method for monte carlo depletion uncertainty propagation

aslak stubsgaard

March 21, 2025

Abstract

monte carlo code, such as openmc, can perform discretized iterative depletion calculations with intermediate calculations to update the transition matrix, however, such codes often don't estimate uncertainty or uncertainty propagation. here the analytical uncertainty propagation is derived for the matrix exponential solution, with the intent that this can be implemented to estimate uncertainty propagation in monte carlo depletion simulation tools.

introduction

when performing monte carlo depletion simulations the initial material quantity will have a certain confidence or uncertainty while nuclear data, material properties, material transfer (i.e. potential online or offline processing schemes), and numerical calculation will all affect the uncertainties of the calculated depletion material quantity at a some later time. furthermore, these uncertainties are compounding for every depletion step requiring uncertainty propagation to estimate the total final uncertainty. these uncertainties can be estimated using different methods, such as sensitivity analysis, running calculation with a large set of input parameters and comparing the final results, alternatively sensitivity analysis or other methods of estimating the individual timestep transition matrices can be performed and their uncertainties propagated. here the latter method is explored.

considering the system of $N_1, N_2, \ldots, N_n$ isotopes, that obeys the first order linear ordinary differential equations

$$\frac{dN_i}{dt} = \sum_{j \neq i} \lambda_{ij} N_j - \lambda_i N_i, \quad \forall i \in \{1, \ldots, n\} \tag{1}$$

where λ_i is the decay constant for N_i and λ_{ji} is the reaction rate yielding N_i from isotope N_j.

assuming $\lambda_{ij} \in \mathbb{R}$, writing $\boldsymbol{N}(t) = (N_1(t), N_2(t), \ldots, N_n(t))^T \in \mathbb{R}^n$, with $(\boldsymbol{A})_{ij} = \lambda_{ij},\ \forall i \neq j$, and $(\boldsymbol{A})_{ii} = \lambda_i$, such that $\boldsymbol{A} \in \mathbb{R}^{n \times n}$. then the differential equation can be written in matrix notation as

$$\dot{\boldsymbol{N}}(t) = \boldsymbol{A}\boldsymbol{N}(t). \tag{2}$$

this has the well known matrix exponential solution

$$\boldsymbol{N}(t) = e^{\boldsymbol{A}t}\boldsymbol{N}(0), \tag{3}$$

where the exponential can be expressed as a taylor series $e^{\boldsymbol{A}t} = \sum_{i=0}^{\infty} \frac{1}{i!}(\boldsymbol{A}t)^i$ and the zeroth order $\boldsymbol{A}$ is just the identity matrix.

performing an eigenvalue decomposition on $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$, assuming that $\boldsymbol{A}$ is diagonalizable, one gets $\boldsymbol{A}^i = \boldsymbol{V}\boldsymbol{\Lambda}^i\boldsymbol{V}^{-1}$, where $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ holds the eigenvectors and $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times n}$ holds the eigenvalue on its diagonal.

using the decomposition, the problem can be written without any high order matrix multiplication, namely

$$e^{\boldsymbol{A}t} = \boldsymbol{V}e^{\boldsymbol{\Lambda}t}\boldsymbol{V}^{-1} = \boldsymbol{V}\begin{pmatrix} e^{\boldsymbol{\lambda_1}t} & & \\ & \ddots & \\ & & e^{\boldsymbol{\lambda_n}t} \end{pmatrix}\boldsymbol{V}^{-1}. \tag{4}$$

since the reaction rates in the off diagonal terms of $\boldsymbol{A}$ depends on $\boldsymbol{N}$, the above solution is only accurate for timestep sizes where $\boldsymbol{N}$ doesn't change substantially, and to obtain $\boldsymbol{N}(t)$ at an arbitrary time t, the calculation needs to be discretized, $\boldsymbol{A}$ recalculated at each timestep, and a new matrix decomposition performed at each timestep. in practice depletion codes use truncated taylor series approximations to calculate the matrix exponential.

to make the multi isotope derivation for the uncertainty propagation easier to follow, the simplified single isotope case is derived first.

