23.06.2019

Perturbation calculations in cross section homogenization with OpenMOC

<Autor des Dokuments (optional)>

<Untertitel des Dokuments (optional)>

Table of contents

Abbildungsverzeichnis (Formatvorlage „Überschrift 1“)

Tabellenverzeichnis (Formatvorlage „Überschrift 1“)

1. Introduction

The purpose of this project is to use the open source code “OpenMOC” [] for uncertainty and sensitivity analyses in the generation of macroscopic cross sections for light water reactors (LWRs).

OpenMOC belongs to the class of spectral codes used in reactor core design. Their main application is to generate few energy-group, spatially homogenized cross sections (also called homogenized macroscopic cross sections) from microscopic cross section data. These macroscopic cross sections can then be used in full-core reactor simulators to calculate the power distribution and its burnup dependence. These simulators typically apply nodal neutron transport solvers in which the reactor core is approximated as a set of a small number of coupled nodes. Each of these nodes is assumed to have a homogenous material composition and a few-energy group dependence as described by its macroscopic cross sections. Different nodes can have different material properties because of different burnups (i.e. nuclide composition) or because they belong to different fuel assembly types (i.e. UOX or MOX fuel).

In principle a full-core reactor power distribution can also be calculated directly from microscopic cross section data with Monte Carlo codes such as MCNP [] or OpenMC []. In 2019 this still requires high performance computer clusters and days of computation time while the method of combining results from a spectral code together with a nodal neutron transport solver takes only a few minutes of calculations and it was already successfully used in the licensing of LWRs in the 1970s []. One drawback of the latter method is the difficulty to quantify the impact of the microscopic cross section uncertainty on the core power distribution.

A focus of ongoing research is to augment best-estimate microscopic cross sections with covariance data to enable uncertainty and sensitivity analyses of core power distributions []. One of the newest approach for this purpose is the “Total Monte Carlo” method (TMC) []: given the covariance data an ensemble of microscopic cross sections is generated (i.e. sampled according to the covariance information), then an ensemble of macroscopic cross sections is derived and finally a set of resulting reactor core power distributions is determined. The TMC method could directly and naturally be used together with classical Monte Carlo codes like MCNP – but the computational burden would still be infeasible. Hence the research focus currently is on enabling spectral codes and nodal reactor simulators for this purpose.

In this project the generation of an ensemble of homogenized macroscopic cross sections without a reactor core simulation is pursued. The primary reason for using OpenMOC is its

open-source status while all other spectral codes known to the author are proprietary. This enables open communication and transparency of results and source code modification. Another advantage of OpenMOC is its ability to utilize GPUs []. Hence its computations are very fast compared to all other spectral codes known by the author and is therefore particularly suited to generate large ensembles of cross sections.

The structure of the text is as follows: the introduction continues with an explanation of the basic theory of neutron transport in LWRs, the description of the microscopic cross section data available for LWR applications, the structure and method of the OpenMOC code and of the microscopic data reader NJOY [].

Chapter 2 describes the OpenMOC code in detail together with the coded Python scripts to generate and analyze ensembles of macroscopic cross sections, given an ensemble of microscopic cross sections.

In chapter 3 the methods of the microscopic cross section reader NJOY are explained together with the coded Python scripts to generate an ensemble of microscopic cross sections, given the covariance data.

Finally, chapter 4 contains the results of the TMC method done with OpenMOC and a comparison with best estimate results from a proprietary spectral code.

Chapter 5 gives recommendations for improvements in future work and an outlook how to use the results in a reactor core simulator.

1.1. Boltzmann equation for neutron transport

The Boltzmann equation of neutron transport [] determines the neutron flux in a reactor core and in the most general case is formulated as follows (for an explanation of the different terms see for example []):

(1) $\Sigma_t(\vec{x},E,t)\varphi\left(\vec{x},\widehat{\Omega},E,t\right)+\widehat{\Omega}\cdot\nabla\varphi+\frac{1}{v(E)}\cdot\frac{\partial\varphi}{\partial t}=\int_0^{4\pi}d\widehat{\Omega}'\int_0^{\infty}dE'\,\Sigma_s\left(\vec{x},E'\rightarrow E,\widehat{\Omega}'\rightarrow\widehat{\Omega},t\right)\varphi+$
$+\frac{\chi(\vec{x},E)}{4\pi}\int_0^{\infty}dE'v(\vec{x},E',t)\Sigma_f(\vec{x},E',t)\varphi$

