification of TNT

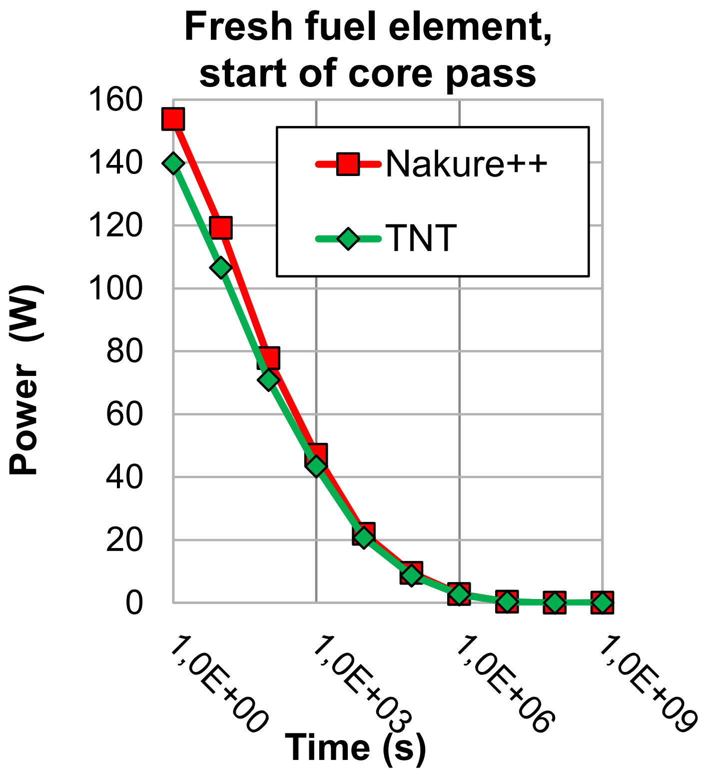
Example: U-235 fission (results, both with ORIGEN 2.2 library):

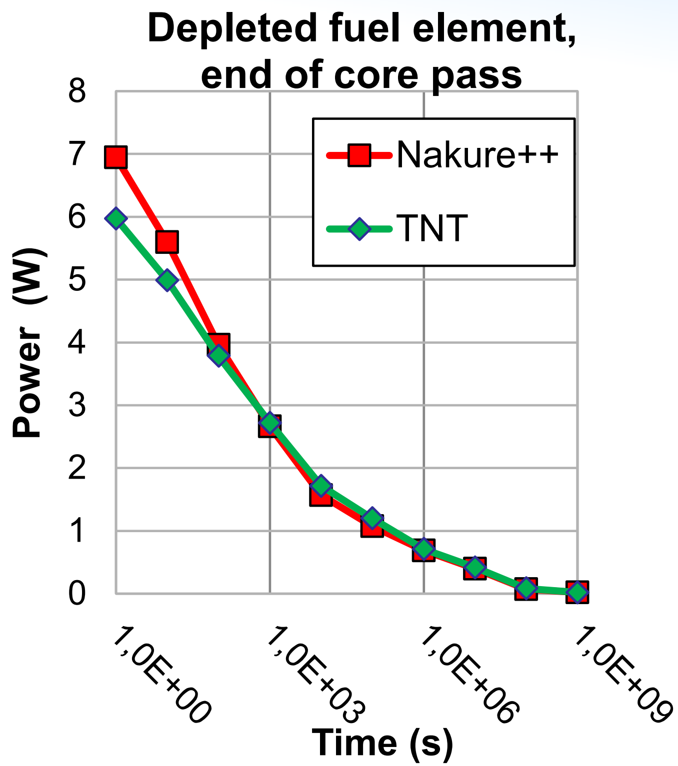
100 mol U-235, Φ=1.0e+13 1/cm2/s, t = 1.0 y

