UPPSALA
UNIVERSITET

Serienummer
Examensarbete 15 hp
Månad och år

Evaluation of nuclear data using the Half Monte Carlo technique

A framework for implementation of the HMC technique

Tyra Axén, Olle Diös, Viktor Lindström, Axel Wohlin

Civilingenjörsprogrammet i teknisk fysik

UPPSALA
UNIVERSITET

Evaluation of nuclear data using the Half Monte Carlo technique

Tyra Axén, Olle Diös, Viktor Lindström, Axel Wohlin

Abstract

The total Monte Carlo technique and the Sandwich method are two techniques used to propagate uncertainties in nuclear data to the uncertainty in macroscopic quantities. Both methods have advantages and drawbacks. The TMC-method can be considered exact but is computationally expensive, and the Sandwich method is fast but is simplified in a way that affects the accuracy of the results.

A new method, the half Monte Carlo method (HMC) takes advantage of both TMC and Sandwich method to create a method that is drastically faster than the TMC, but with a higher accuracy than the sandwich method. The method has shown promising results in the article J. Dyrda et al.

In this work, we demonstrate how the HMC technique can be used to calculate Δk_{eff} in the setting of the integral experiment Godiva. Furthermore, a framework that implemented the half Monte Carlo method for uncertainty propagation from random ACE files and Δk_{eff} sensitivities was developed.

A comparison of the results from the HMC-technique was done with TMC and Sandwich method showcasing that HMM with a high degree of accuracy replicates the results of TMC, whilst being less computationally taxing.

The reaction specific results of HMC compared to the Sandwich method also provide insight as to which reactions may be causing the discrepancies in results between TMC and the Sandwich method.

Teknisk-naturvetenskapliga fakulteten

Uppsala universitet, Utgivningsort Uppsala/Visby

Handledare: Erik Andersson Sundén Ämnesgranskare: Alf Göök, Henrik Sjöstrand

Examinator: Martin Sjödin

Contents

1 Introduction

A common technique for uncertainty propagation of k_{eff} is the total Monte Carlo method (TMC). TMC is extensively used as it captures both the non-linear relations and allows the cross section distribution to be non-Gaussian. It has previously been suggested by Dyrda et al that k_{eff} sensitivities to nuclear data perturbations remain linear over a large range[**?**]. However the same cannot be said for nuclear data parameters such as cross sections. This was the base idea when developing the half Monte Carlo method (HMC), which utilises the fact that there exists readily available sampled cross-section files. HMC is significantly faster than TMC, often producing results within minutes compared to days. However it needs to be further established to what degree the results of HMC are comparable to those of TMC.

A simpler and classic approach to computing uncertainty is the so-called Sandwich method, which also assumes linearity of the k_{eff} sensitivities, and utilizes the covariances of the nucelar cross-sections.

In "A comparison of uncertainty propagation techniques using NDaST: full, half or zero Monte Carlo?", by Dyrda, Hill, Fiorito, Cabellos, and Soppera, HMC was compared to TMC and the Sandwich method by testing them with the Jezebel (PMF001) integral experiment from the ICSBEP handbook. It was concluded that HMC achieves results close to those of TMC, which is indicative that the linearity assumption of the k_{eff} sensitivities is sensible. HMC also showcases which reactions are the primary contributors to $\Delta\text{k}_{\text{eff}}$, without additional computational burden.

In this paper we present a framework in python for HMC using already processed random files in the ACE-format. The sensitivity vectors were taken from the Database for ICSBEP (DICE). The Godiva experiment [**?**], and random ^{235}U files were used to evaluate the framework's functionality. The same files and experiment were used for comparing total Monte Carlo in OpenMC with half Monte Carlo, using the mean value, standard deviation, kurtosis, and skewness. Δk_{eff} was calculated using the sensitivities of the following different reactions:

- (n,2n)
- (n,3n)
- (n,4n)
- Fission
- Elastic
- Inelastic
- n,gamma
- Prompt nubar
- Nubar

1.1 Problem description

The aim of the project is first and foremost to create a framework in python that implements the half Monte Carlo method (HMC) for Δk_{eff} calculations. Ideally these results should be comparable with the ones created from the total Monte Carlo method (TMC), and more accurate than the results of the Sandwich method. The goal is to publish a well documented and intuitive framework, so that the department of applied nuclear physics at Uppsala University can continue using the HMC via the framework for uncertainty propagation of k_{eff}.

2 Theory

2.1 k_{eff}

k_{eff} is the effective neutron multiplication factor which is the average number of neutrons from one fission that causes another fission. The value of k determines how the nuclear chain reaction will proceed. When $k_{\text{eff}} < 1$ the system is subcritical and the chain reaction dies out over time. If $k_{\text{eff}} = 1$ the system is critical and the chain reaction is selfsustaining, and if $k_{\text{eff}} > 1$ the number of fission reactions increases exponentially and the system is supercriticall. The changes in k_{eff}, Δk_{eff} tells us how sensitive the reaction is to changes in the nuclear model parameters.

2.2 Random files

Random files are randomly generated nuclear data files that are generated by variating the model parameters such as the angular distribution, energies, and spins of the particles involved in the nuclear reaction. These random files reflect the covariances of the cross sections. Therefore they can be used to propagate the uncertainty of k_{eff}, via TMC for example.

2.3 Central file

The central ace file refers to the nuclear data's ace files that serves as the baseline or reference file for the perturbations. The random files are distributed around the central file.

2.4 Total Monte Carlo

The total Monte Carlo Method uses the random nuclear data files for a particle transport simulation. The outcome from each simulation is a k_{eff} value. From this, uncertainties can be propagated from the nuclear data's random files.

This method is computationally expensive, and often takes multiple days to complte for nuclear data applications[**?**]. The method can be considered the golden standard since it captures all interactions, including nonlinear interactions and non-Gaussian parameter distributions[**?**].

Additionally, the method can be implemented using total Monte Carlo particle transport simulators, such as MCNP, SERPENT, or OpenMC. Thus, no processing of covariance or Δk_{eff} sensitivities are needed. However, the individual effects of the different reactions and energies are not easily obtained. Post-processing the data is required to study the correlations to k_{eff} from individual nuclear data parameters. Another problem is the shortage of easily available random files, especially in the ACE-format.

In conclusion, if k_{eff} has significant nonlinear dependencies then total Monte Carlo is needed for sufficient accuracy. Otherwise, if it is sensible to assume that the sensitivities of k_{eff} are methods are preferred, considering the high computational cost of TMC.

