



Hands off/on introduction to coupled calculations with Serpent 2.1.27

Serpent UGM, Politecnico di Milano

Sept. 26, 2016

V. Valtavirta

VTT Technical Research Center of Finland

Background

Serpent 2 has been designed for **coupled multi-physics calculations** and there are many features supporting them:

- On-the-fly treatment of temperature effects on cross sections using the TMS treatment¹.
- On-the-fly temperature interpolation of $S(\alpha, \beta)$ data (thermal scattering)².
- Treatment of non-uniform material density distributions³.
- Universal multi-physics interface** for separation of temperature and density data from geometry definition **and** for easy I/O of relevant fields.⁴
- File based (**set comfile**) and POSIX based (**set ppid**) communication for **external coupling** without the need to restart Serpent for each iteration.
- Standardized solution flow** suggested for externally coupled applications.
- Internal, separately licensed, **fuel behavior module FINIX**⁵ for multi-physics applications.

¹T. Viitanen. "Development of a stochastic temperature treatment technique for Monte Carlo neutron tracking." PhD thesis. Aalto University, 2015.

²T. Viitanen and J. Leppänen. "New Interpolation Capabilities for Thermal Scattering Data in Serpent 2." In proc. PHYSOR 2016. Sun Valley, ID, USA, May 2016.

³J. Leppänen. "Modeling of Non-uniform Density Distributions in the Serpent 2 Monte Carlo Code." Nucl. Sci. Eng. **174** (2013), 318. DOI: [10.13182/NSE12-54](https://doi.org/10.13182/NSE12-54). URL: http://www.ans.org/pubs/journals/nse/a_17107.

⁴J. Leppänen, T. Viitanen, and V. Valtavirta. "Multi-physics Coupling Scheme in the Serpent 2 Monte Carlo Code.." Trans. Am. Nucl. Soc. **107** (2012), 1165.

⁵T. Ikonen et al. "Module for thermomechanical modeling of {LWR} fuel in multiphysics simulations." Annals of Nuclear Energy **84** (2015). Multi-Physics Modelling of {LWR} Static and Transient Behaviour, 111–121. ISSN: 0306-4549. DOI: [10.1016/j.anucene.2014.11.004](https://doi.org/10.1016/j.anucene.2014.11.004). URL: <http://www.sciencedirect.com/science/article/pii/S0306454914005842>.

Overview

Coupled multi-physics simulations can be now executed in criticality source, burnup and transient modes.

The purpose of this presentation is to give a primer on coupled calculations with Serpent using either

- The internally coupled [FINIX fuel behavior module](#).
- An externally coupled solver, here the [Minimal Serpent Coupling Script](#).

FINIX

FINIX is a lightweight fuel behavior solver developed at VTT. It is built especially for multi-physics simulations.

FINIX simulates the time-dependent thermal and mechanical behavior of the rod, taking into account the dependence of material properties such as thermal conductivity, thermal expansion, Young's modulus, etc., on various parameters, including temperature and burnup.

The thermal and mechanical solutions are coupled via the gas pressure and conductance of the pellet-cladding gap. The numerical models have been chosen to be computationally inexpensive, but on par with other currently employed one dimensional fuel performance codes.

Developed especially for transients. Currently no models for accumulating burnup, but initialization for transients at non-zero burnup should be done with, e.g., FRAPCON⁶.

FINIX has been validated against the FRAPTRAN fuel performance code as well as experimental Halden reactor data^{7,8}. The comparison shows very good agreement, especially for the temperature distribution, which is the most relevant output for coupled neutronics simulations.

⁶FINIX can read FRAPCON/FRAPTRAN restart files

⁷T. Ikonen et al. "Module for thermomechanical modeling of {LWR} fuel in multiphysics simulations." *Annals of Nuclear Energy* 84 (2015). Multi-Physics Modelling of {LWR} Static and Transient Behaviour, 111–121. ISSN: 0306-4549. DOI: 10.1016/j.anucene.2014.11.004. URL: <http://www.sciencedirect.com/science/article/pii/S0306454914005842>.

⁸H. Loukusa. *Validation of the FINIX fuel behavior code version 0.13.9*. Tech. rep. VTT Technical Research Centre of Finland, 2013.