In (1) the angular flux φ is a function of location $\vec{x}$, direction $\widehat{\Omega}$, energy E and time t. It is defined as the product of the angular neutron density n and the neutron speed $\mathbb{v}$:

(2) $\varphi\left(\vec{x},\widehat{\Omega},E,t\right)=\mathbb{v}(E)\cdot n\left(\vec{x},\widehat{\Omega},E,t\right)$

The macroscopic cross sections Σ are the product of the number density ρ of the material and its microscopic cross section σ:

(3) $\Sigma = \rho(\vec{x},t) \cdot \sigma(\vec{x},E,t)$

The energy spectrum of the fission neutrons is $\chi(\vec{x},E)$ and the neutron numbers per fission event is $\upsilon(\vec{x},E,t)$.

In (1) the total cross section Σ_t, the scattering cross section Σ_s and the fission cross section Σ_f are occurring. These macroscopic cross sections are a function of energy and space (and sometimes of time, too) and must be spatially averaged and homogenized and reduced to a few-group energy dependence for subsequent use in a nodal solver for a reactor core simulation.

If a material region consists of a homogenous mixture of materials, the cross section definition for type Σ_x (x = fission/scatter/total) is:

(4) $\Sigma_x = \sum_k \rho_k(\vec{x}) \cdot \sigma_{x,k}(\vec{x},E)$

The sum is over all nuclide types k within the homogenous mixture.

If the solution φ_{true} to (1) is known, a few-energy group structure can be formulated (stationary solution and factorization assumption $\varphi_{true} \approx f_1(\vec{x},E)f_2(\vec{x},\hat{\Omega})$):

(5) $\Sigma_{t,g}(\vec{x})\varphi_g(\vec{x},\hat{\Omega}) + \hat{\Omega}\cdot\nabla\varphi_g = \int_0^{4\pi} d\hat{\Omega}' \sum_{g'} \Sigma_{s,g'\to g}(\vec{x},\hat{\Omega}'\to\hat{\Omega})\varphi_{g'} + \frac{\chi(E_g)}{4\pi}\sum_{g'} \upsilon_{g'}\Sigma_{f,g'}(\vec{x})\,\varphi_{g'}$

$\varphi_g = \int_{E_g}^{E_g+\Delta} dE' \varphi_{true}(\vec{x},\hat{\Omega},E')$

$\Sigma_{x,g}(\vec{x}) \cdot \varphi_g(\vec{x},\hat{\Omega}) = \int_{E_g}^{E_g+\Delta} dE' \Sigma_x(\vec{x},E')\varphi_{true}(\vec{x},\hat{\Omega},E')$

The sum g is over G groups. In LWR cases G=2 or 4, usually.

In practice the following procedure has been applied for LWR applications: a high-resolution energy solution φ_{cell} is determined for a unit cell like a fuel rod plus its surrounding subchannel and the location and angular dependence is averaged out. This result $\varphi_{true} \approx \varphi_{cell}$ is then used in the integrals in (5) and in practice is has been shown that this approach is often sufficiently accurate. Programs like NJOY contain reference functions for φ_{cell} for this purpose.

The next step is to reduce the dependency of (5) on the neutron direction $\hat{\Omega}$. Since the fission reaction is to a very good approximation isotropic and the scattering cross section depends only on the change in direction, (5) can be further simplified. The angular flux can be ex-

pressed in terms of spherical harmonics and the scattering cross section as a Legendre polynomial:

(6) $\varphi_g(\vec{x},\hat{\Omega}) = \sum_{l=0}^{L} \frac{2l+1}{4\pi} \sum_{m=-l}^{+l} \phi_{g,l,m}(\vec{x}) Y_{l,m}(\hat{\Omega})$;

$$\Sigma_{s,g'\to g}(\vec{x},\hat{\Omega}' \to \hat{\Omega}) = \Sigma_{s,g'\to g}(\vec{x},\hat{\Omega}' \cdot \hat{\Omega}) = \sum_{m=-l}^{+l} \Sigma_{s,g'\to g,l}(\vec{x}) P_l(\hat{\Omega}' \cdot \hat{\Omega})$$