2.5 Linear propagation/Sandwich formula

The classic Sandwich formula is:

$$\Delta k_{\text{eff}} = J \cdot C \cdot J^T \tag{1}$$

where J is the k_{eff} sensitivity vector for the nuclear data parameters and C is the covariance matrix for the nuclear data parameters. The covariance matrix consists of the uncertainties in the nuclear data parameters and correlations between them.

Even though the Sandwich method is drastically cheaper in computations than TMC it is still not widely used due to poor accuracy. This is due to the assumption made of the cross section distributions being Gaussian by using the covariance matrix, whereas in reality this may not be the case. HMC and TMC do not come with this assumption. The Sandwich method also assumes linearity within the k_{eff} sensitivities,as does HMC, which is not as problematic of an assumption as claimed by Dyrda and Hill [?].

These covariance matrices can be found in many existing libraries. For example the SCALE6-44g covariance library as used in this paper, or processed directly from reaction data ENDF-files.

2.6 Half Monte Carlo

The half Monte Carlo method also uses random ACE files as in the TMC-method, but instead uses pre-calculated k_{eff} sensitivity matrices as opposed to the expensive neutron transport calculations done in TMC. Therefore, if possible, half Monte Carlo would be a great option if the accuracy of its results is sufficient.

$$\Delta k_{\text{eff}} = \sum_{i=1}^{m} J \cdot (ND_i - ND_0)$$

Where Δk_{eff} is the uncertainty propagation, J is the k_{eff} sensitivity vector, ND_i is the cross-section of the ith random file and ND_0 is the cross-section of the central file. The method makes it possible to use results from integral experiments, by using k_{eff} sensitivity vectors with respect to a nuclear data reaction. The half Monte Carlo method does not assume a Gaussian distribution of nuclear data parameters. However, it assumes that the k_{eff} sensitivities are linear, making it so that the sensitivity vector can be thought of as a Jacobian.

In conclusion, if the assumption of a linear sensitivity is sensible, the half Monte Carlo is a fast and accurate method for determining the uncertainty propagation Δk_{eff}.

2.6.1 Δk_{eff} sensitivities

The k_{eff} sensitivity vectors represent the sensitivity of k_{eff} to changes in various nuclear data parameters such as cross sections or reactions, in other words the sensitivity vector, J, can be though of as a discrete partial derivative over an energy vector, E,

$$J = \frac{\delta \Delta k}{\delta \sigma}(E)$$

3 Method

3.1 Python for Nuclear Engineering (PyNE)

The Nuclear Engineering Toolkit PyNE is a Python module, written in Python and C++[?]. PyNE contains many useful features such as support for reading and parsing cross sections from ACE and ENDF-6 files. This project used PyNE version 0.7.5, which is compatible with Python versions 3.x. All versions of PyNE require a Linux environment to function, if using Windows one can use the Windows Subsystem for Linux[?] or a virtual machine if using Mac OS. For an installation guide see the PyNE team's website[?] or the project repository's *README.me* file.

3.2 DICE

The Δk_{eff} sensitivity vectors used are taken from the Database for The International Criticality Safety Benchmark Evaluation Project (DICE) corresponding to the Godiva experiment. To retrieve the sensitivity vectors the keyword "Godiva" was searched. Then, under sensitivity plots "total within bin" and the library ENDF/B-VII.0 continuous was chosen. The sensitivities for ^{235}U were selected, see figure 1. From the table tab, the values of the sensitivity vectors and the corresponding energies were manually copied and made into a .csv file.

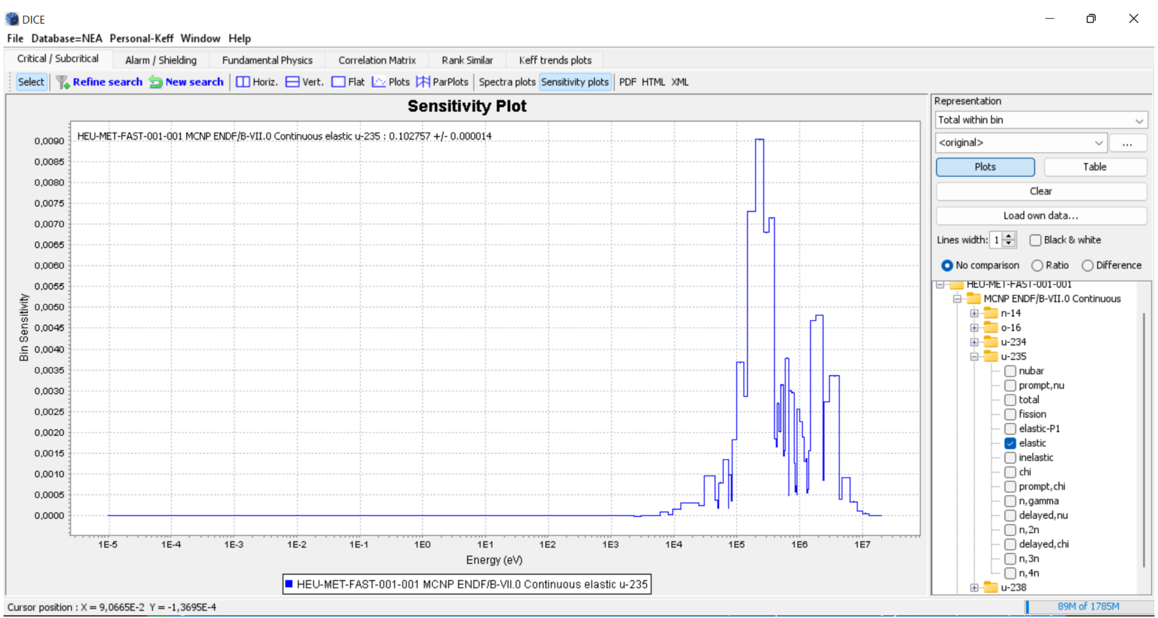


Figure 1: A figure showing the plot of the elastic sensitivity vector for the Godiva experiment in DICE.

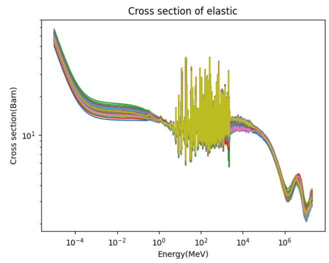
In the DICE, and our.csv file, the right column contains the values of the sensitivity vector and the left column contains the corresponding energies.