single isotope depletion uncertainty propagation

considering the simple system of only one isotope N and m discretized time-steps with transition constants λ_k $\forall k \in \{1, \ldots, m\}$, the transition matrix reduces to λ_k and iterative depletion equation then takes the form

$$N_{k+1} = e^{\lambda_k (t_{k+1} - t_k)} N_k, \quad \forall k \in \{1, \ldots, m\}. \tag{5}$$

assigning λ_k an uncertainty σ_{λ_k} and assigning N_0 an uncertainty σ_{N_0} means that each iteration will have uncertainty σ_{N_k}. given λ_k and σ_{λ_k} $\forall k \in \{1, \ldots, m\}$, and N_0 and σ_{N_0} and the non-linear differential equation for N_{k+1} one can approximate the uncertainty $\sigma_{N_{k+1}}$ as the first order taylor series expansion

$$\sigma^2_{N_{k+1}} = \left(\frac{\partial N_{k+1}}{\partial N_k}\right)^2 \sigma^2_{N_k} + \left(\frac{\partial N_{k+1}}{\partial \lambda_k}\right)^2 \sigma^2_{\lambda_k} + \left(\frac{\partial N_{k+1}}{\partial N_k}\right)\left(\frac{\partial N_{k+1}}{\partial \lambda_k}\right) \mathrm{Cov}\left(N_k, \lambda_k\right). \tag{6}$$

assuming N_k and λ_k are uncorrelated, i.e. $\mathrm{Cov}\left(N_k, \lambda_k\right) = 0$, and that λ_k nondependent on N_k, i.e. $\frac{\partial \lambda_k(N_k)}{\partial N_k} = 0$, gives

$$\sigma^2_{N_{k+1}} = \left(e^{\lambda_k (t_{k+1} - t_k)}\right)^2 \sigma^2_{N_k} + \left((t_{k+1} - t_k) N_{k+1}\right)^2 \sigma^2_{\lambda_k}. \tag{7}$$

this expression allows for the approximate propagation of the uncertainties in each discrete time-steps.

equation 7 is the easiest to implement to get approximate uncertainty propagation in practice, and would likely be sufficient for most applications.

if instead it's assumed that $\frac{\partial \lambda_k(N_k)}{\partial N_k} \neq 0$ and the dependency explicitly expressed as $N_{k+1} = e^{\lambda_k(N_k)(t_{k+1} - t_k)} N_k$. then, using the product rule and chain rule, the first term becomes

$$\begin{aligned}
\frac{\partial N_{k+1}}{\partial N_k} &= (e^{\lambda_k(N_k)(t_{k+1} - t_k)}) \frac{\partial}{\partial N_k}(N_k) + \frac{\partial}{\partial N_k}\left(e^{\lambda_k(N_k)(t_{k+1} - t_k)}\right) N_k \\
&= e^{\lambda_k(N_k)(t_{k+1} - t_k) + \frac{\partial \lambda_k(N_k)}{\partial N_k} \frac{\partial}{\partial \lambda_k(N_k)}\left(e^{\lambda_k(N_k)(t_{k+1} - t_k)}\right) N_k} \\
&= e^{\lambda_k(N_k)(t_{k+1} - t_k)} + \frac{\partial \lambda_k(N_k)}{\partial N_k}(t_{k+1} - t_k) N_{k+1}.
\end{aligned} \tag{8}$$

thus, including λ_k's N_k-dependency, but still assuming $\mathrm{Cov}\left(N_k, \lambda_k\right) = 0$, the that the full expression for the uncertainty propagation is

$$\sigma^2_{N_{k+1}} = \left(e^{\lambda_k (t_{k+1} - t_k)} + (t_{k+1} - t_k) \frac{\partial \lambda_k(N_k)}{\partial N_k} N_{k+1}\right)^2 \sigma^2_{N_k} + \left((t_{k+1} - t_k) N_{k+1}\right)^2 \sigma^2_{\lambda_k}. \tag{9}$$

note that this neglects the numerical uncertainty $\sigma_{\lambda_k(N_k \pm \sigma_{N_k}/2)}$ associated with this calculation.