The direction independent neutron flux $\phi_g = \phi_{g,0,0}$ and the currents can be calculated as (not writing down the energy dependence explicitly) in the P1 approximation:

(7) $\phi(\vec{x}) = \phi_{0,0}(\vec{x}) = \int d\hat{\Omega}\, \varphi(\vec{x},\hat{\Omega})$;

$$J_x = \int d\hat{\Omega}\, (\hat{n}_x \cdot \hat{\Omega}) \varphi(\vec{x},\hat{\Omega}) = \frac{1}{\sqrt{2}} (\phi_{1,-1} - \phi_{1,1})$$

$$J_y = \int d\hat{\Omega}\, (\hat{n}_y \cdot \hat{\Omega}) \varphi(\vec{x},\hat{\Omega}) = \frac{1}{\sqrt{2}} (\phi_{1,-1} + \phi_{1,1})$$

$$J_z = \int d\hat{\Omega}\, (\hat{n}_z \cdot \hat{\Omega}) \varphi(\vec{x},\hat{\Omega}) = \phi_{1,0}$$

Then the angular flux and the current becomes (Fick’s law):

(8) $\varphi_g(\vec{x},\hat{\Omega}) = \frac{1}{4\pi} \phi_g(\vec{x}) + \frac{3}{4\pi} \hat{\Omega} \cdot J_g(\vec{x})$

$$\vec{J}_g(\vec{x}) = -\frac{1}{3\left(\Sigma_{t,g}(\vec{x}) - \bar{\mu}_o \Sigma_{s,g}(\vec{x})\right)} \nabla \phi_g(\vec{x}) = -D_g(\vec{x}) \cdot \nabla \phi_g(\vec{x})$$

The scalar flux (averaged over energy, too) ϕ is often used in a product with the macroscopic cross sections to generate the reaction rates R:

(9) $R(\vec{x}) = \Sigma_x(\vec{x}) \phi(\vec{x})$

As will be shown later one of the most important principles of cross section homogenization is to preserve reaction rates like (9) per node or fuel assembly.

The above approximations finally lead to the diffusion approximation of the Boltzmann equation:

(9) $\Sigma_{t,g}(\vec{x}) \phi_{g'}(\vec{x}) - \nabla \cdot D_g(\vec{x}) \nabla \phi_g(\vec{x}) = \sum_{g'} \Sigma_{s,g'\to g,0}(\vec{x}) \phi_{g'} + \chi(E_g) \sum_{g'} \nu_{g'} \Sigma_{f,g'}(\vec{x})\, \phi_{g'}$

Equation (9) has been the basis for all LWR reactor design calculations in the 1970s and later. There are many instances where it is not valid: for example, near strongly absorbing regions like in the vicinity of control rods or at the core reflector. Also, in a heterogenous reactor core in which high and low burnup fuel assemblies stand side-by-side or in which UOX and MOX fuel assemblies share a common boundary equation (9) is often not a good ap-

proximation. However, modern nodal reactor simulation codes have been augmented with additional equations which allow (9) to be sufficiently accurate in practice (e.g. []).

For most LWR applications the time dependence in (1) is dropped. This is strictly true if the neutron field is determined only for steady state operation. The time dependence enters on a relatively long timescale of many hours and days in the form of burnup: as the energy generation by fission progresses, fission products are accumulating in the fuel which in turn change the number densities of the cross sections in (3). This time dependence is typically treated in a quasi-static manner: (1) is solved without time dependence, a burnup step is calculated with constant neutron field and then (1) is updated with new cross section data. The equations describing the time dependence of the nuclide vector are called the Bateman equations [] describing the generation and destruction of nuclides:

(10)

In a reactor simulator every relevant core region (e.g. a fuel assembly node) is approximated as an infinite, homogenous medium in which fission products are generated and decaying and higher actinides are bred through neutron capture, and alpha and beta decay channels.

