Kraken workshop

Short overview of KrakenTools

The KrakenTools Python package

Collection of Python scripts, functionalities and definitions useful for the Kraken framework.

Examples:

- Producing water and fuel material definitions for Serpent.
- Producing material volumes for Serpent.
- Interacting with Serpent binary restarts.
- Condensation of Serpent group constant data after Serpent calculation.

- Radial reflector homogenization.

- krakentools.ants
- krakentools.meshes
- krakentools.plotting

Producing water and fuel material definitions for Serpent

Producing water compositions for Serpent

krakentools.utils.write_water_composition()

krakentools.utils.write_p_t_water_composition()

krakentools.utils.write_saturated_water_composition()

KrakenTools/tests/ water_composition_no_therm, water_composition_with_therm, water_composition_saturated, water_composition_pT

Feed in basic data: (p, T, boron, …)

```
pressure = 15e6 # Pa
temperature = 550 # K
boron = 1000 # ppm (weight)
```

Get Serpent material definition:

```
mat cool_1000B_76D -0.76978 tmp 550
                        moder lw550K 1001
                        rgb 200 200 255
O-16.03c   3.323385e-01
O-17.03c   1.265963e-04
80180      6.829536e-04
H-1.03c    6.662196e-01
H-2.03c    7.662406e-05
B-10.03c   1.105860e-04
B-11.03c   4.451225e-04
```

Producing fuel compositions for Serpent

krakentools.utils.write_fuel_composition()
KrakenTools/tests/ fuel_composition

Feed in basic data: (density, enrichment, gadolinium, …)

```
density = 10.2 # g/cm3
wt_frac_U234 = 0.0002 # of U
wt_frac_U235 = 0.04 # of U
wt_frac_U236 = 0.0002 # of U
wt_frac_Gd2O3 = 0.02 # of fuel density
```

Get Serpent material definition:

```
mat fuel_20U4_40U5_20U6_20GO -10.2 tmp 800
92234.06c   -1.727707e-04
92235.06c   -3.455413e-02
92236.06c   -1.727707e-04
92238.06c   -8.289537e-01
8016.06c    -1.187942e-01
64152.06c   -3.352815e-05
64154.06c   -3.702705e-04
64155.06c   -2.530124e-03
64156.06c   -3.522011e-03
64157.06c   -2.709997e-03
64158.06c   -4.328777e-03
64160.06c   -3.857778e-03
```

Producing material volumes for Serpent

krakentools.utils.process_material_volumes()

- Serpent needs information on the volumes of each depletion zone for burnup calculation.
- Depletion zone numbering may be hard to follow.
- Clever use of symmetries increases statistics, but means that some depletion zones can have 2, 4 or 8 times the base volume.
- Serpent can estimate volumes based on Monte Carlo sampling, but that leaves statistical uncertainties.

```
% --- Material volumes:

% Produced Wed Jul 17 10:40:00 2019 by MC volume calculation routine by
% sampling 10000000 random points in the geometry.

set mvol

fuelNoGad     35 1.98263E+00 % (0.005)
fuelNoGad     34 1.96811E+00 % (0.004)
fuelNoGad     33 3.98636E+00 % (0.003)
fuelNoGad     32 1.99161E+00 % (0.005)
fuelNoGad     31 3.96966E+00 % (0.004)
fuelNoGad     30 3.98779E+00 % (0.003)
fuelNoGad     29 3.96966E+00 % (0.003)
...
```

krakentools.utils.process_material_volumes()