since $\lambda_k(N_k)$ or $\lambda_{k|N_k}$ doesn't have a analytical expression the derivative $\frac{\partial \lambda_k(N_k)}{\partial N_k}$ can be approximated numerically as

$$\frac{\partial \lambda_k(N_k)}{\partial N_k} = \frac{\lambda_{k|N_k + {}^{\sigma_{N_k}}/_2} - \lambda_{k|N_k - {}^{\sigma_{N_k}}/_2}}{\sigma_{N_k}}, \tag{10}$$

assuming σ_{N_k} is small. this is akin to a sensitivity analysis of λ_k for each time step, and would in practice mean doing three calculations per time step, with the three inputs $N_k - {}^{\sigma_{N_k}}/_2$, N_k, and $N_k + {}^{\sigma_{N_k}}/_2$ instead of just N_k. more practically, the $N_k + \sigma_{N_k}$ and N_k can be used for the basis of the approximation so only one additional monte carlo transport step is needed per timestep.

alternatively, openmc has the capability of calculating the tally derivatives which could also be used to estimate $\frac{\partial \lambda_k(N_k)}{\partial N_k}$.

using equation 9 instead of equation 7 would yield more accurate uncertainty propagation but also involve substantially higher computational cost.

with this simplified case in mind, the method can easily be extended to the multi-isotope system and simplified using some matrix notation.

matrix depletion uncertainty propagation

assuming m distritized timesteps, that $(\boldsymbol{A}_k)_{ij} \equiv \lambda_{ij,k}$, and that $\lambda_{ij,k}$ is the decay, reaction, and transfer rates $\forall k \in \{1,\dots,m\}$, such that $\boldsymbol{A}_k \in \mathbb{R}^{n\times n}$, $\boldsymbol{N}_k = (N_{1,k}, N_{2,k}, \dots, N_{n,k})^T \in \mathbb{R}^n$, and $\dot{\boldsymbol{N}}_k = \boldsymbol{A}_k \boldsymbol{N}_k$ $\forall k \in \{1,\dots,m\}$. then the iterative depletion takes the form

$$\boldsymbol{N}_{k+1} = e^{\boldsymbol{A}_k(t_{k+1}-t_k)}\boldsymbol{N}_k, \quad \forall k \in \{1,\dots,m\}. \tag{11}$$

assume $(\boldsymbol{\sigma}_{\boldsymbol{N}_k})_i \equiv \sigma_{N_{i,k}}$ and $(\boldsymbol{\sigma}_{\boldsymbol{A}_k})_{ij} \equiv \sigma_{\lambda_{ij,k}}$, where $\sigma_{N_{i,k}}$ and $\sigma_{\lambda_{ij,k}}$ are the respective uncertainties of $N_{i,k}$ and $\lambda_{ij,k}$, $\forall k \in \{1,\dots,m\}$.

following the same approach as in single isotope depletion uncertainty propagation, the p'th isotopes abundance uncertainty at time t_{k+1} can be approximated by the first order taylor series expansion

$$\begin{aligned}\left[(\boldsymbol{\sigma}_{\boldsymbol{N}_{k+1}})_p\right]^2 = &\sum_q \left[\frac{\partial(\boldsymbol{N}_{k+1})_p}{\partial(\boldsymbol{N}_k)_q}\right]^2 \left[(\boldsymbol{\sigma}_{\boldsymbol{N}_k})_q\right]^2 + \sum_j \left[\frac{\partial(\boldsymbol{N}_{k+1})_p}{\partial\lambda_{pj,k}}\right]^2 \sigma^2_{\lambda_{pj,k}} \\ &+ \sum_q \sum_j \left[\frac{\partial(\boldsymbol{N}_{k+1})_p}{\partial(\boldsymbol{N}_k)_q}\right]\left[\frac{\partial(\boldsymbol{N}_{k+1})_p}{\partial\lambda_{pj,k}}\right] \mathrm{Cov}\left((\boldsymbol{N}_k)_q, \lambda_{pj,k}\right) \\ &+ \sum_{j\neq l} \left[\frac{\partial(\boldsymbol{N}_{k+1})_p}{\partial\lambda_{pj,k}}\right]\left[\frac{\partial(\boldsymbol{N}_{k+1})_p}{\partial\lambda_{pl}}\right] \mathrm{Cov}\left(\lambda_{pj,k}, \lambda_{pl}\right).\end{aligned} \tag{12}$$

