HAL
open science

Accelerating Monte Carlo particle transport with adaptively generated importance maps

Michel Nowak

To cite this version:

Michel Nowak. Accelerating Monte Carlo particle transport with adaptively generated importance maps. Computational Physics [physics.comp-ph]. Université Paris-Saclay, 2018. English. NNT : 2018SACLS403 . tel-02135102

HAL Id: tel-02135102

https://theses.hal.science/tel-02135102v1

Submitted on 21 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2018SACLS403

Accelerating Monte Carlo particle transport with adaptively generated importance maps.

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

Ecole doctorale n°576 Particules, Hadrons, Énergie, Noyau, Instrumentation, Image, Cosmos et Simulation (PHENIICS)
Spécialité de doctorat : Énergie Nucléaire

Thèse présentée et soutenue à Gif Sur Yvettes, le 12 Octobre 2018, par

MICHEL NOWAK

Composition du Jury :

M. Kenneth Burn ENEA, Bologne	Rapporteur
M. Tony Lelievre CERMICS, ENPC	Rapporteur
Mme. Mariya Brovchenko IRSN	Examinatrice
M. Pierre Desesquelles CSNSM, Paris Sud	Président
M. Davide Mancusi CEA, Saclay	Examinateur
M. Marc Verderi CNRS/IN2P3/LLR	Examinateur
M. Jamal Atif LAMSADE, Paris Dauphine	Directeur
M. Eric Dumonteil IRSN	Directeur

Thèse de doctorat

0.0 Acknowledgements

I want to thank all the people who shared my path during this work at SERMA.

First, I am grateful to the reviewers of this thesis who gave me their feedback until final convergence. I thank Tony Lelièvre for the discussions we had during the thesis and for his final review. He brought the mathematical rigorousness that one needs to tackle AMS and more generally variance reduction techniques. I am grateful to Kenneth Burn, also reviewer, for the many exchanges we had on the manuscript. His remarks made me think and re-think on how to make this work more precise and more pedagogical. I finally thank Mariya Brovchenko from IRSN, who gave me relevant comments on my work.

I now thank my supervisors who followed the progress of my work and directed it when needed. I thank Jamal Atif for giving me insights on which Machine Learning algorithms to use. To my advisor and mentor Eric Dumonteil: our history started with an internship on *clustering*. He maintained me aware of the context of this PhD, stressing on the industrial importance of this work. He help me build my path through research and made me enjoy it. To my day-to-day supervisor, Davide Mancusi, who was always present when needed. For the many things you said that brought me back on *tracks*. For your many skills, ranging from computation, to supervising, thank you for having played a big role in this project. I also want to thank Andrea Zoia for partnering on clustering and playing with equations in the bus.

This work is partly the result of a collaboration between the Monte Carlo team at SERMA, and the deterministic team. I want to thank Emiliano Masiello, Igor Zmijarevic (Youpi!) and Daniele Sciannandrone for guiding me on the deterministic path.

I want to thank the people that showed me that our work might have a real impact on the industry, especially Antoine and Stéphane from the LPEC laboratory.

Less formally, I want to thank Cheikh Diop: it was a pleasure to hear his anecdotes about the beginning of SERMA. Also, some of his phrases on research are still present in my mind.

Merci à Audrey et à Milo pour avoir partagé presque 2 ans en coloc et pour avoir toujours été présents. Merci également à mes autres colocs pour m'avoir toujours laissé la porte ouverte: Antonio, Alex, Daniele, Enrica et Laura.

Laurent, toujours partant pour une rando ;)

A ceux qui ont partagé mon bureau et plus encore: Henri, merci d'avoir fait ce bout de chemin avec moi. J'ai adoré travaillé avec toi sur le flux adjoint pour l'AMS. On s'en souviendra.:w Antonio, merci d'avoir ramené un peu de déterminisme dans mon bureau. Cela nous a permis de brûler quelques étapes ensemble. A toute l'équipe de ski: Margaux, Thomas, Wesley, Esteban, Paul, Damien, Valentin, à très bientôt.

Merci au Plan B pour l'accueil durant la période de rédaction.

Enfin, merci à mes parents et à mon frère qui m'ont toujours soutenu lors de cette

aventure.

Contents

Page

Acknowledgements 2

List of Figures v

List of Tables xi

I Introduction **1**

Introduction 3

1. Particle transport 7

1.1 The forward perspective 8
1.1.1 The integro-differential formulation 9
1.1.2 The integral formulation 11
1.2 The backward perspective 14
1.2.1 The integro-differential formulation 15
1.2.2 The integral formulation 16
1.3 On the solution of these equations 16

2. Handling rare events in Monte Carlo shielding calculations 17

2.1 Monte Carlo for particle transport 18
2.1.1 Sampling scheme 19
2.1.2 Analog transport 23
2.1.3 TRIPOLI-4® 23
2.1.4 Monte Carlo failure to solve shielding problems 23
2.2 Variance reduction: importance sampling 24
2.2.1 Implicit capture and Russian roulette 25
2.2.2 Exponential transform 26
2.2.3 From the integro-differential formulation 26
2.2.4 *Biased* Sampling 27
2.2.5 TRIPOLI-4® 28
2.2.6 Intelligent random numbers. 29
2.3 Variance reduction: splitting methods 29
2.3.1 Weight Windows 29
2.3.2 Adaptive Multilevel Splitting 29
2.4 Conclusion 34

II The importance map **35**

3. Optimal importance map 37

3.1 Importance and adjoint quantities 38

3.2 Zero variance scheme and a way out? ... 39
3.3 Practices for the generation of the importance map ... 41
3.3.1 Deterministic importance ... 41
3.3.2 Weight Window Generator ... 42
3.3.3 DSA ... 42
3.3.4 Serpent2 ... 43
3.3.5 INIPOND ... 43
3.3.6 What is done outside the nuclear sphere? ... 47
3.3.7 Adaptive Multilevel Splitting ... 47
3.4 Conclusion ... 47

4. Deterministic solutions from IDT to TRIPOLI-4® ... 49

4.1 The multi-group formalism ... 50
4.1.1 Direct ... 50
4.1.2 Adjoint ... 51
4.2 Coupling ... 52
4.2.1 General ... 52
4.2.2 Mesh & geometry ... 52
4.2.3 Cross sections ... 53
4.2.4 Adjoint source ... 53
4.2.5 IDT solver parameters ... 53
4.3 Verification ... 54
4.4 Results ... 55
4.4.1 Strong attenuation ... 55
4.4.2 Streaming : the bunker ... 62
4.4.3 Nucifer at Osiris ... 63
4.5 Conclusion ... 65

5. Implementation of a forward-weighted adjoint Monte Carlo estimator ... 69

5.1 Illustration ... 70
5.2 Derivation of the importance ... 70
5.2.1 Physical interpretation ... 72
5.3 The analog adjoint estimator ... 73
5.3.1 The variance ... 75
5.3.2 Implementation of the score in TRIPOLI-4® ... 75
5.4 Scoring the adjoint flux with importance sampling ... 78
5.4.1 The estimator ... 78
5.4.2 Implementation in TRIPOLI-4® ... 79
5.5 Scoring the adjoint flux with Adaptive Multilevel Splitting ... 79
5.5.1 The estimator ... 79
5.5.2 Implementation in TRIPOLI-4® ... 80
5.6 Verification ... 83
5.6.1 A homogeneous two-group infinite sphere ... 83

5.6.2 A two dimensional, multi-group, heterogeneous problem 85
5.7 Relation to previous work. 85
5.7.1 Weight Window Generator 85
5.7.2 CADIS-**Ω** 87
5.8 Conclusions 87

6. Using the adjoint score as an importance map 89

6.1 Two-step scheme 90
6.1.1 Strong attenuation 90
6.1.2 Coupled Neutron-Gamma problem 91
6.1.3 Conclusions 94
6.2 Fixed-step alternating scheme 96
6.2.1 Strong attenuation 97
6.2.2 Neutron Gamma 99
6.2.3 Conclusions 102
6.3 Bootstrap scheme 102
6.3.1 Strong attenuation 104
6.3.2 Neutron Gamma 106
6.3.3 Capturing a variance jump? 106
6.3.4 Conclusions 111
6.4 A few words on the parallelisation of these strategies. 111
6.5 Conclusions 113

7. Conclusion and Perspectives 115

7.1 Findings 116
7.2 Perspectives 117

APPENDIX A. Benchmarks 129

A.1 Sphere 1 group 129
A.2 Sphere 2 groups 129
A.3 Water+Iron 130
A.4 Bunker 130
A.5 Strong Attenuation 130
A.6 Neutron-gamma problem. 134
A.7 Bypass 135
A.8 Nucifer at Osiris 136

APPENDIX B. Choice of the smoothing algorithm 139

B.1 Requirements 140
B.2 The dataset 140
B.2.1 A small condensation 142
B.3 The training phase and performance metrics 142

B.4 Discussion on models 143
B.4.1 Some small tests 144
B.5 Conclusion 144

APPENDIX C. Kernel Density Estimators 147

C.1 The model 148
C.2 Implementation in TRIPOLI-4® 149
C.2.1 Time monitoring 149
C.3 Acceleration on GPU 151
C.3.1 Motivation 151
C.3.2 Implementation in TRIPOLI-4 152
C.4 Results 152
C.4.1 Strong attenuation 152
C.4.2 Neutron-Gamma 152
C.5 Conclusion 154

List of Figures

Figure **Page**

1.1 Illustration of the optical distance along the characteristic. 11

1.2 Illustration of the integral formulation of the Boltzmann equation for the emission density $\chi(\mathbf{r}, E, \mathbf{\Omega})$ 13

1.3 Illustration of the integral formulation of the Boltzmann equation for the collision density $\psi(\mathbf{r}, E, \mathbf{\Omega})$ 14

2.1 A dummy shielding scheme. A source, a detector and attenuating materials or pipes in between. 23

2.2 Illustration of path stretching. The importance is minimum near the source and increases along the axis towards the detector. The lengths of the particle flights are stretched if they move towards higher importance zones (green), and shortened otherwise (red). 28

2.3 Illustration of the AMS algorithm 33

3.1 Zero variance track estimator 40

3.2 Geometry of a theoretical 2D model for the illustration of INIPOND. Two materials are present. The detector where the response is sought is in the top right corner and we are interested in estimating the importance of a particle leaving an arbitrary point in the phase space $(\mathbf{r}, E, \mathbf{\Omega})$ 43

3.3 Spectrum used to condense multi-group cross sections for INIPOND. A fission spectrum is used over 1 MeV. A simple slowing down behaviour is chosen between 1E−7 MeV and 1 MeV. A Maxwell spectrum is chosen for thermal energies lower than 1E−7 MeV. 45

3.4 Illustration of the spatial component of the INIPOND module. The spatial component in the $(E, 3)$ cell of the importance in the sense of INIPOND is $\mathcal{I}_{\text{INIPOND}} = e^{-k_1(\ell_1+\ell_2)+k_2\ell_3}$ 46

4.1 Water-iron benchmark geometry.. 55

4.2 Direct flux comparison for the water-iron benchmark. The first row represents the flux computed by IDT, the second one with TRIPOLI-4®. The last figure represents the discrepancy between IDT and TRIPOLI-4® as a ratio between their difference and the standard deviation of the adjoint score with TRIPOLI-4®. The energy groups correspond to those present in [Appendix A.3] 56

4.3 Geometry for the strong attenuation benchmark. A mono-directional neutron source, sampled on a Watt spectrum is placed in a 3 m-long tank of water. The detector is a 20 cm-wide part of a 120 cm-long concrete wall. Reflection conditions are imposed on all surfaces except for the outer surface of the detector, from which neutrons can leak out. 57

4.4 Ratio of mean response for each case to the reference ET with the importance map computed by IDT with cross sections produced by APOLLO3. The shaded areas represent the confidence intervals at two standard deviations. They are symmetric (distortions due to the log scale). All calculations lie within two standard deviations from the reference solution, except for the simulation ET with the importance map computed by IDT with cross sections condensed by TRIPOLI-4®, which is clearly inconsistent below 1 eV. 59

4.5 Spectrum for the strong attenuation problem. The flux spectrum is plotted against the energy at three different positions: (a) between x=50 cm and x=60 cm from the source, (b) between x=2.3 m and x=2.4 m of water, (c) at the entrance of the detector between x=4.0 m and x=4.20 m, for the 6 tested configurations: AMS (blue) and ET (green), for the three importance maps, INIPOND tunes (DATA), INIPOND with cross sections from T4 (IDT INI), and IDT with cross sections homogenized in a prior direct deterministic simulation (IDT ANI). 60

4.6 Figure of Merit for the strong attenuation problem. FOMs are plotted against the energy for 6 configurations. 61

4.7 Bunker geometry 62

4.8 Z=0 cm-slice of the importance map generated by INIPOND AUTO for the Bunker problem. The mesh in not regular and contains $28 \times 25 \times 25$ cells 63

4.9 Z=0 cm-slice of the importance map generated by IDT for the Bunker problem. The mesh is regular cartesian and consists of $200 \times 200 \times 100$ cells . . 64

4.10 Nucifer detector next to Osiris 66

4.11 Importance map for the benchmark: Nucifer detector next to Osiris computed by IDT 66

5.1 Illustration of two methods for the computation of the adjoint flux on a theoretical benchmark. 71

5.2 Two analog non branching tracks stored in TRIPOLI-4® 76

5.3 Illustration of the AMS estimator and of the track structure during an AMS simulation. 81

5.4 Geometry of the 2-group sphere problem 83

5.5 Adjoint flux scored on a mesh with regular 0.1 cm-spaced cells along the radial direction. The score is performed with AMS. Black lines are reference solutions obtained by a numeric integration. The blue (green) curve shows the result for the fast (slow) group with associated confidence intervals at three standard deviations. 84

5.6 Adjoint flux comparison for the water-iron benchmark. The first row represents the flux computed by IDT, the second one with TRIPOLI-4®. The last figure represents the discrepancy between IDT and TRIPOLI-4® as ratio between their difference and the standard deviation of the adjoint score of TRIPOLI-4®. 86

6.1 Two-step adjoint response update scheme. 90

6.2 Comparison between adjoint fluxes computed by IDT and scored with AMS. 92

6.3 Geometry of the coupled neutron gamma problem. 94

6.4 Adjoint score for the neutron-gamma coupled benchmark. 95

6.5 Fixed-step alternating adjoint response update scheme. 96

6.6 Fixed-step alternating scheme for the strong attenuation benchmark. Each exploration phase lasts 5 batches and is performed with AMS running with the importance map computed with the cross sections coming from APOLLO3®. Each exploitation phase lasts five batches and is performed with ET. . . . 98

6.7 Fixed-step alternating scheme for the strong attenuation benchmark. Each exploration phase lasts 5 batches and is performed with AMS running with geometrical importance map (inverse of distance to detector). The exploitation phase lasts five batches and is performed with ET. 100

6.8 Fixed-step alternating scheme for the neutron-gamma benchmark. Each exploration phase lasts 10 batches and is performed with AMS running with a geometrical importance map(inverse of distance to detector). The exploitation phase lasts 10 batches and is performed with AMS. 101

6.9 Fixed-step alternating scheme for the neutron-gamma benchmark. Each exploration phase lasts 10 batches and is performed with AMS running with a geometrical importance map(inverse of the distance to the detector). The exploitation phase lasts 10 batches and is performed with ET. 103

6.10 Bootstrap scheme 104

6.11 Bootstrap scheme for strong attenuation benchmark. Each exploration phase lasts 5 batches and is performed with AMS running with a geometrical importance map (inverse of the distance to the detector). The exploitation phase lasts 5 batches and is performed with ET. After the second phase, the exploration phase starts using the scored importance map. 105

6.12 Bootstrap scheme for coupled neutron-gamma problem. Each exploration phase lasts 5 batches and is performed with AMS starting with a geometrical importance map (inverse of the distance to the detector) and continuing with the scored importance map. The exploitation phase lasts 5 batches and is performed with ET. 107

6.13 Geometry of the theoretical bypass problem. 108

6.14 Importance maps for the bypass benchmark. 108

6.15 Convergence of the bypass problem with the fixed-step alternating strategy and the bootstrap strategy 110

6.16 Illustration of the convergence of the bypass problem with the bootstrap scheme. Figures a,c,e,g and i illustrate the trajectories sampled during a given batch. Alternatively, figures b,d,f,h and j show the state of the importance map at a given batch. 112

A.1 Geometry of infinite medium with constant cross sections 129

A.2 Water+Iron geometry. 130

A.3 Bunker geometry. 131

A.4 Geometry for the strong attenuation benchmark. A mono-directional neutron source, sampled on a Watt spectrum is placed in a 3 m-long tank of water. The detector is a 20 cm-wide part of a 120 cm-long concrete wall. Reflections conditions are imposed on all surfaces except for the outer surface of the detector, from which neutrons can leak out. 135

A.5 Geometry of the coupled neutron-gamma problem. Polyethylene slabs are colored in red, whereas stainless steel slabs are in blue. The scale in X and Y are in cm. 135

A.6 Geometry of the theoretical bypass problem. 136

A.7 Nucifer detector next to Osiris 137

B.1 Distribution of points generated by AMS for the adjoint score. 145

B.2 Distribution of the condensed contribution and emission density. 146

B.3 Comparison of three models for the emission density in the multi group water-iron benchmark. From left to right: emission density projected on a histogram, emission density interpolated during one batch by XGBoost, emission density interpolated with a neural network 146

B.4 Illustration of the model projection method. Left: emission density from XGBoost trained on a single batch. Right: Averaged XGBoost models on 100 batches. 146

C.1 KDE on the sphere 2 groups 150

C.2 Grid of KDE prediction for the Water+Iron benchmark. From up to down, the bandwidth in both directions increases and takes the values $h = (1, 3, 4, 5, 7, 8, 10)$. From right to left, the number of cells in each direction of the mesh increases $N = (5, 10, 20, 40, 60, 80, 100)$ 150

C.3 Simulation time for the KDE on the Water-Iron benchmark. The bandwidth takes values from $h = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ and the mesh size from $(5, 10, 20, 40, 60, 80, 100)$ 151

C.4 Score evolution for a fixed-step alternating strategy. Each exploration phase of 10 batches is run with AMS starting from a geometrical initial importance map(the inverse of the distance to the detector.) Each exploitation phase is run with ET. The simulation using the adjoint score smoothed with a KDE is plotted in red and compared to the reference simulation taking the adjoint score on a histogram. The energy groups are listed in A.5153

C.5 Prediction of KDE compared to the adjoint score on a histogram after 1000 batches for three energy groups. 153

C.6 Fixed step alternating strategy applied to the neutron-photon problem. Each exploration phase is performed with AMS, uses a spatial importance map and lasts 10 batches. Each exploitation phase is done with ET 154

C.7 Bootstrap scheme applied to the neutron-photon problem. Each exploration phase is made with AMS that starts with a spatial importance map. The adjoint KDE score is collected and projected over a mesh. Each exploitation phase alternates with the exponential transform. After the first two phases AMS re-uses the KDE score as an importance map 155

List of Tables

Table **Page**

4.1 Results for the strong attenuation problem. We present the integrated mean response in the detector, as well as the estimate of its standard error after a certain simulation time. The figures of merit can be compared for both the AMS and the ET method used with the three importance maps : INIPOND in manual mode, IDT with cross-sections from INIPOND and IDT with cross sections from APOLLO3 with anisotropy order 5. 58

4.2 Results for the Bunker problem. We present the integrated mean of the target response, as well as the estimate of the standard error after a certain simulation time. The figures of merit can be compared for both the AMS and the ET method used with two importance maps : INIPOND in automatic mode and IDT with cross sections from TRIPOLI-4®. 65

4.3 Nucifer integrated results 67

6.1 Results for the strong attenuation problem. We present the integrated mean response in the detector, as well as the estimate of its standard error after a certain simulation time. The figures of merit can be compared for both the AMS and the ET method used with the four importance maps : INIPOND in manual mode, IDT with cross-sections condensed with TRIPOLI-4®, IDT with cross sections from APOLLO3® with anisotropy order 5 and the adjoint response scored with AMS. 93

6.2 FOM comparisons for the coupled neutron-gamma setup. We report the mean and standard error for the sought response (gamma flux in a cylinder), as well as the CPU time, for four different calculation methods. FOMs are normalised to the analog calculation. 96

Part I

Introduction

Introduction

Radiation protection is the ensemble of actions, measures, and studies undertaken that contribute to estimating and reducing the impact of radiation on humans and on the environment. This domain intervenes in various applications where the priority is always to protect operators exposed to radiation. Consider first the production of energy in nuclear power plants. In these facilities, a serious challenge is to design robust structures that can withstand the exposure to high-intensity radiation. In nuclear research reactors, operators carry out numerous experiments where they are easily exposed to radiation fields. Finally, in the medical domain, smaller radiation sources are used for imagery or radiation therapy.

It is clear that whenever radiation is present, it is necessary to formulate accurate estimates of the amount of radiation dose received. This is done by modelling and solving *radiation transport* problems.

The *Service d'Étude des Réacteurs et de Mathématiques Appliquées* (SERMA) of the Nuclear Energy Division of CEA, in Saclay, France, develops computer codes that simulate, among other, the transport of particles and their effect on matter.

These tools are essentially divided into two different families. Tools based on *deterministic* methods implement numerical schemes for solving the Boltzmann transport equation over a discretised phase space. On the other hand, *Monte Carlo* methods accurately simulate the transport of particles within the most detailed representation of a problem and of the physics of radiation transport, without introducing any further approximation. Particle trajectories are created with the generation of a sequence of pseudo-random numbers, and this gives rise to fluctuations. Apart from the inherent statistical uncertainties resulting from this process, their only limitations arise from the description of the geometry and the nuclear data. Monte Carlo codes are therefore considered reference codes that can be used to simulate complex 3D geometries, fine spectrum effects, or to validate deterministic codes.

So what is the main limitation with using Monte Carlo methods to solve shielding problems? By construction, shielding problems involve the estimation of the probability of occurrence of very rare events. These events occur whenever particles have to travel several mean free paths before contributing to the detection. This is therefore always the case for shielding structures involving absorbing materials. In this case, the so-called "analog simulation" - that samples particle trajectories according to their natural probability

of occurrence - promptly faces the limits of the computational and financial resources that one is willing, or able, to deploy.

Therefore, these limits are overcome by the use of mathematical "tricks" gathered under the illustrative name of *variance reduction* methods. The idea is to preferentially simulate trajectories that are likely to contribute the most to the observable of interest. The leitmotif of these methods is that they all need to estimate how likely a given particle trajectory is to contribute to the sought response, in order to decide whether it is worth to spend computational effort on its simulation. The estimation is often encoded as an *importance function*. The use of an inadequate importance function may generally lead to widely inaccurate response estimation and even to outright failure of the variance reduction method.

Many studies have already aimed at proposing a suitable importance function for these methods. Among them, Kahn and Harris (1951) showed that, for a certain class of variance reduction methods, there exists an optimal importance function that leads to a zero-variance game - i.e. complete suppression of Monte Carlo's intrinsic fluctuations. The determination of the optimal importance function is equally difficult as the solution to the original problem. However, approximate importance functions may be sufficient to accelerate Monte Carlo calculations and make them efficient.

Keeping this result in mind, Wagner (1997) proposed to compute an approximation of this function with deterministic methods. This methodology - known as Consistent Adjoint Driven Importance Sampling (CADIS) is probably the most recognized and used method by Monte Carlo users today.

Complementarily, efforts have been made to use Monte Carlo transport codes to estimate the optimal importance function themselves. This has the clear potential advantage of providing an exact estimate, without introducing the approximations that are typical of deterministic methods. T. E. Booth and J. S. Hendricks (1984) estimate the importance function during the Monte Carlo simulation itself. They showed that the importance could be estimated on a mesh during a direct simulation and stated that significant accelerations could be obtained if the estimator had converged enough.

To summarize, the state of the art methods that generate importance maps for Monte Carlo shielding calculations are well-oiled. However, their limitation is that they require the intervention of an expert eye to be able to actually make Monte Carlo simulations converge properly.

This justifies the scope of this work, which aims at adaptively generating importance maps on the fly during the direct Monte Carlo simulation itself by using machine learning algorithms. This work is organized as follows. Chapter 1 presents the mathematical formulation of the problem and is thus the occasion to set the notations that will be used further. Chapter 2 focuses on the Monte Carlo method itself and presents the main variance reduction methods. Chapter 3 reviews the main characteristics of the importance function and how it was derived so far. In Chapter 4, we look at the potential accelerations of variance reductions methods when used with a solution generated by a deterministic code. This chapter will also show how we automatically generate reference adjoint solutions for the verification of the next chapter. Chapters 4 to 6 represent the core contributions of this thesis. Chapter 5 presents the implementation of a forward-weighted

adjoint Monte Carlo estimator. Chapter 6 discusses how the adjoint score can be used as an importance function and to what extent this results in accelerations. Finally, Appendices B and C explore the interest of *smoothing* the adjoint flux. We conclude with a discussion in Chapter 7 and present suggestions for future research.

Chapter 1

Particle transport

Contents

1.1 The forward perspective **8**

1.1.1 The integro-differential formulation 9

1.1.2 The integral formulation 11

1.2 The backward perspective **14**

1.2.1 The integro-differential formulation 15

1.2.2 The integral formulation 16

1.3 On the solution of these equations **16**

The aim of this first chapter is to introduce the standard mathematical formalism for the description of particle transport. We essentially focus on the transport of neutrons and photons which are the principal types of particles involved in reactor physics or shielding. We start by introducing the most intuitive perspective in Section 1.1: the *forward* approach. Section 1.2 dives into the *backward* perspective and introduces the adjoint formalism. Equations and observables will be written in an exhaustive form so as to facilitate the derivation of numerical schemes in the following chapters.
Finally, a discussion on the solutions to these equations is started in Section 1.3.