Serpent and FINIX

Easy way to incorporate fuel behavior to Serpent simulations. No need for additional programming or worrying about solution flow.

Power relayed from Serpent to FINIX and temperatures from FINIX to Serpent automatically.

Several boundary condition options (axial distribution for each rod):

- Temperature at cladding surface.
- Heat flux at cladding surface.
- Coolant bulk temperature and heat transfer coefficient wrt. cladding.
- Coolant bulk temperature, heat transfer coefficient calculated by FINIX.

Some fuel information should be known or guessed:

- Fuel grain size.
- Fuel and cladding surface roughness.
- Plenum length.
- Fractional fuel density.
- Clad cold work.
- Fill gas pressure and composition.
- etc.

Can be used in criticality source and transient simulations⁹.

⁹Also in burnup simulations, but the fuel behavior solution will be for fresh fuel.

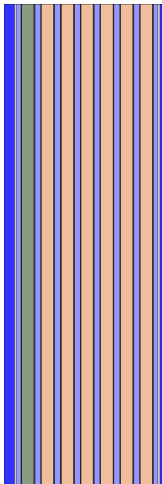
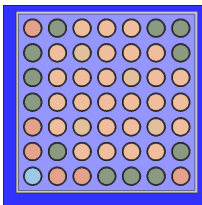
Serpent and FINIX

Modifications to normal input

Peach Bottom 2 type 3D assembly.

Input available at [the wiki](#).

- Fuel at 900 K.
- Cladding at 600 K.
- Coolant at 556 K.



Serpent and FINIX

SS example

```
01ssFIN/
├─ uncoupled/
│   ├── input
│   └─ include/
│       └─ ...
└─ coupled/
    ├── input
    └─ include/
        └─ ...
```

Fix the `acelib` path to the two input-files.

Run the uncoupled input.

Run the coupled input.

Compare the results.

Serpent and FINIX

Modifications to normal input

Materials, whose temperature comes from solver should use the "tft <Tmin> <Tmax>" card instead of the "tmp <T>" card.

```
mat fuel1 -10.42 tmp 900
92235.03c -0.02583
92238.03c -0.85567
 8016.03c -0.11850
```

```
mat fuel1 -10.42 tft 556 2500
92235.03c -0.02583
92238.03c -0.85567
 8016.03c -0.11850
```

```
mat Zirc2 -6.56 tmp 600
 8016.03c -1.19E-03
 8017.03c -4.83E-07
24050.03c -4.16E-05
...
```



```
mat Zirc2 -6.56 tft 556 1000
 8016.03c -1.19E-03
 8017.03c -4.83E-07
24050.03c -4.16E-05
...
```

This will set up the temperature bounds for each material for the on-the-fly TMS temperature treatment.

Tighter limits will result in more efficient TMS sampling, but temperatures outside of the limits will result in unphysical sampling and will terminate the simulation if the command line option `-nofatal` (see [wiki:Installing and running Serpent](#)) was not used.

Serpent and FINIX

Modifications to normal input

Create FINIX-solvers for chosen pins.

```
% --- Assembly lattice
lat 100 1 0.23876 0.23876 9 9 1.875
99 99 99 99 99 99 99 99 99
99 1 2 3 4 5 6 7 99
99 2 8 9 10 11 12 13 99
99 3 9 14 15 16 17 18 99
99 4 10 15 19 20 21 22 99
99 5 11 16 20 23 24 25 99
99 6 12 17 21 24 26 27 99
99 7 13 18 22 25 27 28 99
99 99 99 99 99 99 99 99 99

% --- Fuel pins
pin 1
fuel4 0.60579
void 0.62103
Zirc2 0.71501
cool

pin 2
fuel3 0.60579
void 0.62103
Zirc2 0.71501
cool
...
```