looking at the derivative in the first expression on the right hand side of equation 12, and explicitly writing $\boldsymbol{A}_k$'s dependency on $\boldsymbol{N}_k$, gives

$$\begin{aligned}\frac{\partial(\boldsymbol{N}_{k+1})_p}{\partial(\boldsymbol{N}_k)_q} &= \frac{\partial}{\partial(\boldsymbol{N}_k)_q}\left(e^{\boldsymbol{A}_k(\boldsymbol{N}_k)(t_{k+1}-t_k)}\boldsymbol{N}_k\right)_p = \frac{\partial}{\partial(\boldsymbol{N}_k)_q}\left(\sum_j \left(e^{\boldsymbol{A}_k(\boldsymbol{N}_k)(t_{k+1}-t_k)}\right)_{pj}(\boldsymbol{N}_k)_j\right) \\ &= \sum_j \left(\frac{\partial(\boldsymbol{A}_k)_{pj}}{\partial(\boldsymbol{N}_k)_q}\right)(t_{k+1}-t_k)\left(e^{\boldsymbol{A}_k(\boldsymbol{N}_k)(t_{k+1}-t_k)}\right)_{pj}(\boldsymbol{N}_k)_j \\ &+ \left(\sum_j \left(e^{\boldsymbol{A}_k(\boldsymbol{N}_k)(t_{k+1}-t_k)}\right)_{pj}\frac{\partial(\boldsymbol{N}_k)_j}{\partial(\boldsymbol{N}_k)_q}\right).\end{aligned} \tag{13}$$

looking at the derivative in the second term on the right hand side of equation 12

$$\begin{aligned}\frac{\partial(\boldsymbol{N}_{k+1})_p}{\partial\lambda_{pj,k}} &= \frac{\partial}{\partial\lambda_{pj,k}}\left(e^{\boldsymbol{A}_k(\boldsymbol{N}_k)(t_{k+1}-t_k)}\boldsymbol{N}_k\right)_p = \frac{\partial}{\partial\lambda_{pj,k}}\left(\sum_l \left(e^{\boldsymbol{A}_k(\boldsymbol{N}_k)(t_{k+1}-t_k)}\right)_{pl}(\boldsymbol{N}_k)_l\right) \\ &= \sum_l \left(\frac{\partial(\boldsymbol{A}_k)_{pl}}{\partial\lambda_{pj,k}}\right)(t_{k+1}-t_k)\left(e^{\boldsymbol{A}_k(\boldsymbol{N}_k)(t_{k+1}-t_k)}\right)_{pl}(\boldsymbol{N}_k)_l \\ &= (t_{k+1}-t_k)\left(e^{\boldsymbol{A}_k(\boldsymbol{N}_k)(t_{k+1}-t_k)}\right)_{pj}(\boldsymbol{N}_k)_j\end{aligned} \tag{14}$$

neglecting correlations and the $\frac{\partial\lambda_{pj,k}(\boldsymbol{N}_k)}{\partial(\boldsymbol{N}_k)_q}$ term, equation 12 reduces to

$$\left[(\boldsymbol{\sigma}_{\boldsymbol{N}_{k+1}})_p\right]^2 = \sum_q \left[\left(e^{\boldsymbol{A}_k(t_{k+1}-t_k)}\right)_{pq}\right]^2 \left[(\boldsymbol{\sigma}_{\boldsymbol{N}_k})_q\right]^2 + \sum_j \left[(t_{k+1}-t_k)\left(e^{\boldsymbol{A}_k(\boldsymbol{N}_k)(t_{k+1}-t_k)}\right)_{pj}(\boldsymbol{N}_k)_j\right]^2 \sigma^2_{\lambda_{pj,k}} \tag{15}$$