TNT 1.0 vs. ORIGEN 2.2

Nuclide	ZAM	Rate	Amount	
As-081	330810	2.1661e-02	4.3461e-08	4.3456E-08
Cd-119	481190	1.2290e-03	2.1274e-08	2.1271E-08
Cs-141	551410	2.7726e-02	7.4884e-07	7.4875E-07
Ra-224	882240	2.1922e-06	4.5926e-14	4.5420E-14
Th-232	902320	7.4000e-11	4.2814e-08	4.2870E-08
U-233	922330	5.7877e-09	2.6722e-10	2.6761E-10
U-235	922350	6.8088e-09	8.0677e+01	8.0667E+01
U-236	922360	5.2001e-11	2.7959e+00	2.7977E+00
U-238	922380	2.7000e-11	7.9653e-06	7.9760E-06
U-240	922400	1.3655e-05	7.0582e-18	7.0677E-18

TNT: NAKURE vs. TNT (single HTR pebble)

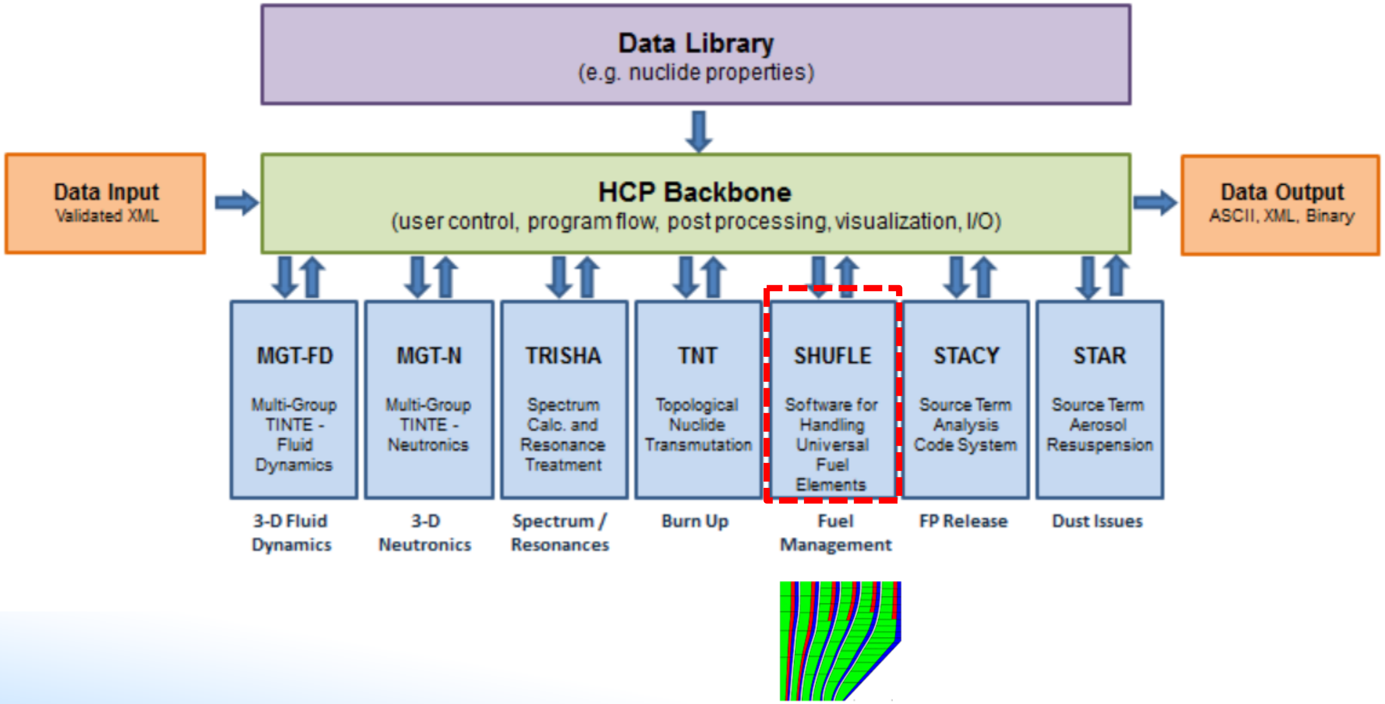




OTTO, burn up target = 100 MWd/kg (Benchmark case documented in DIN standard)