Serpent and FINIX

Modifications to normal input

Create FINIX-solvers for chosen pins:

```
finrod 1  rod_pb2_nogd  options_pb2_all  scenario_pb2_all  powerbins1  nrods  1
finrod 2  rod_pb2_nogd  options_pb2_all  scenario_pb2_all  powerbins1  nrods  2
finrod 3  rod_pb2_nogd  options_pb2_all  scenario_pb2_all  powerbins1  nrods  2
...
finrod 14 rod_pb2_yesgd options_pb2_all  scenario_pb2_all  powerbins1  nrods  1
...
finrod 28 rod_pb2_nogd  options_pb2_all  scenario_pb2_all  powerbins1  nrods  1
```

Link FINIX-input files:

```
% --- FINIX file names

set finrodfile  "./include/rod_library.txt"
set finoptionsfile  "./include/options_library.txt"
set finscenariofile  "./include/scenario_library.txt"
```

Define the powerbinning:

```
% --- FINIX power binning definitions

finbin powerbins1 0.0 0.60579 5 -182.88 182.88 24
```

Serpent and FINIX

Modifications to normal input

Set coupled calculation scheme:

```
% --- Number of coupled calculation iterations
```

```
set ccmxiter 3
```

```
% --- Use fission source passing with 10 inactive cycles between iterations
```

```
set fsp 1 10
```

Set FINIX temperature file to be used for first neutron transport solution if available:

```
% --- FINIX initialization interface file has to be created
```

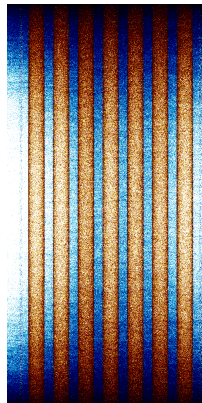
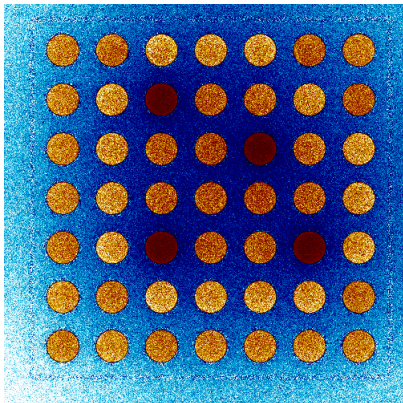
```
%set fininitfile "./Finix0.ifc"
```

Run the simulation!

1.7 min (CC) vs 0.7 min (no CC) as the materials no longer have pre-calculatable material total cross sections.

Serpent and FINIX

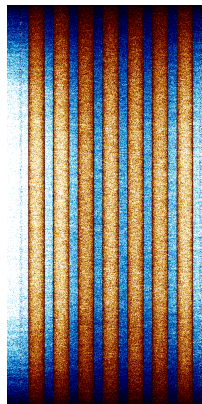
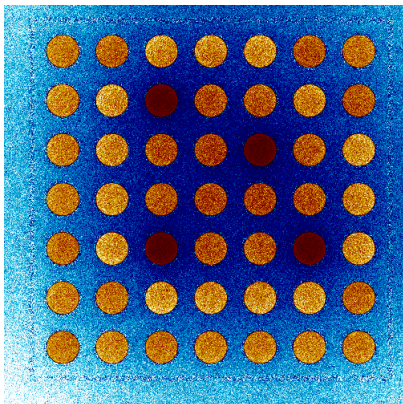
Results



Fission rate distribution from non-coupled simulation.

Serpent and FINIX

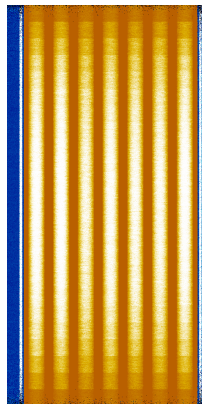
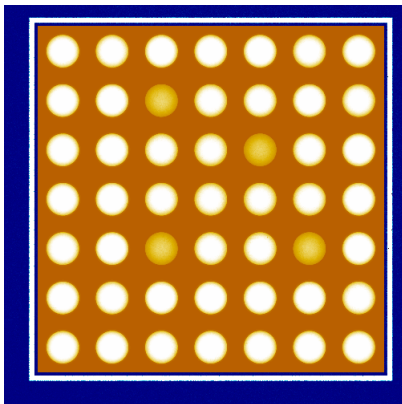
Results



Fission rate distribution from coupled simulation.