1.1 The forward perspective

The first perspective, and the most natural one, is to consider the transport of particles in matter as it actually occur in real life. Particles are emitted from arbitrary sources - for example from a radioactive sample, or nuclear reactors - and propagate in the surroundings. The forward perspective simply consists in following particles in time from their birth to their absorption. To formally describe their behaviour, it is imperative to delimit the space in which they evolve. We refer to *phase-space* $\mathcal{P}$ the space formed of all possible states in which a particle can be. The state of a particle of mass m moving at speed v is fully described by its position in the *configuration space*

$$\mathbf{r} \in \mathbb{R}^3,$$

its direction of motion

$$\mathbf{\Omega} \in S^2,$$

and its energy

$$E = \frac{1}{2}mv^2 \in \mathbb{R}^+$$

The particle state evolves through time in a non-deterministic manner. The lack of determinism is partly due to our ignorance of the exact conditions of the system, and partly to the quantum-mechanical nature of the microscopic nuclear and atomic scattering processes that govern the interaction of radiation with matter. One must therefore give up any hope of describing the individual particle trajectories and must settle instead for a collective description of the motion of a large ensemble of particles. In this context, a natural observable of interest is the expected number of particles that can be found at a particular point $P = (\mathbf{r}, E, \mathbf{\Omega})$ in the phase space at time t. By defining the *angular particle density* $n(\mathbf{r}, E, \mathbf{\Omega}, t)$, we can express the total number of particles as an integral over the whole phase space

$$\iiint n(\mathbf{r}, E, \mathbf{\Omega}, t) d^3\mathbf{r}\, dE\, d^2\mathbf{\Omega}. \tag{1.1}$$

We define the *angular flux* $\Phi(\mathbf{r}, E, \mathbf{\Omega}, t)$ as the product of the particle density with its speed v.

$$\Phi(\mathbf{r}, E, \mathbf{\Omega}, t) = vn(\mathbf{r}, E, \mathbf{\Omega}, t), \tag{1.2}$$

as well as the *scalar flux* $\Phi(\mathbf{r}, E, t)$ as its integral over all angles $\mathbf{\Omega}$

$$\Phi(\mathbf{r}, E, t) = \iint \Phi(\mathbf{r}, E, \mathbf{\Omega}, t) d^2\mathbf{\Omega}. \tag{1.3}$$

The particle angular flux is a local quantity and its evolution can thus be written in the form of a balance equation. The balance equation can be expressed from different points of view. We express the integro-differential formulation before deriving an equivalent integral form afterwards.

1.1.1 The integro-differential formulation

We begin by describing the flux balance with the direct integro-differential form. For simplicity, in the following, we restrict ourselves to the case of one particle species interacting with matter. Let us consider a small region of the phase space δP around point $P = (\mathbf{r}, E, \mathbf{\Omega})$ at time t. What can happen around δP?

First, particles can simply be produced in δP by a fixed source. Let

$$S(\mathbf{r}, E, \mathbf{\Omega}, t)\, d^3\mathbf{r}\, dE\, d^2\mathbf{\Omega}\, dt$$

denote the number of particles emitted by the source in the infinitesimal neighbourhood δP between times t and $t + dt$. Then, particles can enter and exit δP without undergoing any interaction inside. The rate of change of the angular particle density can be expressed as a streaming term

$$-\mathbf{\Omega} \cdot \nabla\Phi(\mathbf{r}, E, \mathbf{\Omega}, t).$$

Contrarily, some particles may experience an interaction. The probability of a particle undergoing any interaction at phase-space point P is captured by the total macroscopic cross section $\Sigma_t(\mathbf{r}, E)$. The total rate of interactions occurring in δP is:

$$-\Sigma_t(\mathbf{r}, E)\Phi(\mathbf{r}, E, \mathbf{\Omega}, t).$$

Finally, particles can enter δP by undergoing a collision in $\mathbf{r}$ and changing energy and angle from $E', \mathbf{\Omega}'$ to $E, \mathbf{\Omega}$. The probability of undergoing such collision at $(\mathbf{r}, E')$ is described by the scattering cross section $\Sigma_s(\mathbf{r}, E')$, which captures the probability of any scattering interaction to occur, and the scattering kernel $f_s(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega})$, which provides the final-state distribution of scattering events at $\mathbf{r}$. By multiplying it with its angular distribution f_s, we can write the rate at which scattering occurs:

$$\int \Sigma_s(\mathbf{r}, E') f_s(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega})\Phi(\mathbf{r}, E', \mathbf{\Omega}', t) dE' d^2\mathbf{\Omega}'$$

The sum of all the rates of change of the angular particle density described so far must

be equal to the total rate of change of the particle density, which simply expresses as

$$\frac{1}{v}\frac{\partial \Phi}{\partial t}.$$

We regroup these terms into a balance equation where the sum of production rates equals the sum of disappearance rates in δP.

$$\frac{1}{v}\frac{\partial \Phi}{\partial t} + \mathbf{\Omega} \cdot \nabla \Phi(\mathbf{r}, E, \mathbf{\Omega}, t) + \Sigma_t(\mathbf{r}, E)\Phi(\mathbf{r}, E, \mathbf{\Omega}, t) = \int \Sigma_s(\mathbf{r}, E') f_s(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega})\Phi(\mathbf{r}, E', \mathbf{\Omega}', t) dE' d^2\mathbf{\Omega}' + S(\mathbf{r}, E, \mathbf{\Omega}, t) \quad (1.4)$$

This equation is known as the direct Boltzmann equation for neutral particle transport. Boundary conditions are associated to this equation, and are customarily assumed to be leakage, reflection or periodic. In the simplest case of leakage at infinity, the boundary condition is expressed as

$$\int \frac{\Phi}{v}\, d^3\mathbf{r}\, dE\, d^2\mathbf{\Omega} < \infty. \quad (1.5)$$

In what follows, we shall restrict ourselves to the search of stationary solutions of the Boltzmann transport equation. We make the assumption that the particle source is time independent and we look for stationary solutions of Eq. (1.4). The time derivative disappears from the equation. The final form of the stationary Boltzmann transport equation finally reads:

$$\mathbf{\Omega} \cdot \nabla \Phi(\mathbf{r}, E, \mathbf{\Omega}) + \Sigma_t(\mathbf{r}, E)\Phi(\mathbf{r}, E, \mathbf{\Omega}) = \int \Sigma_s(\mathbf{r}, E') f_s(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega})\Phi(\mathbf{r}, E', \mathbf{\Omega}') dE' d^2\mathbf{\Omega}' + S(\mathbf{r}, E, \mathbf{\Omega}) \quad (1.6)$$

It is interesting for future considerations to point out a remarkable property of this equation. The energy and angular dependence cannot be described by any other mean than with an integral operator. This means that the energy is a particular direction in the phase space along which particles can move discontinuously.

In order to simplify the use of this equation, we define the direct transport operator

$$\mathbf{L} = \mathbf{\Omega} \cdot \nabla \cdot + \Sigma_t(\mathbf{r}, E) \cdot - \iint \Sigma_s(\mathbf{r}, E') f_s(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}) \cdot dE' d^2\mathbf{\Omega}', \quad (1.7)$$

so that Eq. (1.4) can be rewritten in compact form as:

$$\mathbf{L}\Phi = S. \quad (1.8)$$

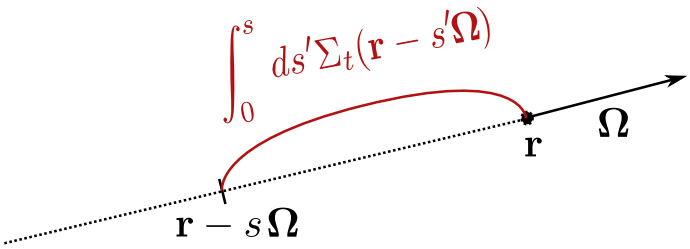


Figure 1.1. Illustration of the optical distance along the characteristic.

Although this formulation already allows for particle transport to be treated with some class of deterministic methods, we turn to the integral form of the transport equation, which is the natural setting to describe the Monte Carlo algorithm.

1.1.2 The integral formulation

As explained by Bell and Glasstone (1970), it is possible to transform Eq. (1.6) into a purely integral form thanks to the method of characteristics. In order to do so, we firstly rewrite Eq. (1.6) by expressing the particles' position in function of its curvilinear abscissa s along its trajectory. We have a look at the small variations of Φ around a point $\mathbf{r} - s\mathbf{\Omega}$.

$$\begin{aligned} &-\frac{d}{ds}\Phi\left(\mathbf{r} - s\mathbf{\Omega}, E, \mathbf{\Omega}\right) + \Sigma_t\Phi(\mathbf{r}, E, \mathbf{\Omega}) = \\ &\iint \Sigma_s(\mathbf{r} - s\mathbf{\Omega}, E')f_s(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}')\Phi(\mathbf{r} - s\mathbf{\Omega}, E, \mathbf{\Omega})dE'd^2\mathbf{\Omega}' + S(\mathbf{r} - s\mathbf{\Omega}, E, \mathbf{\Omega}) \end{aligned} \tag{1.9}$$

The idea is now to integrate the variations of the flux from $\mathbf{r} - s\mathbf{\Omega}$ to $\mathbf{r}$. In order to do so, we define the optical distance illustrated in Figure 1.1.

$$\rho(s) = \int_0^s ds'\Sigma_t(\mathbf{r} - s'\mathbf{\Omega}).$$

Let us write the variations of the attenuated flux $\Phi(\mathbf{r}, E, \mathbf{\Omega})$ before it reaches $\mathbf{r}$.

$$\begin{aligned} \frac{d}{ds}\left[e^{-\int_0^s ds'\Sigma_t(\mathbf{r}-s'\mathbf{\Omega})}\Phi(\mathbf{r} - s\mathbf{\Omega}, E, \mathbf{\Omega})\right] &= e^{-\int_0^s ds'\Sigma_t(\mathbf{r}-s'\mathbf{\Omega})} \\ &\times \left(-\frac{d}{ds}\left[\Phi(\mathbf{r} - s\mathbf{\Omega}, E, \mathbf{\Omega})\int_0^s ds'\Sigma_t(\mathbf{r} - s'\mathbf{\Omega})\right] + \frac{d}{ds}\Phi(\mathbf{r} - s\mathbf{\Omega}, E, \mathbf{\Omega})\right) \\ &= e^{-\int_0^s ds'\Sigma_t(\mathbf{r}-s'\mathbf{\Omega})}\left(-\Sigma_t(\mathbf{r} - s\mathbf{\Omega})\Phi(\mathbf{r} - s\mathbf{\Omega}, E, \mathbf{\Omega}) + \frac{d}{ds}\Phi(\mathbf{r} - s\mathbf{\Omega}, E, \mathbf{\Omega})\right) \end{aligned} \tag{1.10}$$

By combining this equation with Eq. (1.9), we find

$$-\frac{d}{ds}\left[e^{-\int_0^s ds'\Sigma_t(\mathbf{r}-s'\mathbf{\Omega})}\Phi(\mathbf{r}-s\mathbf{\Omega},E,\mathbf{\Omega})\right]=e^{-\int_0^s ds'\Sigma_t(\mathbf{r}-s'\mathbf{\Omega})}\chi(\mathbf{r}-s\mathbf{\Omega},E,\mathbf{\Omega}), \tag{1.11}$$

where we have set

$$\chi(\mathbf{r},E,\mathbf{\Omega})=-\Sigma_t(\mathbf{r})\Phi(\mathbf{r},E,\mathbf{\Omega})+\frac{d}{ds}\Phi(\mathbf{r},E,\mathbf{\Omega}).$$

We now integrate Eq. (1.11) over s, from 0 to ∞

$$-\int_0^\infty ds\frac{d}{ds}\left[e^{-\int_0^s ds'\Sigma_t(\mathbf{r}-s'\mathbf{\Omega})}\Phi(\mathbf{r}-s\mathbf{\Omega},E,\mathbf{\Omega})\right]=$$
$$\int_0^\infty ds\, e^{-\int_0^s ds'\Sigma_t(\mathbf{r}-s'\mathbf{\Omega})}\chi(\mathbf{r}-s\mathbf{\Omega},E,\mathbf{\Omega}).$$

The integral on the left hand side yields $\Phi(\mathbf{r},E,\mathbf{\Omega})$ because the contribution at ∞ vanishes thanks to the boundary condition Eq. (1.5). We can now write the final integral form of the Boltzmann equation:

$$\Phi(\mathbf{r},E,\mathbf{\Omega})=\int_0^\infty e^{-\int_0^s ds'\Sigma_t(\mathbf{r}-s'\mathbf{\Omega})}\chi(\mathbf{r}-s\mathbf{\Omega},E,\mathbf{\Omega}). \tag{1.12}$$

This equation suggests that if we regard $\chi(\mathbf{r},E,\mathbf{\Omega})$ as the density of the particles emitted at $(\mathbf{r},E,\mathbf{\Omega})$, then the flux at any point of phase space can be expressed as the sum of the emissions along the characteristics, weighted by the attenuation factor $e^{-\int_0^s ds'\Sigma_t(\mathbf{r}-s'\mathbf{\Omega})}$.

For the future, we introduce two useful kernels so as to simplify the notations. First, the *transition kernel* expresses the probability density for a particle at $\mathbf{r}'$ to experience its next collision at $\mathbf{r}$.

$$T(\mathbf{r}'\to\mathbf{r},E,\Omega)=\Sigma_t(\mathbf{r},E)e^{-\int_0^s ds\Sigma_t(\mathbf{r}-s\Omega,E)}\left[\frac{\delta\left(\mathbf{\Omega}-\frac{\mathbf{r}-\mathbf{r}'}{|\mathbf{r}-\mathbf{r}'|}\right)}{|\mathbf{r}-\mathbf{r}'|^2}\right]$$

Similarly, the *collision kernel* expresses the probability for a particle entering a collision at $(\mathbf{r},E',\mathbf{\Omega}')$ to be scattered in direction $\mathbf{\Omega}$ and energy E.

$$C(\mathbf{r},E'\to E,\mathbf{\Omega}'\to\mathbf{\Omega})=\frac{\Sigma_s(\mathbf{r},E')f_s(E'\to E,\mathbf{\Omega}'\to\mathbf{\Omega})}{\Sigma_t(\mathbf{r},E')} \tag{1.13}$$

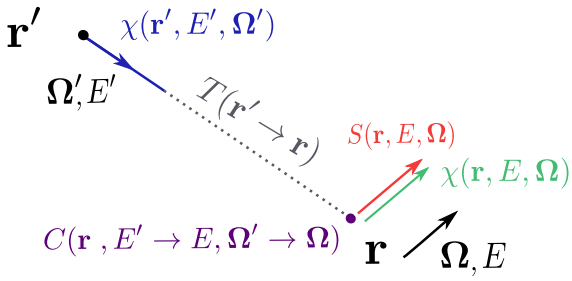


Figure 1.2. Illustration of the integral formulation of the Boltzmann equation for the emission density $\chi(\mathbf{r}, E, \mathbf{\Omega})$

Let $\mathbf{T}$ and $\mathbf{C}$ be the integral operators associated to these kernels. We can write:

$$\Phi(\mathbf{r}, E, \mathbf{\Omega}) = \frac{S_1}{\Sigma_t} + \int \mathbf{TC}\Phi(P')dP'. \tag{1.14}$$

where

$$S_1 = \mathbf{T}S$$

is the source of first collision.

Once the Boltzmann equation is cast into its integral form Eq. (1.14), it is easy to derive other equivalent formulations - some of which do not have any integro-differential counterpart. The integral Boltzmann equation can be written for two interesting physical quantities. The first formulation concerns the **emission density** already introduced. $\chi(P)$ is the rate at which particles are emitted from a point $P \in \mathcal{P}$. The integral equation for $\chi(P)$ reads:

$$\chi(P) = S(P) + \int C(P' \to P)T(P'' \to P')\chi(P'')dP'dP''. \tag{1.15}$$

The observables involved in this equation are illustrated in Figure 1.2. The emission density at P is the sum of the source at P and the transported emission density from any point P'' in phase space that has first experienced a displacement via the transition kernel, and finally undergone a collision at $\mathbf{r}$.

The second one is the **collision density** $\psi(\mathbf{r}, E, \mathbf{\Omega})$, which is the rate at which particles undergo a collision at $P \in \mathcal{P}$. Its relation to the particle flux is straightforward:

$$\psi(\mathbf{r}, E, \mathbf{\Omega}) = \Sigma_t(\mathbf{r}, E, \mathbf{\Omega})\Phi(\mathbf{r}, E, \mathbf{\Omega}) \tag{1.16}$$

The integral equation for ψ reads:

$$\psi(P) = S_1(P) + \int T(P' \to P)C(P'' \to P')\psi(P'')dP'dP'' \tag{1.17}$$

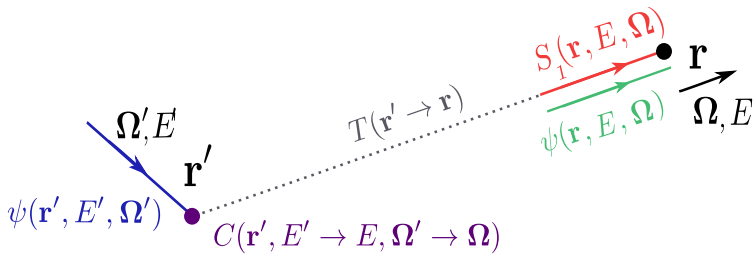


Figure 1.3. Illustration of the integral formulation of the Boltzmann equation for the collision density $\psi(\mathbf{r}, E, \mathbf{\Omega})$

This expression is illustrated in Figure 1.3. The collision density at P is the sum of the first collision source at P and the transported collision density at any point P" in phase space moving it from P'' to P'.

The observables introduced so far stand for the most intuitive perspective on the transport of particles: from a real source through real materials. However, there exists another handy mathematical perspective on the transport of particles codified into the **adjoint theory**.

1.2 The backward perspective

Because this perspective is crucial to understand optimal variance reduction games, we introduce its mathematical formulation. In order to talk about the adjoint flux, we must introduce a scalar product. Let θ be a function of $L_1(\mathbb{R}^6)$ and κ be a test function, whose precise properties will be defined in a moment. Let also $\mathbf{A}$ be a linear operator on $L_1(\mathbb{R}^6)$.

The scalar product between θ and κ is the integral over the entire phase space

$$(\theta, \kappa) = \int \theta(P)\kappa(P)dP. \tag{1.18}$$

And we assume that it converges. We define the adjoint operator $\mathbf{A}^\dagger$ of $\mathbf{A}$ as the provable unique linear operator that satisfies the following relation:

$$(\kappa, \mathbf{A}\theta) = (\theta, \mathbf{A}^\dagger\kappa) \tag{1.19}$$

Formally, the adjoint operator $\mathbf{A}^\dagger$ acts on the dual space of $L_1(\mathbb{R}^6)$, which is the space where the solutions of the direct Boltzmann equation live.

Given these definitions, we suppose that such an integral exists on the phase space of particles for Φ, χ or ψ. We define the operator $\mathbf{L}^\dagger$ adjoint to $\mathbf{L}$, as well as the adjoint observables $\Phi^\dagger$, $\chi^\dagger$ or $\psi^\dagger$, as the solutions of the adjoint Boltzmann equation:

$$\mathbf{L}^\dagger\Phi^\dagger = S^\dagger \tag{1.20}$$

where $S^\dagger(\mathbf{r}, E, \mathbf{\Omega})$ represents an adjoint source. So how does the adjoint transport and

adjoint collision kernels behave? Let us write Eq. (1.19) for $\mathbf{C}$.

$$
\begin{aligned}
&(\kappa^\dagger, \mathbf{C}\theta) \\
&= \int_{\delta P} d^3\mathbf{r}\ dE\ d^2\mathbf{\Omega}\, \kappa^\dagger(P) \int_{\delta P'} \delta(\mathbf{r}-\mathbf{r}') \frac{\Sigma_s(\mathbf{r}, E') f_s(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega})}{\Sigma_t(\mathbf{r}, E)} \theta(P') d^3\mathbf{r}' dE' d^2\mathbf{\Omega}' \\
&= \int_{\delta P'} d^3\mathbf{r}'\ dE'\ d^2\mathbf{\Omega}'\, \theta(P') \int_{\delta P} \delta(\mathbf{r}-\mathbf{r}') \frac{\Sigma_s(\mathbf{r}, E') f_s(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega})}{\Sigma_t(\mathbf{r}, E)} \kappa^\dagger(P) d^3\mathbf{r} dE d^2\mathbf{\Omega} \\
&= (\theta, \mathbf{C}^\dagger \kappa^\dagger)
\end{aligned}
$$

From this, we derive the adjoint collision kernel:

$$
C^\dagger(\mathbf{r}, E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}) = \frac{\Sigma_s(\mathbf{r}, E) f_s(E \to E', \mathbf{\Omega} \to \mathbf{\Omega}')}{\Sigma_t(\mathbf{r}, E)} \tag{1.21}
$$

A similar procedure leads to the adjoint transport kernel $T^\dagger$:

$$
T^\dagger(\mathbf{r} \to \mathbf{r}', E, \Omega) = T(\mathbf{r}' \to \mathbf{r}, E, \Omega) \tag{1.22}
$$

The adjoint transport operator takes the form:

$$
\mathbf{L}^\dagger = -\mathbf{\Omega} \cdot \nabla \cdot + \Sigma_t(\mathbf{r}, E) \cdot - \iint \Sigma_s(\mathbf{r}, E) f_s(E \to E', \mathbf{\Omega} \to \mathbf{\Omega}') \cdot dE' d^2\mathbf{\Omega}' \tag{1.23}
$$

Now that these quantities are defined, we can write the formulations of the adjoint Boltzmann equation.

1.2.1 The integro-differential formulation

Eq. (1.20) leads to the adjoint integro-differential Boltzmann equation reads:

$$
\begin{aligned}
&-\mathbf{\Omega} \cdot \nabla \Phi^\dagger(\mathbf{r}, E, \mathbf{\Omega}) + \Sigma_t(\mathbf{r}, E) \Phi^\dagger(\mathbf{r}, E, \mathbf{\Omega}) = \\
&\qquad \int \Sigma_s(\mathbf{r}, E) f_s(E \to E', \mathbf{\Omega} \to \mathbf{\Omega}') \Phi^\dagger(\mathbf{r}, E', \mathbf{\Omega}') dE' d^2\mathbf{\Omega}' + S^\dagger(\mathbf{r}, E, \mathbf{\Omega})
\end{aligned} \tag{1.24}
$$

We notice that the only differences with Eq. (1.6) is that the streaming term has a minus sign and the integral on the energy and angle now acts on the final state of the particle after collision.

1.2.2 The integral formulation

From Eq. (1.24), we can write the integral formulation for $\psi^\dagger$.

$$\psi^\dagger(\mathbf{r}, E, \mathbf{\Omega}) = S_1^\dagger(P) + \int T^\dagger(\mathbf{r}' \to \mathbf{r}) C^\dagger(\mathbf{r}', E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}) \psi^\dagger(P') dP', \tag{1.25}$$

as well as the integral formulation for $\chi^\dagger$.

$$\chi^\dagger(\mathbf{r}, E, \mathbf{\Omega}) = S^\dagger(P) + \int C^\dagger(\mathbf{r}', E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}) T^\dagger(\mathbf{r}' \to \mathbf{r}) \chi^\dagger(P') dP'. \tag{1.26}$$

1.3 On the solution of these equations

Solutions to these equations can be found with different methods. The deterministic ones use the integro-differential or the characteristic formulations, whereas Monte Carlo methods rely on the integral formulation. A shielding problem can be tackled either by the forward approach or the backward approach. We chose to use the forward approach because it is the most natural one, and because most of Monte Carlo codes do not implement adjoint transport. So how can we derive a Monte Carlo scheme from the integral equation and what implications does it have for shielding applications?

Chapter 2

Handling rare events in Monte Carlo shielding calculations

Contents

2.1 Monte Carlo for particle transport **18**

2.1.1 Sampling scheme 19

2.1.2 Analog transport 23

2.1.3 TRIPOLI-4® 23

2.1.4 Monte Carlo failure to solve shielding problems 23

2.2 Variance reduction: importance sampling **24**

2.2.1 Implicit capture and Russian roulette 25

2.2.2 Exponential transform 26

2.2.3 From the integro-differential formulation 26

2.2.4 *Biased* Sampling 27

2.2.5 TRIPOLI-4® 28

2.2.6 Intelligent random numbers. 29

2.3 Variance reduction: splitting methods **29**

2.3.1 Weight Windows 29

2.3.2 Adaptive Multilevel Splitting 29

2.4 Conclusion **34**

The goal of this chapter is to present how the Monte Carlo method is used to solve neutral particle transport problems. Section 2.1 introduces the method from the integral formulation of the Boltzmann equation, the Monte Carlo code that we use, as well as simple considerations on its failure to treat shielding problems. Then, Sections 2.2 and 2.3 review the mathematical practices used to circumvent the difficulty.

2.1 Monte Carlo for particle transport

The fundamental idea of the method is to model the transport of a single particle as a sequence of pseudo-random events; simulate particle *histories*; and average quantities of interest over independent realisations.

Recall the integral transport equation for the emission density:

$$\chi(P) = S(P) + \int M(P' \to P)\chi(P')dP'$$

where the transport kernel

$$M(P' \to P) = C(P'' \to P)T(P' \to P'') \tag{2.1}$$

is associated to the transport operator $\mathbf{M}$. The idea is to sequentially sample emissions. We imagine the following scheme in which the emission density is first sampled according to the particle source S. Then the transport operator $\mathbf{M}$ is applied. Subsequent emissions are iteratively generated:

$$\begin{aligned}
\chi_0 &= S \\
\chi_1 &= \mathbf{M}\chi_0 \\
&\vdots \\
\chi_n &= \mathbf{M}\chi_{n-1}
\end{aligned}$$

We understand that the emission density is the sum of these terms and that it admits a limit. The emission density can thus be written as a Neuman series expansion:

$$\chi(P) = \sum_{n=0}^{\infty} \chi_n(P). \tag{2.2}$$

So how does this translate in the facts? Consider that we have one particle starting at a point sampled from the source. It is transported collision after collision until the history ends, which can be because with absorption, leakage or a cut on the physics(on the minimal energy for example). This particle history is a realisation that approximates this integral.

2.1.1 Sampling scheme

Let us concentrate on the intermediate steps of history generation.

Source kernel

First, an emission is initially sampled from $\chi_0 = S$. Sources of interest can be simple localised sources, analytical expressions, uniformly distributed sources or resulting from a criticality calculation. In any case, the probability of sampling a source particle in dP around point P can always be written in this form:

$$\pi_s(P)dP = \frac{S(P)}{\int S(P')dP'}dP, \tag{2.3}$$

where π_s is the probability distribution of the source.

Transition kernel

After being emitted, a particle undergoes a flight. For simplicity, we only treat the case where a particle stays in the same volume during a flight; we also assume that the material traversed by the particle can be considered homogeneous over distances of the order of the particle mean free path. Under this assumption, the probability density is taken as an exponential distribution. Let us consider a particle being emitted at $P = (\mathbf{r}, E, \mathbf{\Omega})$. The total cross section of the material at energy E is noted Σ_t. The probability for a particle being emitted at P to experience its next collision at a distance ρ from its emission position r in the direction Ω is

$$\pi_t(\rho)d\rho = \Sigma_t e^{-\Sigma_t \rho} d\rho. \tag{2.4}$$

We want to easily sample ρ according to its probability density π. Let us take $\xi = \Pi^{-1}(\rho)$, where Π^{-1} is the inverse of the cumulative distribution of π. First sample the random variable ξ according to a uniform probability distribution between zero and one

$$\xi \sim \mathcal{U}_{[0,1]}.$$

Now, $\Pi^{-1}(\xi)$ follows the same probability distribution as ρ.

$$\rho = \Pi^{-1}(\xi) = -\frac{1}{\Sigma_t}\ln(1-\xi) \tag{2.5}$$

This way, ρ is sampled with minimum effort. The particle can be displaced from its initial position to the position where it will undergo the following collision. Of course, if a particle reaches a boundary condition, it is treated accordingly to its nature: for reflective boundary conditions, the particle is scattered back into the geometry. For leakage conditions, the particle is stopped being followed.