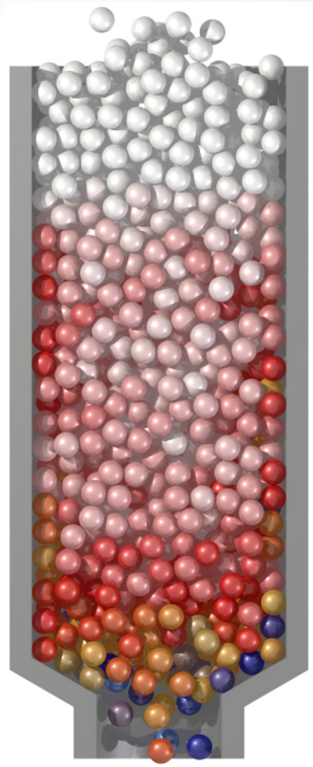
HTR-Physics Modules

Software for Handling Universal Fuel Elements (SHUFLE)



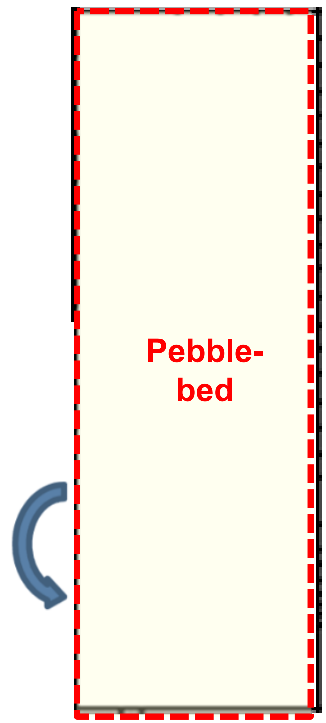
SHUFLE - Introduction

- Module SHUFLE (**S**oftware for **H**andling **U**niversal **F**ue**l** **E**lements) performs fuel management
- Code does not treat single spheres (like a Discrete Element Method) but depicts the movement of lumped volumes of pebbles with same properties
- Two modes are offered:
  - Flow tube model
  - Seepage model → not supported to be runned yet with other HCP modules



Fuel Management Modes in SHUFLE

- Pebble-bed volume is divided in flow channels (limited by flow lines in radial direction)
- Each channel is sub-divided in nodes which contain the same solid volume
- Continuous shuffling is depicted by moving content of each node to the node below recursively, starting at the lower



New Aspects with Respect to Legacy Codes

- Modeling of discharge tube
- Space and time dependent filling factor / densification of pebble bed
- Tracking of single batches (multi fuel pass fuel strategy)
- Simulation of conical pebble heap(s)

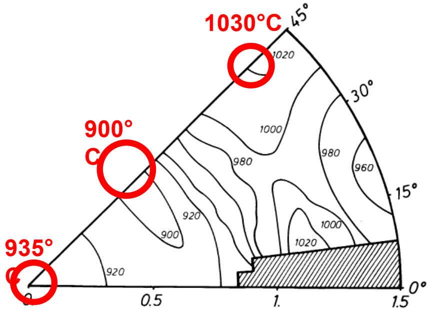


Fig.: Fuel temperature distribution [2]