Serpent and FINIX

Results



Temperature distribution from coupled simulation.

mesh 10 3 500 500

Serpent and FINIX

Output

`finix_data_structures.dbg` FINIX initial data-structure contents. Good for checking that all options are set up correctly.

`input_fin.m` MATLAB readable file containing the fuel behavior solution: Temperature distributions, power density distributions, cold and hot radial node positions, heat flux at cladding if calculated.

`Finix.ifc` Fuel behavior interface file containing the latest FINIX solution. Can be used (after renaming) for initial conditions for next simulation using [set fininitfile](#)

`Finixifcout.m` Fuel behavior interface output file containing the latest power tallies for the FINIX rods.

`Finix.ifc_Pconv0.m` contains convergence criteria calculated for the tallied power distribution of the FINIX rods from each iteration.

Serpent and FINIX

Transients

Transient simulation (or burnup calculation) setup with FINIX is exactly the same as in the steady state case:

Materials, whose temperature comes from solver should use the "tft <Tmin> <Tmax>" card instead of the "tmp <T>" card.

Create FINIX-solvers for chosen pins.

Link FINIX-input files.

Define the powerbinning.

Set coupled calculation scheme.

Set FINIX temperature file to be used as an initial condition (important from transients starting from power).

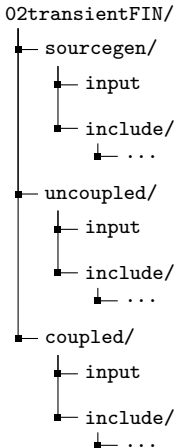
Note that creating the FINIX solvers turns on the coupled calculation mode, which means you'll be running the coupled dynamic simulation mode instead of the uncoupled one¹⁰.

The rest of the transient simulation should be defined as normally. See [wiki:Transient_simulations](#).

¹⁰All neutron batches are first simulated through first interval etc.

Serpent and FINIX

Transients



Fix the `acelib` path to the three input-files.

Run the `sourcegen` input.

Run the `uncoupled` input.

Run the `coupled` input.

Serpent and FINIX - Transient

Transients

3x3 rod unit cell, infinite lattice.

AIC control rod in the middle.

50 cm in height with black axial boundary conditions.

Fuel is 3.6 wt-% UO₂.

Start from cold conditions (300 K) at very high power (10 MW).

Close to critical ($k_{\text{eff}} = 1.00001 \pm 0.00014$) with CR 28.54 cm inserted.

$\beta_{\text{eff.}} \approx 730\text{pcm}$ and $\lambda_{\text{gen.}} = 72\ \mu\text{s}$.

Same initial neutron and precursor source can be used for coupled and uncoupled transient. It has to be created with a criticality source simulation. See

[Wiki:Transient_simulations#Generating_the_steady_state_source_distributions](#)

Serpent and FINIX - Transient

Source generation simulation:

Criticality source simulation with the `set savesrc` option set.

2.4 minutes running time generates 102 000 live neutrons and 3400 delayed neutron precursors. Generated source size scales linearly with simulated neutron population (and thus the simulation time).

Transient simulation:

At $t = 0$ the control rod receives an upwards velocity of $100\,000 \frac{\text{cm}}{\text{s}}$, which is maintained until the control rod is completely removed from the geometry, resulting in an insertion of 7.9 \$ of reactivity. After this, the control rod is inserted back with 10 % of the removal speed until the critical position is reached again.

The simulation time is set to 50 ms with 500 time intervals for population control and coupled calculation. No iteration is used between the solvers.

Approx. 2 hour simulation time (80 000 initial neutrons on each time interval).

Serpent and FINIX - Transient

Transient input – uncoupled

How to move from the steady state input to transient input?

Instead of generating a dynamic source, we will link a pre-generated one.

```
set savesrc "./source" 0.75 ⇒ set dynsrc "../sourcegen/source" 1
```

We'll create a time binning for the simulation:

```
tme tbin1 2 200 0.0 0.05
```

We'll switch `set pop` to `set nps` and link the time binning to it.