Another reason why time dependence enters (1) is due to the delayed neutrons which are emitted by neutron rich fission products. They are the ultimate reason why the chain reaction in commercial reactors can be controlled by practical means. In this case (1) is augmented by the following set of equations:

$$\text{(11) } \Sigma_t(\vec{x},E,t)\varphi\left(\vec{x},\widehat{\Omega},E,t\right)+\widehat{\Omega}\cdot\nabla\varphi+\frac{1}{v(E)}\cdot\frac{\partial\varphi}{\partial t}=\int_0^{4\pi}d\widehat{\Omega}'\int_0^{\infty}dE'\,\Sigma_s\left(\vec{x},E'\rightarrow E,\widehat{\Omega}'\rightarrow\widehat{\Omega},t\right)\varphi+$$

$$+(1-\beta)\frac{\chi_p(E)}{4\pi}\int_0^{\infty}dE'v(\vec{x},E',t)\Sigma_f(\vec{x},E,t)\varphi+\frac{\chi_d(E)}{4\pi}\sum_k\lambda_k C_k(\vec{x},t)$$

$$\frac{dC_k}{dt}=\beta_k\int_0^{\infty}dE'v(\vec{x},E',t)\Sigma_f(\vec{x},E,t)\varphi\text{ - }\lambda_k C_k$$

The fraction of delayed neutrons is β and the energy spectrum of the delayed and prompt neutrons has been named χ_d and χ_p, respectively.

Typically, a k = 6 or 8 group structure of delayed neutron precursors is considered, i.e. all relevant fission products are grouped according to their decay constants λ_k before emitting a neutron. It is $\sum_k \beta_k = \beta$.

From (11) the classical point reactor kinetics equations can be derived. Details of the derivation can be found in [], for example. For the angular velocity the following Ansatz is made: $\varphi\left(\vec{x},\widehat{\Omega},E,t\right)=N(t)\cdot\Psi\left(\vec{x},\widehat{\Omega},E,t\right)$. The angular integrated flux of Ψ is Φ. The corresponding adjoint solutions are $\Psi^{\dagger}$ and $\Phi^{\dagger}$. Let Ψ fulfill the following eigenvalue equation (neglecting the delayed neutron contributions):

$$(12)\ \Sigma_t(\vec{x},E,t)\Psi\left(\vec{x},\widehat{\Omega},E,t\right)+\widehat{\Omega}\cdot\nabla\Psi=\int_0^{4\pi}d\widehat{\Omega}'\int_0^{\infty}dE'\,\Sigma_s\left(\vec{x},E'\to E,\widehat{\Omega}'\to\widehat{\Omega},t\right)\Psi+$$
$$+(1-\rho(t)\,)\frac{\chi_p(E)}{4\pi}\int_0^{\infty}dE'\upsilon(\vec{x},E',t)\Sigma_f(\vec{x},E,t)\Psi$$

The time dependence of Ψ enters through the time dependence of the cross sections and not because of any time derivative in (12). The quantity $\rho(t)$ plays the role of fundamental eigenvalue and must be determined so that there exists a positive solution Ψ. The main assumption for the following equations is that the time dependent cross sections vary on the same slow timescale on which the neutron precursors decay and that the system is near criticality at all times. The following definitions are made:

$$(13)\ \theta(t)=\frac{\int\ dV\int_{4\pi}d\widehat{\Omega}\int\ dE\Psi^{\dagger}\frac{1}{v}\cdot\frac{d\Psi}{dt}}{\int\ dV\int_{4\pi}d\widehat{\Omega}\int\ dE\Psi^{\dagger}\frac{1}{v}\Psi}$$

$$\frac{1}{\Lambda(t)}=\frac{\int\ dV\left(\int\ dE\Phi^{\dagger}\chi_p\right)\left(\int\ dE\upsilon\Sigma_f\Phi\right)}{4\pi\int\ dV\int_{4\pi}d\widehat{\Omega}\int\ dE\Psi^{\dagger}\frac{1}{v}\Psi}$$

$$\beta(t)=\frac{\int\ dV\left(\int\ dE\Phi^{\dagger}\chi_p\right)\left(\int\ dE\upsilon\beta\Sigma_f\Phi\right)}{\int\ dV\left(\int\ dE\Phi^{\dagger}\chi_p\right)\left(\int\ dE\upsilon\Sigma_f\Phi\right)}$$

$$c_j(t)=c_j(0)e^{-\lambda_j t}+\frac{\int\ dV\left(\int\ dE\Phi^{\dagger}\chi_p\right)\left[\int_0^t dt'e^{-\lambda_j(t-t')}\left(\int\ dE\upsilon\beta_j\Sigma_f\Phi\right)N(t')\right]}{4\pi\int\ dV\int_{4\pi}d\widehat{\Omega}\int\ dE\Psi^{\dagger}\frac{1}{v}\Psi}$$