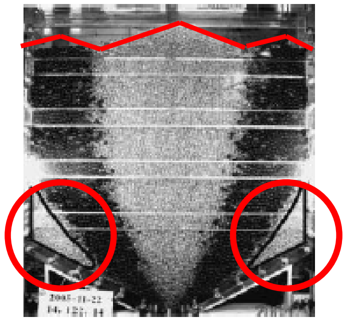
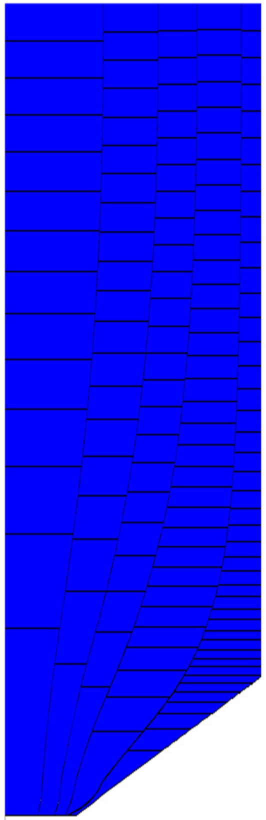


Fig.: Stagnant zones [1]

[1] Yang, X.: *Experimental Investigation on Feasibility of Two-Region-Designed Pebble-Bed High-Temperature Gas-Cooled Reactor*, INET, 2008
[2] Verfondern, K.: *Numerische Untersuchung der 3-dimensionalen stationären Temperatur- und Strömungsverteilung im Core eines Kugelhaufen-Hochtemperaturreaktors*, 1984

HCP Module: SHUFLE

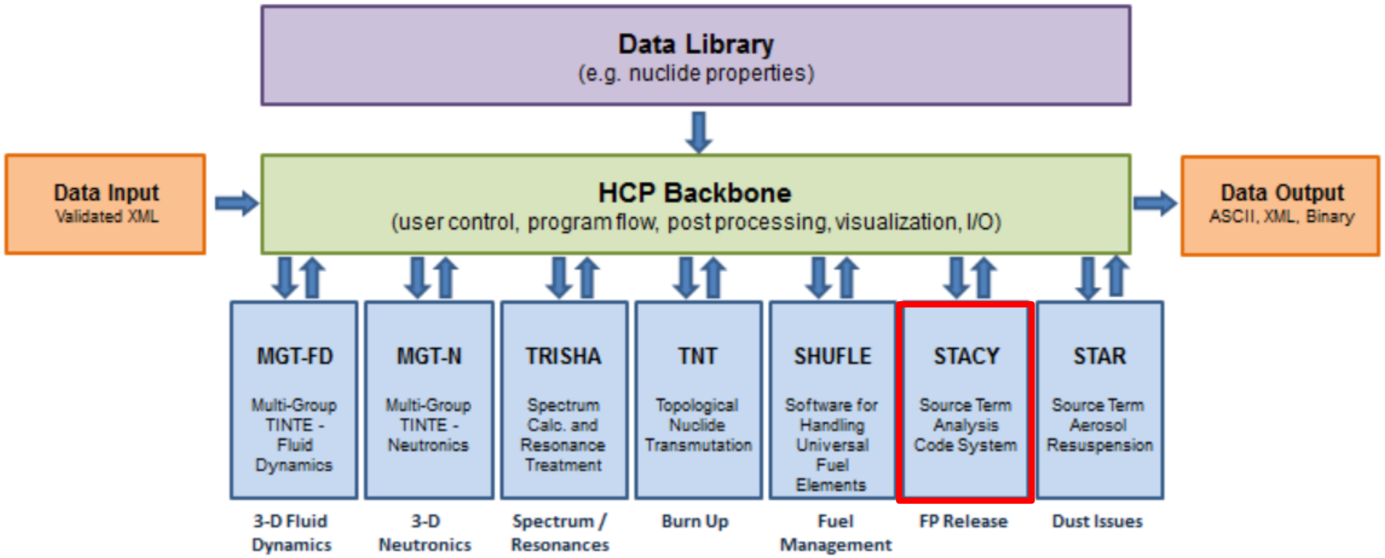
(Extended Flow Tube Model)

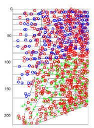


(pebble flow visualised with MAVIS, not drawn to scale,
calculation zone has been mirrored for better illustration)

HTR-Physics Modules

Source Term Analysis Code System (STACY)