We'll add some detectors to keep track of things. Time binning is only needed in uncoupled simulations.

```
det FLX du 0 di tbin1 % integral flux
det NSF dr -7 void di tbin1 % fission neutron production
det CAPT dr -2 void di tbin1 % capture
det FISS dr -6 void di tbin1 % fission
det NUXN dr -16 void di tbin1 % scattering production
det LEAK dr -17 void di tbin1% neutron leakage
```

$$\text{prod./loss} = \frac{\text{NSF}}{\text{FISS} + \text{CAPT} - \text{NUXN} + \text{LEAK}}$$

Serpent and FINIX - Transient

Transient input – coupled

Link the dynamic source `set dynsrc "../sourcegen/source" 1`

Create time binning, insert `set nps` and link the time binning to it.

Add some detectors to keep track of things. Time binning is only needed in uncoupled simulations.

```
det FLX  du 0          % integral flux
det NSF  dr -7 void    % fission neutron production
det CAPT dr -2 void    % capture
det FISS dr -6 void    % fission
det NUXN dr -16 void   % scattering production
det LEAK dr -17 void   % neutron leakage
```

Set detectors to output their results after each time interval!

```
set outp 1
```

$$\text{prod./loss} = \frac{\text{NSF}}{\text{FISS} + \text{CAPT} - \text{NUXN} + \text{LEAK}}$$

Serpent and FINIX - Transient

Transient input – coupled

Materials, whose temperature comes from solver should use the "tft <Tmin> <Tmax>" card instead of the "tmp <T>" card.

Create FINIX-solvers for the two pins.

Link FINIX-input files.

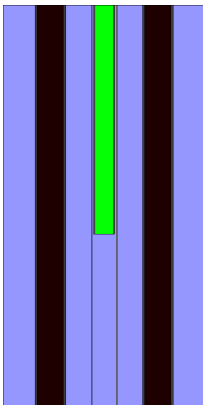
Define the powerbinning.

Set coupled calculation scheme.

Set FINIX temperature file to be used as an initial condition (now using cold conditions).

Serpent and FINIX - Transient

Results



Serpent and FINIX - Transient

Output

`input_det0.m` Detector output of first time interval.

`input_det1.m` Detector output of second time interval.

...

In coupled simulations, detector results are cleared after each coupled calculation iteration.

`input_geom0_t0.m` Plot of geometry used during first time interval.

`input_geom0_t1.m` Plot of geometry used during second time interval.

...

Plots are generated during the simulation of the first neutron batch¹¹.

¹¹Time dependent geometry plots are produced for: i) dynamic simulations with a transient source (set `dynsrc`). ii) dynamic simulations with coupled calculation (FINIX, set `comfile` or set `ppid`)

Serpent and FINIX - Transient

Results

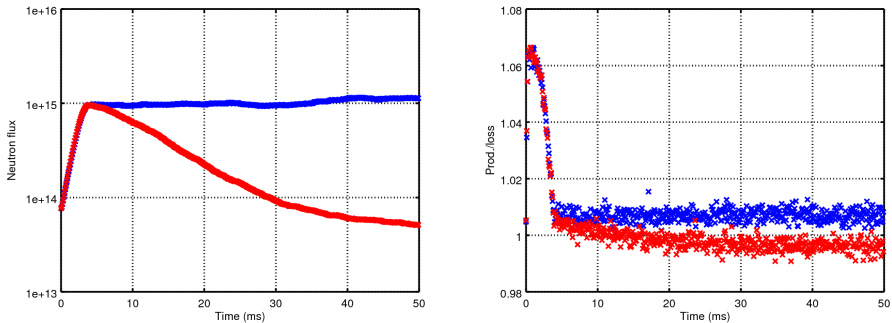


Figure 1: **Left:** Integral neutron flux. **Right:** Momentary neutron production / loss. Coupled (red), uncoupled (blue).

Summary – coupled calculations with FINIX

FINIX fuel behavior module can easily be internally coupled with Serpent to bring in fuel behavior for steady state and transient calculations.

FINIX can be licensed separately from VTT¹²

Moving from uncoupled simulations to coupled simulations is quite easy:

"tmp <T>" ⇒ "tft <Tmin> <Tmax>".

Create FINIX-solvers for chosen pins.

Link FINIX-input files¹³.

Define the powerbinning.

