Coupling TRANSURANUS with Monte Carlo and subchannel codes for pin-by-pin depletion in LWR

M. García, D. Ferraro, R. Tuominen, A. Gommlich
McSAFE EU H2020 project

- **Objectives:**
  - Improve the prediction of local safety parameters through high-fidelity.
  - Develop multiphysics tools based on Monte Carlo particle transport, subchannel thermalhydraulics and fuel-performance analysis.
  - Solve steady-state, depletion and transient problems.
  - Optimize the codes for massive problems such as full-core burnup.
  - Validate with PWR and VVER plant data.

- **Participants:**
  - KIT (SUBCHANFLOW).
  - JRC, HZDR (TRANSURANUS).
  - VTT (Serpent2).
  - CEA (Tripoli), DNC (MCNP), AMEC (MONK), NRI, KTH.
  - EKK, CEZ, EdF.

- Continuation of the NURESAFE and HPMC projects.
**Serpent2-SCF-TU**

- **Codes:**
  - Serpent2: continuous-energy Monte Carlo particle transport.
  - SUBCHANFLOW: MATRA- (COBRA-) based subchannel code.
  - TRANSURANUS.

- **Objectives:**
  - Perform fully coupled pin-by-pin depletion calculations.
  - Optimize the system for High Performance Computing (HPC).
  - Develop a suitable modelling approach (pin-by-pin, hot channel) for full-core.
  - Validate with PWR and VVER experimental data from the industry partners.

- **Coupling approach:**
  - Pin-level feedback, fully coupled calculation scheme.
  - TRANSURANUS replaces the rod solver of SUBCHANFLOW.
  - Object-oriented design.
  - Mesh-based (geometry agnostic) field exchange.
  - Pre- and post-processing capabilities for PWR and VVER reactors.
Object-oriented coupling

- Object-oriented design:
  - **Advantages:**
    - The codes are kept completely separate and maintainability is enhanced.
    - Suitable model for collaborative development projects.
    - The coupling through the supervisor is flexible and somehow generic.
Mesh-based feedback

- Serpent2:

Multiphysics interfaces:
- Superimposed meshes.
- Not linked directly to the tracking geometry.
- Define temperatures and densities and tally power.
Mesh-based feedback

- **SUBCHANFLOW:**

![Diagram showing mesh-based feedback with coolant and fuel models.](image)
Mesh-based feedback

- TRANSURANUS:

  Multiphysics mesh:
  - Used to gather results and set input variables.
  - Contains only the fuel rods.
Calculation scheme

- Semi-implicit depletion scheme.
- Solution of the Bateman equations both in TU and Serpent2.
- The only Serpent2 data used in TU is the power distribution.
- The TU neutronic model is still used.
- SCF does not model the rods.
Developments on TRANSURANUS

- No changes to the physical models.

- Code structure:
  - Kernel implemented as a library for flexible execution.
  - C++ wrapper to define the API used in the coupling scheme.
  - Support for multiple rods.
  - An unstructured mesh is used to gather results and handle input variables.

- Input:
  - Coolant boundary conditions ($h_{\text{clad-cool}}$, $T_{\text{cool}}$, $p$) from SUBCHANFLOW.
  - Linear heat rate from Serpent2.

- Output:
  - Fuel temperature to Serpent2.
  - Gap conductance, gap width, fission gas release (Xe, Kr, etc).

- Optimization:
  - MPI parallelization using domain decomposition.
  - Dynamic memory allocation (~12GB → ~1.5GB for a VVER-1000 fuel).
Results: steady state


### Difference: Serpent-SCF-TU – Serpent-SCF

<table>
<thead>
<tr>
<th>Field</th>
<th>Maximum difference</th>
<th>Average difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power (W)</td>
<td>19.83 (2.52%)</td>
<td>5.24 (0.48%)</td>
</tr>
<tr>
<td>Coolant density (kg/m³)</td>
<td>6.08 (0.89%)</td>
<td>1.06 (0.15%)</td>
</tr>
<tr>
<td>Coolant temperature (°C)</td>
<td>2.31 (0.39%)</td>
<td>0.42 (0.07%)</td>
</tr>
<tr>
<td>Fuel temperature (°C)</td>
<td>46.36 (5.51%)</td>
<td>24.22 (3.36%)</td>
</tr>
</tbody>
</table>
Results: steady state

- VVER: 30AV5 VVER-1000 fuel-assembly type.

<table>
<thead>
<tr>
<th>Field</th>
<th>Maximum difference</th>
<th>Average difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power (W)</td>
<td>22.12 (2.90%)</td>
<td>4.61 (0.46%)</td>
</tr>
<tr>
<td>Coolant density (kg/m³)</td>
<td>0.43 (0.06%)</td>
<td>0.09 (0.01%)</td>
</tr>
<tr>
<td>Coolant temperature (°C)</td>
<td>0.17 (0.03%)</td>
<td>0.04 (0.01%)</td>
</tr>
<tr>
<td>Fuel temperature (°C)</td>
<td>28.86 (3.62%)</td>
<td>11.75 (1.73%)</td>
</tr>
</tbody>
</table>
Results: depletion

- VVER: 30AV5 VVER-1000 fuel-assembly type (t = 60 days).

Runtime: ~48hs in 40x20 cores for a 360-day cycle.
Open issues

- Neutronic feedback to TRANSURANUS:
  - Currently only the linear heat rate is used, the simplest possible feedback.
  - Using power and flux radial profiles not feasible, at least not with the current modelling approach, i.e. having all pins in TRANSURANUS.
  - Other average parameters? Isotope compositions?

- Doppler feedback to Serpent2:
  - Currently volume averaging or empirical formulas are used.
  - Analysis of radial fuel temperature profiles in the near future.

- Modelling approach:
  - With the current approach we’d simulate ~60,000 rods for full-core cases!
  - Hot-channel methodology? Average pin for each fuel-assembly?

- Memory bottleneck for Serpent2 depletion:
  - A full core takes about 2TB for pin-by-pin depletion.
  - Domain decomposition in progress.
Conclusions

- **Current status:**
  - Three code coupling with pin-level feedback implemented.
  - Verification with single-fuel-assembly PWR and VVER steady-state cases.
  - Analysis of single-fuel-assembly depletion cases underway.
  - Optimization towards full-core capabilities in progress.

- **Open issues:**
  - Analysis of the TRANSURANUS side of the depletion scheme, including adding more neutronic feedbacks from Serpent2.
  - Analysis of the Doppler feedback to Serpent2.
  - Optimization of the modelling in TRANSURANUS (and the other codes).

- **McSAFE user group open to new participants!**
Questions? Comments?