3.3 Random acefiles

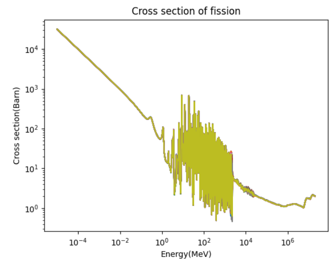
The random acefiles [?], including the central acefile, were gathered from the TENDL-2021 website [?]. These random acefiles are ENDF files from the ENDF/B-VII.1 processed in 187 groups produced by using the NUSS approach, as described in this NSE paper [?]. The acefiles are used are divided into two datasets labeled dataset A and B for this project. All acefiles include cross section for reactions for energies up to 20 MeV.

3.3.1 Dataset A

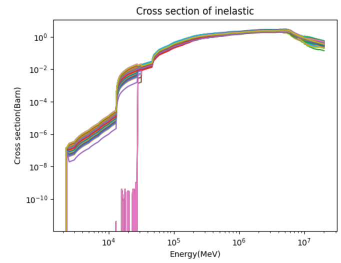
Dataset A consists of a central acefile and 99 random acefiles. The cross sections used in the propagation of the criticality coefficient in this report are visually represented in figures 2.



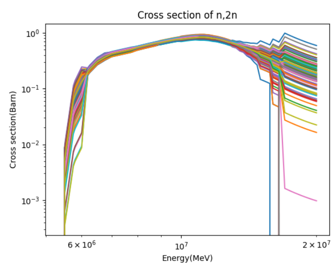
(a) elastic



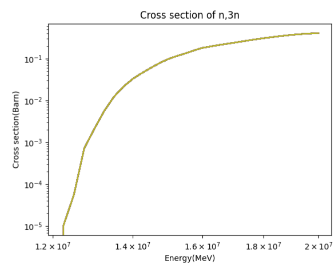
(b) fission



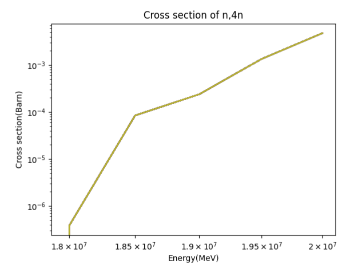
(c) inelastic



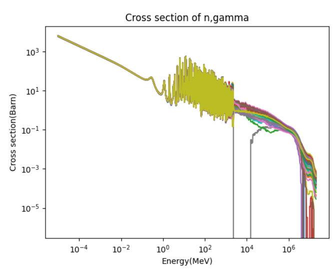
(d) n,2n



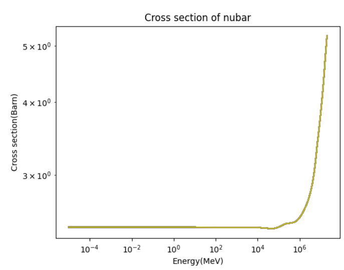
(e) n,3n



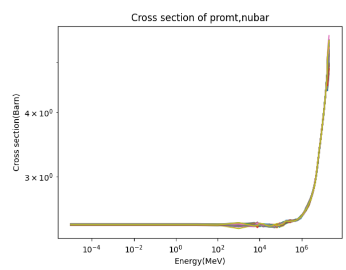
(f) n,4n



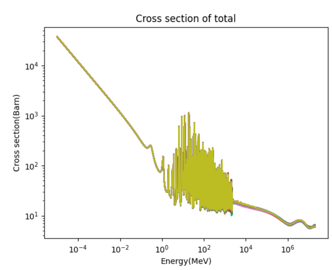
(g) n,gamma



(h) nubar



(i) prompt,nubar

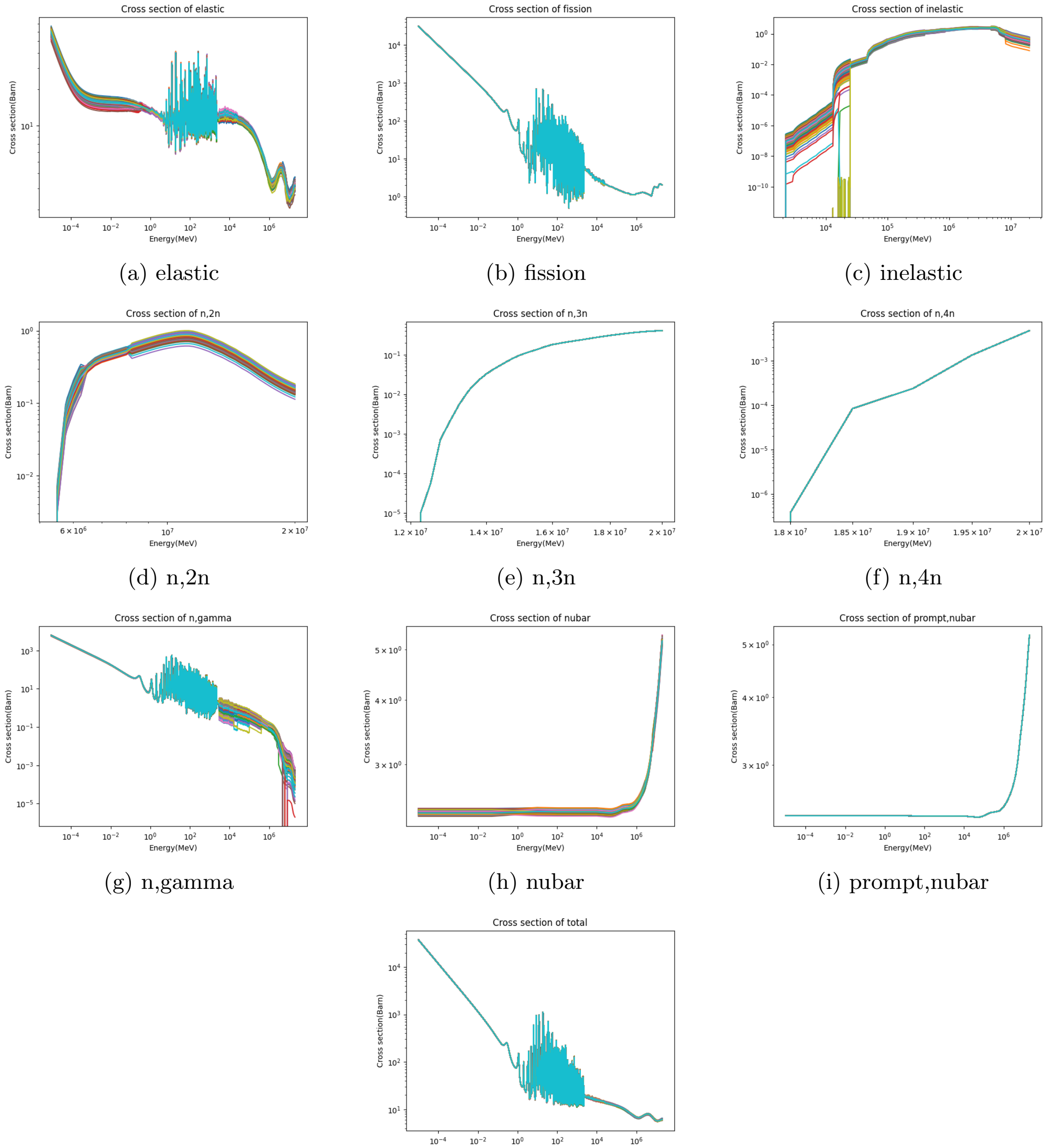


(j) total

Figure 2: Plots for the random acefiles from dataset A of the different reactions with the cross section (Barn) on the x-axis and Energy (MeV) on the y-axis.

3.3.2 Dataset B

Dataset B consists of a central acefile and 299 random acefiles. The cross sections used in the propagation of the criticality coefficient in this report are visually represented in figures 3.



(a) elastic (b) fission (c) inelastic

(d) n,2n (e) n,3n (f) n,4n

(g) n,gamma (h) nubar (i) prompt,nubar

(j) total

Figure 3: Plots for the random acefiles from dataset B of the different reactions with the cross section (Barn) on the x-axis and Energy (MeV) on the y-axis.

3.4 HMC implementation

HMC was implemented using the k_{eff} sensitivity vectors from DICE and the ACE-files mentioned in 3.3. Mathematically Δk_{eff} can be expressed as the dot product between the sensitivity vector and the difference between the nuclear data of the random acefile and the central acefile. The sensitivity vector utilized in this dot product is binned, meaning its sensitivity value is constant for specific stretches of energy. Each energy bin is represented as a sum over l, the i random acefiles are represented as a sum over i and both the sensitivity vector and the nuclear data, ND, is evaluated at discrete energies, E_j, this is shown in equation 2.

$$\Delta k_{\text{eff}} = \sum_{l=1}^{q}\sum_{i=1}^{n}\sum_{j=1}^{m} J_l(E_j) \cdot (ND_i(E_j) - ND_0(E_j)) \tag{2}$$

3.4.1 Sensitivity bins

The randomfiles from the ENDF B-VII.0 used in this paper had an energy vector consisting of 84420 elements in dataset A or 76518 elements in dataset B, ranging from 10^{-5} MeV to $2 \cdot 10^7$ MeV in both datasets. This is in stark contrast to the sensitivity vectors 239 discrete energy bins, also ranging from 10^{-4} MeV to $2 \cdot 10^6$ MeV. The framework approaches this by finding the mean difference between all cross section differences between central acefile and given random acefile for each bin. This in turn makes the dot product between the sensitivity bins and the cross section values consider almost all relevant values.