using the Hadamard product $\circ$ [1]and denoting the Hadamard product of the quantity with itself as ^{②}, the equation can be written as

$$\boldsymbol{\sigma}^{②}_{\boldsymbol{N}_{k+1}} = \left(e^{\boldsymbol{A}_k(t_{k+1}-t_k)}\right)^{②}\boldsymbol{\sigma}^{②}_{\boldsymbol{N}_k} + (t_{k+1}-t_k)^2\boldsymbol{\sigma}^{②}_{\boldsymbol{A}_k}\boldsymbol{N}^{②}_{k+1} \tag{16}$$

thus, in order to calculated the propagated uncertainty of $\boldsymbol{\sigma}_{\boldsymbol{N}_{k+1}}$ it's required of the user to specifying $\boldsymbol{\sigma}_{\boldsymbol{N}_0}$ and to specify and calculating $\sigma_{\boldsymbol{A}_i}$ $\forall i \in \{1,\dots,m\}$, along with using the solved the matrix exponential solution for

[1]For $A, B \in \mathbb{M}^{m\times n}$ the Hadamard product: $A \circ B \in \mathbb{M}^{m\times n}$ is defined by $(A\circ B)_{ij} = (A)_{ij}(B)_{ij}$ and $A^{②} \equiv A \circ A$

each timestep, already given by the depletion routine. thus, the added computational cost for the uncertainty propagation to estimate $\sigma_{\boldsymbol{A}_i}$ for each timestep is minimal.

this method ignore the error introduced from the numerical calculation of the matrix exponential solution, e.g. errors introduced from the taylor series expansion or particle transport calculations.

the ignored $\frac{\partial(\boldsymbol{A}_k)_{pj}}{\partial(\boldsymbol{N}_k)_q} = \frac{\partial\lambda_{pj,k}(\boldsymbol{N}_k)}{\partial(\boldsymbol{N}_k)_q}$ term can be approximated numerically as

$$\frac{\partial\lambda_{pj,k}(\boldsymbol{N}_k)}{\partial(\boldsymbol{N}_k)_q} = \frac{\lambda_{pj,k}(\ldots,(\boldsymbol{N}_k)_q + (\sigma_{\boldsymbol{N}_k})_q/2,\ldots) - \lambda_{pj,k}(\ldots,(\boldsymbol{N}_k)_q - (\sigma_{\boldsymbol{N}_k})_q/2,\ldots)}{(\sigma_{\boldsymbol{N}_k})_q}. \tag{17}$$

this approach can more loosely be generalized to get the following approximate expression

$$\boldsymbol{\sigma}^{②}_{\boldsymbol{N}_{k+1}} = \left(e^{\boldsymbol{A}_k(t_{k+1}-t_k)} + (t_{k+1}-t_k)\frac{\partial\boldsymbol{A}_k}{\partial\boldsymbol{N}_k}N_{k+1}\right)^{②}\boldsymbol{\sigma}^{②}_{\boldsymbol{N}_k} + (t_{k+1}-t_k)^2\boldsymbol{\sigma}^{②}_{\boldsymbol{A}_k}\boldsymbol{N}^{②}_{k+1}. \tag{18}$$

two isotope matrix form

as an example, consider the two isotope system

$\boldsymbol{N}_k = \begin{bmatrix} N_{1,k} \\ N_{2,k} \end{bmatrix}$ and $\boldsymbol{A}_k = \begin{bmatrix} \lambda_{11,k} & \lambda_{12,k} \\ \lambda_{21,k} & \lambda_{22,k} \end{bmatrix}$,