Collision kernel

Our particle has now finished its flight and is ready to collide with a nucleus. At point $\mathbf{r}, E, \mathbf{\Omega}$, we note Σ_t the total macroscopic cross section at energy E, which is the sum of all microscopic total cross sections σ_i of the N nuclei present in the material at energy E, each weighted by its atomic concentration n_i. The discrete probability of interacting with nuclus i is:

$$p_i = \frac{\sigma_i n_i}{\sum_{j=1}^{N} \sigma_j n_j}$$

A nucleus k is chosen. Once a nucleus is chosen, an interaction k needs to be sampled in the set of all possible interaction j, associated to a cross section $\sigma_{k,j}$ on this nucleus.

$$p_{k,j} = \frac{\sigma_{k,j}}{\sigma_j}$$

We generalise these choices and define π_c the collision probability density. If the interaction chosen is an absorption, then the trajectory is ended.

Secondary particles

Some captures can generate new particles, for example (n, γ), (n, xn) or fission reactions. When such interactions happen, new particles are simply generated and followed in turn.

After the collision, a new flight is sampled the process goes on until the particle is stopped. Let us assume that a particular particle history underwent n collisions resulting in the particle being scatterd at $\mathbf{r}_1$ towards $E_1, \mathbf{\Omega}_1$, then towards $E_n, \mathbf{\Omega}_n$ (i.e. $E_i, \mathbf{\Omega}_i$ is the final state of the i-th scattering event). This particle history then provides Monte Carlo estimates of $\chi(\mathbf{r_1}), E_1, \mathbf{\Omega}_1$ etc

One particle history provides with Monte Carlo estimates of:

$$\chi_0, \chi_1, \cdots$$

χ is build from them with Eq. (2.2). Of course the collision density ψ is concurrently sampled with this strategy (Hoogenboom, 2008).

Another way to look at this sampling cascade is to think about it as if it were the latent mechanism for sampling a bigger object:

$$\text{a trajectory } \mathcal{T}.$$

We define a trajectory as a set of collisions and emissions:

$$\mathcal{T} = \{\chi_1, \psi_2, \chi_3, \dots\} \tag{2.6}$$

Where χ_i, and ψ_i are respectively chosen to denote points emitted from a collision and the points entering a collision. A trajectory can end on a collision or on an emission (if it leaves the problem geometry). The last stop is to sample as many particles as necessary to faithfully represent the neutron population in the problem.

Monte Carlo estimators

With these definitions, Monte Carlo estimators can be written to estimate the average of a response in any region $\mathcal{D}$ of the phase space. Most types of target scores can be written as:

$$R = \int_{\mathcal{D}} \eta_\Phi(P)\Phi(P)dP \tag{2.7}$$

where η_Φ is the response function of the detector exposed to a flux Φ. For example, we wish to compute the total flux in a detector. The response is clearly

$$\eta_\Phi(P) = \mathbb{1}_{\mathcal{D}}(P)$$

So how can we build a Monte Carlo estimator for this response? Recall that the Monte Carlo scheme samples χ and ψ. We would prefer to relate our estimator to a functional over these quantities. We thus define them η_χ and η_ψ.

$$\eta_\psi(P) = \Sigma_t(P)\eta_\Phi(P)$$

and η_χ is skipped voluntarily because its mathematical expression is of no physical interest for now (Hoogenboom, 2008).

The response can be alternatively described by:

$$R = \int \eta_\chi(P)\chi(P)dP, \tag{2.8}$$

or

$$R = \int \eta_\psi(P)\psi(P)dP. \tag{2.9}$$

Adopting one perspective or another rely upon ones' needs. Among the variety of scores available in a MC code, two stand out for the estimation of the flux: the collision estimator

$$\eta_\psi(P) = \frac{1}{V\Sigma_t(P)}\mathbb{1}_{\mathcal{D}}(P). \tag{2.10}$$

and the track-length estimator:

$$\eta_\chi(P) = \frac{l_\chi}{V}. \tag{2.11}$$

where l_χ is the length crossed by a particle in the detector volume V.

Consider a set T of N sampled trajectories.

$$T = \{\mathcal{T}_1, \ldots, \mathcal{T}_N\} \tag{2.12}$$

The Monte Carlo estimators for the response based on the support of the emission (resp. collision) density $\hat{\varphi}_\chi$ (resp. $\hat{\varphi}_\psi$) are defined:

$$\hat{\varphi}_\chi(\mathcal{T}) = \sum_{P \in \mathcal{T}} \eta_\chi(P) \mathbb{1}_\chi(P) \tag{2.13}$$

and

$$\hat{\varphi}_\psi(\mathcal{T}) = \sum_{P \in \mathcal{T}} \eta_\psi(P) \mathbb{1}_\psi(P) \tag{2.14}$$

The response reads as the expectation of the observable $\hat{\varphi}$:

$$R = \mathbb{E}[\hat{\varphi(\mathcal{T})}]$$

With N sampled trajectories, the mean estimators associated become M_χ and M_ψ.

$$M_\chi = \frac{1}{N} \sum_{\mathcal{T} \in T} \hat{\varphi}_\chi(\mathcal{T}) \tag{2.15}$$

and

$$M_\psi = \frac{1}{N} \sum_{\mathcal{T} \in T} \hat{\varphi}_\psi(\mathcal{T}) \tag{2.16}$$

The unbiased *absolute* variance estimator for these means reads:

$$\hat{\sigma}^2 = \frac{1}{N} \frac{1}{N-1} \sum_{\mathcal{T} \in T} (\hat{\varphi}(\mathcal{T}) - M)^2 \tag{2.17}$$

where M is either M_χ or M_ψ. In the following, we will use $\hat{\sigma}^2_{\%}$ to denote the variance of the response estimator in percentage:

$$\hat{\sigma}^2_{\%} = \frac{100}{M} \sigma^2 \tag{2.18}$$

2.1.2 Analog transport

The procedure that we just described is customarily called the *analog* scheme in the sense that particles are transported in the exact way in which their interactions with matter are understood. For example, if a fission occurs and three outgoing neutrons are sampled, then three different particles are followed. The term *analog simulation* is typically only used to refer to simulations where the fundamental conservation laws (energy, momentum, angular momentum) are satisfied at each event.

2.1.3 TRIPOLI-4®

The work of this thesis was carried out in the Monte Carlo transport code developed at SERMA, CEA, Saclay, France: TRIPOLI-4® (Brun et al., 2015). Its main application fields are reactor physics, instrumentation, safety-criticality and radiation protection. Its capabilities to efficiently model radiation protection setups; the fact that it is easy to handle (the main language is C++) makes it a good choice for research and development around new variance-reduction techniques, such as those discussed in the present work.

2.1.4 Monte Carlo failure to solve shielding problems

The main focus of this work is on the particular case of shielding. A shielding problem is conceptually composed of a particle source, and a detector. They may be separated by large blocks of attenuating material, with the possible presence of cavities through which particles can stream. Figure 2.1 schematically illustrates such types of problems.

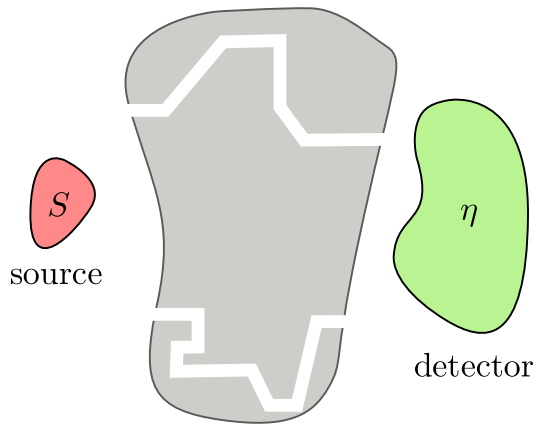


Figure 2.1. A dummy shielding scheme. A source, a detector and attenuating materials or pipes in between.

If the absorbing material is sufficiently thick and the cavities are sufficiently small, then clearly particles will struggle to reach the detector. To illustrate this idea, we present the first realistic benchmark (Boireau et al., 2016). The objective of this study was to determine the amount of neutrinos produced by the fission reaction in the Osiris reactor. A neutrino detector (Nucifer) is placed outside the reactor. The calibration of the detector requires an estimate of the noise produced by fast neutrons in the detector. Monte Carlo

calculations were needed to tell if neutrons would flood the detector and hide the detection of neutrinos. This is a typical application where the probability for neutrons born in the reactor to hit the detector is of the order of 10^{-15}. This means that analog MC require the sampling of 10^{15} histories to see (on average) one particle hitting the detector. This motivates the development of methods that preferentially simulate particles contributing to the target response.

In order to quantitatively assess the performance of a Monte Carlo simulation, we classically define a quality factor, also called the *Figure of Merit* (FOM), that penalises long simulation times T for a given estimate of the variance of the response R:

$$FOM = \frac{1}{\sigma_R^2 T} \tag{2.19}$$

Kenneth Burn (1997) proposes one that takes into account the mean value of the response, but we will stick to this definition and comment on the possible under estimations separately for the sake of simplicity.

Because analog Monte Carlo breaks down for slightly complex problems, we must use mathematical tricks, called *variance reduction* methods. They can be split into two categories: importance sampling methods, and splitting methods.

2.2 Variance reduction: importance sampling

The first type of methods that accelerates Monte Carlo simulations are called **importance sampling** methods (Kahn and Harris, 1951). Assume again that we are interested in computing a detector response R, Analog trajectories are sampled according to the probability distribution θ, which is in general difficult to write out in closed form because it results from unbounded sequences of samplings from the transport kernel. In shielding problems, very few trajectories ever contribute to the detector response. The idea of importance sampling is to sample more frequently trajectories that contribute to the desired response. Let us imagine that we have an idea of where trajectories should pass in order to contribute to the response of interest. For example, we know that in a particular region of the phase space, particles need to scatter on a roof to bypass a wall. Because our trajectories are the result of successive samplings, we could imagine to force particles to bypass the wall by sampling scattering angles from an appropriate distribution.

Formally, this means that we are sampling trajectories $\bar{\mathcal{T}}$ according to a new distribution $\bar{\theta}$. Trajectories generated by this alternative probability density are usually qualified as *biased*. In the following, quantities that are the result of a sampling from $\bar{\theta}$ will be noted with an overbar. The response R that was originally expressed as the expectation of the estimator φ over the probability density θ now can be written as an expectation

over $\bar{\theta}$.

$$\begin{aligned} R = \mathbb{E}[\varphi(\mathcal{T})] &= \int \varphi(t)\theta(t)dt \\ &= \int \varphi(t)\frac{\theta(t)}{\bar{\theta}(t)}\bar{\theta}(t)dt \\ &= \bar{\mathbb{E}}\left(\varphi(t)\frac{\theta(t)}{\bar{\theta}(t)}\right) \end{aligned}$$

The response R now writes as the expectation of φ times the likelihood ratio $\frac{\theta(t)}{\bar{\theta}(t)}$ over the proposal distribution $\bar{\theta}$. The whole problem is to appropriately choose $\bar{\theta}$ that samples trajectories of interest. $\bar{\theta}$ must respect the dominance property with respect to θ. This means that if $\theta(t) > 0$, then $\theta(t)\varphi(t) > 0$.

Because in practice, we cannot write the distribution θ over which trajectories are sampled, it is not straightforward to implement the importance sampling scheme in a Monte Carlo particle transport code. Alternatively, we focus on the successive distributions from which trajectories are built (π_s, π_t, π_c). We state that altering one of these distributions so as to preferentially sample contributing trajectories yields an unbiased result as far as particles are assigned a *statistical weight* that compensates the altered probability with which the event was sampled.

2.2.1 Implicit capture and Russian roulette

If strongly absorbing media are present between the source and the detector, forcing particles to scatter instead of being absorbed can encourage particles to get through the difficult part of the problem and eventually contribute to the detector. Let σ_a be the absorption cross section of a certain isotope on which a particle is colliding. The absorption event is sampled with a rate

$$\frac{\sigma_a}{\sigma_t}.$$

Instead, we may choose to sample a scattering event and change the weight of the particle exiting the collision w_{out} as a function of its weight entering the collision w_{in} as:

$$w_{\text{out}} = \left(1 - \frac{\sigma_a}{\sigma_t}\right) w_{\text{in}}$$

This technique is named **implicit capture** and it belongs to the importance sampling framework. In its naive formulation, it is quite inefficient because it leads to particles with monotonically decreasing weights, that however are only killed when they leak out of the system, this leads to spending large amounts of CPU time on the simulatoin of particles that yield negligible contributions. We thus use the *russian roulette* method to

truncate particle histories. If the weight of a particle decreases below a fixed threshold ($w_{\mathrm{RR}} = 0.8$ is the default value in TRIPOLI-4®) the procedure is: (a) sample a number ξ between zero and one. (b) If ξ is below w_{RR}, then kill the particle. Otherwise, let it survive with a survival weight w_{surv}. Russian roulette limits the simulation time of particles with negligible weights and can thus increase the FOM. There are many variations on the theme of Russian Roulette, depending on how one chooses the threshold, the survival weight, etc.

2.2.2 Exponential transform

The exponential transform (ET) was first introduced by (Levitt, 1968). He intuitively inferred that the particle flight length should be stretched in the directions of interest by studying purely attenuating slabs. The transformation he proposes is to set a parameter $a \in]0;1]$

$$\Sigma_t \leftarrow a\Sigma_t$$

2.2.3 From the integro-differential formulation

In order to apply the importance sampling scheme to the simulation of a certain response - for example the particle flux - we introduce a function I on the phase space. This function is called the *importance function*. We then replace $\psi(P)$ with $\bar{\psi}(P)/I(P)$ in the integro-differential equation 1.6. It is equivalent to change $\Phi(P)$ into $\bar{\Phi}(P)/I(P)$. By doing so and multiplying by $I(\mathbf{r}, E, \mathbf{\Omega})$ throughout, we obtain

$$I(\mathbf{r}, E, \mathbf{\Omega})\mathbf{\Omega} \cdot \nabla \left(\frac{\bar{\Phi}(\mathbf{r}, E, \mathbf{\Omega})}{I(\mathbf{r}, E, \mathbf{\Omega})} \right) + \Sigma_t(\mathbf{r}, E)\bar{\Phi}(\mathbf{r}, E, \mathbf{\Omega}) = \int C(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}) \frac{I(\mathbf{r}, E, \mathbf{\Omega})}{I(\mathbf{r}, E', \mathbf{\Omega}')} \bar{\Phi}(\mathbf{r}, E', \mathbf{\Omega}') dE' d^2\mathbf{\Omega}' + I(\mathbf{r}, E, \mathbf{\Omega}) S(\mathbf{r}, E, \mathbf{\Omega})$$

The gradient term can be split and rearranged so that the equation takes the same form as the original Boltzmann equation.

$$\mathbf{\Omega} \cdot \nabla \bar{\Phi}(\mathbf{r}, E, \mathbf{\Omega}) + \left(\Sigma_t(\mathbf{r}, E) - \mathbf{\Omega} \cdot \frac{\nabla I(\mathbf{r}, E, \mathbf{\Omega})}{I(\mathbf{r}, E, \mathbf{\Omega})} \right) \bar{\Phi}(\mathbf{r}, E, \mathbf{\Omega}) = \int \bar{C}(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}) \bar{\Phi}(\mathbf{r}, E', \mathbf{\Omega}') dE' d^2\mathbf{\Omega}' + \bar{S}(\mathbf{r}, E, \mathbf{\Omega}) \quad (2.20)$$

The biased source reads:

$$\bar{S}(\mathbf{r}, E, \mathbf{\Omega}) = I(\mathbf{r}, E, \mathbf{\Omega}) S(\mathbf{r}, E, \mathbf{\Omega}) \quad (2.21)$$

Eq. (2.20) is formally similar to Eq. (1.6) provided that we interpret

$$\bar{\Sigma}_t(\mathbf{r}, E) = \Sigma_t(\mathbf{r}, E) - \mathbf{\Omega} \cdot \frac{\nabla I(\mathbf{r}, E, \mathbf{\Omega})}{I(\mathbf{r}, E, \mathbf{\Omega})} \tag{2.22}$$

as an *apparent* total cross section. The biased collision kernel reads:

$$\bar{C}(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}) = C(E' \to E, \mathbf{\Omega}' \to \mathbf{\Omega}) \frac{I(\mathbf{r}, E, \mathbf{\Omega})}{I(\mathbf{r}, E', \mathbf{\Omega}')} \tag{2.23}$$

2.2.4 *Biased* Sampling

Now that we have derived all the information that the Boltzmann equation could give on the shape of the biased kernels, let us write the sampling strategy for $\bar{\chi}$ and $\bar{\psi}$.

Sampling the corrected source

Sampling from the corrected source is easy. Remark though, that it normalises to a different value than S:

$$\bar{\pi}_s(P)dP = \frac{\bar{S}(P)}{\int \bar{S}(P')dP'} dP \tag{2.24}$$

Sampling the corrected transition kernel (*path stretching*)

The integro-differential form of the equation highlighted an apparent macroscopic total cross section.

$$\bar{\Sigma}_t(\mathbf{r}, E) = \Sigma_t(\mathbf{r}, E) - \mathbf{\Omega} \cdot \frac{\nabla I(\mathbf{r}, E, \mathbf{\Omega})}{I(\mathbf{r}, E, \mathbf{\Omega})}$$

We write the corrected sampling distribution for the flight length:

$$\bar{\pi}_t(\rho)d\rho = \bar{\Sigma}_t e^{-\bar{\Sigma}_t \rho} d\rho \tag{2.25}$$

This means that if the particle is about to go in a direction of interest ($\mathbf{\Omega} \cdot \nabla I > 0$), its flight length is stretched. On the other hand, it is shortened if it goes in the opposite direction ($\mathbf{\Omega} \cdot \nabla I < 0$). Figure 2.2 illustrates the idea.

Note that Clark (1966) reports that the cross section can become negative and proposes solutions to remedy this problem.

Sampling the biased collision kernel (*collision biasing*)

Sampling the collision with importance sampling is more tricky. One needs to integrate the importance of all possible outcomes from the collision and sample from this distribution. Formally, this is similar to sampling from the biased source; however, contrary to the source case, the biased distribution must be constructed *ex novo* at each collision, at

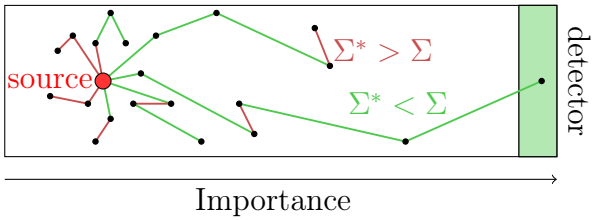


Figure 2.2. Illustration of path stretching. The importance is minimum near the source and increases along the axis towards the detector. The lengths of the particle flights are stretched if they move towards higher importance zones (green), and shortened otherwise (red).

least for the typical case where the scattering kernel is a function of $\mathbf{\Omega} \cdot \mathbf{\Omega}'$. Since this is computationally expensive, this is rarely done

2.2.5 TRIPOLI-4®

TRIPOLI-4® implements a variant of the exponential transform (ET) algorithm. Assume for now that TRIPOLI-4® uses a scalar importance map $I(r, E)$ defined on a mesh and on a multi-group energetic grid. TRIPOLI-4® implements importance sampling for the source, path stretching and a light version of collision biasing. The path stretching is made with the corrected expression of the total cross section. The gradient of the importance is computed once for all at the beginning of the simulation with finite differences for each energy group and each cell of the mesh.

Because particle weights change along the history, splitting/roulette is played along with path stretching. Splitting and roulette are performed depending on the discrepancy Δw between the weight of the particle w and the weight required by the importance function $w_r(P) = 1/I(P)$:

$$\Delta w = \frac{w}{w_r}$$

If $\Delta w \leq 0.5$, russian roulette is played with a survival weight of w_r. If $\Delta w \geq 2$, the particle is split into $int(\Delta w)$ particles. The number of splits is capped at 10 in order to avoid extreme scenarios of massive particle splits. This trades off some biasing efficiency for some guarantees on resource usage.

Particles are deposited along the trajectory with the *stratified sampling* method. It consists in forcing collisions to occur along the flight path, in order to improve coverage of phase space (Thomas E. Booth, 2005). Another option allows to perform splitting exclusively at surfaces between volumes.

There exists a version of the collision biasing scheme implemented in TRIPOLI-4®. Because it is difficult to integrate the importance over all possible outcomes, TRIPOLI-4® samples a fixed number (three) of collisions at each collision, evaluates the importance for each of the three sampled final states and selects the one that has the highest importance. This *sampling-rejection* technique is questionable because it approximates the integral over all possible outcomes by only three outcomes. Moreover, the importance is scalar.

2.2.6 Intelligent random numbers.

Another method that implements importance sampling is the one proposed by Thomas E Booth (1986), who proposes learning in the random number space. The author takes the example of the exponential transform: he says that instead of sampling ξ uniformly in $]0, 1]$ to sample from $\bar{T}$, one could sample directly ξ according to the appropriate probability density. He showed that this approach has some advantages and drawbacks. The author tried to get rid of the physical phase space and learn the optimal importance sampling distribution in the random number space. After some successes and failures, he decided to upgrade his technique by including the physical phase space, which gave birth to an hybrid importance sampling scheme, later implemented in MCNP (Thomas E. Booth, 1988).

2.3 Variance reduction: splitting methods

The second family of variance reduction methods are splitting methods. They consist in duplicating particles that are likely to contribute to the detector, and removing the ones that don't. Two methods are presented. The weight windows and the last one that was implemented in TRIPOLI-4®, Adaptive Multilevel Splitting.

2.3.1 Weight Windows

Weight windows were first implemented by T. Booth for the particle-transport code MCNP. Thomas E Booth (2006) describes how the idea came to his mind. The original idea was to split particles that travel in important regions of the phase space, and kill particles that traverse less importance regions. This ensures to simultaneously increase the population near the detector and control the particle population in order to keep the required resources (memory and computational time) in check.

In their first incarnation, weight windows were defined per volume. This means that every time a particle leaves a volume, the question is asked: is the importance in the new volume lower or higher that the importance in the previous volume? If it were lower, roulette is played. Contrarily, particles are split if the importance of the new volume is higher than the one the particle is coming from.

Later, improvements were made to superimpose a mesh on the geometry, such that the importance could be refined within the volumes itself (Evans and Hendricks, 1998). Both method require the definition of an **importance function** that estimates the contribution to the detector response of a particle starting in this volume.

2.3.2 Adaptive Multilevel Splitting

Adaptive Multilevel Splitting (AMS) is born from the hope of improving multilevel splitting methods (Cérou and Guyader, 2005; Cérou, Guyader, et al., 2013). Its strength relies on its capability of adaptively fixing splitting levels during the simulation, without the intervention of the code user. It was successfully applied to particle transport by Louvin (2017). TRIPOLI-4® is the only Monte Carlo particle transport code implementing this algorithm. How does it work?

The philosophy of AMS resembles the one of importance sampling in the sense that it aims at sampling trajectories that effectively contribute to a target response. From this perspective, it generates trajectories from a set of probability distributions

$$\theta_{\text{AMS}}.$$

The approach differs from importance sampling in the fact that θ_{AMS} is *iterative*.

Initialisation phase

A set $S_0 \in \mathbb{S} = \mathbb{T}^N$ is constructed by the sampling of N trajectories following a strategy θ_T. For now, this strategy is simply taken to be the *analog* one. Formally Louvin (2017) showed that the implicit capture can also be used. The use of different trajectory strategies will be discussed in the perspectives of this PhD thesis. For now, this formally writes:

$$\mathcal{T}^{(0)}_{i\in[1;N]} \sim \theta_T(\mathcal{T}|x_0 \sim \pi_s) \tag{2.26}$$

where π_s is the source distribution and x_0 the starting point of trajectory $\mathcal{T}$.

$$S_0 = \{\mathcal{T}^{(0)}_1, \cdots, \mathcal{T}^{(0)}_N\} \tag{2.27}$$

The plan is to make this set of trajectories iteratively evolve so that is has more chances to contribute to the detector response. So how is S_1 sampled from S_0? In order for the algorithm to favor important trajectories, it needs to be fed with a function that marks each trajectory $\mathcal{T}$. This function is called the *reaction coordinates* ξ:

$$\xi : \mathbb{T} \to \mathbb{R}. \tag{2.28}$$

It maps any trajectory in the trajectory or path space $\mathbb{T}$ to a real value. It quantifies how well a trajectory performed in getting close to the target. Because a trajectory is a sequence of points $P \in \mathcal{P}$, a second importance function can be defined on the phase space:

$$\xi : \mathcal{P} \to \mathbb{R}. \tag{2.29}$$

The trajectory importance is simply defined as the maximum importance achieved by its points:

$$\xi(\mathcal{T}) = \max_{P\in\mathcal{T}} \xi(P) \tag{2.30}$$

This function must respect one condition for particles to be driven towards a certain target. There must be a part of the target where the importance is maximum (z_{max}). In TRIPOLI-4®, the importance is set to infinity whenever a particle enters the target volume (defined by the user).

Sorting phase

With this definition, AMS performs a permutation σ_ξ of S_0 and returns an ordered set $S' \in \mathbb{S}$. The permutation:

$$\begin{aligned}\sigma_\xi :& \mathbb{S} \to \mathbb{S} \\ & S \to S' = \{\mathcal{T}'_1, \cdots, \mathcal{T}'_N\} \\ & = \{\mathcal{T}_{\sigma_{\xi(1)}}, \cdots, \mathcal{T}_{\sigma_{\xi(N)}}\}\end{aligned}$$

is defined such that

$$\xi(\mathcal{T}_{\sigma_\xi(1)}) \leq \cdots \leq \xi(\mathcal{T}_{\sigma_\xi(N)})$$

Selection phase

A selection phase follows. An arbitrary number $k < N$ is fixed by the user. We define the **AMS level** at iteration 0:

$$Z_0 = \xi(\mathcal{T}'_k)$$

as the importance of the k-th element of S'. S' is split into two sets depending on the condition:

$$\xi(\mathcal{T}') \leq Z_0$$

Let us call K_0 the number of trajectories that respect this condition. Because more than one trajectories can have its importance equal to Z_0, K_0 can be greater than k. Then, the set of *unactive* trajectories is defined as

$$S_0^{\text{off}} = \{\mathcal{T}'_1, \ldots, \mathcal{T}'_{K_0}\}$$

and the set of *active* trajectories as:

$$S_0^{\text{on}} = \{\mathcal{T}'_{K_0+1}, \ldots, \mathcal{T}'_N\}$$

The idea is to remove the trajectories that did not succeed in reaching level Z_0.