3.5 TMC OpenMC implementation

To compare the results from the half Monte Carlo method, total Monte Carlo was implemented in OpenMC. A script was created, defining the same geometry, material, density, and radius as the Lady Godiva device [?]. This was an unshielded sphere with a diameter of 6.848 inches or 17.39392 cm, 93.7% ^{235}U, 1% ^{234}U, 5.3% ^{238}U and a density of $18.7 g/cm^3$. A vacuum cube was defined with a side length of 50cm around the sphere to capture potential neutrons reentering the sphere. The number of batches used was 100, with 10 inactive batches. The discarding of the first 10 batches is because the first values are inaccurate since all particles are placed in the middle of the sphere, creating a bigger k_{eff} value. The number of particles used was 600 000. This was determined by running the central file for different amounts of particles, and seeing when the results converged, see figure 4.

To run simulations in OpenMC a library of .hdf5 files is needed. For all random ACE files a new library was created, using the OpenMC functions, .data.IncidentNeutron.from_ace() and .data.DataLibrary(). A new simulation for each library was run. The .hdf5 files for ^{234}U and ^{238}U were taken from JEFF 3.3 from OpenMC's library [?], and were kept constant for all ACE files. The proportional difference between the k_{eff} values for each run and the central file was calculated to get the Δk_{eff} values, as seen in equation 3.

$$\Delta k_{\text{eff}} i = \frac{k_{\text{eff}}0 - k_{\text{eff}} i}{k_{\text{eff}}0} \tag{3}$$

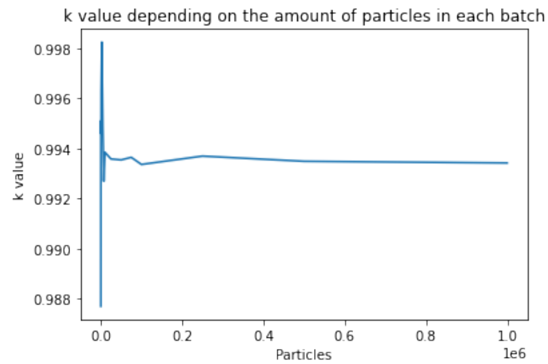


Figure 4: k_{eff} criticality coefficient for different amounts of particles in TMC implemented in OpenMC for the central file.

3.6 Sandwich method

NDaST was used for the implementation of the Sandwich method. Available cross section covariances from the JANIS database for ENDF/B-VII.0, ENDF/B-VII.1, ENDF/B-VIII.0 were:

- Fission
- Elastic
- Inelastic
- n,gamma
- n,2n.

These were combined in NDaST with the sensitivity vectors used in the HMC implementation as described in 3.4, using the Sandwich formula 1. It is important to note that the libraries used for covariances in this Sandwich method implementation is not exactly the same as the libraries used when creating the random ACE files used in the HMC implementation 3.4. However, these results can still be used as reference point for evaluating the discrepancies between the results of HMC and the Sandwich method as compared to the results of TMC.

4 Results and discussion

With the Half Monte Carlo method implemented in code the script is able to produce information about the error propagation of the criticality coefficient. The results are represented in graphs and tables where the deviation from the central file is defined with pcm (percent mille) which is one one-thousandth of a percent.