Then it follows:

$$(14)\ \frac{dN}{dt}+\theta(t)N(t)=\frac{\rho(t)-\beta(t)}{\Lambda(t)}\cdot N(t)+\sum_j\ \lambda_j c_j(t)$$

And if the time dependence of $\Phi^{\dagger}$ is small:

$$(15)\ c_j(t)=c_j(0)e^{-\lambda_j t}+\int_0^t dt'e^{-\lambda_j(t-t')}\frac{\beta_j(t')}{\Lambda(t')}N(t')$$

If $\theta(t)$ is small, then equation (14) and (15) are the classical point reactor kinetic equations.

(12)

1.2. Microscopic cross sections

Neutrons can either be scattered or absorbed. Scattering can be elastic or inelastic, but in this case, they survive the interaction. Absorption can mean capture and creation of a daughter nuclide which subsequently may experience beta decay. Also, photons are often emitted during de-excitation of internal nuclide states. Absorption can also mean re-emission of a secondary particle or of a set of particles like another neutron, proton or alpha particle together with photons. Finally, absorption can also lead to fission and other reactions.

The microscopic scattering cross sections are proportional to the probability that a given reaction channel is occurring. Therefore $\sigma(E, E', \hat{\Omega}, \hat{\Omega}')$ is proportional to the probability that a neutron with energy E and direction $\hat{\Omega}$ is scattered into energy E' and direction $\hat{\Omega}'$. Absorption reactions like (n,γ), (n,α), (n,2n') or (n, fission) depend only on the incoming neutron energy and cross sections have a $\sigma(E)$ functional dependence. If the secondary particle field is of interest, too, probabilities for secondary particle energy and direction must be given as additional cross section dependences.

Microscopic cross section data for neutron interactions are available in a number of libraries, for example ENDF/B-VII.1 [] or JEFF-3.3 []. The cross sections are stored in a standardized format, called the ENDF (evaluated nuclear data file) format libraries []. For practical use the data must be preprocessed and the standard code for this purpose is often NJOY []. To inspect the data tools like JANIS (https://www.oecd-nea.org/janis/) or Sigma (https://www.nndc.bnl.gov/sigma/) can be used.

The formats were upgraded with each version to handle new features, for example, the extension from the original upper limit of 15 MeV to 20 MeV, the addition of photon production information, the introduction of new resonance formats, or the addition of charged-particle

data. Control over the ENDF formats has been retained by the US Cross Section Evaluation Working Group and the format specifications are published through the National Nuclear Data center (https://www.nndc.bnl.gov/).

1.3. Cross section homogenization

Koepke’s generalized equivalence theory [] today has become the standard method to improve the accuracy of (9) and is the basis of all existing nodal solvers. The following assumptions are made:

- .
- .

1.4. Methods of characteristics

The method of characteristics (MOC) is one technique for solving partial differential equations. MOC is one of the most common methods with real world applications in commercial reactor physics spectral codes used today [Smith]. MOC is used to solve the Boltzmann equation in 2D by discretizing both polar and azimuthal angles and integrating the multi-group form of the equation for a particular azimuthal and polar angle quadrature [Boyd].

Als es die ersten Hügel des Kursivgebirges erklommen hatte, warf es einen letzten Blick zurück auf die

1.5. Uncertainty and sensitivity analysis

Eines Tages aber beschloss eine kleine Zeile Blindtext, ihr Name war Lorem Ipsum, hinaus zu gehen in die weite Grammatik. Der große Oxmox riet ihr davon ab, da es dort wimmele von bösen Kommata, wilden Fragezeichen und hinterhältigen Semikoli, doch das Blindtextchen ließ sich nicht beirren. Es packte seine sieben Versalien, schob sich sein Initial in den Gürtel und machte sich auf den Weg.

Als es die ersten Hügel des Kursivgebirges erklommen hatte, warf es einen letzten Blick zurück auf die

1.6. The OpenMOC code

OpenMOC is a Method of Characteristics (MOC) neutral particle transport code for reactor physics criticality calculations (https://mit-crpg.github.io/OpenMOC/). It is capable of simulating 2D assembly and full-core reactor models based on constructive solid geometry with second-order surfaces. High-performance parallel solvers for multi-core CPUs and GPUs are actively pursued as part of the OpenMOC project.