Duplication phase

In order to keep the same number of histories from one iteration to another, K_0 trajectories are restarted on the remaining $N - K_0$ trajectories. For this, K_0 trajectories are randomly chosen from S_0^{on}. For each trajectory

$$\mathcal{T} = \{P_1, \cdots, P_n\},$$

chose the splitting point that has an importance greater than the AMS level.

$$\{P \in \mathcal{T} \in S_0^{\text{on}} \mid \xi(P) > Z_0\}$$

This assures that all trajectories sampled at the next iteration have at least crossed the level Z_0. We choose P_{split} as being the first point crossing the AMS level.

$$P_{\text{split}} = P_{\inf\{i \in [1,n] | \xi(P_i) > Z_0\}}$$

Next iteration

Now, K_0 trajectories are sampled from $\theta_T(\mathcal{T}|x_0 = P_{\text{split}})$, and we are at the next iteration with N trajectories forming an updated set S_1.

On the stopping criteria

Iterations are performed until a stopping criterion is met. If Z_q is greater than the importance in the detector ($z_{\max}$), then it is assumed that enough particles have contributed to the target response. Also, if all trajectories have the same importance at an iteration, AMS is stopped. This means that the combination of the AMS parameters along with the importance function must be adequate for the algorithm to sample trajectories that contribute to the target response.

Illustration

Because AMS has only been recently implemented in a Monte Carlo particle transport code (TRIPOLI-4®), we illustrate its mechanisms on a very simple example Figure 2.3. A source emits particles on the left side of the axis. The detector is on the right hand side of the axis, and is delimited by a squared region. Three particles ($N = 3$) are emitted from the source. k is set to one. At the first iteration, the worse trajectory is removed from the active set and resampled on one of the two remaining trajectories. Iteration after iteration, splitting points are chosen just above the AMS level and trajectories finish to reach the target.

On the estimators

The set of trajectories sampled from θ_{AMS} is complex, but an unbiased estimator for any observable is given by Brehier et al. (2016). Consider the full AMS simulation, the set of AMS trajectories sampled until iteration Q is:

$$T^{\text{AMS}} = \left(\bigcup_{q=0}^{Q} S_q^{\text{off}} \right) \bigcup S_Q^{\text{on}} \tag{2.31}$$

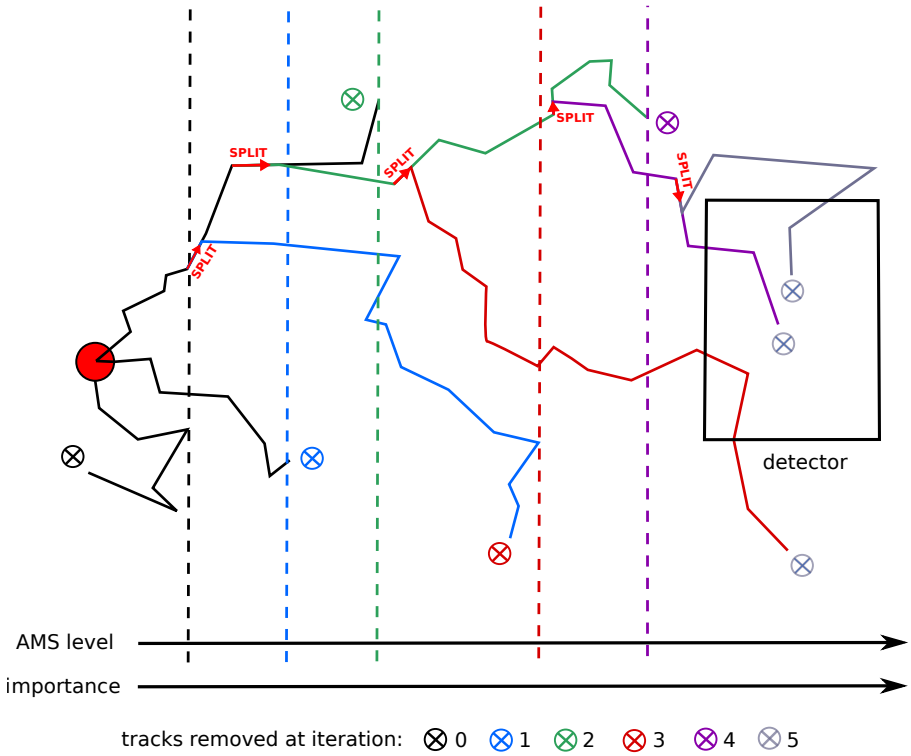


Figure 2.3. Illustration of the AMS algorithm

Set $q_{\text{off}} : \mathcal{T} \to \mathbb{N}$ the iteration at which the trajectory $\mathcal{T}$ has been put into the unactive set of trajectories. Then, the AMS weight of a trajectory is defined as:

$$w_{\text{AMS}}(\mathcal{T}) = \frac{1}{N} \prod_{i=0}^{q_{\text{off}}} \left(1 - \frac{K_i}{N}\right), \text{ for } \mathcal{T} \in T_{\text{AMS}} \tag{2.32}$$

The AMS estimator for the collision density reads:

$$\hat{\varphi}_{\text{AMS},\,\psi} = \sum_{\mathcal{T} \in T^{\text{AMS}}} w_{\text{AMS}}(\mathcal{T})\ \hat{\varphi}_{\psi}(\mathcal{T}) \tag{2.33}$$

and for the emission density:

$$\hat{\varphi}_{\text{AMS},\,\chi} = \sum_{\mathcal{T} \in T^{\text{AMS}}} w_{\text{AMS}}(\mathcal{T})\ \hat{\varphi}_{\chi}(\mathcal{T}). \tag{2.34}$$

The on the fly scoring optimization

An interesting optimisation was proposed by Louvin (2017) in his PhD. It applies when one wishes to collect scores in intermediate regions that are not the AMS target. The algorithm assigns an importance to the region of the phase space in which a response

is desired. This importance is first set to minus infinity, and updated as the minimum importance of all particles that have entered this region. At each iteration, it is compared to the AMS level. If the AMS level is higher than the importance of the region, it means that at least one trajectory that entered the region will be de-activated at the next iteration. It is chosen to collect the score at this moment, in order for all the contributions in the region to be weighted by the same AMS weight. Indeed, if one more iteration were made, there would be trajectories contributing with different AMS weights, which is computationally annoying.

On candidates for splitting trajectories

Note that from the definition of the splitting candidates, any point on the trajectory can be taken for duplication. However, the default option in TRIPOLI-4® is that only emission points are considered to be splitting candidates. Adding the points before collision did not show any significant improvement (Louvin, 2017). However, adding crossing points between volumes really helps in strongly attenuated problems.

2.4 Conclusion

We derived the Monte Carlo scheme from the integral formulation of the Boltzmann equation. We saw that the analog strategy fails to treat shielding problems. This pushed us to use variance reduction methods. We reviewed the main methods implemented in state of the art Monte Carlo particle transport codes and saw that most of them require the definition of an importance function. Most studies show (Levitt, 1968) that these methods strongly rely on the definition of a good importance map. So how are importance maps computed in practice? Which ones perform best?

Part II

The importance map

Chapter 3

Optimal importance map

Contents

3.1 Importance and adjoint quantities **38**

3.2 Zero variance scheme and a way out? **39**

3.3 Practices for the generation of the importance map **41**

3.3.1 Deterministic importance 41

3.3.2 Weight Window Generator 42

3.3.3 DSA 42

3.3.4 Serpent2 43

3.3.5 INIPOND 43

3.3.6 What is done outside the nuclear sphere? 47

3.3.7 Adaptive Multilevel Splitting 47

3.4 Conclusion **47**

Because the importance map is the cornerstone of every variance reduction scheme, colossal amount of work was directed at its generation. In particular, studies have focused on the existence of an optimal importance map. This chapter reviews the methodologies examined to generate the importance map. Section 3.1 derives the physical interpretation of the adjoint flux. Section 3.2 explains why the solution to the adjoint Boltzmann equation has been at the center of research for the past sixty years. Then, Section 3.3 gathers some practices used to generate importance maps.

3.1 Importance and adjoint quantities

So far, we have designated the importance as being the arbitrary function that we choose to feed our variance reduction methods with. In the literature, it is sometimes interpreted as the expected detector response. Let us explain the origin of this interpretation.

Let us consider a problem with a direct source S that gives rise to an emission distribution χ, and a detector with sensitivity to the emission density η_χ. Recall from Chapter 2 that the sought response reads:

$$R = \int \eta_\chi(P)\chi(P)dP.$$

We now interpret η_χ as being the adjoint source $S^\dagger$ of *another problem formulated in the backward formalism* (Bell and Glasstone, 1970).

$$R = \int S^\dagger(P)\chi(P)\,dP \tag{3.1}$$

We are now treating two problems: the direct one, with source S, producing χ and described by the integral transport equation Eq. (1.15) and an adjoint one with source $S^\dagger$ that entails the adjoint emission density $\chi^\dagger$ described by the adjoint integral equation Eq. (1.26). Multiplying the direct equation by $\chi^\dagger$ and the adjoint one by χ leads to the following relation after their substraction:

$$\int S^\dagger(P)\chi(P)\,dP = \int S(P)\chi^\dagger(P)\,dP \tag{3.2}$$

This relation is known as the *duality* or *reciprocity* relation. It relates the observables of two different problems: one is direct and the other is adjoint. From this relation we derive an interpretation of $\chi^\dagger$ by considering the source of the direct problem to be a dirac δ_0 (Bell and Glasstone, 1970). χ_0 is the emission density resulting from choosing the direct source $S = \delta_0$.

$$\int \delta_0(P)\chi^\dagger(P) = \int S^\dagger(P)\chi_0(P)dP$$

Because $S^\dagger = \eta_\chi$, we can finally write:

$$\chi^\dagger(P) = \int \eta_\chi(P)\chi_0(P)dP = R_0 \tag{3.3}$$

The adjoint emission density generated by setting the adjoint source to be the sensitivity to the emission density is thus the response R_0 in the detector to a unit source δ_0. This explains why adjoint quantities are frequently references to as the importance of a point in the phase space. It can be interpreted as the *contribution* of a point in phase space to the response.

3.2 Zero variance scheme and a way out?

When Kahn and Harris (1951) introduced importance sampling, they stated that there exists an optimal form of the importance map for which the sampling scheme results in a zero variance estimator. Kahn and Harris (1951), Sweezy et al. (2005), Coveyou, Cain, and Yost (1967) and Hoogenboom (2008) have mainly contributed to the understanding of these schemes. It can be shown that zero-variance game can be achieved if the following conditions are respected:

- biased source
- biased flight kernel
- biased collision kernel
- infinite trajectory (cannot be rouletted, absorbed nor leak out the geometry).
- the importance is chosen to be the solution to the adjoint Boltzmann equation

In his recent review of zero-variance schemes, Hoogenboom (2008) proposes to derive schemes for different MC estimators. We briefly discuss two of them: the collision estimator and the track estimator.

An optimal collision estimator

The zero-variance collision estimator relies on the definition of the response as a function of the collision density ψ:

$$R = \int \eta_\psi(P)\psi(P)\, dP. \tag{3.4}$$

The domain $\mathcal{D}_\psi$ on which $\eta_\psi(P)$ is not null is simply the region of the configuration space delimited by the detector. The optimal importance function is chosen to be:

$$I_\psi^{\text{opt}}(P) = \frac{R}{\psi^\dagger(P)}$$

where $\psi^\dagger(P)$ is the solution to the adjoint Boltzmann equation with a source $S^\dagger = \eta_\psi$.

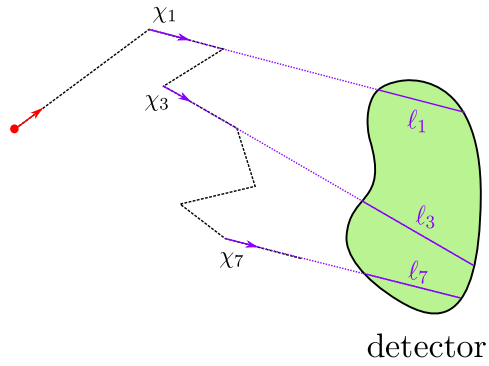


Figure 3.1. Zero variance track estimator

It can be shown that choosing I_ψ^{opt} as an importance map implies that the trajectory contributes to the collision estimator with a constant value R. So the scheme yields a zero-variance estimator.

An optimal track estimator

A track estimator that yields a zero-variance estimate can also be defined. The response now needs to be tackled at the emissions:

$$R = \int \eta_\chi(P)\chi(P)\, dP. \tag{3.5}$$

Take a detector $\mathcal{B}$. The domain D_χ on which the response is not null is defined as the ensemble of points that respect the following condition:

$$P \in \mathcal{D}_\chi = \{P \in \mathbb{R}^6 | \{\mathbf{r} + \lambda \mathbf{\Omega} : \lambda > 0\} \cap \mathcal{B} >\neq \emptyset\},$$

In other words, we draw virtual rays after each emission. The response at each emission point must be the expected contribution in the detector if it had travelled its way through the detector. Figure 3.1 illustrates the zero variance track domain in which η_ψ is not null. Then, the optimal importance reads:

$$I_\chi^{\text{opt}}(P) = \frac{R}{\chi^\dagger(P)}$$

where $\chi^\dagger$ is the solution to the adjoint Boltzmann equation where the adjoint source is taken to be η_χ It can be shown that the resulting track estimator always yields the same response, and has thus zero-variance.

In reality, a MC code is far from respecting the zero-variance scheme. What are the main differences? How do they affect the variance of the MC estimator?

On the impracticality of generating the optimal trajectory

Because our computational budget is not infinite, we are forced to truncate trajectories earlier with roulette. We are simply left with N trajectories per batch; each did their best before being rouletted. Paradoxically, it is worth waiting indefinitely long for the trajectory to be optimum. Hoogenboom (2008) indeed showed that lowering the roulette threshold, decreases the variance of the estimator effectively.

The support of the importance map

There are several reasons why the importance map is never optimal in realistic shielding calculations. First, the importance map is projected on a discretised support (usually a mesh in space and in energy). Second, deterministic methods make approximations to the physics and will never describe exactly the Monte Carlo problem.

The zero-variance track estimator is expensive

Although the zero-variance collision estimators are analogous to the ones implemented in MC codes, the track estimator is not. Implementing it would require to track a particle from its emission to the detector, and compute the length travelled in the detector. This would introduce a large overhead on the simulation time.

Although MC sampling schemes are not optimal, the literature showed that getting a good approximation of the adjoint flux for the importance map still provides excellent accelerations. So how did people generate approximate solutions to the adjoint Boltzmann equation, and what were the alternatives tried?

3.3 Practices for the generation of the importance map

We start by covering the Weight Window method, because it is the most widely used. Although there is no theoretical proof of an optimum importance function for the Weight Window, the adjoint flux has been shown to be a good guess. The two following subsections treat two methods to approximate the adjoint flux of a problem. The first one uses deterministic modules whereas the second implements a Monte Carlo score.

3.3.1 Deterministic importance

Deterministic codes are flexible and can provide the solution to the adjoint Boltzmann equation. All the scattering information is contained within a transition matrix. Inversing the collision kernel simply requires the transposition of this direct scattering matrix. Using a deterministic adjoint flux as an importance map is referred to as the *Consistent Adjoint Driven Importance Sampling* (CADIS) methodology. It is discussed in detail in the next chapter.

3.3.2 Weight Window Generator

In order to estimate the importance, the so called **Weight Window Generator** was implemented in MCNP. It consists in looking at the ratio between two quantities over a discretized phase space (spatial and energetic mesh):

$$\frac{\text{the total contribution to the response of trajectories entering a cell}}{\text{the total weight of trajectories entering a cell for the first time}} \tag{3.6}$$

It is difficult to write this estimator as a function of physical observables (J. Hoogenboom and Légrády, 2005). Evans and Hendricks (1998) compared this importance score to the real adjoint flux in some simple situations, and large discrepancies were observed. The authors corrected the score by counting the trajectories re-entering the scoring cells. They conclude on the fact that their score now correctly estimates the importance and reads:

$$\frac{\text{the total contribution to the response of trajectories entering a cell}}{\text{the total weight of trajectories entering a cell}} \tag{3.7}$$

Although the original implementation did not exactly compute the importance (Thomas E Booth, 2004), we would not qualify this as a "pathology" as Evans and Hendricks (1998) did. We state that the initial implementation of the Weight Window is similar to the "**first visit Monte Carlo policy**". The first visit Monte Carlo policy is a method that estimates the *value* function in the reinforcement learning (RL) framework (Sutton and Barto, 1998). From this point of view, the reinforcement learning *agent* estimates the *value* function of a particular state s in the state space $\mathcal{S}$. If we consider a cell of the importance map as a state in the context of reinforcement learning, then what Booth does is exactly "first visit Monte Carlo". The corrected estimator written by Evans and Hendricks (1998) also has an equivalent in the RL framework called the "**every visit Monte Carlo policy**".

A more limiting aspect of the Weight Window Generator strategy is that this score does not necessarily converge if the initial importance map is ill-conditioned. One way for MCNP users to overcome this issue, is to perform iterations over the MC WWG runs (that may themselves be iterations). They start with a calculation where they reduce the density of their materials and score the WWG. The density is iteratively increased and brought back to its original value. This way, sufficient statistics are accumulated on the path from the source to the detector and particles succeed in reaching the detector.

3.3.3 DSA

The direct statistical approach is an alternative to the estimation of the Weight Window parameters. This method was frequently upgraded with new features over time (K.W. Burn, 1990; K. W. Burn, 1992; Kenneth Burn, 1997). The idea is to optimize the splitting parameter at each surface or phase space region. This splitting parameter ν_s is a multiplication factor assigned to a *specific* surface s. If it is greater than one, particles of statistical weight w_{in} crossing s are split into ν_s particles, each with an updated weight

equal to w_{in}/ν_s . If it is lower than one, particles of statistical weight w_{in} entering s are rouletted and survive with a weight equal to w_{in}/ν_s. It was shown that optimizing these parameters during the simulation yields better results than exclusively relying on the adjoint flux.

3.3.4 Serpent2

Serpent2's variance reduction strategy also relies on Weight Windows. Leppänen, Viitanen, and Hyvönen (2017) introduced a method that aims at estimating the importance in each cell of a mesh. It is called the "response matrix" method. It relies on the computation of currents from a cell to its neighbours. The score is formed by summing the contribution of source particles emitted in a cell with the contribution resulting from the inward currents in this cell.

3.3.5 INIPOND

INIPOND is a TRIPOLI-4® module initially developed to support the Importance Sampling (IS) scheme (Both, Nimal, and Vergnaud, 1990). The goal of INIPOND is to give the importance $I_{\text{INIPOND}}(\mathbf{r}, E, \mathbf{\Omega})$ of a particle emitted at any point $(\mathbf{r}, E, \mathbf{\Omega})$ in the phase space.

The description of the INIPOND module will be illustrated by a theoretical 2D geometry depicted in Figure 3.2. Two materials M_1 and M_2 are present. Let us imagine that we are willing to compute the flux in the detector represented by the green region. Variance reduction methods require the estimation of the importance at any point in the phase space. So we ask the question: how does INIPOND estimate the importance of a particle leaving position $\mathbf{r}$ with energy E and in the direction $\mathbf{\Omega}$?

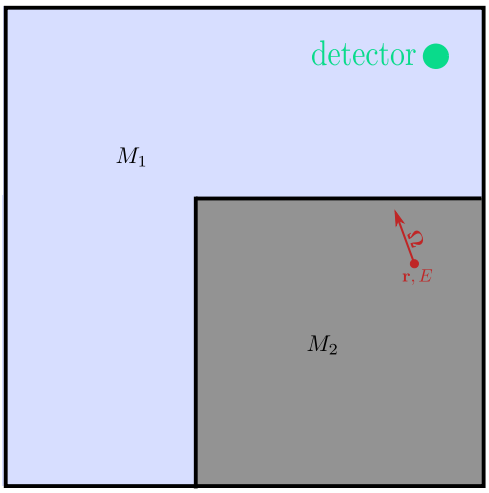


Figure 3.2. Geometry of a theoretical 2D model for the illustration of INIPOND. Two materials are present. The detector where the response is sought is in the top right corner and we are interested in estimating the importance of a particle leaving an arbitrary point in the phase space $(\mathbf{r}, E, \mathbf{\Omega})$

INIPOND factors the importance function as the product of a factor depending only on energy, a factor depending on space and energy and a factor depending on space and

angle.

$$I_{\text{INIPOND}}(\mathbf{r}, E, \mathbf{\Omega}) = \mathcal{I}_{\text{energetic}}(E) \times \mathcal{I}_{\text{spatial}}(\mathbf{r}, E) \times \mathcal{I}_{\text{angular}}(\mathbf{r}, \mathbf{\Omega}). \tag{3.8}$$

The energetic component

The energetic component is treated as in multi-group deterministic codes. The energy range is partitioned into n energy bounds delimited by the energy ranges:

$$E_n \leq \cdots \leq E_1.$$

The way INIPOND distinguishes the materials' compositions is particular and requires splitting the explanation into two steps.

The first step concerns the cross section condensation on the energetic mesh. INIPOND condenses point-wise cross sections for all materials into multi-group cross sections on the n energy groups defined by the user. This condensation is performed over a spectrum representative of a Pressurized Water Reactor(PWR). Figure 3.3 shows this condensation spectrum. It is separated into three parts:

1. A thermal region, where a Maxwell spectrum is assumed.
2. A slowing down region for epi-thermal energies
3. A fission spectrum region for high energies.

Remark that for problems where this spectrum is not appropriate, multi-group cross sections might be poorly estimated, and thus the obtained solution of the importance map might not be suitable for the variance reduction methods used further. Cross sections are generated for either neutrons or photons. In both cases, a P1 anisotropy is modelled with a transport correction to the isotropic cross sections. For problems where thermal neutrons are present, or photons, this represent a real bottleneck for the quality of the cross sections, because higher orders of anisotropy should be used.

The second step of the cross section processing by INIPOND is the conversion of multi-group cross sections into coefficients assigned to each material, called Placzek coefficients. For a given material and a given energy group, the scattering cross section - Σ_s - and the total cross section - Σ_t - are condensed. Then, the Placzek coefficient k is computed as the solution to the equation:

$$\frac{\Sigma_s}{2k} \ln\left(\frac{\Sigma_t + k}{\Sigma_t - k}\right) = 1. \tag{3.9}$$

In some cases, when a material with a very low density is present, the user can force this material to be assigned a Placzek coefficient that is systematically half lower than the ones of all the other materials. This allows to prevent flat importance maps in regions where Σ_t is too low. In the following we will refer to this heuristic technique as the *void trick*.

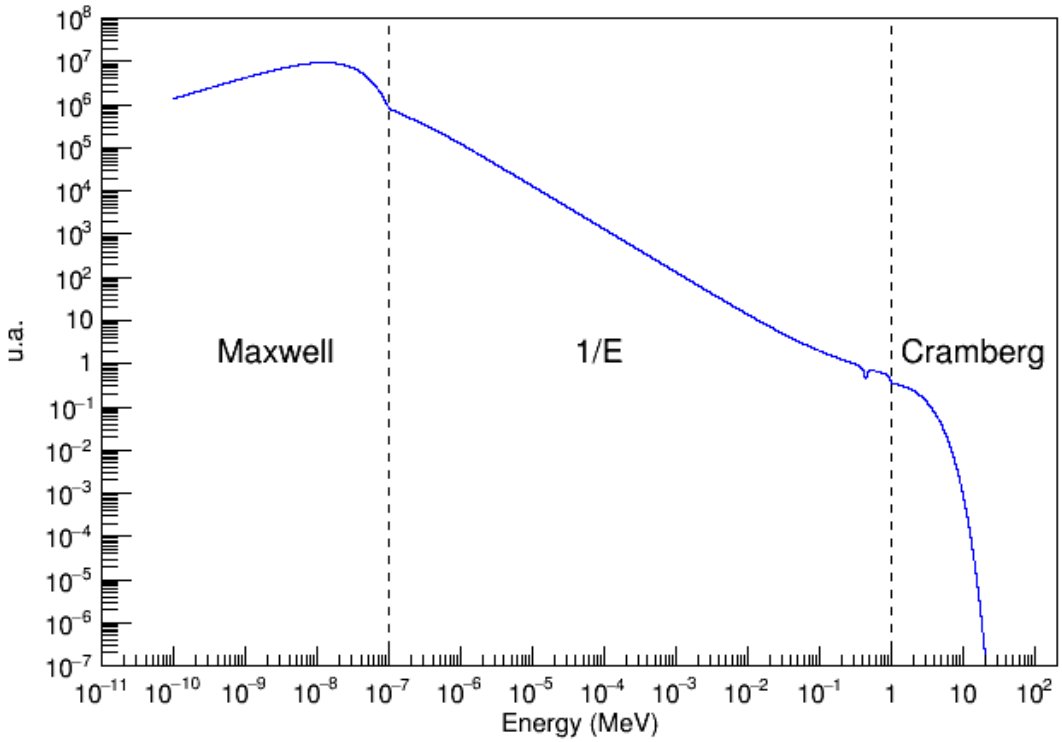


Figure 3.3. Spectrum used to condense multi-group cross sections for INIPOND. A fission spectrum is used over 1 MeV. A simple slowing down behaviour is chosen between 1E−7 MeV and 1 MeV. A Maxwell spectrum is chosen for thermal energies lower than 1E−7 MeV.