4.1 Histograms and tables of HMC results

The script has produced 10 graphs for each dataset with each representing each reaction available and the 10th and final graph the total reaction which is the summation of all reaction channels (not the cross-section sometimes reffered to as 'total'). The data in the graphs are presented in a histogram where the deviation from the central file in criticality coefficient(k_{eff}) represents x-axis and the number of randomfiles/cases in each bin represents the y-axis. The histograms can be found in figure 5.

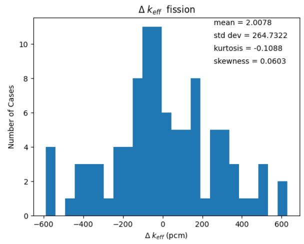
The values produced along with the histograms such as mean, standard deviation, kurtosis and skewness can be found for each reaction in table 1. The units for all the mentioned measurements are all in percent mille(pcm).

4.2 HMC and TMC results comparison

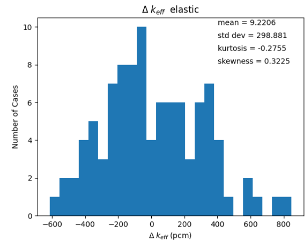
From the same library of cross section data the two aforementioned methods of total monte carlo and half monte carlo methods have been utilized to propagate the error of the criticality coefficient. As results from the TMC method are assumed to be reflective of the correct answer the comparison between the results is in interest to accredit the produced framework of the half monte carlo method. From figure 10 the histograms of the criticality coefficient from the TMC and HMC methods cannot effectively be compared visually, which is why measurements such as mean, standard deviation, kurtosis and skewness are showcased. The results can be found in table 1

Table 1: Dataset A results in percent mille(pcm)

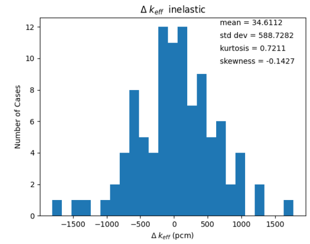
Reactions:	mean	std dev	kurtosis	skewness
n, 2n	-0.3287	6.5326	-0.09546	-0.0744
n, 3n	0.0	0.0	NaN	NaN
n, 4n	0.0	0.0	NaN	NaN
Fission	2.0078	264.7322	-0.1088	0.06032
Elastic	9.2206	298.8810	-0.2755	0.3225
Inelastic	34.6112	588.7282	0.7211	-0.1427
n,gamma	-74.6649	782.2485	0.01220	0.1869
Prompt nubar	-8.3408	94.6210	-0.8251	0.8240
Nubar	0.0	0.0	NaN	NaN
Total HMC	**-37.4949**	**998.8476**	**0.3200**	**-0.2971**
Total TMC	**-14.9571**	**1045.7883**	**-0.0592**	**0.3876**



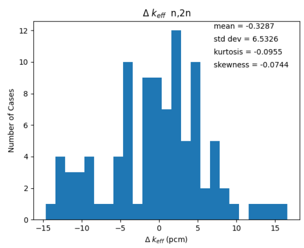
(a) Histogram for the fission criticality coefficient error propagation.



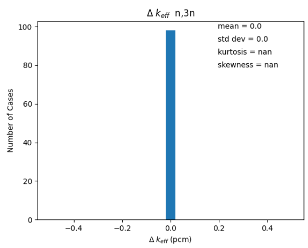
(b) Histogram for the elastic criticality coefficient error propagation.



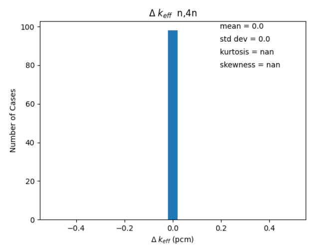
(c) Histogram for the inelastic criticality coefficient error propagation.



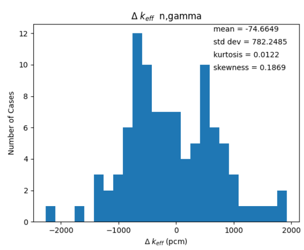
(d) Histogram for the n,2n criticality coefficient error propagation.



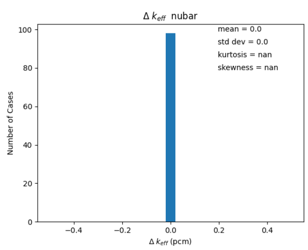
(e) Histogram for the n,3n criticality coefficient error propagation.



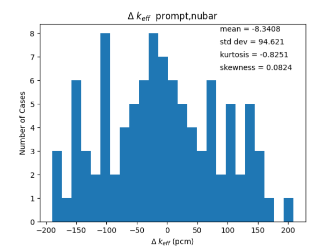
(f) Histogram for the n,4n criticality coefficient error propagation.



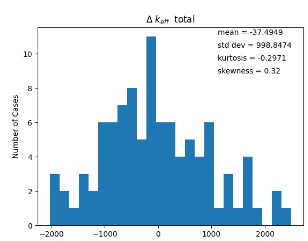
(g) Histogram for the n,gamma criticality coefficient error propagation.



(h) Histogram for the nubar criticality coefficient error propagation.

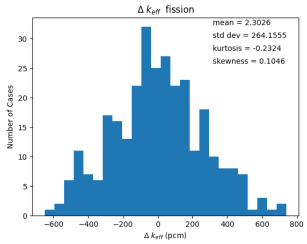


(i) Histogram for the prompt,nubar criticality coefficient error propagation.

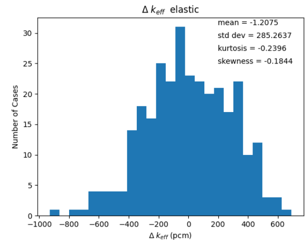


(j) Histogram for the total criticality coefficient error propagation.

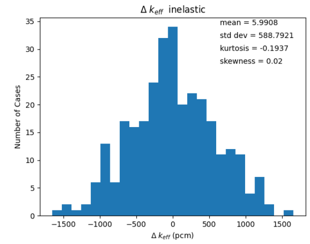
Figure 5: Histograms of the HMC error propagation of the criticality coefficient for the different reactions for dataset A with Δk_{eff} on the x-axis and number of cases on the y-axis.