Set coupled calculation scheme.

Set FINIX temperature file to be used as an initial condition (important from transients starting from power).

For further information see: [wiki:FINIX_fuel_behavior_module](https://wiki.vtt.fi/Finix_fuel_behavior_module)

¹²Contact the main developer Timo.Ikonen@vtt.fi

¹³Writing these input files is the hardest part (not very hard)

Externally coupled solver

Serpent can be coupled with existing solvers using the external coupling approach with file based transfer of the relevant solution fields.

The TH-fields can be brought to Serpent using any of the [multi-physics interface](#) formats.

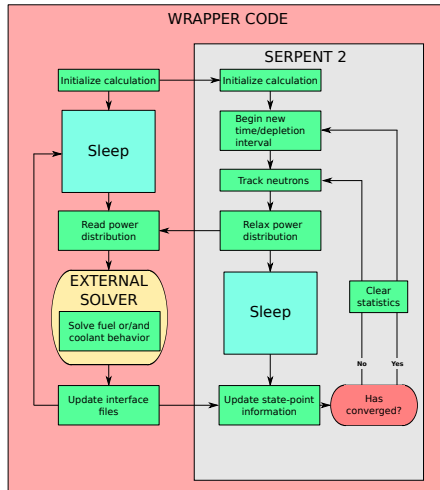
The simplest way to do this is just re-run the Serpent simulation with updated interface definition for each coupled calculation iteration.

However, input processing, cross section preparation and inactive source cycles take a fixed amount of time for each such iteration.

It is possible to update the coupled solution without restarting Serpent by using the file or POSIX-signal based communication between Serpent and an external wrapper program¹⁴.

¹⁴You could do the communication manually, but it is not very convenient.

Solution flow & signals with external coupling



SSS->EXT

EXT->SSS

SIGUSR1 – update interfaces for current time step.
 SIGUSR2 – move to next time step and update ifcs.
 SIGTERM – simulation over, Serpent stopping.

SIGUSR1 – interfaces updated.
 SIGUSR2 – interfaces updated, move to next time step.
 (wrapper is satisfied with convergence)

Setting up an external coupling

Materials, whose temperature comes from solver should use the "tft <Tmin> <Tmax>" card instead of the "tmp <T>" card.

A multi-physics interface file should be linked to the input using the `ifc`-card

```
ifc cool.ifc
```

Set up the coupled calculation scheme:

```
% --- Number of coupled calculation iterations  
set ccmxiter 3
```

```
% --- Use fission source passing with 10 inactive cycles between iterations  
set fsp 1 10
```

Set up one way to communicate with Serpent:

```
% --- Communication files for signals into / out of Serpent  
set comfile com.in com.out
```

```
% --- Process id of wrapper program for POSIX signalling  
% This isn't known before the wrapper program is running  
%set ppid XXX
```

Minimal Serpent Coupling Script

See: [Wiki:Minimal Serpent Coupling Script](#)

A python script giving a minimal example of a wrapper program (and an integrated TH-solver) that can be used to run externally coupled simulations with Serpent.

Since the script is supposed to be minimal, it is not general, but specific to the input it is created for.

Nevertheless, it is a good way to familiarize oneself with the signaling, interface I/O and solution updating in the externally coupled simulations.

TH-solutions can be replaced with calls to external solver.

Two variants:

- Steady state MSCS (< 300 lines with comments)
- Transient MSCS (< 400 lines with comments)

Burnup version upcoming.

Minimal Serpent Coupling Script – Usage

Get the MSCS script you want to use from [Serpent Wiki](#) and save it to MSCS.py. Copy the input file from Serpent wiki into the same directory using the name `input`.

Add a `set aceLib` definition to the input-file.

Add the path to the Serpent executable to be used to MSCS.py.

Familiarize yourself with killing running processes in case something goes wrong¹⁵.

Run the script from the subdirectory with `"python MSCS.py"`

¹⁵MSCS starts the Serpent simulation in the background with the `&` option, making it impossible to kill with Ctrl+C

Steady state MSCS

[Link](#)

200 cm long fuel rod in infinite lattice. Axially black boundary conditions. Coolant temperature and density solved by MSCS using simple correlations.