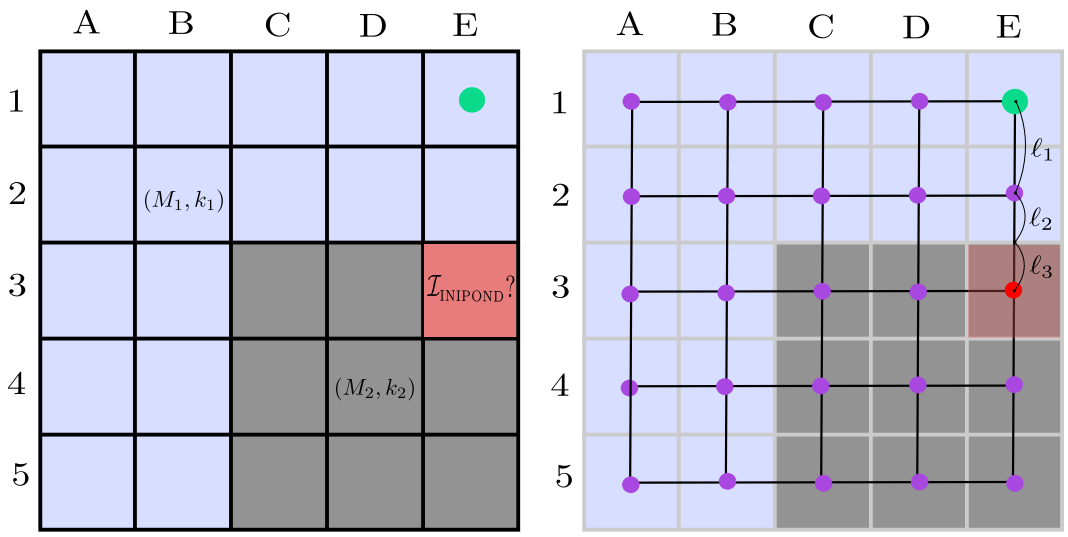
Now that we know how the energy is treated in INIPOND, we can express the energetic component of the importance as a function of the parameter β corresponding to the energy group to which E belongs:

$$\mathcal{I}_{\text{energetic}}(E) = \mathbb{1}_{E_{i+1} \leq E \leq E_i} \times \left(\frac{1}{\beta + 1} \frac{E_i^{\beta+1} - E_{i+1}^{\beta+1}}{E_i - E_{i+1}} \right). \tag{3.10}$$

The spatial component

For each energy group, a spatial mesh is defined. INIPOND estimates the importance in each of the cells of this mesh by viewing the importance mesh as a *graph*. The centers of the cells become the nodes of the graph. The graph is undirected and edges are are drawn between neighbouring cells. The user may choose to draw edges between first neighbours, second neighbours, etc. For example, in Figure 3.4b, the user chose to draw edges between first neighbours only and there are only 4 neighbours considered for each node. Then, a *root node* is defined as the node of the cell containing the detector (in green in Figure 3.4a)

INIPOND works in two steps. The first step consists in computing all the distances ℓ_i from one node to its neighbours. The second step, is the heart of the algorithm. It finds the shortest optical path from one node to the root node considering the optical length



(a) Mesh of the importance map

(b) Graph used to compute the importance map

Figure 3.4. Illustration of the spatial component of the INIPOND module. The spatial component in the $(E, 3)$ cell of the importance in the sense of INIPOND is $\mathcal{I}_{\text{INIPOND}} = e^{-k_1(\ell_1+\ell_2)+k_2\ell_3}$

on the graph.

$$\mathcal{I}_{\text{spatial}} = w_{\text{detector}} \times \exp\left(-\sum_{i \in \text{shortest path}} k_g \times \ell\right) \tag{3.11}$$

At the root node (the cell in which the detector is present), the spatial component of the importance is thus w_{detector}. In the cell $(E, 3)$ the spatial component of the importance is:

$$\mathcal{I}_{\text{spatial}}(E, 3) = w_{\text{detector}} \times e^{-k_1(\ell_1+\ell_2)+k_2\ell_3} \tag{3.12}$$

In other cells, the values of k_1 and k_2 determine the shortest optical path and thus the value of the spatial component takes more complicated expressions.

The angular component

Finally, an angular information is added to the importance.

$$\mathcal{I}_{\text{angular}}(\mathbf{r}, E, \mathbf{\Omega}) = \frac{\Sigma_t}{\Sigma_t - k\mathbf{\Omega} \cdot \mathbf{\Omega_0}} \tag{3.13}$$

Here $\mathbf{\Omega_0}$ is the privileged direction of interest at point $\mathbf{r}$, i.e. the unit vector that points in the direction of the gradient of the importance map (computed by finite differences). The influence of this component is easy to understand as long as one considers the two conditions on the scalar product of $\mathbf{\Omega}$ and $\mathbf{\Omega_0}$.

- Consider the case where $\mathbf{\Omega} \cdot \mathbf{\Omega_0} \geq 0$, then, $\mathcal{I}_{\text{angular}}(\mathbf{r}, E, \mathbf{\Omega}) \geq 1$.

- Consider now the case where $\mathbf{\Omega} \cdot \mathbf{\Omega_0} < 0$, then, $\mathcal{I}_{\text{angular}}(\mathbf{r}, E, \mathbf{\Omega}) < 1$.

The more the particle is emitted in the *good* direction $\mathbf{\Omega_0}$, the more the angular importance will be increased. Contrarily, it will be decreased if the particle goes in the opposite direction.

Extension to multiple detectors, or volumes

If the detector is not a point, or a single cell of a mesh, but rather an extended volume, then constructs the importance map by summing the importance maps computed as previously described placing a detector in each of the cells that intersect the target volume.

Additional details about INIPOND can be found in the appendix of Louvin (2017).

3.3.6 What is done outside the nuclear sphere?

Although Monte Carlo simulations have their origin in nuclear applications, the method has now largely spread across other communities. It is interesting to consider how rare events are simulated in other domains. And it is indeed interesting to see that *automatic* Monte Carlo schemes have emerged to find the best importance map. These methods are named *adaptive importance sampling*. A review of them can be found in Juneja and Shahabuddin (2006). The subject of rare event simulation is of course accompanied with discussions on zero variance schemes. Tuffin and L'Ecuyer (2008) particularly focus on tilted, or approximate zero-variance schemes. Their intuition is that an importance map that is *close* to the optimal importance map still yields excellent accelerations compared to arbitrary choices. They discuss on how "close" should be defined.

3.3.7 Adaptive Multilevel Splitting

Adaptive Multilevel Splitting is a relatively young algorithm compared to the others. It is certain that AMS can not result in a zero-variance scheme. Plus, at the moment, there is no characterisation of the optimal importance map. Until now, the importance functions proposed in shielding calculations were taking the form of simple geometrical functions (Louvin, Dumonteil, and Lelièvre, 2017). In his work, H. Louvin studied the use of an importance map generated by TRIPOLI-4®'s INIPOND module. It was observed that this combination yields acceptable results without manually tuning the parameters of INIPOND. An open question: how well does AMS perform when the importance map is taken to be the adjoint flux?

3.4 Conclusion

In this chapter, we discussed the existence of a zero variance scheme for Monte Carlo shielding simulations. We explained how it has motivated research around the effective approximation of the adjoint flux. We reviewed the main strategies taken by state of the art Monte Carlo codes. A part of these strategies relies on the Monte Carlo estimation of the importance, whereas another part relies on the use of deterministic modules (that solve for the adjoint flux (CADIS) or some approximation (INIPOND)).

By doing so, we concluded that the importance map is at the center of variance reduction studies and the first concern of a Monte Carlo code user. However, the actual implementations have limitations:

1. Splitting/roulette based on the Weight Windows is difficult to tune when no a priori knowledge of the system is available. For some problems, it requires an expertise in order to iterate over different simulations (on the density for example)

2. This makes it hard to develop a strategy that continuously learns from the simulation.

3. Finally, without the guarantee to reach the detector, it is hard for the simulation to start.

The aim of this thesis is to overcome a part of these limitations by learning from trajectories sampled by Adaptive Multilevel Splitting. AMS has the main advantage to be able to sample trajectories that contribute to the detector response, even with a crude approximation of the importance map. The idea is to *continuously* learn from the simulation.

The first objective is thus to retrieve an estimation of the adjoint flux, because it is the optimal importance map for ET. In order to confidently develop this first score, we first develop tools that will allow to easily generate adjoint fluxes thanks to deterministic methods.

Chapter 4

Deterministic solutions from IDT to TRIPOLI-4®

Contents

4.1 The multi-group formalism **50**
4.1.1 Direct 50
4.1.2 Adjoint 51
4.2 Coupling **52**
4.2.1 General 52
4.2.2 Mesh & geometry 52
4.2.3 Cross sections 53
4.2.4 Adjoint source 53
4.2.5 IDT solver parameters 53
4.3 Verification **54**
4.4 Results **55**
4.4.1 Strong attenuation 55
4.4.2 Streaming : the bunker 62
4.4.3 Nucifer at Osiris 63
4.5 Conclusion **65**

This chapter has two goals. The first one is to describe the implementation of a coupling between a deterministic code and TRIPOLI-4® with the aim of automatically generating solutions to the adjoint Boltzmann equation. This will later serve as a reference for Monte Carlo estimators of the adjoint flux. The second objective is to see if the solution to the adjoint Boltzmann equation computed by a deterministic solver can effectively accelerate the two variance reduction methods implemented in TRIPOLI-4® (i.e. ET and AMS). This chapter starts with a reminder of the multi-group formalism Section 4.1. The second Section 4.2 presents this coupling and verifies its implementation on a simple test case. Finally, more realistic configurations are studied to assess the efficiency of ET and AMS with deterministic adjoint fluxes in Section 4.4.

4.1 The multi-group formalism

4.1.1 Direct

Recall the integro-differential formulation of the direct Boltzmann equation (1.6) from Chapter 1, which we rewrite here for convenience:

$$\boldsymbol{\Omega} \cdot \nabla \Phi(\mathbf{r}, E, \boldsymbol{\Omega}) + \Sigma_t \Phi(\mathbf{r}, E, \boldsymbol{\Omega}) = \int \Sigma_s(\mathbf{r}, E', \boldsymbol{\Omega}' \to \mathbf{r}, E, \boldsymbol{\Omega}) \Phi(\mathbf{r}, E', \boldsymbol{\Omega}') d\boldsymbol{\Omega}' dE' + S(\mathbf{r}, E, \boldsymbol{\Omega}) \quad (4.1)$$

IDT (Zmijarevic and Sanchez, 2001) is a deterministic code that solves this direct transport equation with a multi-group formalism. It is part of the deterministic suite of SERMA(APOLLO3® (Schneider et al., 2016)).

In the multigroup formalism, the energy domain is split into G groups

$$\{E_g, E_{g-1}\}_{g \in [1,G]}.$$

The multigroup flux in group g is defined by Φ_g.

$$\Phi_g(\mathbf{r}, \boldsymbol{\Omega}) = \int_{\Delta u_g} \Phi(\mathbf{r}, u, \boldsymbol{\Omega}) du \quad (4.2)$$

Where $u = \ln\left(\frac{E_0}{E}\right)$ is the lethargy given for an arbitrary energy E_0, and $\Delta u_g = \ln(E_g/E_{g-1})$. Total cross sections are constant in each energy group and must thus respect:

$$\Sigma_t^g = \frac{\int_{\Delta u_g} \Sigma_t(u) \Phi(u) du}{\int_{\Delta u_g} \Phi(u) du} \quad (4.3)$$

Similarly, the scattering cross section from group g' to g is:

$$\Sigma_s^{g\leftarrow g'} = \frac{\int\limits_{\Delta u_g} du \int\limits_{\Delta u_{g'}} du' \Phi(u')\Sigma_s(u \leftarrow u')}{\int\limits_{\Delta u_{g'}} \Phi(u')du'} \tag{4.4}$$

The integro-differential formulation of the direct multi-group transport equation reads :

$$\mathbf{\Omega} \cdot \nabla \Phi_g(\mathbf{r}, \mathbf{\Omega}) + \Sigma_t^g(\mathbf{r})\Phi_g(\mathbf{r}, \mathbf{\Omega}) = \sum_{g'=1}^{G} \Sigma_s^{g\leftarrow g'} \Phi_{g'}(\mathbf{r}, \mathbf{\Omega}) + S_g(\mathbf{r}, \mathbf{\Omega}) \tag{4.5}$$

The direct multigroup source is:

$$S_g = \frac{1}{\Delta E_g} \int S(E)dE \tag{4.6}$$

4.1.2 Adjoint

The advantage of deterministic codes is that very few modifications need to be made to the direct solver for it to treat the adjoint equation. Recall the adjoint integro-differential equation from Chapter 1:

$$\begin{aligned} -\mathbf{\Omega} \cdot \nabla \Phi^\dagger(\mathbf{r}, E, \mathbf{\Omega}) + \Sigma_t \Phi^\dagger(\mathbf{r}, E, \mathbf{\Omega}) = \\ \int \Sigma_s(\mathbf{r}, E, \mathbf{\Omega} \rightarrow \mathbf{r}, E', \mathbf{\Omega}')\Phi^\dagger(\mathbf{r}, E', \mathbf{\Omega}')d\mathbf{\Omega}'dE' + S^\dagger(\mathbf{r}, E, \mathbf{\Omega}) \end{aligned} \tag{4.7}$$

The multi-group adjoint transport equation reads :

$$-\mathbf{\Omega} \cdot \nabla \Phi_g^\dagger(\mathbf{r}, \mathbf{\Omega}) + \Sigma_t^g(\mathbf{r})\Phi_g^\dagger(\mathbf{r}, \mathbf{\Omega}) = \sum_{g'=1}^{G} \Sigma_s^{g'\leftarrow g} \Phi_{g'}(\mathbf{r}, \mathbf{\Omega}) + S_g^\dagger(\mathbf{r}, \mathbf{\Omega}) \tag{4.8}$$

Where the multi-group adjoint flux $\Phi^\dagger(\mathbf{r}, \mathbf{\Omega})$ is defined as:

$$\Phi_g^\dagger(\mathbf{r}, \mathbf{\Omega}) = \frac{1}{\Delta u_g} \int_{\Delta u_g} \Phi^\dagger(\mathbf{r}, E, \mathbf{\Omega})dE, \tag{4.9}$$

and the multi-group adjoint source as:

$$S_g^\dagger(\mathbf{r}) = \int_{4\pi} \int_{E_g}^{E_{g-1}} S^\dagger(\mathbf{r}, E, \mathbf{\Omega})\, dE\, d^2\mathbf{\Omega} \tag{4.10}$$

IDT solves this equation on a discretised phase space. The spatial component is discretized over a 3D cartesian mesh and solved with the nodal method, finite differences, or short characteristics. Several methods can be used to accelerate the convergence under specific conditions (Zmijarevic and Sanchez, 2001).

4.2 Coupling

A previous attempt to use adjoint fluxes in TRIPOLI-4® was done (Giffard, 2000) and particular cases were studied. Our approach is strictly independent and more systematic. In this section, we describe the methodology with which we coupled IDT to TRIPOLI-4®. In short, IDT solves a discrete version of the (adjoint) Boltzmann equation. A vast body of literature is dedicated to the assessment and the continuous improvement of the solution methods employed in codes such as IDT. These themes are not covered at all in this thesis - we shall essentially use IDT as a black box for solving the Boltzmann equation(s) without paying much attention to its user-adjustable parameters (except for the most important). Of course a phase of thorough study and optimization will in general be required, along with a gain of experience from the user.

4.2.1 General

For the sake of simplicity, the directive that triggers the computation of the adjoint flux is set within the TRIPOLI-4® input file. Similarly to MAVRIC (Peplow, 2011), an IDT input file is automatically generated and the solver is called immediately afterwards. The resulting adjoint flux is sent back to TRIPOLI-4® in memory in order to avoid shuffling large files to and from the disk. The user is responsible for a few configuration choices (discretisation, etc. . .) which are detailed in the following sections.

4.2.2 Mesh & geometry

The importance map is computed on a user-defined cartesian mesh that can be either regular or variable. Each cell is assumed to be homogeneously filled with the material found at its center. The same strategy was chosen by Sweezy et al. (2005). As a comparison, we review the other strategies employed by a few common couplings between deterministic and MC codes. ADVANTG performs ray tracing on the geometry to estimate the ratio of materials present in each cell (Mosher et al., 2013). Another way to go is the solution chosen by MCBEND (Cowan et al., 2009) in which a mesh is automatically generated by analysing the diffusion lengths. It can be tuned or improved by adding additional information, such as the “principal attenuation directions”, or excluding empty parts from the mesh.

It is the user’s responsibility to ensure that the chosen mesh coarseness is suitable for

the problem description. Note however that IDT internally uses a finer calculation mesh.

4.2.3 Cross sections

Each material needs to be described with a set of group-wise cross sections. This condensation is performed in TRIPOLI-4® by weighting the point-wise cross sections by an arbitrary spectrum representative of a PWR reactor. The same condensation module is used for INIPOND. The total, the scattering and the (n,2n) cross sections are condensed. This is done according to equations (4.3) and (4.4). Note that *upscattering* is not modelled. The anisotropy if taken into account with a transport correction (theory P1) for both neutrons and photons. As results will show, this condensation probably represents the main flaw of the method. Most of other coupling implementations use deterministic cross section libraries. For example Sweezy et al. (2005) uses cross sections on the SAILOR deterministic energy grid, while Peplow (2011) uses a default anisotropy order of 3. The coupling is in theory capable of handling pure photon problems, although in this case the quality of the resulting importance maps may be limited by the rough treatment of the cross-section anisotropy.

4.2.4 Adjoint source

In an adjoint calculation, the source $S^\dagger$ of the problem is the kernel of the response R in the detector

$$\begin{aligned} R_{\mathcal{D}} &= \iint \eta(\mathbf{r}, E)\Phi(\mathbf{r}, E)d^3\mathbf{r}dE \\ &= \iint S^\dagger(\mathbf{r}, E)\Phi(\mathbf{r}, E)d^3\mathbf{r}dE \end{aligned}$$

Since we want to represent R as a convolution over the multi-group adjoint flux.

$$\sum_{g=1}^{G} \int S_g^\dagger(\mathbf{r}, E)\Phi_g(\mathbf{r}, E)d^3\mathbf{r} \tag{4.11}$$

We choose to build this adjoint source automatically from the score of interest. The user is asked to assign the keyword TARGET to the score. TRIPOLI-4® can then build the adjoint source automatically.

4.2.5 IDT solver parameters

Now that we have described the condensation and homogenization of TRIPOLI-4®'s point-wise cross sections, we need to fix the user-adjustable parameters specific to the deterministic solver. Among them, we focus on the following :

Angular quadrature

A common issue arises when one tries to solve propagation problems with deterministic solvers that rely on the S_N angular discretisation method. Ray effects appear when the adjoint source is very localized and when the materials scatter weakly (Lathrop, 1968). It materialises in a "star" shape of the flux around point-wise sources.

Sweezy et al. (2005) and Čufar et al. (2017) reported poor convergence due to ray effects in using the CADIS methodology. In these studies, the weight window generator of MCNP performed better than the deterministic coupling. Customarily, ray effects are smoothed with the two following methods :

(a) increase the number of directions in the angular quadrature.

(b) implement the first collision method. It consists in decomposing the flux into two components, the uncollided flux and the multiple collided one. The first contribution can be computed with minimal effort in each cell of the calculation mesh, thus providing a well distributed initial source for the calculation.

We chose the first brute-force method. The Chebyshev-Legendre quadrature, which is not limited in the number of directions, was thus selected for the coupling. But the second method was implemented in IDT (Zmijarevic and Sciannandrone, 2017) and its integration is forthcoming in this coupling. The default quadrature order is set to 8, similarly to what is done in MAVRIC (Peplow, 2011).

Calculation mesh

The result mesh used for the importance map is not the same as the computation mesh. The computation mesh is usually finer than the result one, so as to guarantee appropriate convergence of the adjoint flux. Its maximum size can either be manually set by the user, or it can be automatically computed by IDT by analysing the shortest optical distance in materials. As it is mostly a user-adjustable parameter, much work has already been carried out to automate the mesh construction with the CADIS method. Complex geometries usually benefit from an automated method as proven by the study on ITER (Mosher et al., 2013).

Finally, we can define reflection or leakage boundary conditions. This was not possible with INIPOND which imposed leakage boundary conditions.

4.3 Verification

Now that the coupling is in place, we would like to check if the discretisation is consistent with the problem described in the Monte Carlo simulation. For that, we introduce the first simple benchmark. Note that all benchmarks will be described in [Appendix A] and illustrated in Figure 4.1. The benchmark we introduce here is a $2D$ problem composed of iron and water [Appendix A.3]. We first verify the coupling for a simple direct flux calculation. Two calculations were performed, one with AMS of TRIPOLI-4®, and one with TRIPOLI-4® driven by the IDT coupling. The direct fluxes discretised over 5 energy groups are plotted in Figure 4.2. Except where the direct flux is poorly converged in the Monte Carlo simulation, the discrepancies between the deterministic and the Monte Carlo solution are contained within a few standard deviations.

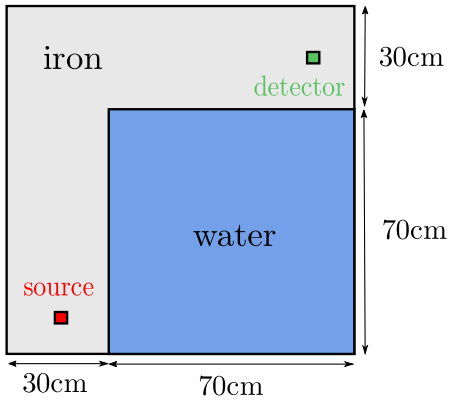


Figure 4.1. Water-iron benchmark geometry..

4.4 Results

In order to illustrate the efficiency of our CADIS method, we focus our attention on three problems; each of them represents a class of difficulty encountered in shielding calculations. The first case is a slab geometry which will help demonstrate the difficulty to obtain an acceleration on the whole range of energy spectrum. The second case will show a streaming problem with more than one path to the detector. Lastly, we will finish on a more realistic strong-attenuation problem that consists in quantifying the fast flux in the Nucifer detector resulting from the neutrons born in the Osiris reactor.

For each problem, the AMS and the ET will be tested with different importance maps.

4.4.1 Strong attenuation

Description

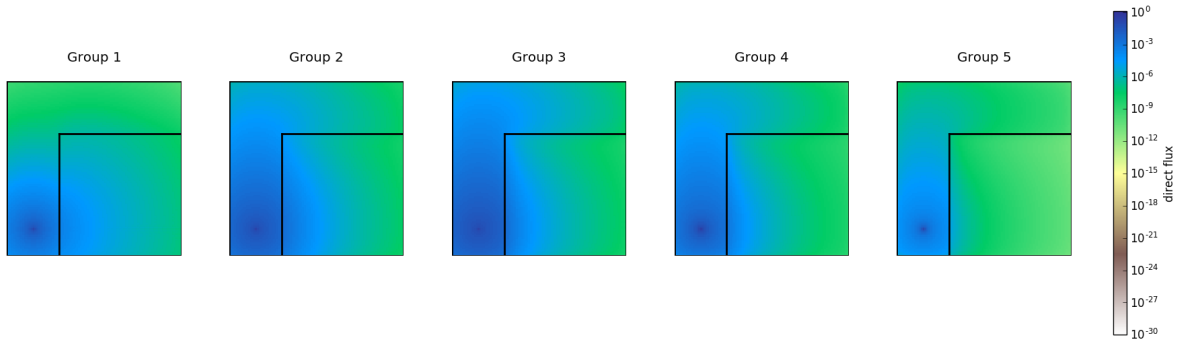
The first problem studied consists of a mono-directional neutron source placed at the beginning of a 3 m-deep water tank [Appendix A.5]. Its energy is sampled from a Watt distribution. The detector is a 20 cm-long zone at the end of a concrete wall of 1.2 m and we are interested in the total flux. Reflection boundary conditions are imposed on all surfaces except on the one behind the detector. See geometry in Figure 4.3.

Importance maps

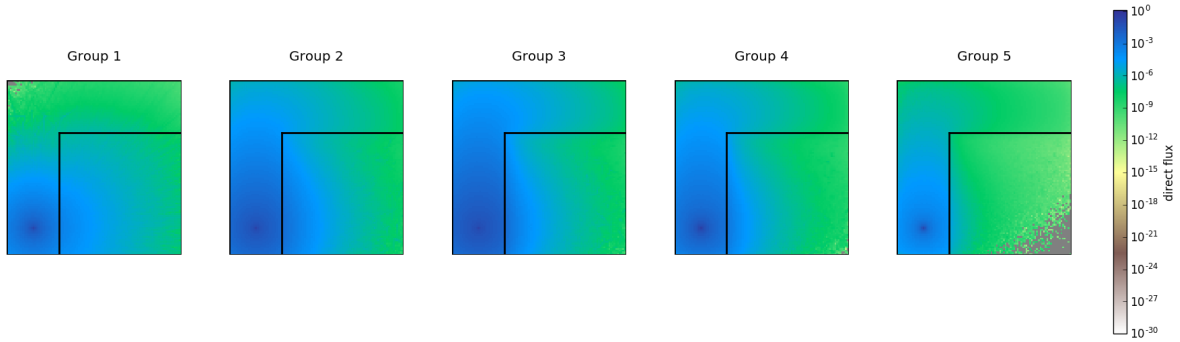
For this configuration, we explored four importance maps.

1. A first importance map was generated with the automatic mode of the INIPOND module with 6 energy groups (there are several reasons in the generation of the INIPOND map that constrain us to choose very few energy groups (Brun et al., 2015)). Using this importance map, the ET is unable to yield any result in the concrete part of the geometry within a reasonable time.

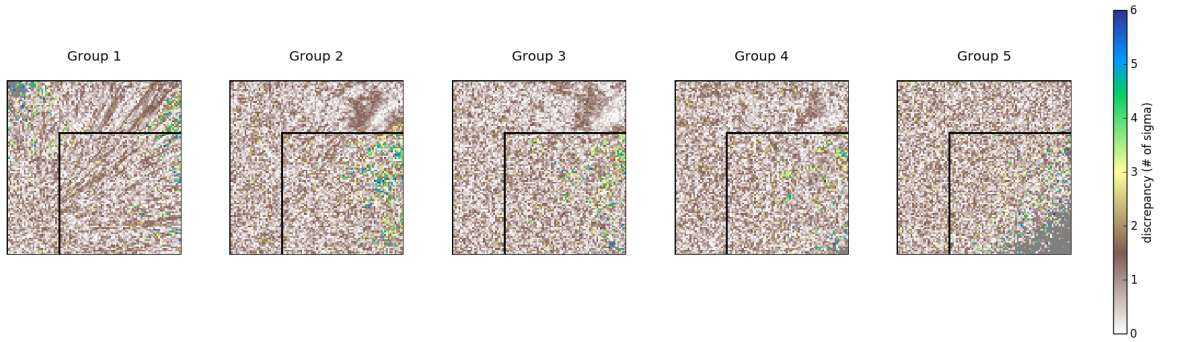
2. The second importance map is also generated by INIPOND with the same energy groups but with *manual adjustments* of its parameters. This time, ET manages to sample trajectories that reach the detector.



(a) IDT



(b) TRIPOLI-4®



(c) Discrepancies

Figure 4.2. Direct flux comparison for the water-iron benchmark. The first row represents the flux computed by IDT, the second one with TRIPOLI-4®. The last figure represents the discrepancy between IDT and TRIPOLI-4® as a ratio between their difference and the standard deviation of the adjoint score with TRIPOLI-4®. The energy groups correspond to those present in [Appendix A.3]

3. The third importance map is produced by the IDT- TRIPOLI-4®coupling. IDT can easily be run over a larger number of groups, so we chose 57 energy groups.

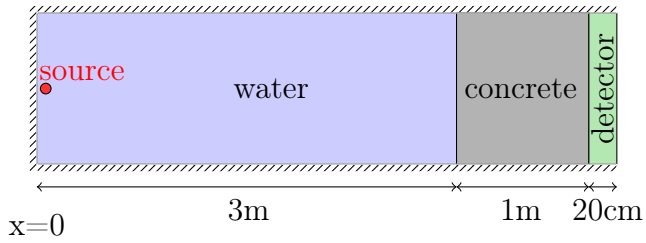


Figure 4.3. Geometry for the strong attenuation benchmark. A mono-directional neutron source, sampled on a Watt spectrum is placed in a 3 m-long tank of water. The detector is a 20 cm-wide part of a 120 cm-long concrete wall. Reflection conditions are imposed on all surfaces except for the outer surface of the detector, from which neutrons can leak out.