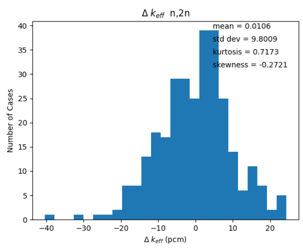
(a) Histogram for the fission criticality coefficient error propagation.



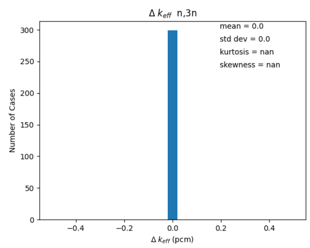
(b) Histogram for the elastic criticality coefficient error propagation.



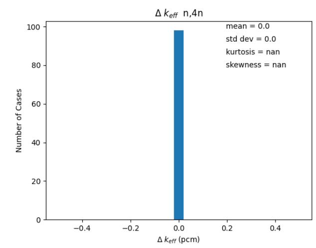
(c) Histogram for the inelastic criticality coefficient error propagation.



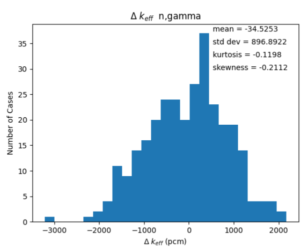
(d) Histogram for the n,2n criticality coefficient error propagation.



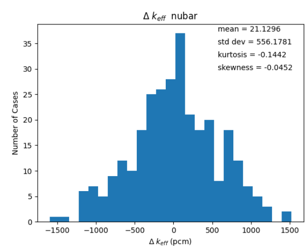
(e) Histogram for the n,3n criticality coefficient error propagation.



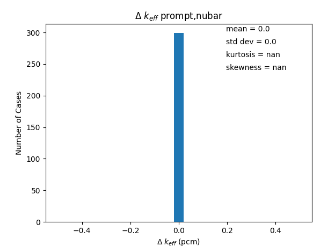
(f) Histogram for the n,4n criticality coefficient error propagation.



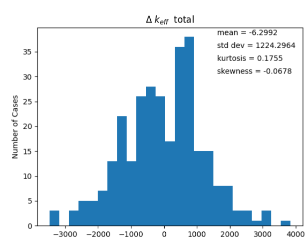
(g) Histogram for the n,gamma criticality coefficient error propagation.



(h) Histogram for the nubar criticality coefficient error propagation.

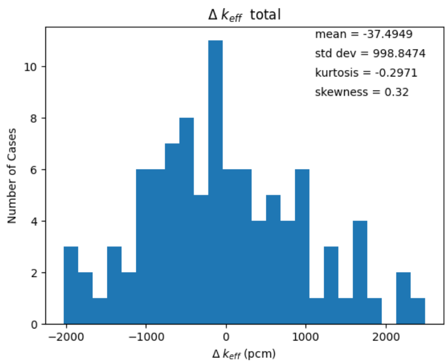


(i) Histogram for the prompt,nubar criticality coefficient error propagation.

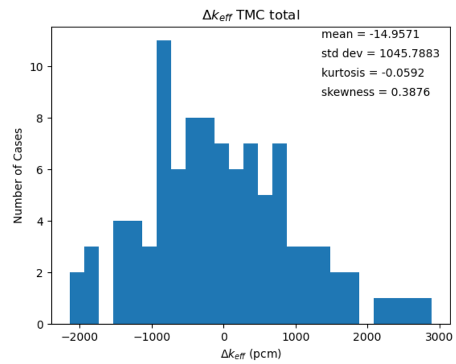


(j) Histogram for the total criticality coefficient error propagation.

Figure 6: Histograms of the HMC error propagation of the criticality coefficient for the different reactions for dataset B with Δk_{eff} on the x-axis and number of cases on the y-axis.



(a) Histogram for the HMC total criticality coefficient error propagation for dataset A.

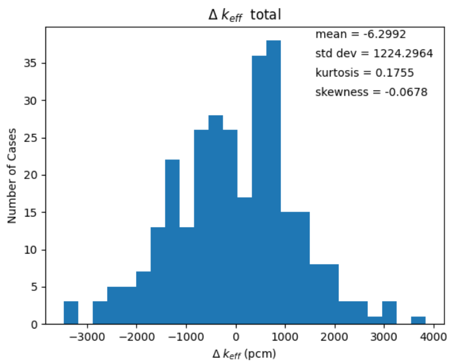


(b) Histogram for the TMC total criticality coefficient error propagation for dataset A.

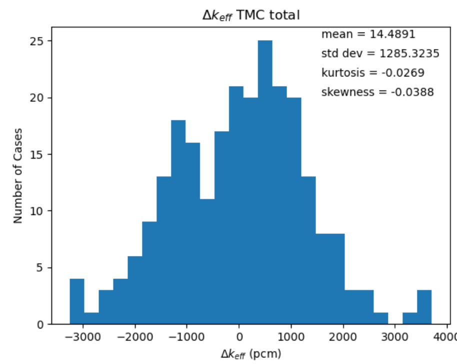
Figure 7: Histograms of the error propagation of the criticality coefficient calculated from HMC and TMC for dataset A with Δk_{eff} in percent mille(pcm) on the x-axis and number of cases on the y-axis.

Table 2: Dataset B results in percent mille(pcm)

Reactions:	mean	std dev	kurtosis	skewness
n, 2n	0.01058	9.8009	0.7173	-0.2721
n, 3n	0.0	0.0	NaN	NaN
n, 4n	0.0	0.0	NaN	NaN
Fission	2.3026	264.1555	-0.2324	0.1046
Elastic	-1.2075	285.2637	-0.2396	-0.1834
Inelastic	5.9908	588.7921	-0.1937	0.0200
n,gamma	-34.5253	896.8922	-0.1198	-0.2112
Prompt nubar	0.0	0.0	NaN	NaN
Nubar	21.1296	556.1781	-0.1442	-0.0452
Total HMC	**-6.2992**	**1224.2964**	**0.1755**	**-0.0678**
Total TMC	**14.4891**	**1285.3235**	**-0.0269**	**-0.0388**



(a) Histogram of the HMC total criticality coefficient error propagation for dataset B.

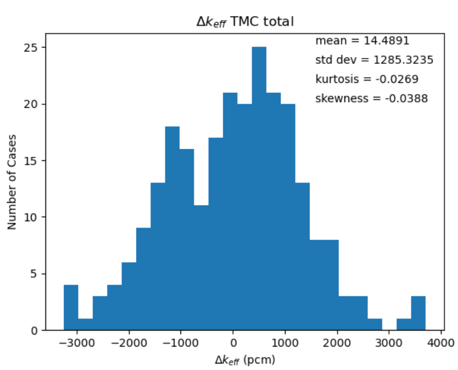


(b) Histogram for the TMC total criticality coefficient error propagation for dataset B.