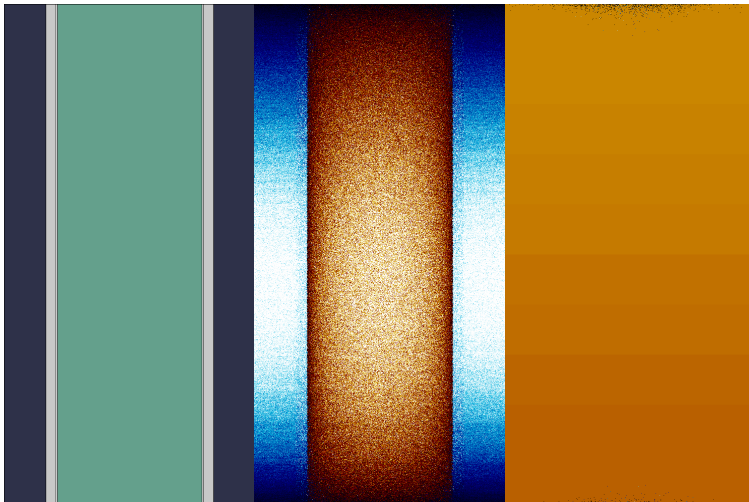
Average linear power of $600 \frac{\text{W}}{\text{cm}}$ to get something interesting. TH-fields use 10 axial layers.

MSCS creates the initial interface for the coolant, appends the interface definition and communication mode to the input file and starts the Serpent simulation.

MSCS will wait for a signal from Serpent (read from the communications file). After receiving the signal, MSCS will read the power distribution from the interface output, calculate the new steady state TH solution, update the interface file and send a SIGUSR1 to Serpent to continue the iteration.

Steady state MSCS

5 coupled calculation iterations with 500k active neutrons per iteration:



Transient MSCS

Link

200 cm long fuel rod in infinite lattice. Axially black boundary conditions. Fuel temperature solved by MSCS assuming no heat conduction out of fuel.

Initial linear power of $600 \frac{\text{W}}{\text{cm}}$ with a supercritical system. Initial source of 1 MeV neutrons at the center of the fuel rod.

MSCS creates the initial interface for the coolant and the fuel, appends the interface definitions and communication mode to the input file and starts the Serpent simulation.

MSCS will wait for a signal from Serpent (read from the communications file). After receiving the signal, MSCS will:

- read the power distribution for the current time interval from the interface output
- calculate the new EOI fuel temperature based on BOI temperature and deposited heat
- update the interface file
- send a SIGUSR1 to Serpent to continue the iteration¹⁶.

When MSCS receives a SIGUSR2 it will move to the next time interval, meaning that:

- EOI temperature field becomes BOI temperature field.
- some files are archived.

¹⁶ MSCS could check, whether the fuel temperature changed significantly during current iteration and send SIGUSR2 if not to indicate that no further iteration is needed.

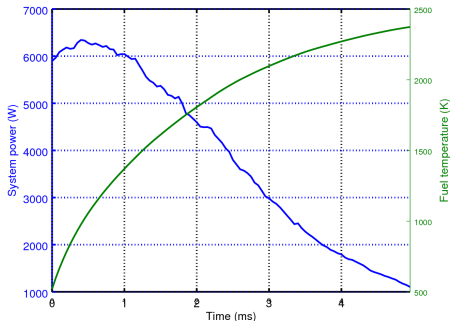
Transient MSCS

Output:

`fuel.ifc<t>` – Fuel temperature at time point `<t>`. 0 corresponds to initial field. 1 corresponds to EOI of first interval etc.

`coolifc.out<t>` – The power distribution output written by Serpent for the multi-physics interface from interval `<t>`. 0 corresponds to first interval etc.

Note: Normally, the interface and the interface output are overwritten during the coupled calculation, but here the fields are archived to `filename<t>` by MSCS.