The cross sections used for these calculations are produced by the TRIPOLI-4® condensation module. They are isotropic and without up-scattering.

4. The fourth importance map is also produced by IDT, but with different cross sections. In order to exhibit the sensitivity to the quality of the macroscopic cross section condensation, we manually (with a prior direct deterministic calculation) homogenize cross sections for the two materials present in this problem (namely water and concrete). Cross sections are condensed over the same 57 energy groups, with an anisotropy order of 5, and with upscattering with APOLLO3®.

Results

AMS and ET are run with these importance maps. The results of these simulations are gathered in Table 4.1.

AMS converges 40% faster with IDT - with cross sections from TRIPOLI-4® - than with INIPOND - with manual adjustments - and twice as fast with IDT - with cross sections from APOLLO3® - compared to INIPOND (with manual adjustments).

Concerning the exponential transform, we note that the simulation run with ET and the importance map produced by IDT (with cross sections from TRIPOLI-4®) results in a smaller mean integrated response than all the other calculations; the differences with the simulations: ET with INIPOND (with manual adjustments) and AMS (with IDT and cross sections from APOLLO3®) are so large that it can not be attributed to statistics alone. This discrepancy will be studied shortly. Improving the quality of the cross sections by producing them with APOLLO3® fixes the problem and adds up a significant factor of 1808 to the FOM compared to the simulation ET with INIPOND (manually adjusted).

Our explanation for the discrepancy of the simulation: ET and IDT (with cross sections condensed by TRIPOLI-4®) is that it originates from the undersampling of thermal groups. In order to verify this statement, we score the flux over a 100 equi-lethargic energy grid. Figure 4.4 shows the ratio of the flux for all cases over the reference solution - taken as the most converged solution, namely ET with IDT and cross sections condensed by APOLLO3® as a function of the neutron energy. The confidence intervals for the non problematic calculations are shaded to better highlight the reference solution and the

Importance map (properties)	**Mean**($n/cm^3/s$)	**error**(%)	**time**(s)	**FOM** (arbitrary units)
Adaptive Multilevel Splitting				
INIPOND (Manual mode)	2.58×10^{-15}	9.90	1.67×10^5	9
IDT (cross sections from TRIPOLI4)	2.61×10^{-15}	9.83	1.20×10^5	13
IDT (cross sections from APOLLO3)	2.78×10^{-15}	7.11	1.59×10^5	16
Exponential Transform				
INIPOND (Manual mode)	2.55×10^{-15}	6.51	9.41×10^4	39
IDT (cross sections from TRIPOLI4)	2.04×10^{-15}	6.60	2.39×10^5	23
IDT (cross sections from APOLLO3)	2.81×10^{-15}	0.82	3.27×10^3	576

Table 4.1. Results for the strong attenuation problem. We present the integrated mean response in the detector, as well as the estimate of its standard error after a certain simulation time. The figures of merit can be compared for both the AMS and the ET method used with the three importance maps : INIPOND in manual mode, IDT with cross-sections from INIPOND and IDT with cross sections from APOLLO3 with anisotropy order 5.

problematic simulation ET with IDT and cross sections from TRIPOLI-4®. All confidence intervals overlap with the reference solution, except for ET / IDT (cross sections from TRIPOLI-4®), that clearly underestimates the flux in the thermal groups. The spectrum results are consistent from 20 MeV down to 1 eV, but start diverging from the other calculations below this energy.

In order to understand the origin of these discrepancies in the target, the spectrum was plotted along the slab from the source to the detector in intermediate 10 cm-wide cells. Figure 4.5 shows the spectrum at three different depths along the main axis : 50 cm, 230 cm and 410 cm. In intermediate volumes, all the AMS calculations show regular shapes across the complete energy range, while the ET calculations yield zero flux in epithermal and thermal groups. This is not surprising because the ET optimizes transport for the target score, i.e. the total flux in the detector in this case. It means that mainly fast neutrons were transported until they have reached the concrete wall and they start to thermalize at this point. Thermal and epi-thermal neutrons are immediately rouletted before they reach the concrete. This causes a direct under-estimation - if not any estimation - of the spectrum in intermediate volumes. From this point on, the importance map favours splitting of low energy neutrons and thus gradually pushes the fast neutrons towards the thermal groups.

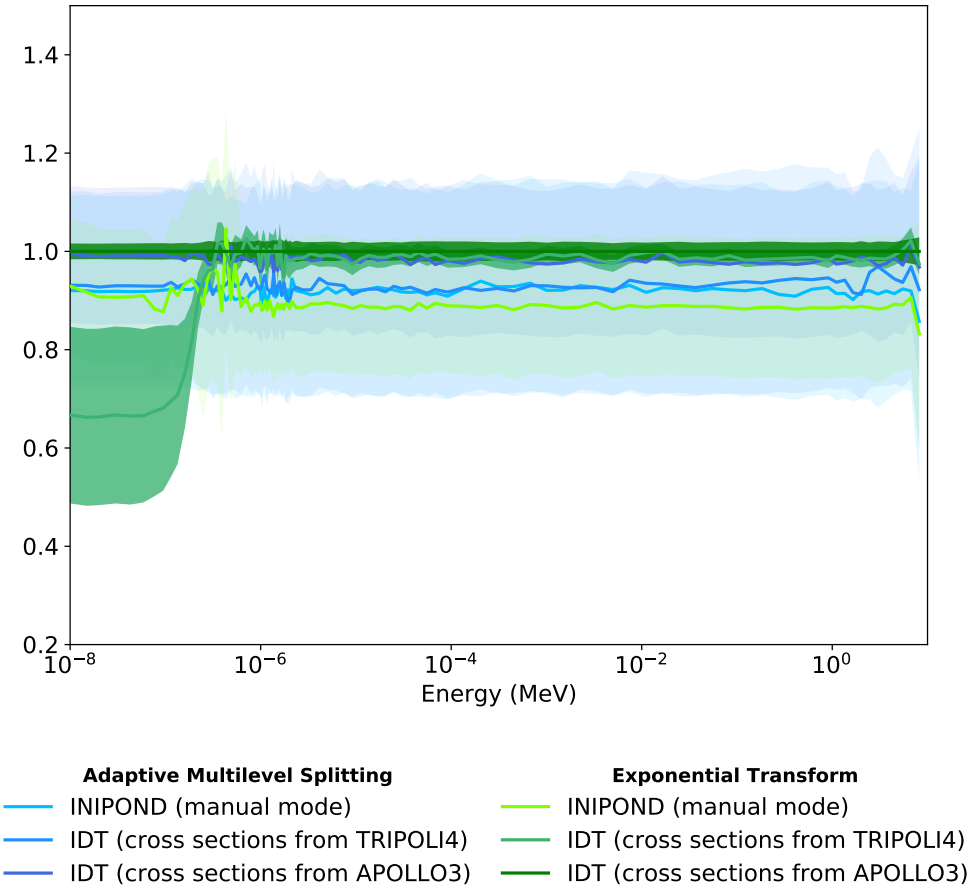


Figure 4.4. Ratio of mean response for each case to the reference ET with the importance map computed by IDT with cross sections produced by APOLLO3. The shaded areas represent the confidence intervals at two standard deviations. They are symmetric (distortions due to the log scale). All calculations lie within two standard deviations from the reference solution, except for the simulation ET with the importance map computed by IDT with cross sections condensed by TRIPOLI-4®, which is clearly inconsistent below 1 eV.

So why does ET (with IDT and anisotopic cross sections) converge perfectly and ET (with IDT and isotropic cross sections) does not? For the sake of conciseness, we will only give the three reasons that we believe are crucial.

(a) it has been observed that TRIPOLI-4® makes some crude approximations in the condensation of the scattering matrix, which leads to underestimated scattering cross sections, and results in a more absorbing media. In the case of strong attenuation problems, this means that the adjoint flux gradients are largely over-estimated. This should mislead the splitting/roulette step and kill thermal histories too soon.

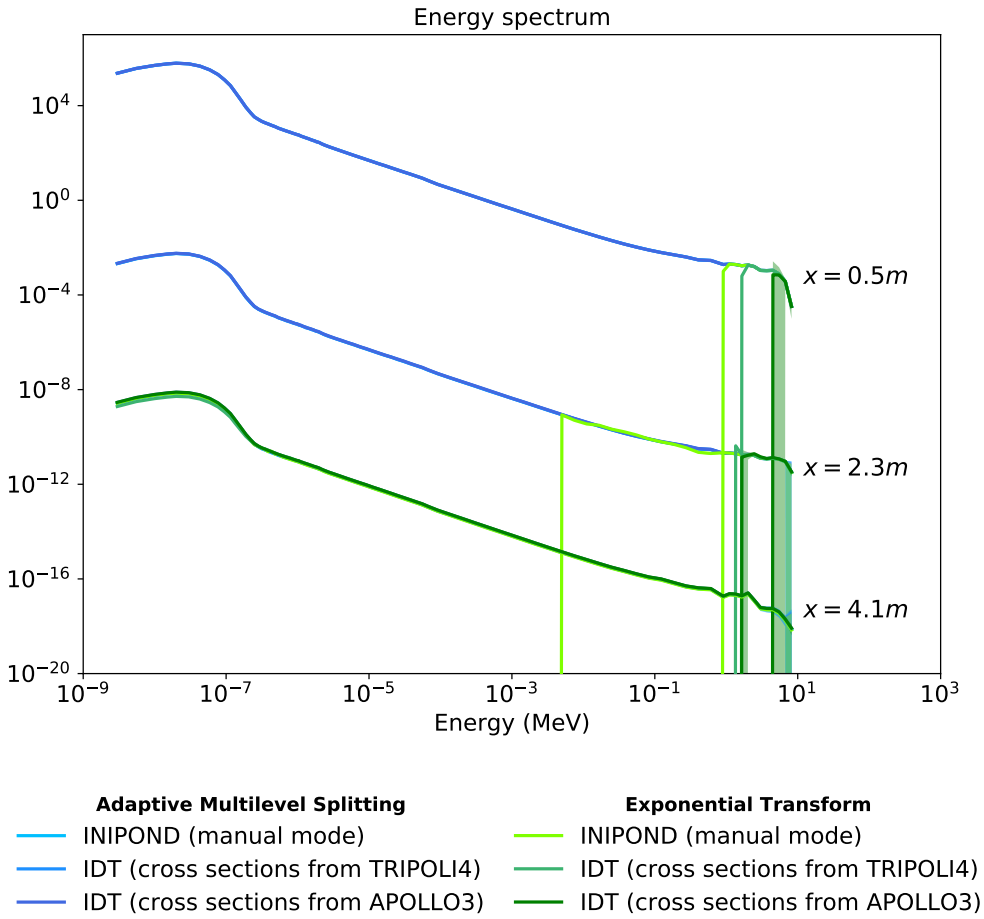


Figure 4.5. Spectrum for the strong attenuation problem. The flux spectrum is plotted against the energy at three different positions: (a) between x=50 cm and x=60 cm from the source, (b) between x=2.3 m and x=2.4 m of water, (c) at the entrance of the detector between x=4.0 m and x=4.20 m, for the 6 tested configurations: AMS (blue) and ET (green), for the three importance maps, INIPOND tunes (DATA), INIPOND with cross sections from T4 (IDT INI), and IDT with cross sections homogenized in a prior direct deterministic simulation (IDT ANI).

(b) TRIPOLI-4® does not take into account upscattering during condensation.

(c) anisotropy plays an important role here, and a comparison between isotropic and anisotropic cross sections showed that this should not be under-rated in the CADIS approach.

Finally, Figure 4.6 shows the FOM obtained for each energy group and simulation. We note that the gain in the FOM is uniformly distributed across the whole range of energy

for all our simulations except for ET with IDT and cross sections from TRIPOLI-4®.

We conclude the following: AMS takes limited advantage of the IDT importance map. Improving the quality of the cross sections does help a little though. This supports the idea that the AMS is robust. On the other hand, the ET is very sensitive to the importance map and yields excellent accelerations with better cross section quality. For such an axial problem, the adjoint flux is essentially characterised by its slope along the axis and a slight under/over estimation of the cross sections used in the deterministic calculation leads to poor importance sampling behaviour.

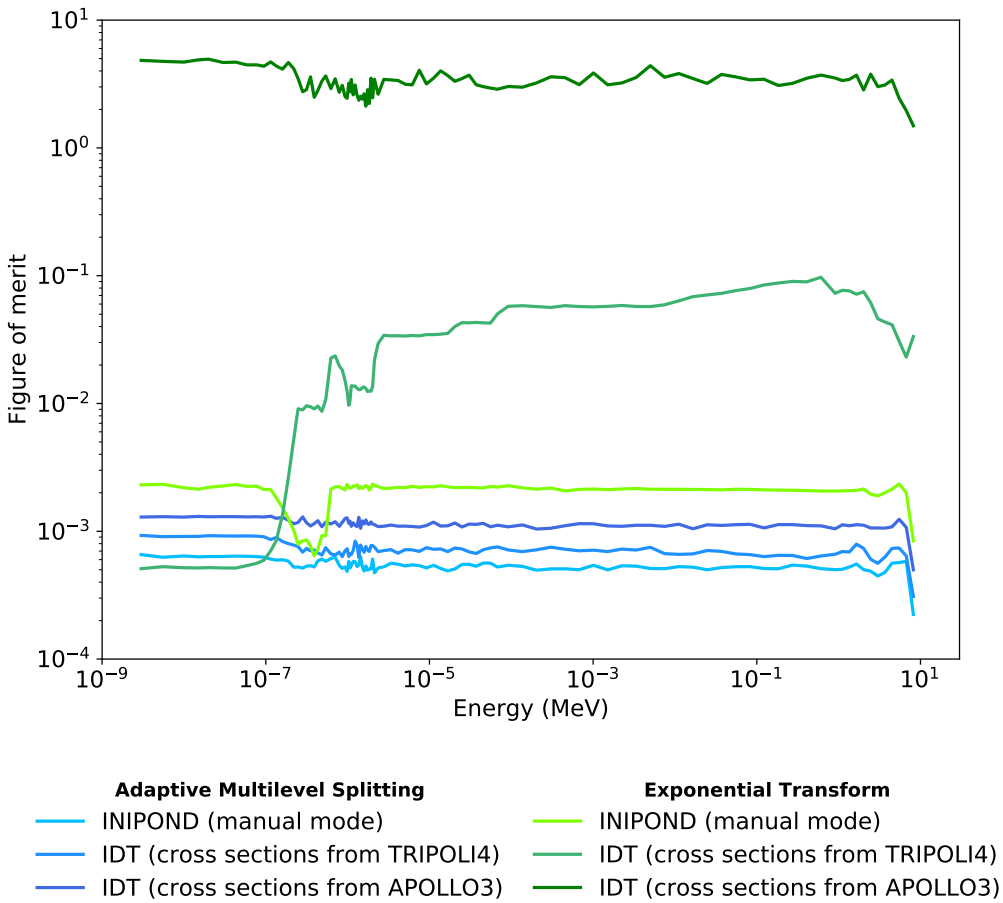


Figure 4.6. Figure of Merit for the strong attenuation problem. FOMs are plotted against the energy for 6 configurations.

4.4.2 Streaming : the bunker

Description

This problem demonstrates the difficulty to account for two privileged paths for particles from the source to the detector. It was proposed by K. W. Burn (1992) and is summarised in [Appendix A.4]. A mono-directional 14 MeV neutron source impinges on a stainless steel sphere. Neutrons are scattered and either go around the wall or penetrate the narrow cylindrical duct in the wall. We are interested in the fast ($\geq$ 9.12 keV) outgoing current through the room exit, in the south-eastern corner. Figure 4.7 shows the geometry of the problem.

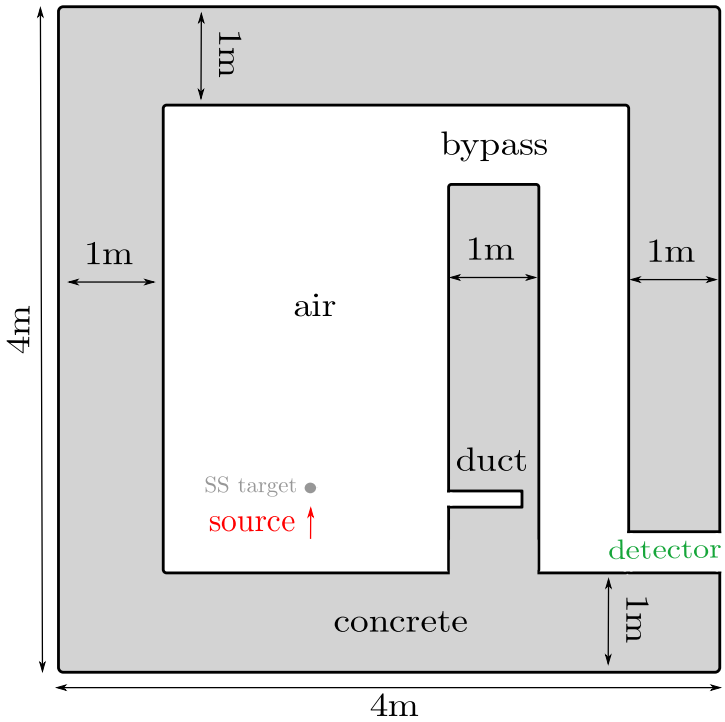


Figure 4.7. Bunker geometry

The importance map

In this problem, the energy is split into the following energetic mesh:

$$1\text{E}{-}11\,\text{MeV} \rightarrow 1\text{E}{-}6\,\text{MeV} \rightarrow 5\text{E}{-}2\,\text{MeV} \rightarrow 5\text{E}{-}1\,\text{MeV} \rightarrow 2\,\text{MeV} \rightarrow 20\,\text{MeV}$$

The automatic mode of INIPOND was run with a mesh with 28x25x25 cells and yielded satisfactory results. However, as many low density cells (filled with air) are present in the geometry, we perform a special *void trick*, which is described in Section 3.3.5. Figure 4.8 shows a slice of the resulting map. The target score is the total current above 9.12 keV.

Alternatively, IDT allowed to significantly refine the mesh - which is now 200x200x100 cells - thus gaining in geometrical precision. Figure 4.9 shows the resulting importance

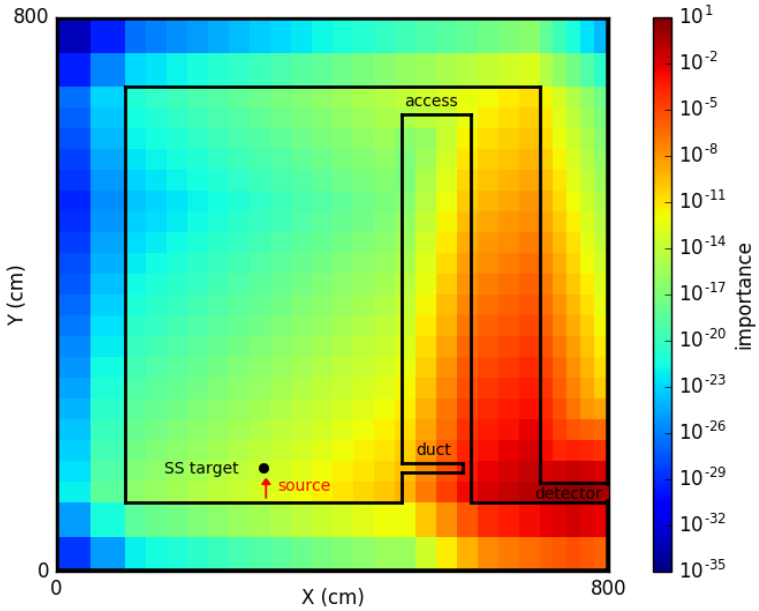


Figure 4.8. Z=0 cm-slice of the importance map generated by INIPOND AUTO for the Bunker problem. The mesh in not regular and contains 28 × 25 × 25 cells

map. Note that although the two maps are qualitatively similar, the gradient in the INIPOND map is stronger(note the different z scale) than the one computed by IDT INI. This should affect the ET more than the AMS.

Results

The problem is treatable without variance reduction methods; however, AMS or the ET yield sizeable improvements in the FOM.

Table 4.2 gathers the results for the current problem. Figures of merit are normalized to the analog result. As expected, AMS is not very sensitive to the importance map from IDT in this situation (only a factor 6 compared to the analog calculation). Unsurprisingly, the ET converges much faster with the IDT map, (factor ×57 compared to the analog) than with INIPOND in automatic mode (factor ×8 compared to the analog).

4.4.3 Nucifer at Osiris

Description

This problem consists in propagating fast neutrons leaking from the Osiris research reactor towards an ex-core neutrino detector, Nucifer. See [Appendix A.8] for full description. Elastic scattering of fast neutrons in the Nucifer sensitive volume is one of the background sources for neutrino detection. The results of this study have already been published (Boireau et al., 2016), but at the time, few details were provided about the

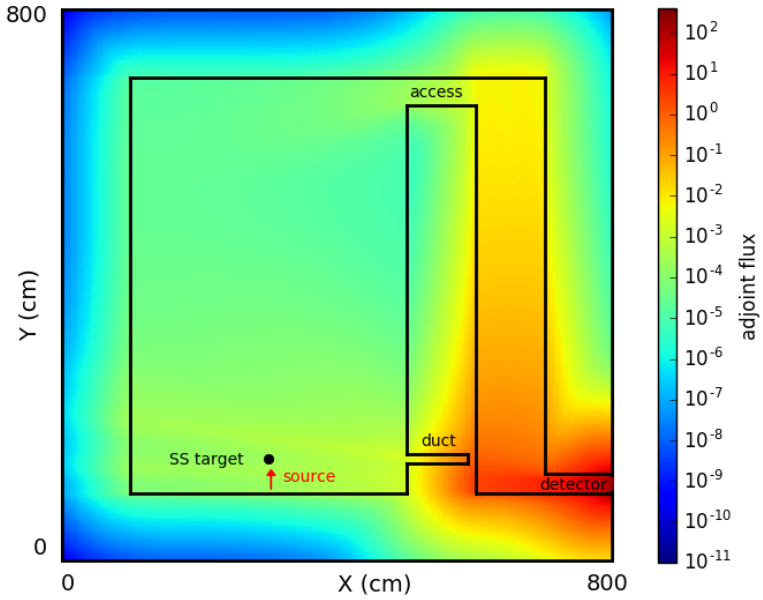


Figure 4.9. Z=0 cm-slice of the importance map generated by IDT for the Bunker problem. The mesh is regular cartesian and consists of $200 \times 200 \times 100$ cells

variance reduction method. The problem is solved in two stages. First, a criticality calculation is performed on a detailed model of Osiris and fast neutrons leaking from the reactor are stored. These neutrons are then used as a fixed source in a second shielding stage in the full Osiris+Nucifer geometry, which includes a model of the reactor pool, of the surrounding walls, as well as a detailed model of the Nucifer detector. The geometry of this problem is shown in Figure 4.10. This problem is similar to the strong attenuation one, except we are only interested in elastic scattering of fast neutrons above 2 MeV.

The importance map

Similarly to the strong attenuation problem 4.4.1, the importance map automatically generated by INIPOND is unable to drive any neutron to the Nucifer detector within a reasonable time. A manual tuning of the INIPOND parameters was made and allowed the ET to converge faster. As in section 4.4.1, only 6 energy groups were used to discretize the energy domain of the importance map, which is acceptable given the small range of energy of interest. As for the Bunker, the problem is truly three dimensional, so IDT was fed with a finely spaced $150 \times 150 \times 150$ uniformly spaced mesh and differs in this way from INIPOND.

In this problem, ray effects were visible for low angular quadrature numbers but quickly disappeared by increasing it to a reasonable number (Chebyshev-Legendre 64). Figures 4.11a and 4.11b show the importance map computed by IDT on a vertical and horizontal cross section respectively, for the highest energy group.

Importance map (parameters)	**Mean**$(n/cm^2/s)$	**error**(%)	**time(s)**	$\frac{\mathbf{FOM}}{\mathbf{FOM}_{Analog}}$
Analog				
-	2.41×10^{-10}	1.84	6.79×10^4	1.00
Adaptive Multilevel Splitting				
INIPOND (automatic mode)	2.43×10^{-10}	1.96	7.77×10^3	7.73
IDT (cross sections from TRIPOLI4)	2.43×10^{-10}	1.85	1.12×10^4	5.98
Exponential Transform				
INIPOND (automatic mode)	2.45×10^{-10}	1.44	1.37×10^4	8.08
IDT (cross sections from TRIPOLI4)	2.44×10^{-10}	1.05	3.63×10^3	57.8

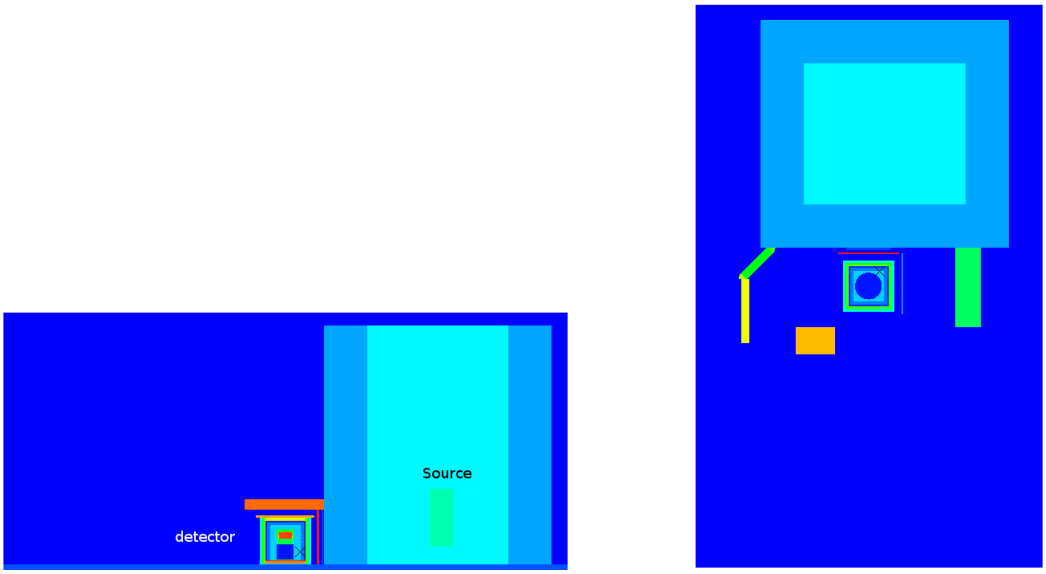
Table 4.2. Results for the Bunker problem. We present the integrated mean of the target response, as well as the estimate of the standard error after a certain simulation time. The figures of merit can be compared for both the AMS and the ET method used with two importance maps : INIPOND in automatic mode and IDT with cross sections from TRIPOLI-4®.