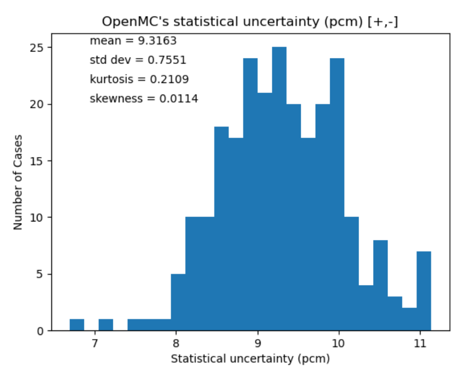
Figure 8: Histograms of the error propagation of the criticality coefficient calculated from HMC and TMC for dataset B with Δk_{eff} in percent mille(pcm) on the x-axis and number of cases on the y-axis.

4.3 TMC results

The total Monte Carlo simulation in OpenMC was produced only for the total change in the criticality coefficient. This result can be compared to the sandwich method and the total HMC result.

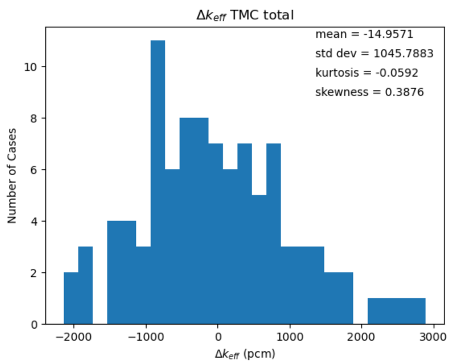


(a) Histogram of the TMC total criticality coefficient error propagation for dataset B.

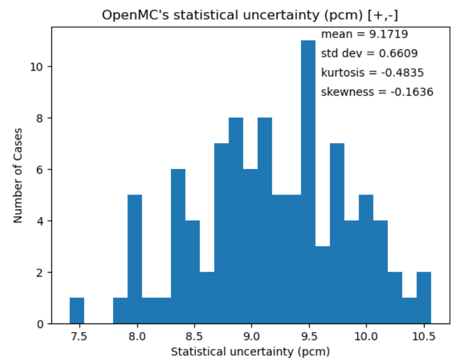


(b) Histogram of the statistical uncertainty in OpenMC for dataset B.

Figure 9: Histograms of the error propagation of the criticality coefficient calculated from TMC and the statistical uncertainty from the TMC calculation. Dataset B was used for these figures.



(a) Histogram of the TMC total criticality coefficient error propagation for dataset A.



(b) Histogram of the statistical uncertainty in OpenMC for dataset A.

Figure 10: Histograms of the error propagation of the criticality coefficient calculated from TMC and the statistical uncertainty from the TMC calculation. Dataset A was used for these figures.

4.4 Sandwich method results

These results were produced using NDaST as described in 3.6, using the uncertainty propagation calculation with included off-diagonal terms, also known as 'CK'.

[pcm]	**elastic**	**inelastic**	**n,2n**	**fission**	**gamma**	**std dev**	**HMC σ A**	**HMC σ B**
elastic	**0.759**	0.174	0.00356	0.146	0.0489	275.6	298.8	285.3
inelastic	0.174	**0.597**	0.00982	0.165	0.0183	244.3	588.7	588.8
n,2n	0.00355	0.00982	**0.00451**	0.00363	0.000465	21.2	6.53	9.801
fission	0.146	0.165	0.00363	**6.21**	0.0659	788.1	264.7	264.2
gamma	0.0490	0.0184	0.000456	0.0659	**0.791**	281.2	782.24	896.9

Table 3: Sandwich method results

4.4.1 Sandwich method results analysis

Table 3 compares the results of NDaST with those of HMC for each cross section. The large discrepancies for some cross sections in the two rightmost columns of table 3 may give insight as to why HMC produces total results closer to those of TMC than the Sandwich method. This could provide further insight as to for which cross section distributions it is reasonable to assume that they are distributed as a Gaussian. One can therefore see that reactions inelastic, fission and gamma differ vastly between the sandwich method and HMC, which likely attributes to why the results of HMC are closer to those of TMC than for the Sandwich method.

4.5 Discussion

For both dataset A and B, the reaction channels such as n,3n and n,4n can be observed to be zero both in the measurement of mean and standard deviation. This is likely due to those reactions being high improbable, and should therefore not be considered. Dataset A encounters a similar issue for total nubar reaction channel**Till Erik: Fel i filerna/pyne här? Hur kan prompt,nubar vara varierad i ace filerna men inte nubar?**, curiously this is not seen in dataset B where instead prompt,nubar encounters the same problem. This lack of variation between separate randomfiles can also be reflected in the histograms for the respective reaction channels found in the above figures 5 and 6. Although this might be interpreted as a computational error it could also be due to issues pertaining to the random files.

The cross section graphs in figure 2 and 8 also reflect what could be defects in the random files, where for the aforementioned reaction channels such as n,3n, n,4n and the total nubar for dataset A and prompt,nubar for dataset B seem to lack the variation between randomfiles which the other reactions inhibit. This issue could be speculated to be the product of the alterations of the randomfiles not altering the conditions necessary to shift the cross section for all reaction channels, therefore not displaying any deviation from the central file. Conversely the issue probable for the reaction channels n,3n and n,4n could be the void of data concerning the cross section. Due to the fact that these reactions are solely plausible for a high energy threshold could lead to an insufficient data to properly perform the uncertainty propagation. However, the total nubar and prompt,nubar criticality coefficient manifests identical issues for dataset A and B respectively, yet pertains a complete energy spectrum.

The achieved standard deviation for the total Monte Carlo and half Monte Carlo methods for dataset A was approximately 1046 pcm and 999 pcm, respectively. The mean value was -15 pcm and -37 pcm. The values for HMC ranged from roughly -2000 to 2500 pcm, and for TMC from -2000 to 3000 pcm. For dataset B the standard deviation was 1285 pcm for TMC and 1224 for HMC. The mean values were -6 pcm and 14 pcm. The values for HMC ranged from about -3500 to 4000 pcm and TMC from -3000 to 4000 pcm. These results are comparable to those of the Jezebel experiment conducted by Dyrda et al. as reported in their publication [**?**]. The mean statistical uncertainty of TMC was found to be approximately 9 pcm for both datasets, which is slightly smaller than the difference between TMC and HMC. This discrepancy is attributed to the linear approximation used in the HMC method. Nonetheless, the small difference between the two methods can be attributed to the linearity of the sensitivities over most energies, as reported in the aforementioned publication [**?**].