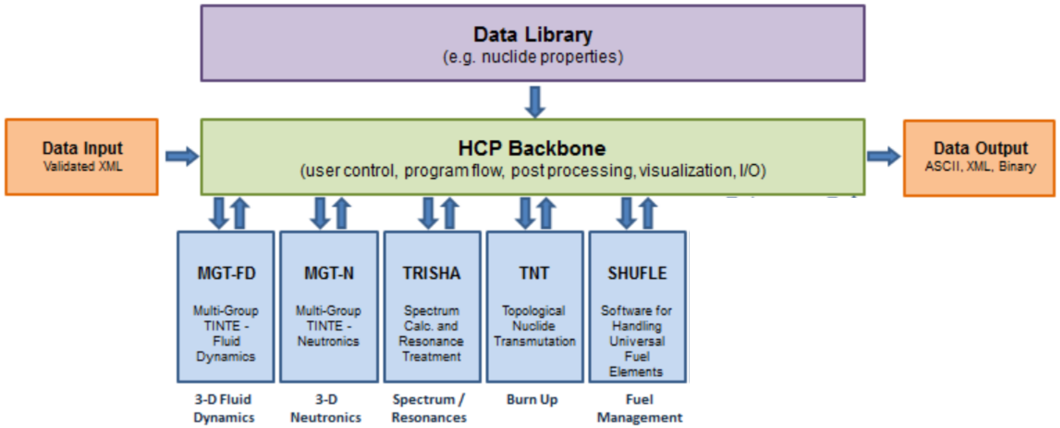


Outline

- Introduction to HTR Code Package (HCP)
- HCP Physics Modules
- HCP Prototype
- Conclusions

HCP Prototype

- HCP Prototype couples
  MGT-FD, MGT-N, TRISHA, TNT and SHUFLE
- Simulation of **sequence of steady state** with one integrated code (replacing capabilities of VSOP)