$\boldsymbol{\sigma}_{\boldsymbol{N}_k} = \begin{bmatrix} \sigma_{N_{1,k}} \\ \sigma_{N_{2,k}} \end{bmatrix}$ and $\boldsymbol{\sigma}_{\boldsymbol{A}_k} = \begin{bmatrix} \sigma_{\lambda_{11,k}} & \sigma_{\lambda_{12,k}} \\ \sigma_{\lambda_{21,k}} & \sigma_{\lambda_{22,k}} \end{bmatrix}$,

and defining $\boldsymbol{M} = e^{\boldsymbol{A}_k(t_{k+1}-t_k)} = \begin{bmatrix} M_{11,k} & M_{12,k} \\ M_{21,k} & M_{22,k} \end{bmatrix}$ as the calculated matrix exponential,

then equation 16, $\boldsymbol{\sigma}^{②}_{\boldsymbol{N}_{k+1}} = \left(e^{\boldsymbol{A}_k(t_{k+1}-t_k)}\right)^{②}\boldsymbol{\sigma}^{②}_{\boldsymbol{N}_k} + (t_{k+1}-t_k)^2\boldsymbol{\sigma}^{②}_{\boldsymbol{A}_k}\boldsymbol{N}^{②}_{k+1}$, takes the form

$$\begin{aligned} \begin{bmatrix} \sigma^2_{N_{1,k}} \\ \sigma^2_{N_{2,k}} \end{bmatrix} &= \begin{bmatrix} M^2_{11,k} & M^2_{12,k} \\ M^2_{21,k} & M^2_{22,k} \end{bmatrix}\begin{bmatrix} \sigma^2_{N_{1,k}} \\ \sigma^2_{N_{2,k}} \end{bmatrix} + (t_{k+1}-t_k)^2\begin{bmatrix} \sigma^2_{\lambda_{11,k}} & \sigma^2_{\lambda_{12,k}} \\ \sigma^2_{\lambda_{21,k}} & \sigma^2_{\lambda_{22,k}} \end{bmatrix}\begin{bmatrix} N^2_{1,k+1} \\ N^2_{2,k+1} \end{bmatrix} \\ &= \begin{bmatrix} M^2_{11,k}\sigma^2_{N_{1,k}} + M^2_{12,k}\sigma^2_{N_{2,k}} + (t_{k+1}-t_k)^2\left(\sigma^2_{\lambda_{11,k}}N^2_{1,k+1} + \sigma^2_{\lambda_{12,k}}N^2_{2,k+1}\right) \\ M^2_{21,k}\sigma^2_{N_{1,k}} + M^2_{22,k}\sigma^2_{N_{1,k}} + (t_{k+1}-t_k)^2\left(\sigma^2_{\lambda_{21,k}}N^2_{1,k+1} + \sigma^2_{\lambda_{22,k}}N^2_{2,k+1}\right) \end{bmatrix}. \end{aligned} \tag{19}$$

coupled materials

in the case of multiple coupled materials, e.g. with transfer rates between different materials, the same approach for uncertainty propagation can be used.

consider s coupled materials $\boldsymbol{N}_{k;i} = (N_{1,k;i}, N_{2,k;i}, \ldots, N_{n_i,k;i})^T \in \mathbb{R}^{n_i}$ with individual transition matrices $\boldsymbol{A}_{k;i} \in \mathbb{R}^{n_i\times n_i}\ \forall i \in \{1,\ldots,s\}$ and sparce transfer rate matrices $\boldsymbol{T}_{k;ij} \in \mathbb{R}^{n_i\times n_j}\ \forall i,j \in \{1,\ldots,s\}$ where $i \neq j$ describing the rates from material j to i.

redefinding $\boldsymbol{N}_k = (\boldsymbol{N}_{k;1}, \boldsymbol{N}_{k;2}, \ldots, \boldsymbol{N}_{k;s})^T \in \mathbb{R}^{\sum_{i=1}^s n_i}$ and $(\boldsymbol{A}_k)_{ii} \equiv \boldsymbol{A}_{k;i}$ and $(\boldsymbol{A}_k)_{ij} \equiv \boldsymbol{T}_{k;ij}$ yielding $\boldsymbol{A}_k \in \mathbb{R}^{\sum_{i=1}^s n_i \times \sum_{i=1}^s n_i}$ and similarly redefining $\boldsymbol{\sigma}_{\boldsymbol{N}_k}$ and $\boldsymbol{\sigma}_{\boldsymbol{A}_k}$, as the respective uncertainty matrices of $\boldsymbol{N}_k$ and $\boldsymbol{A}_k$, equation 16 and 18 still applies. note that the sub-matrices $\boldsymbol{\sigma}_{\boldsymbol{T}_{k;ij}}$ are user defined.