The half Monte Carlo method and our framework seem to work well for the Godiva experiment. The accuracy for other experiments and geometries depends on the linear dependence of the sensitivity and has to be researched before using HMC and the framework.

As seen in figures 7 and 8, when using the 300 random ace files (dataset B), the skewness is similar, while for the 100 random ace files (dataset A) the skewness differs. This might be explained by the insufficient amount of files used in dataset A. For further work, more files should be examined to see if any differences occur.

5 Conclusion

This work has showcased the implementation of the HMC method via a framework implemented in Python. It was showcased how the DICE database can be used to access Δk_{eff} sensitivities which can be combined with random ACE files to implement HMC. The results were compared to those of TMC and the sandwich method.

The results show that HMC achieved a high accuracy (compared to TMC), whilst being many times faster than TMC. The reaction specific results of HMC may also provide insight for the sandwich method as to which reaction parameters could be responsible for the lower accuracy.

In J. Dyrda et al. [**?**], the SANDY tool, which is not freely available at the moment was used to perform 'backwards' HMC. Since this work have made use of open source programs and databases, we stuck to 'forwards' HMC and have created a HMC implementation with a wider availability, which in [**?**] was stated to be a beneficial area of further development.

As discussed when presenting the results in table 3 it is important to acknowledge that the covariances, Δk_{eff} and random Ace files all were sourced from different libraries such as ENDF B-VII.0, ENDF B-VII.1, JANIS and the SCALE6-44g covariance library. This could contribute to some of the discrepancies between the methods. Therefore it would be highly beneficial to establishing the validity of these results if a HMC, TMC and Sandwich method comparison was done with stricter sourcing, making sure all the parameters used originated from the same library where applicable.

In this paper, the produced framework implemented HMC and TMC on the simple spherical symmetry of the Godiva experiment, and only utilized random ACE files for the isotope U^{235}. It is unlikely that a more thorough investigation utilizing random ACE files for U^{234} and U^{238} would result in significantly different results. However it would definitely be beneficial to perform further investigations of other integral experiments, to see if the accuracy of HMC could be more widely applicable.

It is intended that the framework presented in this paper is to be as available and accessible as possible. To make this framework able to be utilized it would be highly beneficial if it could easily be combined with other easily accessible tools to process the random ACE files, Δk_{eff} sensitivities and covariances used. This opportunities for HMC to further be investigated as a viable method for uncertainty propagation would be greatly increased if all tools required could be used in a more holistic setting. These are aspirations for further development of the framework.

6 References

heading=none]references.bib

7 Populärvetenskaplig sammanfattning

I en fissionsprocess kan neutroner ge upphov till kedjereaktioner genom att neutronerna som utsänds vid en inledande fission ger upphov till nya klyvningar. Om förutsättningarna är rätt så kan denna process vara självbevarande. Kriticitetfaktorn, eller k_{eff} är ett mått på hur många neutroner från en fission som i snitt orsakar en ny fission i en fissionsprocess, exempelvis i en kärnreaktor. När $k_{\text{eff}} = 1$ så skapas i snitt en ny klyvning per klyvning, och det råder balans mellan produktion och förluster av neutroner.

Ur ett miljöperspektiv är k_{eff} av stort intresse då det påverkar säkerheten och effekten hos kärnreaktorer. Om k_{eff} är för högt kan reaktorn hamna ur kontroll, detta är något som utnyttjas i kärnvapen. Å andra sidan kan ett k-effektivt värde som är för lågt resultera i att en otillräcklig mängd energi produceras, vilket leder till att reaktionen dör över tid. Detta gör att k_{eff}-värdet, samt förändringar i detta, Δk_{eff} är av högt intresse.

För att kunna skapa en kärnreaktor som uppfyller denna säkerhet krävs att vi vet vilka faktorer som påverkar k_{eff}. En vanlig metod för att propagera osäkerheter i kärndata med avseende på förändringar i makroskopiska faktorer är via så kallade total Monte Carlo metoden. Denna metod använder sig av så kallade randomfiler som är skapade genom av variera olika parametrar som är involverade i en kärnreaktion. Randomfilerna används sedan i en partikeltransport-simulering. Från varje simulering får vi ut ett k_{eff}-värde och utifrån det kan vi propagera osäkerheter i kärndatan. Denna metod är flitigt använd på grund av att den kan göra exakta beräkningar, men tar lång tid att köra.

En annan simplare metod är den vid engelskt namn "Sandwich formula". Denna metod är drastiskt snabbare än TMC men har lägre nogrannhet då den antar att vi har en gaussisk tvärsnittsfördelning vilket i verkligheten inte alltid är fallet. Denna metod antar också linjaritet inom k_{eff}-känsligheterna vilket däremot verkar stämma till stor utsträckning.

En ny metod vid namn Half Monte Carlo-metoden utnyttjar randomfilerna från TMC men är inte beroende av partikeltransport-simuleringarna för att få fram ett resultat. Istället använder den sig av redan uträknade kovarians-matriser eller känslighetsvektorer och antar på så sätt likt "Sandwich formula" linjaritet. Half Monte Carlo metoden tar alltså vara på styrkorna hos båda metoderna och är därför en möjlig ersättare för dessa.

Syftet med detta projekt är att skapa ett ramverk som implementerar half Monte Carlo metoden för att beräkna Δk_{eff} för olika typer av reaktioner, samt att underöka hurvida resultatet är lovande genom att jämföra med TMC-simuleringar samt med beräkningar gjorda med hjälp av "sandwich method". Ramverket skrivs i python och koden har som syfte att kunna användas av institutionen för tillämpad kärnfysik vid Uppsala Universitet.

8 Acknowledgments

We thank Ian Hill for contributions with NDAST-results, Dice and the implementation of the Sandwich method. We thank Dimitri Rochman for providing us our ACE-files. We thank Erik Anderson Sundén for being a great supervisor and for helping us continuously throughout our work. We thank Henrik Sjöstrand for many ideas and insightful discussions and Alf Göök for taking the time to read and review our final report.