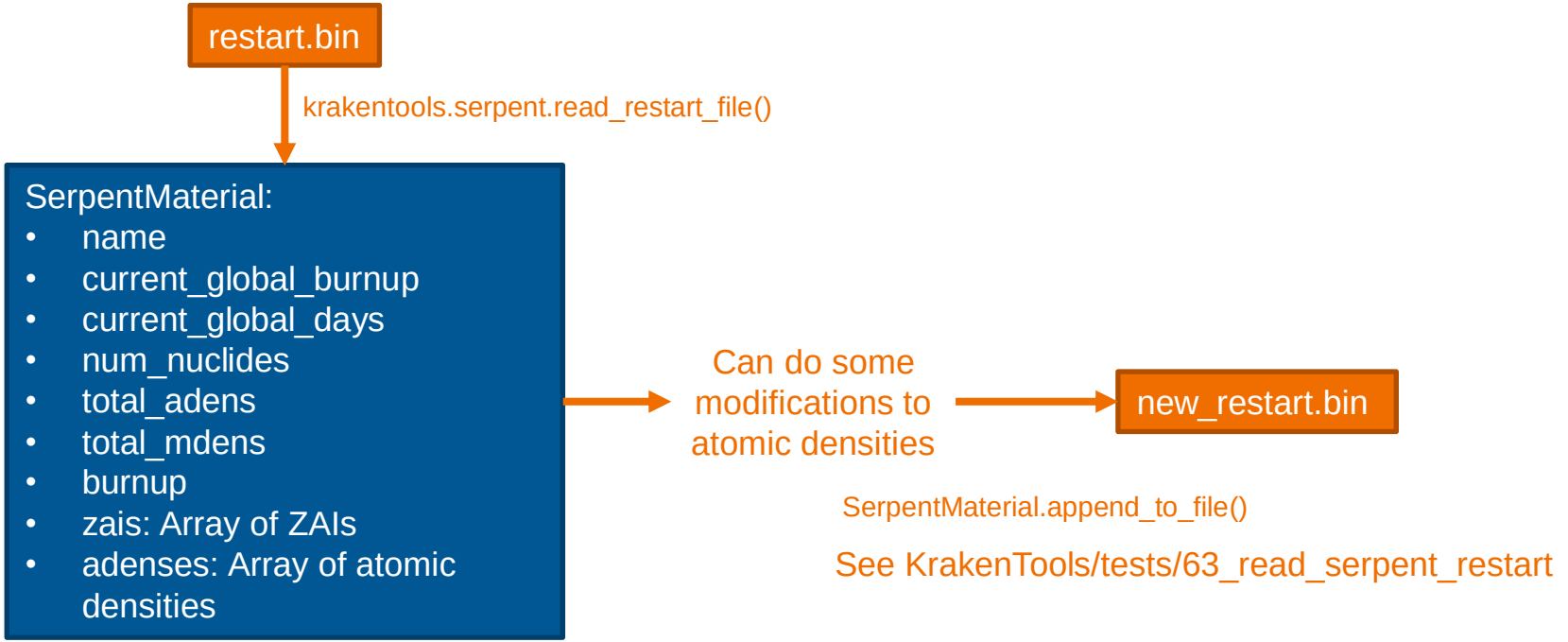
- Some clean up can be made to the Monte Carlo calculated volumes based on basic information on the depletion zone setup.
- KrakenTools/tests/06_material_volume_processing/input.py
- KrakenTools/tests/62_process_arbitrary_fuel_volumes/input.py

```
% --- Material volumes:


set mvol

fuelNoGad       35 1.98557E+00     %% original volume 1.98263E+00 was a multiple of 4 (3.994 as float)
fuelNoGad       34 1.98557E+00     %% original volume 1.96811E+00 was a multiple of 4 (3.965 as float)
fuelNoGad       33 3.97113E+00     %% original volume 3.98636E+00 was a multiple of 8 (8.031 as float)
fuelNoGad       32 1.98557E+00     %% original volume 1.99161E+00 was a multiple of 4 (4.012 as float)
fuelNoGad       31 3.97113E+00     %% original volume 3.96966E+00 was a multiple of 8 (7.997 as float)
fuelNoGad       30 3.97113E+00     %% original volume 3.98779E+00 was a multiple of 8 (8.034 as float)
fuelNoGad       29 3.97113E+00     %% original volume 3.96966E+00 was a multiple of 8 (7.997 as float)
...
```

Interacting with Serpent binary restarts

krakentools.serpent.read_restart_file()

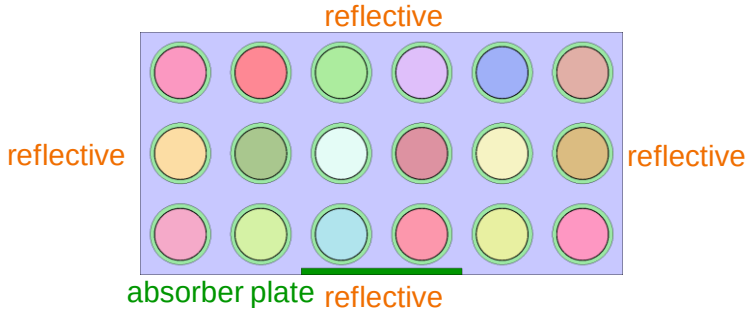
- Serpent can be told to write material compositions into restart files (set rfw) for later utilization in restart simulations (set rfr).
  - In the history part of group constant calculations, the binary restart file (.wrk) is produced automatically.
- Krakentools can be used to extract the material data from such files (including all burnup points) into SerpentMaterial objects:

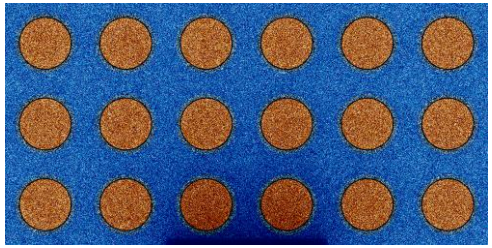


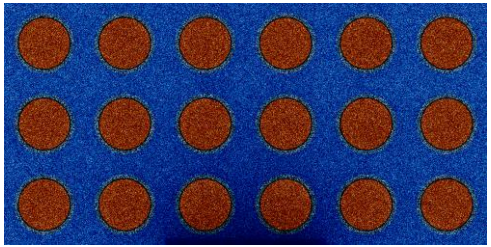
See KrakenTools/tests/63_read_serpent_restart

Example of working with restart files

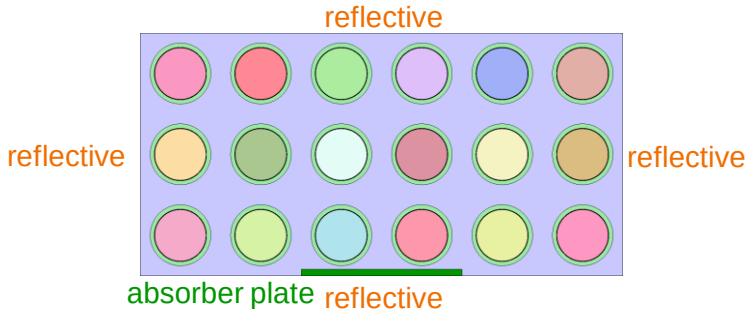
- Simple non-symmetric depletion problem with depletion zone division with:
  - div fuel sep 1
- Positioning of depletion zones can be checked with the -matpos command line parameter of Serpent.
- Deplete up to some burnup and write restart.
  - first.inp.wrk
- Read restart to Python, modify nuclide densities:
  - Move some fuel to another position.
    Apply fresh fuel densities to others.
  - Write modified restart out.
- Read restart to Serpent and continue.







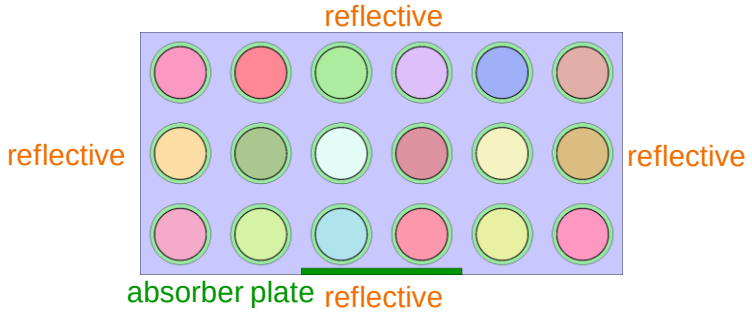
Example of working with restart files



- Simple non-symmetric depletion problem with depletion zone division with:
  - div fuel sep 1
- Positioning of depletion zones can be checked with the -matpos command line parameter of Serpent.
- Deplete up to some burnup and write restart.
  - first.inp.wrk
- **Read restart to Python**, modify nuclide densities:
  - Move some fuel to another position.
    Apply fresh fuel densities to others.
  - Write modified restart out.
- Read restart to Serpent and continue.

```
Material burnups at 40.0 MWd/kgU.
Material fuel    , material burnup 39.97.
Material fuelz18 , material burnup 43.00.
Material fuelz17 , material burnup 42.37.
Material fuelz16 , material burnup 41.86.
Material fuelz15 , material burnup 42.00.
Material fuelz14 , material burnup 42.33.
Material fuelz13 , material burnup 42.93.
Material fuelz12 , material burnup 42.23.
Material fuelz11 , material burnup 41.18.
Material fuelz10 , material burnup 38.87.
Material fuelz9  , material burnup 39.35.
Material fuelz8  , material burnup 40.91.
Material fuelz7  , material burnup 42.12.
Material fuelz6  , material burnup 41.69.
Material fuelz5  , material burnup 38.20.
Material fuelz4  , material burnup 30.17.
Material fuelz3  , material burnup 30.05.
Material fuelz2  , material burnup 38.38.
Material fuelz1  , material burnup 41.79.
```