Results

Results are gathered in Table 4.3. One first appreciable thing to notice is that all calculations converge to the same integrated result and are all consistent with each other within the statistical uncertainties. IDT (with cross sections from INIPOND) improves the FOM by a factor of 2 compared to INIPOND for the AMS. On the other hand, the exponential transform is significantly accelerated with the manually tuned INIPOND parameters. These results are similar to those obtained with the strong attenuation problem. Also, the ET converges with the IDT map, (recall that ET did not converge with the automatically generated importance map from INIPOND. Unfortunately, the IDT map does not improve the FOM for this particular problem compared to the manually tuned INIPOND DATA map. As discussed in section 4.4.1, better cross sections, with higher anisotropy description, would certainly improve the importance map quality, and thus accelerate the Monte Carlo calculation.

4.5 Conclusion

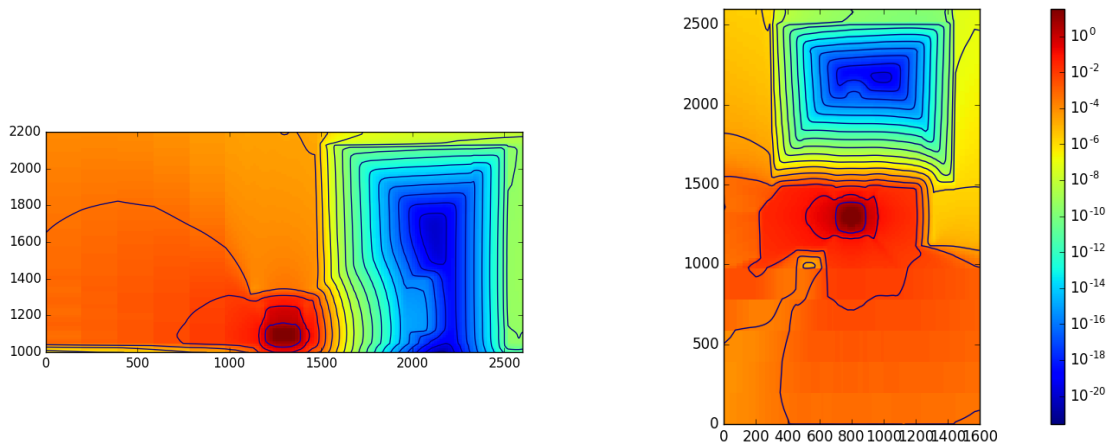
In this chapter, we have described how the deterministic solver IDT has been coupled to TRIPOLI-4®, and we have detailed the ingredients that the user needs to provide to compute an importance map for a given geometry. The most delicate ones are (i) the choice of the spatial and energy mesh, and (ii) condensation of multi-group cross sections for the IDT adjoint calculation. TRIPOLI-4® can condense the point-wise cross sections



(a) Vertical cross section in the middle of the nucifer detector. On the left, the nucifer is placed in a shielding box. On the right, the Osiris reactor.

(b) Horizontal cross section in the middle of the nucifer detector. The detector is placed in a room where a few walls surround it. The room is filled with air.

Figure 4.10. Nucifer detector next to Osiris



(a) Horizontal cross section of the importance map at the middle of the Nucifer detector for the Nucifer-Osiris problem. The color bar is the same as the one used in Figure 4.11b.

(b) Vertical cross section of the importance map at the middle of the Nucifer detector for the Nucifer-Osiris problem

Figure 4.11. Importance map for the benchmark: Nucifer detector next to Osiris computed by IDT

for the user, but clearly the condensation algorithm is too simplistic to produce accurate results in all situations.

In order to illustrate the quality of the importance maps generated by IDT, we stud-

Map	Mean(n/cm^3)	error(%)	time(s)	FOM(a.u.)
Adaptive Multilevel Splitting				
INIPOND (manual mode)	4.01×10^{-9}	26.56	2.79×10^4	3.2
IDT (cross sections from TRIPOLI4)	4.22×10^{-9}	8.75	1.52×10^5	4.8
Exponential Transform				
INIPOND (manual mode)	4.35×10^{-9}	6.70	2.81×10^4	41.7
IDT (cross sections from TRIPOLI4)	4.35×10^{-9}	12.07	1.52×10^5	2.4

Table 4.3. Nucifer integrated results

ied three different problems with the two main variance reduction methods available in TRIPOLI-4®, namely the exponential transform and adaptive multilevel splitting. As a reference, we used importance maps generated by TRIPOLI-4®'s INIPOND module (Brun et al., 2015), either in automatic or in manual mode.

We conclude that the AMS is very robust and is not subject to drastic changes in its behaviour as long as the importance map is not of very poor quality. On the other hand, the ET fails to transport neutrons through thick and attenuating materials when driven by a poor importance map. We also conclude that the gain in the figure of merit can be much larger for the ET than for AMS whenever the importance map is of sufficient quality.

From the point of view of the deterministic solver, we note a few improvements that will be developed in the future. Better cross sections from state of the art self-shielding modules could improve the energetic description of the problem. The first collision method will also be of significant help in reducing the computational time and memory used by IDT for large and complex problems.

Chapter 5

Implementation of a forward-weighted adjoint Monte Carlo estimator

Contents

5.1 Illustration **70**

5.2 Derivation of the importance **70**

5.2.1 Physical interpretation 72

5.3 The analog adjoint estimator **73**

5.3.1 The variance 75

5.3.2 Implementation of the score in TRIPOLI-4® 75

5.4 Scoring the adjoint flux with importance sampling **78**

5.4.1 The estimator 78

5.4.2 Implementation in TRIPOLI-4® 79

5.5 Scoring the adjoint flux with Adaptive Multilevel Splitting **79**

5.5.1 The estimator 79

5.5.2 Implementation in TRIPOLI-4® 80

5.6 Verification **83**

5.6.1 A homogeneous two-group infinite sphere 83

5.6.2 A two dimensional, multi-group, heterogeneous problem 85

5.7 Relation to previous work. **85**

5.7.1 Weight Window Generator 85

5.7.2 CADIS-**Ω** 87

5.8 Conclusions **87**

The aim of this chapter is to derive and describe the implementation of a Monte Carlo estimator of the adjoint flux for a direct Monte Carlo simulation.

In order to achieve this goal, we begin by illustrating the idea on a theoretical model. The estimator is then derived in the purely analog mode -i.e. none of the variance reduction methods introduced in Chapter 2 are used. It is then extended to importance sampling, and finally applied to AMS. The idea is to gradually add features to the adjoint estimator and show that we can handle problems of increasing complexity. We finish this chapter by verifying its implementation against solutions computed with the deterministic coupling and with a discussion about the differences between this score and the other MC importance estimators discussed in Section 3.3.

5.1 Illustration

Let us consider a $2\,\mathrm{m} \times 2\,\mathrm{m}$ box of concrete (see Figure 5.1a). A mono-energetic source is placed at position $r = (50, 100)$. The detector is a simple square region (green box). We are interested in scoring the adjoint flux in a region delimited by the small black cell by only using direct simulations.

The first idea that comes to mind is to uniformly sample particles in the black region of the phase space and simply compute the response in the detector (see Figure 5.1b). This is computationally very expensive because we typically want to have an estimation of the adjoint flux across a mesh composed of lots of cells.

The alternative proposed in this chapter is to take the original problem source S, propagate particles from there to the detector and then score an adjoint estimator by scanning these histories backward. Figure 5.1c illustrates the idea. Particles' trajectories leaving the source are plotted in blue. The ones contributing to the emission density in the scoring cell are colored in red after their first emission in the cell. A collision estimator is used for the target score: the contributions are thus localised at each collision in the detector region. So how is this formally written?

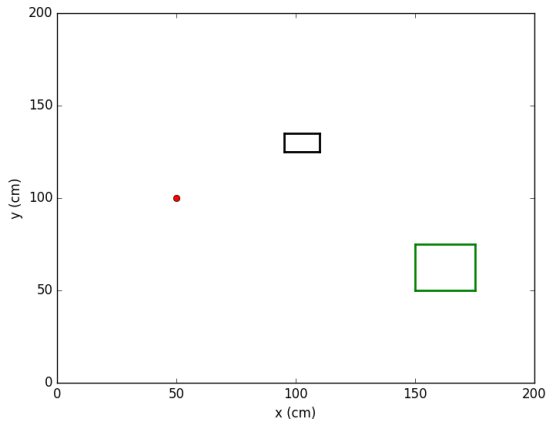
5.2 Derivation of the importance

Let c be the contribution to the detector response generated by a particle created at the phase-space point $(\mathbf{r}, E, \mathbf{\Omega})$. For example $c = 0$ when the particle does not reach the detector. If the particle reaches the detector and realises two collisions inside, then the contribution to the flux is $c = 2/\Sigma_t$. For given values of $\mathbf{r}, E$ and $\mathbf{\Omega}$, c is a random variable described by a conditional probability distribution $k(c|\mathbf{r}, E, \mathbf{\Omega})$. The adjoint emission density $\chi^\dagger(\mathbf{r}, E, \mathbf{\Omega})$ is the expected detector response from point $(\mathbf{r}, E, \mathbf{\Omega})$ in phase space. It reads as

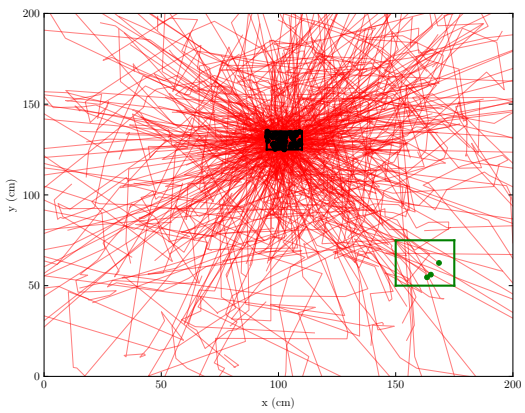
$$\chi^\dagger(\mathbf{r}, E, \mathbf{\Omega}) = \int c \cdot k(c|\mathbf{r}, E, \mathbf{\Omega})\,\mathrm{d}c. \tag{5.1}$$

Let $f(\mathbf{r}, E, \mathbf{\Omega})$ represent the probability density for a particle being produced at $(\mathbf{r}, E, \mathbf{\Omega})$ (by the source or by a collision) during one Monte-Carlo history. Then, by the law of conditional probability, the joint probability distribution for all the variables is given by

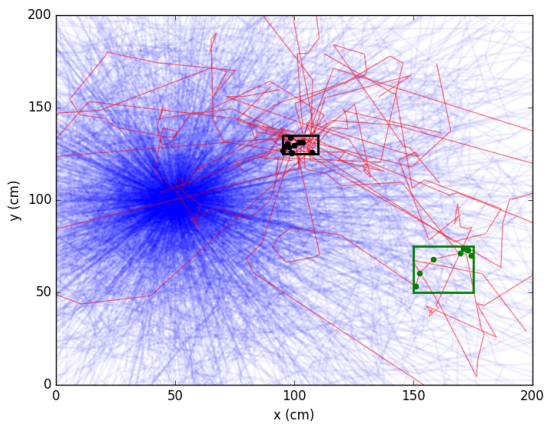
$$d(c, \mathbf{r}, E, \mathbf{\Omega}) = k(c|\mathbf{r}, E, \mathbf{\Omega}) \cdot f(\mathbf{r}, E, \mathbf{\Omega}).$$



(a) Theoretical 2D-problem description



(b) Uniform source



(c) Adjoint score

Figure 5.1. Illustration of two methods for the computation of the adjoint flux on a theoretical benchmark.

It is crucial to observe that d is exactly the distribution of the events generated by the direct Monte Carlo calculation. Therefore, Eq. (5.1) can be rewritten as

$$\chi^\dagger(\mathbf{r}, E, \mathbf{\Omega}) = \frac{\int c \cdot d(c, \mathbf{r}, E, \mathbf{\Omega})\, \mathrm{d}c}{f(\mathbf{r}, E, \mathbf{\Omega})}. \tag{5.2}$$

Eq. (5.2) allows us to construct an approximate estimator for the integral of $\chi^\dagger(\mathbf{r}, E, \mathbf{\Omega})$ over a certain neighbourhood δP of $(\mathbf{r}, E, \mathbf{\Omega})$. In Monte Carlo, such neighbourhoods are usually represented by bins in a histogram.

5.2.1 Physical interpretation

So how does this relate to the physical observables from Chapter 1? An interpretation of our score can be given by following the same procedure as in Chapter 3 for the physical interpretation of the adjoint flux.

Our problem is the following: we perform a Monte Carlo simulation with particles emitted at the source S_0. χ_0 is the resulting emission density. Also, we are interested in computing the response R_0 of a detector with sensitivity $\eta_{\chi,0}$ to the emission density.

Now, consider a second problem where the source is fixed to:

$$S(P) = \frac{\chi_0(P)}{\int_{\delta P} \chi_0(P)\, dP} \mathbb{1}_{\delta P}. \tag{5.3}$$

The emission density resulting from such a source is χ. Recall the duality equation Eq. (3.2) for this problem:

$$R = \int S(P)\chi^\dagger(P)\, dP = \int S^\dagger(P)\chi(P)\, dP$$

Because we are interested in the same detector as for the initial problem 0, take

$$S^\dagger = S_0^\dagger = \eta_{\chi,0}.$$

Now, compute the response R:

$$R = \int \chi(P)\eta_{\chi,0}(P)\, dP.$$

If we replace S by Eq. (5.3) in the duality equation Eq. (3.2), we obtain:

$$R = \frac{\int_{\delta P} \chi^\dagger(P)\chi_0(P)\, dP}{\int_{\delta P} \chi_0(P)\, dP} \tag{5.4}$$

We state that the *trajectories* simulated by a Monte Carlo calculation with source S_0 can be used to estimate the response R_0, but also the response R of a problem defined with source S.

5.3 The analog adjoint estimator

We wish to compute this response Eq. (5.4) with the analog strategy defined in Section 2.1.2. Particles are created at a fixed source S and are transported through the geometry; a few of them will actually reach the detector and contribute to the desired response R.

We consider a set $\mathcal{S}$ of N particle trajectories simulated in a batch

$$\mathcal{S} = \{\mathcal{T}_1, \ldots, \mathcal{T}_N\}.$$

Each trajectory is an ordered set of intermediate states of the Monte Carlo Markov process. Let n_j be the number of states visited by the trajectory $\mathcal{T}_j$.

$$\mathcal{T}_j = \{P_1, \ldots, P_{n_j}\} = \{(c_1, \boldsymbol{r}_1, E_1, \boldsymbol{\Omega}_1), \ldots, (c_{n_j}, \boldsymbol{r}_{n_j}, E_{n_j}, \boldsymbol{\Omega}_{n_j})\}.$$

The $\boldsymbol{r}_i$ should be taken to be **emissions or source sites** of particles with outgoing energy E_i and outgoing direction $\boldsymbol{\Omega}_i$; c_i **represents the contribution to the target score delivered by the particle or its descendants** (which may vanish if the particle does not contribute).

We define two observables on this trajectory.

$\psi_{D,\delta P}$ returns the sum of the future contributions for each point present in a phase space region δP.

$$\psi_{D,\delta P}(\mathcal{T}) = \sum_{P_i \in \mathcal{T}} c_i \, \mathbb{1}_{\delta P}(\boldsymbol{r}_i, E_i, \boldsymbol{\Omega}_i), \tag{5.5}$$

and $\psi_{F,\delta P}$ counts the number of emissions in the neighbourhood δP:

$$\psi_{F,\delta P}(\mathcal{T}) = \sum_{P_i \in \mathcal{T}} \mathbb{1}_{\delta P}(\boldsymbol{r}_i, E_i, \boldsymbol{\Omega}_i). \tag{5.6}$$

The function $\mathbb{1}_{\delta P}(\mathbf{r}, E, \boldsymbol{\Omega})$ is the characteristic function of δP (equal to 1 inside δP and to 0 outside).

The Monte Carlo estimators of these observables with the analog strategy read:

$$\hat{\psi}_{D,\delta P} = \frac{1}{N} \sum_{\mathcal{T} \in \mathcal{S}} \psi_{D,\delta P}(\mathcal{T}) \tag{5.7}$$

$$= \frac{1}{N} \sum_{\mathcal{T} \in \mathcal{S}} \sum_{P_i \in \mathcal{T}} c_i \, \mathbb{1}_{\delta P}(\boldsymbol{r}_i, E_i, \boldsymbol{\Omega}_i) \tag{5.8}$$

and

$$\hat{\psi}_{F,\delta P} = \frac{1}{N} \sum_{\mathcal{T} \in \mathcal{S}} \psi_{F,\delta P}(\mathcal{T}) \tag{5.9}$$

$$= \frac{1}{N} \sum_{\mathcal{T} \in \mathcal{S}} \sum_{P_i \in \mathcal{T}} \mathbb{1}_{\delta P}(\boldsymbol{r}_i, E_i, \boldsymbol{\Omega}_i) \tag{5.10}$$

Their expectations are

$$\mathbb{E}(\hat{\psi}_{D,\delta P}) = \int \mathrm{d}c \int_{\delta P} \mathrm{d}^3\boldsymbol{r}\, \mathrm{d}E\, \mathrm{d}^2\boldsymbol{\Omega}\, [c \cdot d(c, \mathbf{r}, E, \boldsymbol{\Omega})]\,.$$

and

$$\mathbb{E}(\hat{\psi}_{F,\delta P}) = \int_{\delta P} \mathrm{d}^3\boldsymbol{r}\, \mathrm{d}E\, \mathrm{d}^2\boldsymbol{\Omega}\, [f(\mathbf{r}, E, \boldsymbol{\Omega})]\,.$$

We define our approximate estimator of the adjoint flux in phase space region as

$$\hat{\chi}^{\dagger}_{\delta P} = \frac{\hat{\psi}_{D,\delta P}}{\hat{\psi}_{F,\delta P}}.$$

There are two reasons why $\hat{\chi}^{\dagger}_{\delta P}$ is approximate. First, $\hat{D}$ and $\hat{F}$ estimate averages of their integrands over a certain neighbourhood δP of $(\mathbf{r}, E, \boldsymbol{\Omega})$. Second, $\hat{\chi}^{\dagger}_{\delta P}$ is biased because of

$$\mathbb{E}(\hat{\chi}^{\dagger}_{\delta P}) \neq \frac{\mathbb{E}(\hat{\psi}_{D,\delta P})}{\mathbb{E}(\hat{\psi}_{F,\delta P})}.$$

However, $\hat{\chi}^{\dagger}_{\delta P}$ is consistent (asymptotically unbiased) as the sample size N tends to infinity and the size of the neighbourhood δP tends to zero, in this order:

$$\mathbb{E}(\lim_{\delta P \to 0} \lim_{n \to \infty} \hat{\chi}^{\dagger}_{\delta P}) = \chi^{\dagger}(\mathbf{r}, E, \boldsymbol{\Omega}).$$

In summary, we propose the following procedure for scoring the adjoint flux in a phase-space cell:

1. collect all particles emitted by collisions within the cell;
2. associate each point with the contribution to the detector response delivered by the particle and its descendants from that point onwards;
3. calculate $\hat{\psi}_{D,\delta P}$ (Eq. (5.8)) by summing the particle contributions;
4. calculate $\hat{\psi}_{F,\delta P}$ (Eq. (5.10)) by summing the number of collisions and source emissions;

5. divide $\hat{\psi}_{D,\delta P}$ by $\hat{\psi}_{F,\delta P}$ to estimate the adjoint flux in the phase space cell.

5.3.1 The variance

The definition of this estimator should come with an estimator of its variance $\sigma^2_{\hat{\chi}^\dagger_{\delta P}}$. We use a zero order approximation of the variance of the ratio, which, as a function of the variances $\sigma^2_{D,\delta P}$ and $\sigma^2_{F,\delta P}$ reads:

$$\sigma^2_{\hat{\chi}^\dagger_{\delta P}} \approx \frac{\sigma^2_{D,\delta P}}{\hat{\psi}^2_{D,\delta P}} + \frac{\sigma^2_{F,\delta P}\hat{\psi}^2_{D,\delta P}}{\hat{\psi}^4_{F,\delta P}} - 2\frac{\hat{\psi}_{D,\delta P}}{\hat{\psi}^3_{F,\delta P}} cov(\hat{\psi}_{D,\delta P}, \hat{\psi}_{D,\delta P}) \tag{5.11}$$

5.3.2 Implementation of the score in TRIPOLI-4®

This estimator was implemented in TRIPOLI-4® in the most intuitive manner, minimizing the amount of changes from an original TRIPOLI-4® input file.

We assume that the input file already contains a list of scores (on a mesh, or a volume), associated to responses (flux, current,...). The first intervention expected from the user is to define which of these scores are of interest. This is simply done by specifying the keyword **TARGET** in the definition of the score. This assures that all the contributions to these scores will be added to the estimator $\hat{\psi}_{D,\delta P}$.

Then, an **ADJOINT** response needs to be specified in the list of responses. This response can be scored on classical score supports such as volumes, list of volumes or meshes (as well as extended meshes in TRIPOLI-4®).

The algorithm

Our adjoint estimator requires to store the particle history until its end, and scan it backwards afterwards. In the following, we will refer to the *forward phase* as the direct part of the simulation from the source to the detector and to the *backward phase* as the scan from the detector back to the source.

In order for the algorithm to be clear, we first define how the particle history is stored in memory. The solution chosen is to store the consecutive states of the particle trajectory in a *track*. This track is composed of a series of points. Each of these points is labelled with a status. Figure 5.2 shows the transport of two particles from an arbitrary source to a detector in a given geometry with more than one volume. The history starts with the status **source**. It is transported by the transition kernel to a location where it will undergoe a collision. Its status is:

- **depos** before the collision.
- **live** when it exits a collision.
- **kill** when it is absorbed.

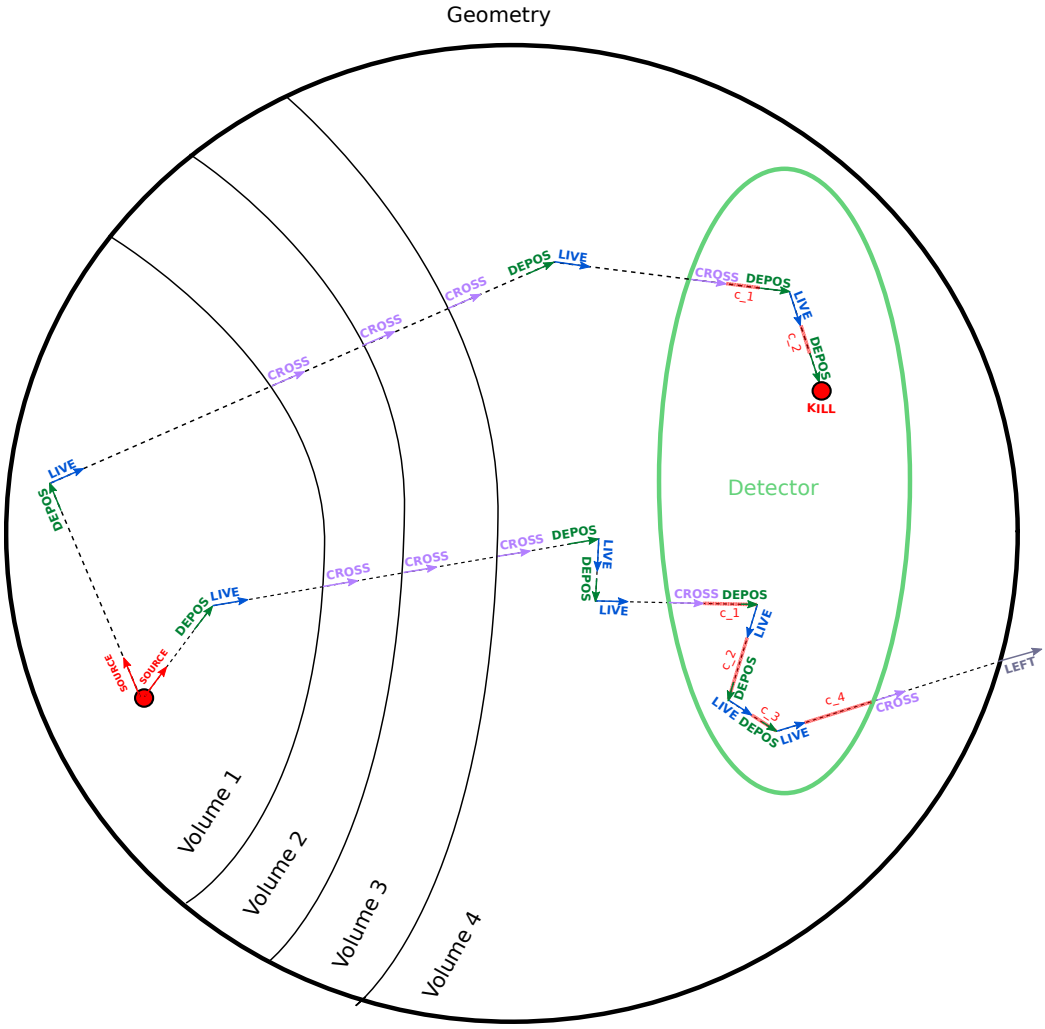


Figure 5.2. Two analog non branching tracks stored in TRIPOLI-4®

- **left** when it leaves the geometry.

Also, our tracks contain additional fictitious points, with status **cross**, which are the locations where a volume was entered. This is necessary if the score in the target is computed with a *track length* estimator.

During the transport, contributions are accumulated to the target scores and stored in some of these points. The strategy is to assign the contribution that was just scored in the target to the next point in the history. For example, if the score in the target is accumulated with a track length estimator, three situations can happen:

- the particle enters the target volume and undergoes a collision, or is killed. Then the point with status **depos** carries the contribution of the flight that has just ended.

- the particle enters the target volume, but exits it without undergoing any collision inside. Then the point with status **cross** will carry the contribution in the target volume.

- if the target volume is on the boundary of the geometry, and a particle enters

the target volume and exits the geometry, the particle with status **left** carries the contribution.

If the score is scored at the collisions, the point with status **depos** carries this contribution. The idea is to always carry the contribution that was just scored in the following point.

With these definitions, we can write Algorithm 1 to score the adjoint response at the end of the simulation of one particle history.

Algorithm 1: Adjoint scores update on a single-branch track

```
/* reset cumulated contribution                                          */
contrib = 0;
points = track.get_points();
/* we loop backward on points                                            */
for point in reversed(points) do
    status = point.get_status();
    /* we update the adjoint scores at each emission (snippet 1)         */
    if status is LIVE or SOURCE then
        point.set_contribution(contrib);
        update_adjoint_scores(point);
    end
    /* we update the contribution if the point carries one (snippet 2)   */
    if status is DEPOS or LEFT or CROSS then
        contrib+ = point.get_contribution();
    end
end
```

Points are scanned backward from the end of the history. The contribution is accumulated at each point of status **depos**, **left** or **cross**. The points with status **live** or **source** are assigned with the current cumulated contribution and are passed to the method *update_adjoint_scores*() which updates all adjoint scores present in the simulation.