All OpenMOC source code is hosted on GitHub (https://github.com/mit-crpg/OpenMOC) and is provided to users as a Python API. The back-end of the code written in C/C++ and CUDA [].

A basic model consists of:

- **Geometry**: a description of the geometry split into regions of homogeneous materials
- **Materials**: a description of the nuclear cross-sections for each material
- **Parameters**: various parameters for the numerical algorithm used in the simulation.

OpenMOC uses constructive solid geometry (CSG) to represent complex reactor models.The constructive solid geometry formulation is the method of choice for many advanced reactor modeling software packages. CSG allows complex spatial models to be built using Boolean operations - such as intersections and unions - of simple surfaces and building blocks termed *primitives*. This is well suited for commercial LWRs whose cores are built out of a rectangular lattice of fuel assemblies, each of which is a rectangular lattice of fuel pins.

The constructive solid geometry formulation in OpenMOC is predicated upon the use of several key objects which allow one to construct a spatial model from simple primitives in a hierarchical fashion. OpenMOC is presently only capable of describing the 2D xy-plane. A typical hierarchy for the way in which surfaces, universes, cells and lattices are constructed to represent a reactor model in OpenMOC is illustrated in figure x.

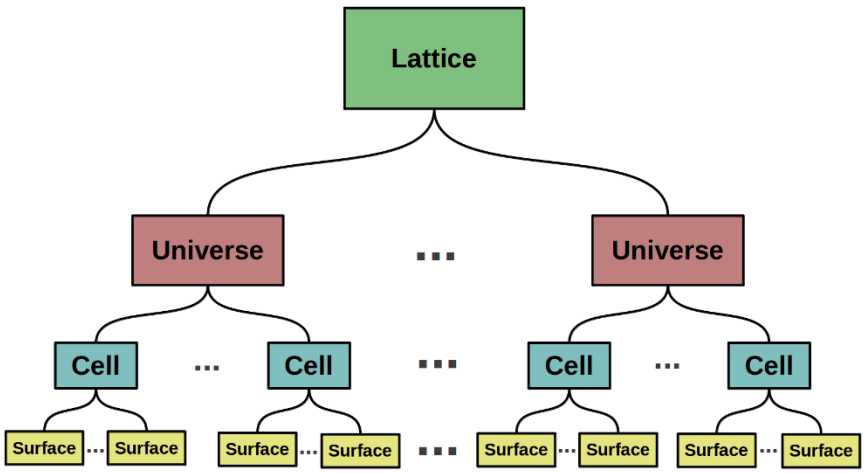

Figure 1: OpenMOC geometry hierarchy (https://mit-crpg.github.io/OpenMOC/_images/csg-primitives-hierarchy.png).

A cell is defined to be the region bounded by a boolean combination of surface halfspaces. Presently, OpenMOC only permits the use of halfspace intersections and does not support other boolean operations such as unions and differences. The region deined by the cell is subsequently filled by either a material or a universe.

A universe is a collection of one or more cells that fill the entirety of the xy-plane. Each cell may be filled with a material or a separate universe. Universes allow unique structures to be created from cells, and for simple replication of that structure throughout a model by placing it in various locations throughout the geometry.

Lattices are useful in the case for reactor cores which typically contain rectangular or hexagonal arrays of fuel pins. OpenMOC currently only contains a single lattice implementation for 2D Cartesian arrays. Each lattice is uniquely specified by the number of array elements along the x and y axes, the width and height of each lattice cell, and the universe to fill each lattice cell. The lattice specification represents a coordinate transformation such that the center of each lattice cell maps to the origin of the universe within it.

OpenMOC uses multi-group macroscopic nuclear cross sections, provided by the user. It does not perform self-shielding or depletion calculations and have to be provided by preprocessing microscopic data with other codes like NJOY [] or ORIGEN [].