Summary – External coupling

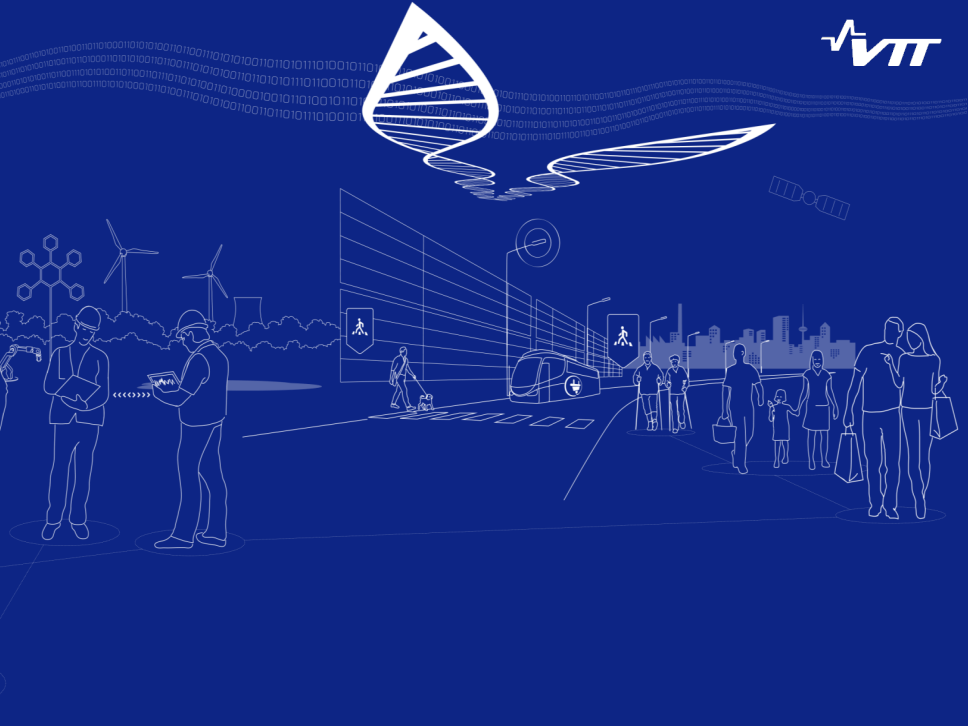
External coupling with file based IO can be used to couple any solver with Serpent.

The TH-fields can be brought to Serpent using any of the [multi-physics interface](#) formats.

Messages controlling the solution flow will be relayed either by files or POSIX-signals.

The external code can estimate the solution convergence and stop the iteration of a certain time-point when a sufficient level of convergence is reached.

The Minimal Serpent Coupling Script serves as an example of externally coupled simulations with Serpent.



T. Ikonen et al. "Module for thermomechanical modeling of {LWR} fuel in multiphysics simulations." *Annals of Nuclear Energy* 84 (2015). Multi-Physics Modelling of {LWR} Static and Transient Behaviour. 111–121. ISSN: 0306-4549. DOI: 10.1016/j.anucene.2014.11.004. URL: <http://www.sciencedirect.com/science/article/pii/S0306454914005842>.

J. Leppänen. "Modeling of Non-uniform Density Distributions in the Serpent 2 Monte Carlo Code." *Nucl. Sci. Eng.* 174 (2013): 49–58. DOI: 10.13182/NSE12-54. URL: http://www.ans.org/pubs/journals/nse/a_17167.

J. Leppänen, T. Viitanen, and V. Valtavirta. "Multi-physics Coupling Scheme in the Serpent 2 Monte Carlo Code." *Trans. Am. Nucl. Soc.* 107 (2012), 1165.

V. Valtavirta, M. Hesan, and J. Leppänen. "Delayed Neutron Emission Model for Time Dependent Simulations with the Serpent 2 Monte Carlo Code – First Results." In proc. Proc. Physor 2016, Sun Valley, ID, USA, (2016).

T. Viitanen and J. Leppänen. "New Interpolation Capabilities for Thermal Scattering Data in Serpent 2." In proc. PHYSOR 2016, Sun Valley, ID, USA, (May 2016).

T. Viitanen. "Development of a stochastic temperature treatment technique for Monte Carlo neutron tracking." PhD thesis. Aalto University, 2015.

H. Loukusa. *Validation of the FINIX fuel behavior code version 0.13.9*. Tech. rep. VTT Technical Research Centre of Finland, 2013.