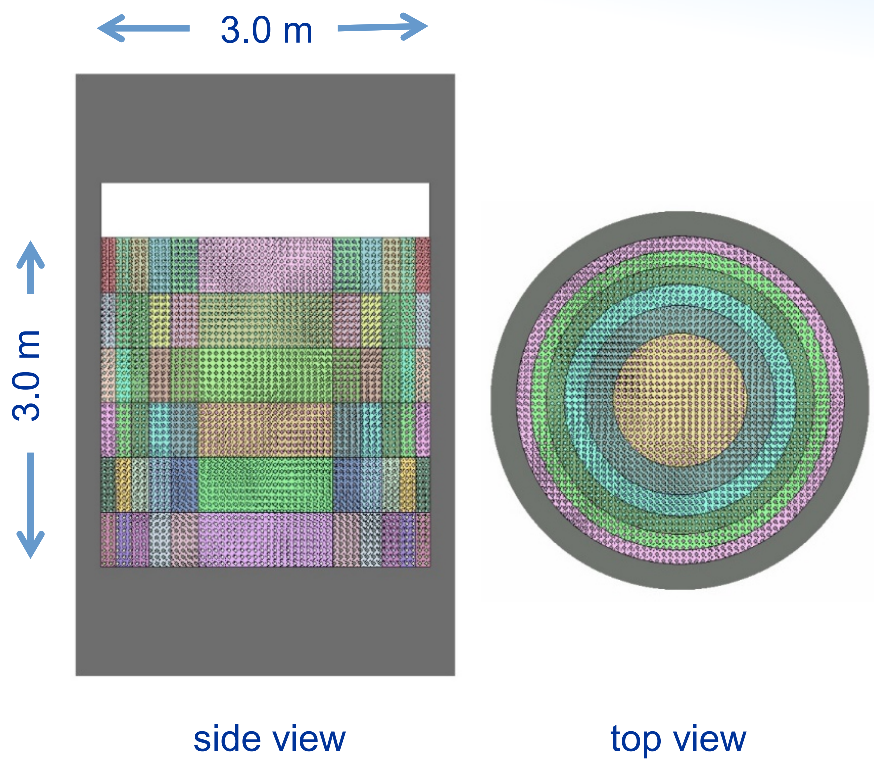


Benchmark: HCP Prototype vs. SERPENT

Simplified model for a pebble bed HTR

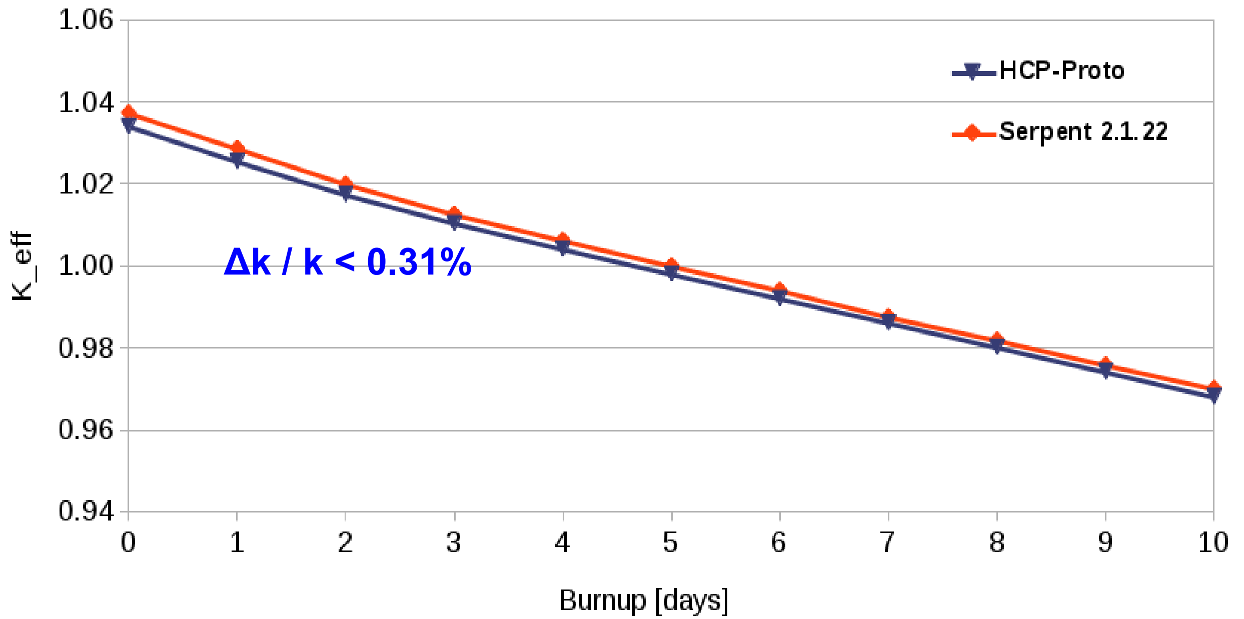
- **Reactor Model:**
  - About 100,000 pebbles
  - 36 distinct burn up zones
  - Graphite reflector & Void
  - LTI-TRISO coated particles
  - T = 900 K, steady state
  - No control rods

- **Simulation Case:**
  - MGT + TRISHA + TNT
  - 10 burn up steps
  - Criticality calculation