OpenMOC includes a module for importing nuclear data cross sections from an HDF5 binary file (http://www.hdfgroup.org/HDF5/). The multi-group cross sections to assign by material or cell must be defined as an HDF5 group with a string name or integer ID to identify the material or cell. The material group must contain the following floating point HDF5 datasets of multi-group cross section data:

- 'total' or 'transport' or 'nu-transport'
- 'nu-scatter matrix' or 'scatter matrix'
- 'chi'
- 'nu-fission'
- 'fission'

Each dataset should be a 1D array of floating-point values ordered by increasing energy group (*i.e.*, from highest to lowest energies). This includes the scattering matrix which should be inner strided by outgoing energy group and outer strided by incoming energy group.

1.7. The NJOY preprocessing code

The NJOY Nuclear Data Processing System is a modular computer code used for converting evaluated nuclear data in the ENDF format into libraries useful for applications calculations. NJOY handles a wide variety of nuclear effects, including resonances, Doppler broadening, heating (KERMA), radiation damage, thermal scattering, gas production, neutrons and charged particles, photo-atomic interactions, photonuclear reactions, self shielding, probability tables, photon production, and high-energy interactions (to 150 MeV). Output can include printed listings, special library files for applications, and Postscript graphics (https://t2.lanl.gov/nis/njoy/index.html).

The original NJOY code was written in FORTRAN77 and is not open source. It was later converted into FORTRAN-90 and is now available as NJOY21 as open source and written in C++ (http://www.njoy21.io/).

The most important modules of NJOY are []:

- **NJOY**: directs the flow of data through the other modules. Subsidiary modules for locale, ENDF formats, physics constants, utility routines, and math routines are grouped with the NJOY module for descriptive purposes

- **RECONR**: reconstructs pointwise (energy-dependent) cross sections from ENDF resonance parameters and interpolation schemes

- **BROADR**: Doppler-broadens and thins pointwise cross sections

- **UNRESR**: computes effective self-shielded pointwise cross sections in the unresolved energy range

- **THERMR**: produces cross sections and energy-to-energy matrices for free or bound scatterers in the thermal energy range

- **GROUPR**: generates self-shielded multigroup cross sections, group-to-group scattering matrices, photon production matrices, and charged-particle multigroup cross sections from pointwise input

- **ERRORR**: computes multigroup covariance matrices from pointwise covariance data.

Each module reads some data from one or more input files, transforms it, and writes the results on one or more output files. The order that the modules are used and the particular input and output files used are specified as in the following example:

```
[copy an ENDF file onto "tape20"]

  reconr
  20 21
    ...specific input for RECONR...
  broadr
  20 21 22
    ...specific input for BROADR...
  groupr
  20 22 0 23
    ...specific input for GROUPR...
  stop

  [copy tape24 to the DTF library]
```

In this case, RECONR reads the original ENDF tape and reconstructs the resonances and nonlinear interpolation schemes to prepare a PENDF (pointwise ENDF) tape on unit 21. BROADR reads the PENDF tape, Doppler broadens the pointwise cross sections, and writes a new temperature dependent PENDF tape on unit 22. Sometimes it needs some data from the ENDF tape, and this is why tape20 is also provided as input. Next, the information from the ENDF and PENDF tapes is run through the multigroup averaging process and written in GENDF format (a special groupwise variation of the ENDF format) on tape23.

2. Cross section homogenization with OpenMOC

In this project a fuel assembly geometry for a typical pressurized water reactor (PWR) is considered. A horizontal plot of such an assembly is shown in figure X. It has 17x17-20 fuel rods and 20 guide tubes for control rod insertion. The fuel rod outer and inner cladding radius is x and x cm. The cladding material is considered standard Zircaloy-4. The fuel pellet outer radius is x cm. The pitch between fuel rods is x cm. The inner and outer diameter of the guide tubes is x and x cm and they are also made of Zircaloy.

The pellets are assumed to be of type UO2 with an enrichment of 4 w/o U235 and a density of x g/cm3. The water and Zircaloy density are x and x g/cm3. An average temperature of the fuel, the H2O moderator and the Zircaloy cladding of x K, x K and x K is assumed.

The pitch between two neighboring fuel assemblies is x cm. For the purpose of spectral calculations and hence cross section homogenization the fuel assemblies are assumed to be of infinite length in the z direction.

2.1. Input generation

The above described fuel assembly geometry can be created with OpenMOC with the following Python script:

In the above script the numpy [] array X holds the few-energy group cross sections which have been generated with NJOY for the UO2 fuel region, the cladding and the H20 moderator.