Example of working with restart files

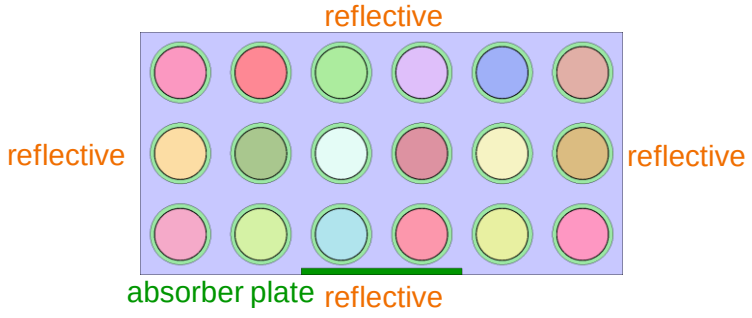


- Simple non-symmetric depletion problem with depletion zone division with:
  - div fuel sep 1
- Positioning of depletion zones can be checked with the -matpos command line parameter of Serpent.
- Deplete up to some burnup and write restart.
  - first.inp.wrk
- **Read restart to Python, modify nuclide densities:**
  - **Move some fuel to another position.**
    Apply fresh fuel densities to others.
  - **Write modified restart out.**
- Read restart to Serpent and continue.

```
Material burnups at 40.0 MWd/kgU.
Material fuel    , material burnup 39.97.
Material fuelz18 , material burnup 43.00.
Material fuelz17 , material burnup 42.37.
Material fuelz16 , material burnup 41.86.
Material fuelz15 , material burnup 42.00.
Material fuelz14 , material burnup 42.33.
Material fuelz13 , material burnup 42.93.
Material fuelz12 , material burnup 42.23.
Material fuelz11 , material burnup 41.18.
Material fuelz10 , material burnup 38.87.
Material fuelz9  , material burnup 39.35.
Material fuelz8  , material burnup 40.91.
Material fuelz7  , material burnup 42.12.
Material fuelz6  , material burnup 41.69.
Material fuelz5  , material burnup 38.20.
Material fuelz4  , material burnup 30.17.
Material fuelz3  , material burnup 30.05.
Material fuelz2  , material burnup 38.38.
Material fuelz1  , material burnup 41.79.
```

```
Material burnups at 40.0001 MWd/kgU.
Material fuel    , material burnup 39.97.
Material fuelz18 , material burnup 42.00.
Material fuelz17 , material burnup 42.33.
Material fuelz16 , material burnup 42.93.
Material fuelz15 , material burnup 0.00.
Material fuelz14 , material burnup 0.00.
Material fuelz13 , material burnup 0.00.
Material fuelz12 , material burnup 39.35.
Material fuelz11 , material burnup 40.91.
Material fuelz10 , material burnup 42.12.
Material fuelz9  , material burnup 0.00.
Material fuelz8  , material burnup 0.00.
Material fuelz7  , material burnup 0.00.
Material fuelz6  , material burnup 30.05.
Material fuelz5  , material burnup 38.38.
Material fuelz4  , material burnup 41.79.
Material fuelz3  , material burnup 0.00.
Material fuelz2  , material burnup 0.00.
Material fuelz1  , material burnup 0.00.
```

Example of working with restart files



- Simple non-symmetric depletion problem with depletion zone division with:
  - div fuel sep 1
- Positioning of depletion zones can be checked with the -matpos command line parameter of Serpent.
- Deplete up to some burnup and write restart.
  - first.inp.wrk
- Read restart to Python, modify nuclide densities:
  - Move some fuel to another position.
    Apply fresh fuel densities to others.
  - Write modified restart out.
- **Read restart to Serpent and continue.**

modified.wrk

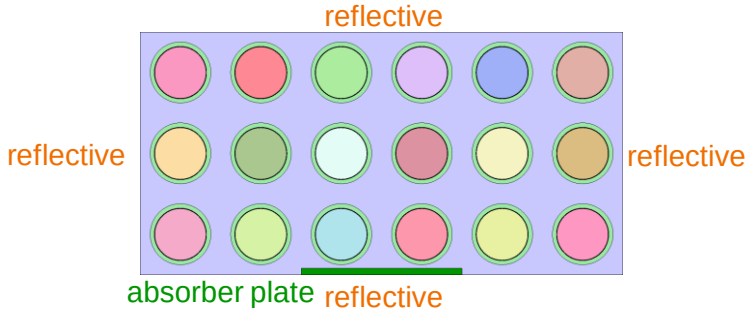
```
Material burnups at 40.0001 MWd/kgU.
Material fuel    , material burnup 39.97.
Material fuelz18 , material burnup 42.00.
Material fuelz17 , material burnup 42.33.
Material fuelz16 , material burnup 42.93.
Material fuelz15 , material burnup 0.00.
Material fuelz14 , material burnup 0.00.
Material fuelz13 , material burnup 0.00.
Material fuelz12 , material burnup 39.35.
Material fuelz11 , material burnup 40.91.
Material fuelz10 , material burnup 42.12.
Material fuelz9  , material burnup 0.00.
Material fuelz8  , material burnup 0.00.
Material fuelz7  , material burnup 0.00.
Material fuelz6  , material burnup 30.05.
Material fuelz5  , material burnup 38.38.
Material fuelz4  , material burnup 41.79.
Material fuelz3  , material burnup 0.00.
Material fuelz2  , material burnup 0.00.
Material fuelz1  , material burnup 0.00.
```