uncertainty estimation

$\boldsymbol{\sigma}_{\boldsymbol{N}_0}$ and the sub-matrices $\boldsymbol{\sigma}_{\boldsymbol{T}_{k;ij}}$ can be specified by the user as part of the material file definition and transfer rate definitions, respectively. only including these factors would not include running additional monte carlo transport calculations or matrix exponential solution, and can thus be achieved with minimal additional computational cost to obtain $\boldsymbol{\sigma}_{\boldsymbol{N}_k}\ \forall i \in \{1,\ldots,m\}$.

$\boldsymbol{\sigma}_{\boldsymbol{A}_k}$ stems from the statistical certainty of the monte carlo calculation and from nuclear data uncertainties propagated through the monte carlo solution, neither of which having a numerical solution. thus, $\boldsymbol{\sigma}_{\boldsymbol{A}_k}$ must be estimated and this can be done in multiple ways, with potentially large computational effort and might be prohibitive to estimate in certain cases.

a crude estimation of $\boldsymbol{\sigma}_{\boldsymbol{A}_k}$ can be achieved by obtaining $\boldsymbol{A}_i$ with a small M sets of different nuclear data libraries, using the same statistics but different pseudorandom number generator seeds, and from these calculating the

uncertainty as $\boldsymbol{\sigma}_{\boldsymbol{A}_k}^{②} \sim {}^{1}/_{M-1} \sum_{i=1}^{M} \left(\boldsymbol{A}_{k|i} - \bar{\boldsymbol{A}}_k\right)^{②}$. alternatively, $\boldsymbol{\sigma}_{\boldsymbol{A}_k}$ can be estimated using more rigurous total monte carlo methods.

if the affect of $\boldsymbol{A}_k$'s dependency of $\boldsymbol{N}_k$ is included, one can use the crude estimate $\frac{\partial \boldsymbol{A}_k}{\partial \boldsymbol{N}_k} \sim \frac{\boldsymbol{A}_{k|\boldsymbol{N}_k + \boldsymbol{\sigma}_{\boldsymbol{N}_k}} - \boldsymbol{A}_{k|\boldsymbol{N}_k}}{\boldsymbol{\sigma}_{\boldsymbol{N}_k}}$, results in two additional monte carlo transport calculations per timestep, assuming $\boldsymbol{N}_k \pm {}^{\boldsymbol{\sigma}_{\boldsymbol{N}_k}}/_{2}$ instead of $\boldsymbol{N}_k$ as input to the monte carlo transport calculations, but ignores recalculating $\boldsymbol{\sigma}_{\boldsymbol{A}_i}$. alternatively, $\frac{\partial \boldsymbol{A}_k}{\partial \boldsymbol{N}_k}$ can be estimated using openmc's tally derivative capabilities.

conclusion

uncertainty propagation theory is employed to obtained a reduced expression for calculating the propagated uncertainties for discretized depletion simulation, including for coupled materials, but assumes that material quantities and transition matrix values are uncorrelated and ignoring the error introduced from the numerical calculation of the matrix exponential solution. the method relies on user defined material uncertainties and transfer rate uncertainties, as well as estimation of the transition matrix uncertainty for each timestep of a monte carlo depletion simulation.

by ignoring the uncertainty of monte carlo transport and only considering user defined uncertainties, an expression for propagating the user defined uncertainties is obtained with minimal additional computation cost.

including the uncertainties of monte carlo transport, an expression for propagating the combined uncertainties is obtained with moderate to significant additional computation cost. the derived expression is invariant to how the transition matrix uncertainties are calculated making it broadly applicable.

it's the expectation that the derived expression could be implemented in monte carlo depletion tools as an optional feature, for uncertainty propagation.