2.2. Ensemble calculation loop

For the execution of the total Monte Carlo method a new, multi-dimensional array X has been defined which stored the ensemble of cross sections generated with NJOY with the help of the covariance information.

The following Python scripts loops over every cross section set and stores the resulting homogenized cross sections in array X.

3. Cross section preparation with NJOY

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen Sie in Buchstabhausen an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort und versorgt sie mit den nötigen Regelialien. Es ist ein paradiesmatisches Land, in dem einem gebratene Satzteile in den Mund fliegen. Nicht einmal von der allmächtigen Interpunktion werden die Blindtexte beherrscht – ein geradezu unorthographisches Leben.

3.1. Boltzmann equation for neutron transport

Eines Tages aber beschloss eine kleine Zeile Blindtext, ihr Name war Lorem Ipsum, hinaus zu gehen in die weite Grammatik. Der große Oxmox riet ihr davon ab, da es dort wimmele von bösen Kommata, wilden Fragezeichen und hinterhältigen Semikoli, doch das Blindtextchen ließ sich nicht beirren. Es packte seine sieben Versalien, schob sich sein Initial in den Gürtel und machte sich auf den Weg.

Als es die ersten Hügel des Kursivgebirges erklommen hatte, warf es einen letzten Blick zurück auf die

4. Total Monte Carlo results

Weit hinten, hinter den Wortbergen, fern der Länder Vokalien und Konsonantien leben die Blindtexte. Abgeschieden wohnen Sie in Buchstabhausen an der Küste des Semantik, eines großen Sprachozeans. Ein kleines Bächlein namens Duden fließt durch ihren Ort und versorgt sie mit den nötigen Regelialien. Es ist ein paradiesmatisches Land, in dem einem gebratene Satzteile in den Mund fliegen. Nicht einmal von der allmächtigen Interpunktion werden die Blindtexte beherrscht – ein geradezu unorthographisches Leben.

4.1. Boltzmann equation for neutron transport

Eines Tages aber beschloss eine kleine Zeile Blindtext, ihr Name war Lorem Ipsum, hinaus zu gehen in die weite Grammatik. Der große Oxmox riet ihr davon ab, da es dort wimmele von bösen Kommata, wilden Fragezeichen und hinterhältigen Semikoli, doch das Blindtextchen ließ sich nicht beirren. Es packte seine sieben Versalien, schob sich sein Initial in den Gürtel und machte sich auf den Weg.

Als es die ersten Hügel des Kursivgebirges erklommen hatte, warf es einen letzten Blick zurück auf die

5. Conclusions

One important task for future work is to extend the OpenMOC code with the Bateman equations to generate cross sections as a function of burnup. In commercial spectral codes cross section tables with burnup steps of about 5 MWd/kgU are generated. Also, for typical reactor core simulations different temperatures and control rod states must be taken into account. This leads to a tree of pre-calculated, homogenized cross sections as exemplified in figure x.

N:\Habitat\R&D\Mohammad\!2018\!nodal diffusion methods\ Andreas Pautz Dissertation Uni München 601539

N:\Habitat\R&D\Mohammad\!2018\!nodal diffusion methods\ MIT-EL-86-002-19764322 development of nodal reactor solutions

N:\Habitat\R&D\Mohammad\!2018\videos\010418\ neutron transport theorey University of Pensilvania nu-ce521_chapter1_reading

N:\Habitat\R&D\Mohammad\!2018\videos\100318\ McMaster UNCERTAINTY IN LATTICE REACTOR PHYSICS

N:\Habitat\R&D\Mohammad\!2017\!03\ Neutronic Analysis For Nuclear Reactor Systems-Springer (2016)

N:\Habitat\R&D\Mohammad\!2017\ neutron numerical solutions BARBARINO_phdthesis

N:\Habitat\R&D\Mohammad\!2017\ review of neutron transport equation nse_v80_n4_pp481-535

N:\Habitat\R&D\Mohammad\!2017\!01\ MOC Development of a 3D MOC methodology JBTThesisFINAL

N:\NuclearLiterature\Chiriac Literatur\ handbook of nuclear engineering

N:\Habitat\R&D\Mohammad\!2019\ Development of a Transient Multiphysics solver for nuclear fuel assemblies 218901

https://t2.lanl.gov/nis/njoy/njoy02.html

https://www.njoy21.io/NJOY2016/