second.inp.wrk

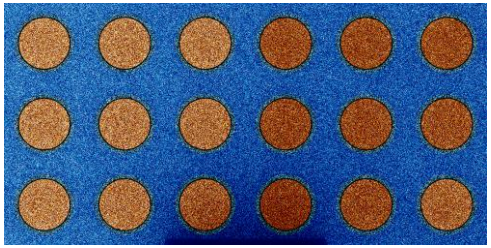
```
Material burnups at 40.0001 MWd/kgU.
Material fuel    , material burnup 19.99.
Material fuelz18 , material burnup 42.00.
Material fuelz17 , material burnup 42.33.
Material fuelz16 , material burnup 42.93.
Material fuelz15 , material burnup 0.00.
Material fuelz14 , material burnup 0.00.
Material fuelz13 , material burnup 0.00.
Material fuelz12 , material burnup 39.35.
Material fuelz11 , material burnup 40.91.
Material fuelz10 , material burnup 42.12.
Material fuelz9  , material burnup 0.00.
Material fuelz8  , material burnup 0.00.
Material fuelz7  , material burnup 0.00.
Material fuelz6  , material burnup 30.05.
Material fuelz5  , material burnup 38.38.
Material fuelz4  , material burnup 41.79.
Material fuelz3  , material burnup 0.00.
Material fuelz2  , material burnup 0.00.
Material fuelz1  , material burnup 0.00.
```

Example of working with restart files

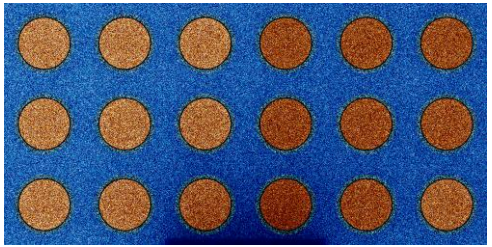
- Simple non-symmetric depletion problem with depletion zone division with:
  - div fuel sep 1
- Positioning of depletion zones can be checked with the -matpos command line parameter of Serpent.
- Deplete up to some burnup and write restart.
  - first.inp.wrk
- Read restart to Python, modify nuclide densities:
  - Move some fuel to another position.
    Apply fresh fuel densities to others.
  - Write modified restart out.
- **Read restart to Serpent and continue.**



40.0001 MWd/kgU



41 MWd/kgU



Condensation of Serpent group constant data

Condensing Serpent group constant data further

- Generate group constants with Serpent into some energy group structure.
- Read the data into NodeBranch objects with
  - krakentools.serpent.read_coe_and_res()
  - krakentools.serpent.read_all_coe_and_res("input*", n_proc=10)

Reading utilizes functionalities from serpentTools: Andrew Johnson, Dan Kotlyar, Stefano Terlizzi, and Gavin Ridley, "serpentTools: A Python Package for Expediting Analysis with Serpent.", Nuclear Science and Engineering, 194 (2020)

- Each krakentools.containers.NodeBranch object contains data for a single universe/node at specific conditions (branch).

- Use NodeBranch.give_condensed_version() to obtain an energy condensed variant (both infinite spectrum and leakage corrected data if available).
- See tests 50 and 51.

NodeBranch.calculate_critical_spectra() evaluates critical (B1, P1 and FM) spectra outside Serpent, but has not been tested thoroughly yet.

Summary

- KrakenTools collects many functionalities for Kraken related tasks.
  - Built primarily to serve VTT internal purposes as is most of the Kraken framework.
    - Not a perfect and comprehensive suite of tools at the moment.

- For general Serpent purposes (producing material compositions etc.) you may have your own scripts already.

- For other purposes you may find the capabilities of KrakenTools useful:
  - Full core 2D radial reflector homogenization.
  - Group constant parametrization.

Serpent users should also consider testing out serpentTools:
https://github.com/CORE-GATECH-GROUP/serpent-tools
https://serpent-tools.readthedocs.io/en/master/overview.html

VTT

beyond
the obvious

Ville Valtavirta
Ville.Valtavirta@vtt.fi

@VTTFinland

www.vtt.fi