HCP Prototype vs. SERPENT – Results

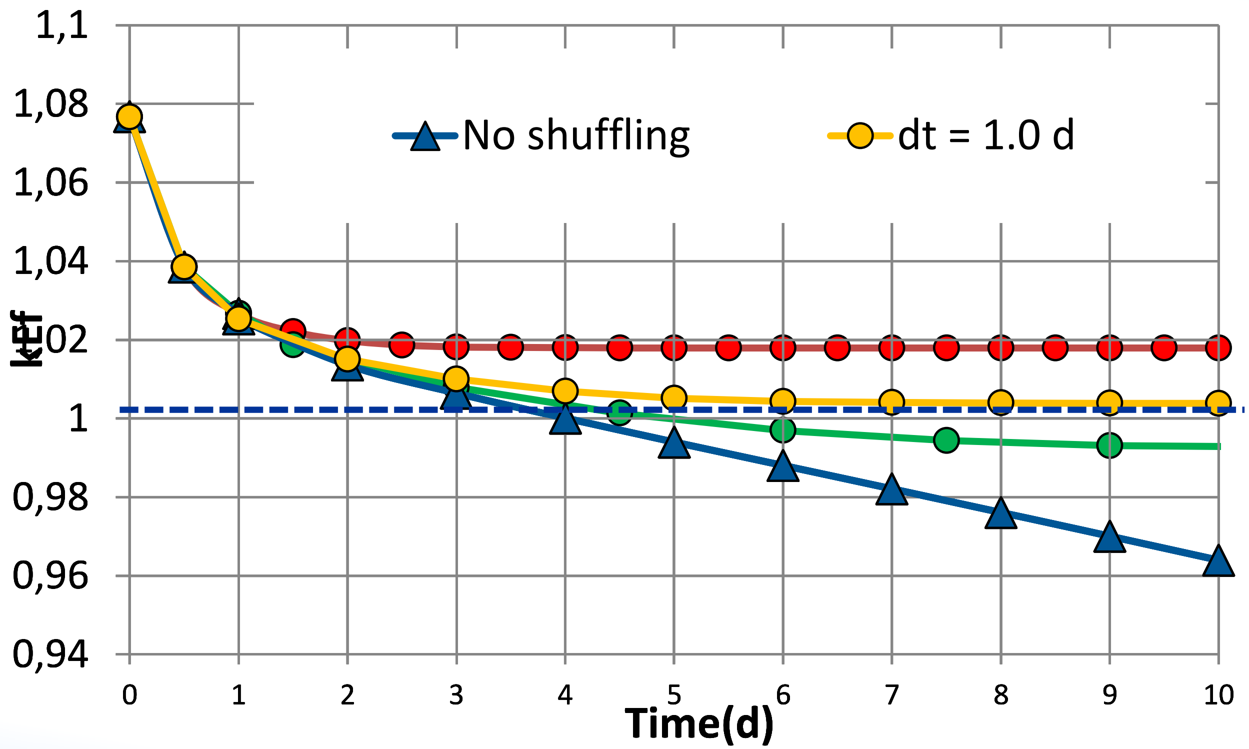
Depletion benchmark incl. Xenon equilibrium



HCP simulation is in very good agreement with SERPENT

HCP Prototype

Depletion benchmark incl. OTTO-Shuffling



dt: time between shuffling steps

Outline

- Introduction to HTR Code Package (HCP)
- HCP Physics Modules
- HCP Prototype
- Conclusions

Conclusions

- To overcome limitations of FZJ legacy codes, these individual programs were integrated at FZJ into a consistent HTR code package (HCP)
- This allows to include new physics like fission product release, neutron spectrum, advanced pebble flow and dust transport models
- For the HCP modern programming techniques and standards (e.g. XML, C++) are applied
- The HCP is a modern HTR physics simulation code (treating both normal operation and accident conditions within one integral framework)
- User friendly consistent input concept based on XML
- An object-oriented data model is used, allowing for future extensions

Conclusions (cont'd)

- Application of object-oriented programming (OOP) encapsulates huge data amounts from the physics and increases readability
- Usage of XML and serialization reduces potential errors for user input allows for an easy data handling
- Heavy use of validated libraries (e.g. boost, Eigen, etc.) reduces amount of own source-code and potential bugs
- Using unit and regression tests reduces the change of including new bugs and keeps the prediction power of codes
- **Technical University of Munich (TUM) and Becker Technologies further developing HCP, closing gaps in the code basis as well as documentation**

IAEA
International Atomic Energy Agency

Dr.-Ing. André Xhonneux
E-Mail: andre.xhonneux@gmail.com

Thank you! Let's stay in touch.