This first algorithm applies only when only one particle type is simulated. For example, only neutrons, or only photons. It needs to be extended for particle histories that experience interactions where particles are emitted by branching events. These events include *fission*, or all the interactions that emits particle of other types. If such branching happens, we adopt the same definition of branches as Louvin (2017). A track is now constituted by a set of branches. Each branch contains the same series of points as the single-branch track previously defined. In order to scan histories backward, each branch needs to carry additional information. First, the *id* of its parent branch, and second, the rank on its parent branch, where it has been emitted.

These definitions suffice to adapt Algorithm 1 to branching events and write Algorithm 2. Each branch is scanned backward on its parent branches until the source is

reached. Contributions and adjoint scores update are performed in the same manner as previously described.

Algorithm 2: Adjoint scores update with analog branching tracks

```
/* loop on each branch of a particle history                                  */
for branch in track.get_branches() do
    contrib = 0;
    branch' = branch;
    last = branch'.get_nb_points();
    /* loop on parent branches                                                 */
    while last > -1 do
        points = branch'.get_points([0, last]);
        for point in reversed(points) do
            /* update adjoint scores (snippet 1)                               */
            /* update contributions (snippet 2)                                */
        end
        /* jump on parent branch (snippet 3)                                   */
        branch' = branch'.get_moter_branch();
        /* at the right rank                                                   */
        /* returns -1 if the branch does not have a parent one                 */
        last = branch'.get_rank_in_mother_branch();
    end
end
```

5.4 Scoring the adjoint flux with importance sampling

5.4.1 The estimator

Although analog simulation is the most reliable strategy, realistic shielding problems are untreatable in a reasonable amount of time to perform studies. In this section, we thus extend the adjoint score to strategies where importance sampling is used. **Implicit capture** and the **exponential transform** fall into this category. For both these methods, the weight w of a particle can change over time. It is straightforward to add the particles' weight into the phase space. We re-write the estimators and note them $\hat{\psi}^{w}_{D,\delta P}$ and $\hat{\psi}^{w}_{F,\delta P}$.

The estimator for $\hat{\psi}^{w}_{D,\delta P}$, now reads:

$$\hat{\psi}^{w}_{D,\delta P} = \frac{1}{N} \sum_{\mathcal{T} \in \mathcal{S}} \sum_{P_i \in \mathcal{T}} w_i \, c_i \, \mathbb{1}_{\delta P}\big(\boldsymbol{r}_i, E_i, \boldsymbol{\Omega}_i\big) \tag{5.12}$$

and the one for $\hat{\psi}^{w}_{F,\delta P}$:

$$\hat{\psi}^{w}_{F,\delta P} = \frac{1}{N} \sum_{\mathcal{T} \in \mathcal{S}} \sum_{P_i \in \mathcal{T}} w_i \, \mathbb{1}_{\delta P}\big(\boldsymbol{r}_i, E_i, \boldsymbol{\Omega}_i\big). \tag{5.13}$$

Their expectations do not change from the previous estimators.

5.4.2 Implementation in TRIPOLI-4®

Although the score is implemented for this method, we think that scoring the adjoint flux with the exponential transform is unsafe. The method deliberately misses out phase-space areas to go straight to the detector. We thus expect the variance of the adjoint estimator to be unsatisfactorily high, hence the focus on AMS.

5.5 Scoring the adjoint flux with Adaptive Multilevel Splitting

Until now, the term trajectory and track were used alternatively for both the analog and the importance sampling strategy. With AMS, we revise the definition of a *track* compared to a *trajectory*. A track is now defined as the series of particle states that begins with an emission *or a splitting point* and ends when the particle is absorbed or leaks out from the geometry. With this definition, a track can thus begin at any iteration at a splitting point. However, this definition is purely established for memory convenience. From the mathematical point of view, a *trajectory* is defined as the track at any iteration *with* its history from the initial emission. For example Figure 5.3a shows the tracks simulated with AMS, whereas Figure 5.3b shows the trajectories actually sampled and used in the estimator.

5.5.1 The estimator

We simulate N trajectories per iteration, with K_i the number of duplications at iteration i. Recall from Section 2.3.2 that AMS produces a set of trajectories

$$T^{\text{AMS}} = \left(\bigcup_{q=0}^{Q} S_q^{\text{off}} \right) \bigcup S_Q^{\text{on}}$$

associated with their weight

$$w_{\text{AMS}}\left(\mathcal{T}\right) = \frac{1}{N} \prod_{i=0}^{j} \left(1 - \frac{K_i}{N}\right), \text{ for } \mathcal{T} \in T^{\text{AMS}}, \tag{5.14}$$

if j is the iteration at which the score is collected. The AMS estimator now reads:

$$\hat{\psi}_{\text{AMS}} = \sum_{\mathcal{T} \in T^{\text{AMS}}} w_{\text{AMS}}\, \psi(\mathcal{T}) \tag{5.15}$$

By replacing our observables from (5.5) and (5.6), in (5.15), and by using the simplified notations, we derive the estimators to be used with AMS:

$$\psi^{\text{AMS}}_{D,\delta P}(T^{\text{AMS}}) = \sum_{\mathcal{T}\in T^{\text{AMS}}} w_{\text{AMS}}(\mathcal{T}) \sum_{P_i\in\mathcal{T}} c_i \, \mathbb{1}_{\delta P}\big(\boldsymbol{r}_i, E_i, \boldsymbol{\Omega}_i\big) \tag{5.16}$$

$$\psi^{\text{AMS}}_{F,\delta P}(T^{\text{AMS}}) = \sum_{\mathcal{T}\in T^{\text{AMS}}} w_{\text{AMS}}(\mathcal{T}) \sum_{P_i\in\mathcal{T}} \mathbb{1}_{\delta P}\big(\boldsymbol{r}_i, E_i, \boldsymbol{\Omega}_i\big) \tag{5.17}$$

5.5.2 Implementation in TRIPOLI-4®

In order to clearly understand the algorithm with AMS, it is necessary to recall how the simulation of the particle is done from the source to the detector - i.e. in the forward phase of the simulation. Algorithm 3 describes the modifications to Algorithm 2 that need to be implemented to score the adjoint flux with AMS. Figure 5.3a illustrates the AMS algorithm and the adjoint estimator. Figure 5.3a shows five iterations of AMS with $N = 3$ and $k = 1$. The branches are colored depending on the iteration at which they were simulated. The cross symbol at the end of the history is colored with the iteration at which it was removed from the set of active trajectories.

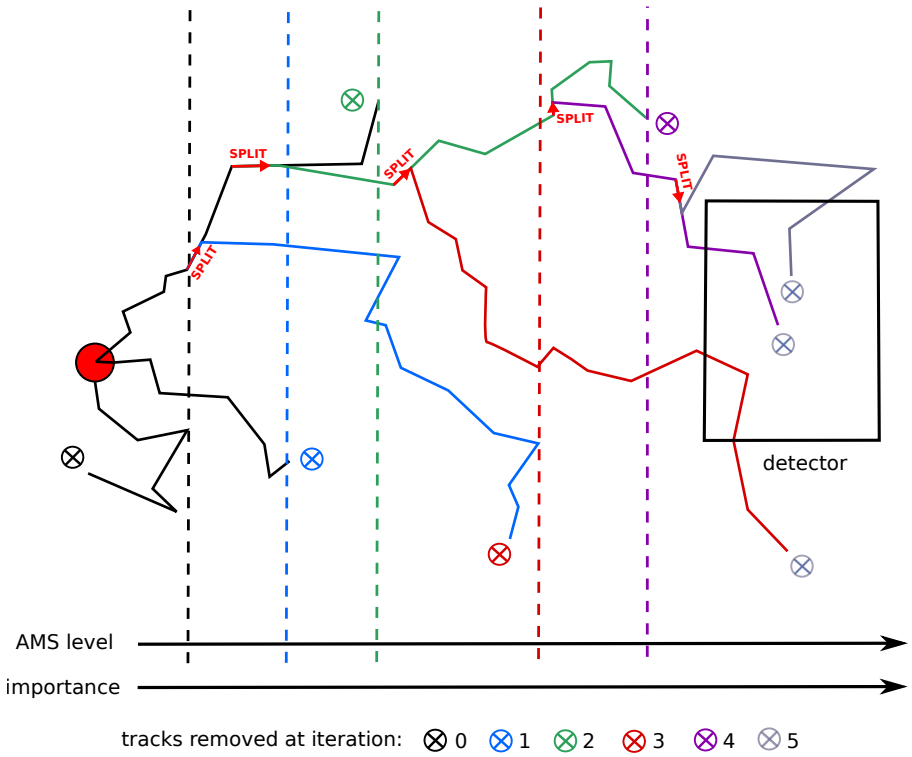
For example, the blue track is the first track duplicated. It survives until the third iteration. The corresponding set of trajectories T^{AMS} used by the AMS estimator are illustrated in Figure 5.3b. The trajectories include the duplicated parts of parent tracks, weighted by the AMS weight w_{AMS}.

In the native implementation of AMS, only the tracks at the current iteration are kept in memory as shown in Figure 5.3a. However, because the estimator (5.17) needs the total contribution of the trajectory, and because this contribution is not known till the history is finished, one must keep the entire history in memory. This is not how this is done in the native implementation of AMS in TRIPOLI-4® by Louvin (see paragraph named *Track storage* in section 3.2.4 of Louvin (2017)). Note finally that in the original implementation, there are points that are not added to the track in two cases in particular:

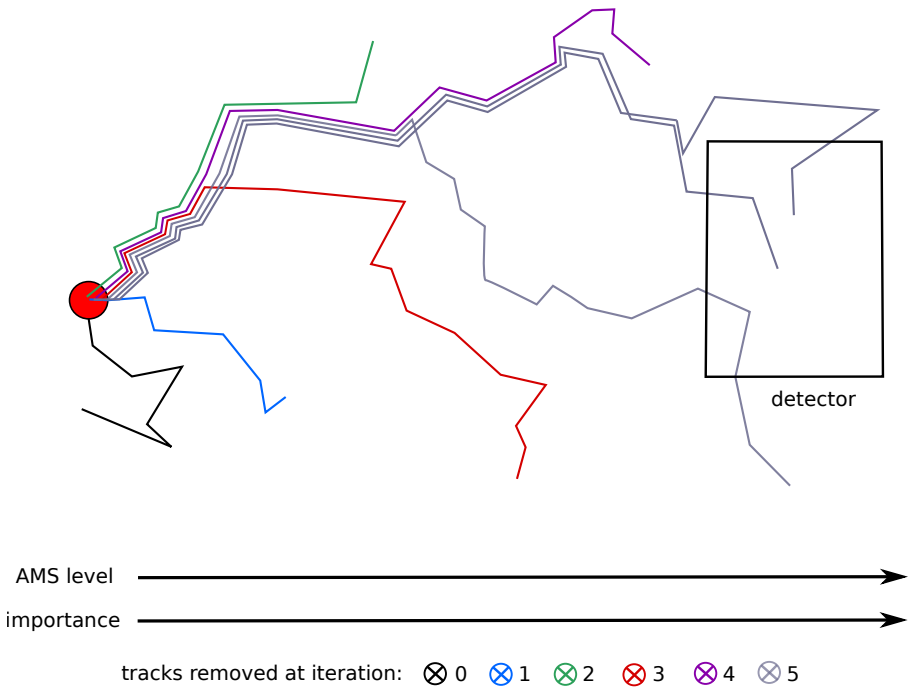
- During the simulation, if the next emission point has an importance lower than the current importance of the track, then the emission is not added to the track. However, estimator 5.17 requires all the emissions to be taken into account. Therefore, if an emission is missing, the estimator will fail to recover the proper estimate.

- Also, consider the situation where branching events can happen, for example gamma particles emitted from (n, γ) interactions. In section 6.2.2 of Louvin (2017) - and also in TRIPOLI-4® - it is clear that if a track branches at some point and needs to be selected for subsequent splitting at an AMS iteration, then *one of the branches* can be removed from memory. We need to keep this part of the history in memory in order to provide with an adjoint estimator.

For these reasons, the optimisations that were present in the code, like resetting pointers on branching tracks' branches were bypassed in order to keep branches of interest in memory.



(a) AMS tracks for $N = 3$ trajectories sampled, $k = 1$ is histories restarted at each iteration. A different color is assigned to each AMS iteration. Tracks are colored with the color corresponding to the iteration at which it was simulated. Cross symbols are of the color of the iteration at which they were killed



(b) AMS trajectories colored with the AMS weight of the iteration at which they were killed.

Figure 5.3. Illustration of the AMS estimator and of the track structure during an AMS simulation.

Algorithm 3: Adjoint scores update on AMS tracks

```
for track in tracks do
    for branch in track.get_branches() do
        contrib = 0;
        branch' = branch;
        while last > −1 do
            points = branch'.get_points([0, last]);
            /* retrieve the weight of the branch when it was removed from the ams
               set of active tracks                                              */
            ams_weight = branch.get_ams_weight();
            for point in reversed(points) do
                status = point.get_status();
                /* we update the adjoint scores at each emission                */
                if status is LIVE or SOURCE then
                    /* the point is assigned with its correct weight, for scoring
                       */
                    point.set_ams_weight(ams_weight);
                    point.set_contribution(contrib);
                    update_adjoint_scores(point);
                end
                /* update the contribution (snippet 2)                          */
            end
            /* jump on parent branch (snippet 3)                                */
        end
    end
end
```

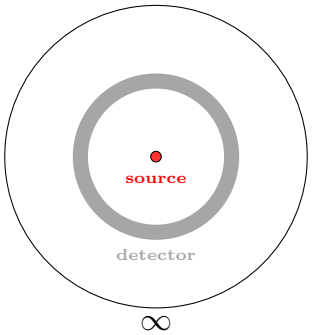


Figure 5.4. Geometry of the 2-group sphere problem

5.6 Verification

We now verify our estimators for simple cases where we can compute the solution to the adjoint Boltzmann equation.

We make the choice not to discretise the angular component, and thus integrate $\hat{\psi}_{D,\delta P}$ and $\hat{\psi}_{F,\delta P}$ over all angles indifferently. What are the risks? If the Monte Carlo simulated emission density is isotropic in a cell, then the score converges exactly to the scalar adjoint emission density. On the other hand, if the angular distribution of the flux and/or the adjoint flux is not isotropic, the score is biased, and the mesh should be refined in the angular direction to minimise the bias.

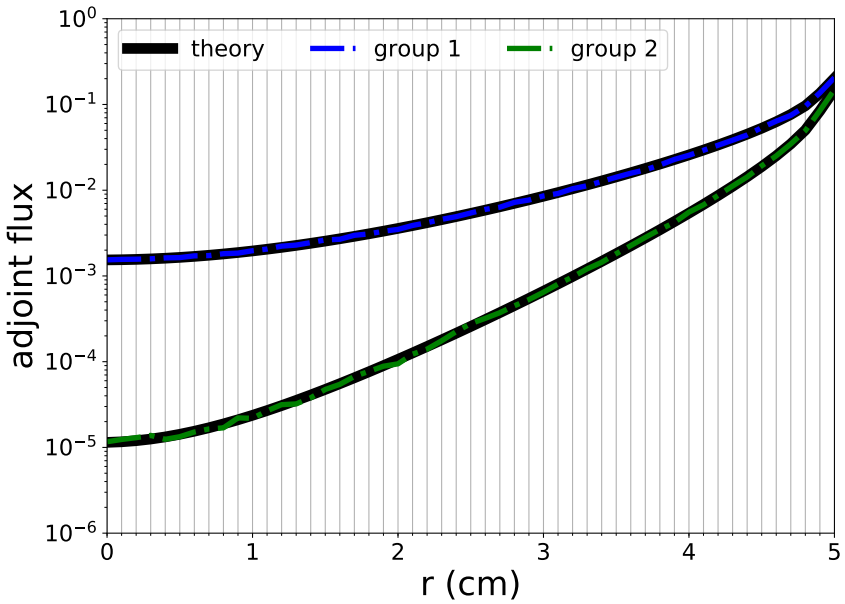
5.6.1 A homogeneous two-group infinite sphere

Let us start looking at a simple problem for which the adjoint flux can be calculated analytically easily. We define an infinite medium filled with a homogeneous material with an isotropic scattering cross section. The macroscopic cross sections for this material are given in two energy groups.

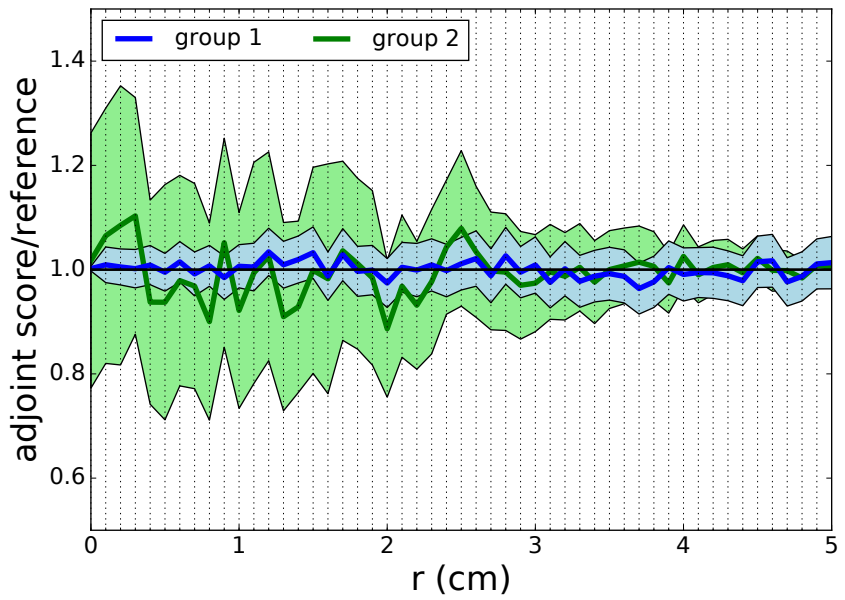
$$\Sigma_t^1 = 1\,\mathrm{cm}^{-1}, \Sigma_s^{1\to1} = 0.2\,\mathrm{cm}^{-1}, \Sigma_s^{1\to2} = 0.4\,\mathrm{cm}^{-1}, \Sigma_t^2 = 2\,\mathrm{cm}^{-1}, \Sigma_s^{2\to2} = 0.5\,\mathrm{cm}^{-1}$$

The sought response is the flux in the shell between $r = 5\,\mathrm{cm}$ and $r = 5.1\,\mathrm{cm}$. In this simple case, the reference adjoint solution can be written in closed form as an integral which can be numerically evaluated (Louvin (2017)).

Figure 5.5 represents the reference solution integrated in 0.1 cm wide spherical shells from the source to the detector. The adjoint score is computed with AMS ($N = 1000$, $k = 10\%$) and is plotted for both energy groups. The score and the reference are perfectly consistent and within 3 standard deviations of the adjoint score. The same computations were performed with the analog strategy and lead to higher variance for the same simulation time.



(a) Adjoint score



(b) Adjoint score over reference

Figure 5.5. Adjoint flux scored on a mesh with regular 0.1 cm-spaced cells along the radial direction. The score is performed with AMS. Black lines are reference solutions obtained by a numeric integration. The blue (green) curve shows the result for the fast (slow) group with associated confidence intervals at three standard deviations.

5.6.2 A two dimensional, multi-group, heterogeneous problem

Now, let us validate the developments against a deterministic adjoint response. We take the same benchmark that was used to verify IDT [Appendix A.3]. In Figure 5.6a, the adjoint solution provided by IDT is plotted for the 5 energy groups of the problem. In order to obtain the adjoint score with TRIPOLI-4®, AMS was run with the importance map computed by INIPOND. The adjoint score was collected on the same mesh that was used with IDT. The simulation was stopped when the convergence was estimated to be sufficient. The resulting scores for the same energy groups as previously(see [Appendix A.3]) are plotted in Figure 5.6b. We observe that even after a long simulation, many regions of the phase space are left unexplored. This is explained by the fact that the forward step is made with a defined *variance reduction* strategy (here AMS/INIPOND) that favours the simulation of a subset of the phase space. The discrepancies is σ of the adjoint flux computed with IDT compared to the adjoint score is shown in Figure 5.6c. We observe that the discrepancies between the adjoint score and the deterministic solution are contained within the tolerances at 3 σ.

5.7 Relation to previous work.

We here compare the implementation of this score to the other implementations of Monte Carlo importance estimators in the literature. Also, we finish this section by comparing to a recent deterministic approach.

5.7.1 Weight Window Generator

Recall from Section 3.3 that the corrected WWG developed by Evans and Hendricks (1998), counts the number of particles *entering the cell* and born from a source in the cell. The main difference with our estimator is that even if a particle collides multiple times in the cell before leaving, all occurrences of emissions are accumulated, whereas the WWG only counts particles entering the cell.

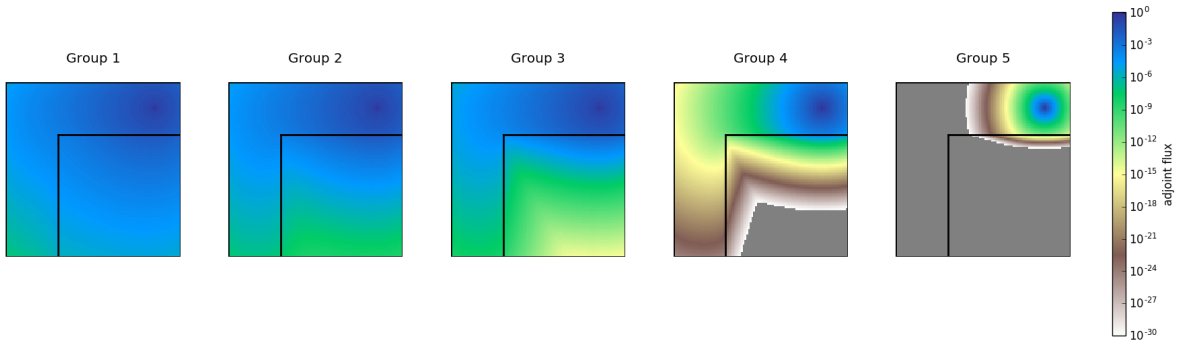
The distribution of our points contributing to the adjoint score is more isotropic than the one of the WWG, because we place ourselves just after the collision kernel.

The WWG also considers all particles streaming through the mesh cells. In brief, they consider a "track" estimator of the importance, which we do not.

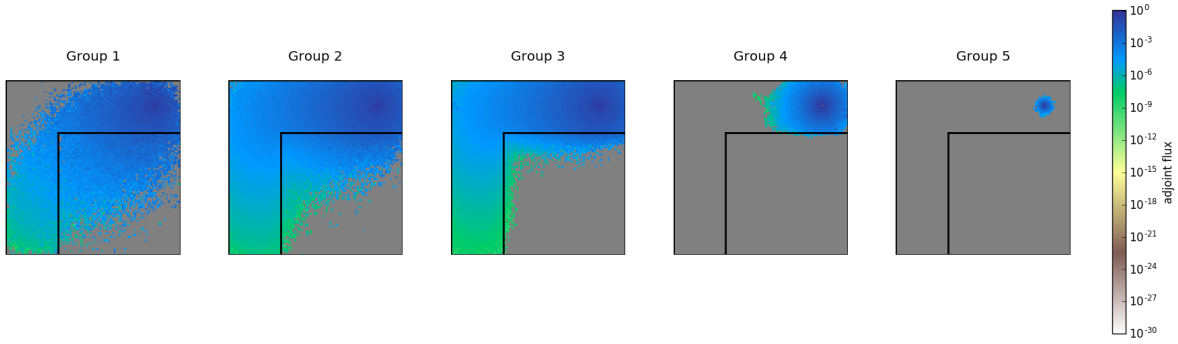
Our score works with AMS, which has the advantage of being less sensitive to the quality of the importance map than other methods. This means that with a crude initial guess of the importance map, our adjoint score with AMS has good chances to converge.

One last remark comparing our scoring method with WWG is the choice of the source. With WWG, it is possible to sample a uniform source across the entire geometry. There are four reasons why doing this would not yield good results with our methodology.

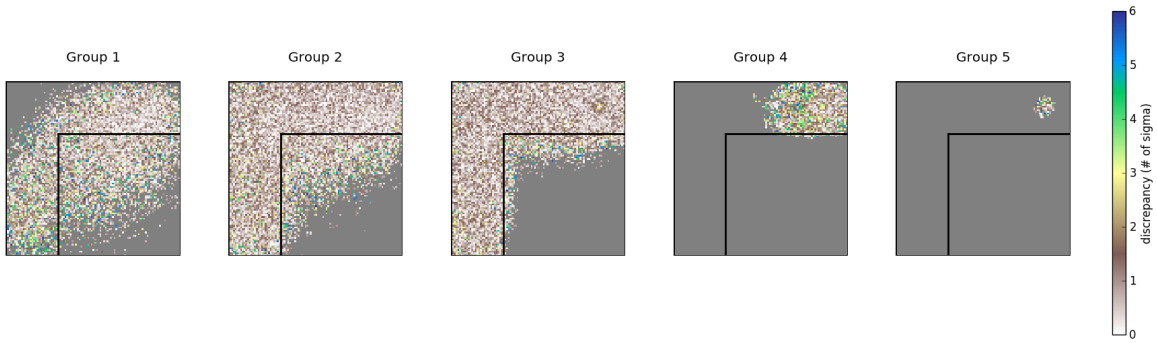
- This would require the user to choose a non trivial virtual source S_0 (refer to Section 5.2.1) .
- There would be many regions of the phase space where particles are simulated and never actually reach the detector. In order to overcome this, one would have to use variance reduction methods. In this sense, WW and ET could converge. But again,



(a) IDT



(b) TRIPOLI-4®



(c) Discrepancies/σ TRIPOLI-4®

Figure 5.6. Adjoint flux comparison for the water-iron benchmark. The first row represents the flux computed by IDT, the second one with TRIPOLI-4®. The last figure represents the discrepancy between IDT and TRIPOLI-4® as ratio between their difference and the standard deviation of the adjoint score of TRIPOLI-4®.

if one does so it means that the importance map is not well known at the beginning, so it could be hard for ET to converge.

- Finally, if AMS is considered, then the trajectories at the end of the first iteration that will be more likely to be splitted are the ones near the detector (because if you sample uniformly across the geometry, there are inevitably particles emitted near the detector). AMS would stop in a few iterations, and the variance of the estimator would be only reduced near the detector. There is no reason that the phase space between the source and the detector is privileged in this scheme.

These questions lead us to a crucial question: where is it important to have a good knowledge of the importance? Where is it worth spending time converging the importance score?

5.7.2 CADIS-Ω

Munk et al. (2016) proposes to compute with deterministic methods the following quantity:

$$\Phi^{\dagger}_{\mathbf{\Omega}}(\mathbf{r}, E) = \frac{\int_{4\pi} \Phi(\mathbf{r}, E, \mathbf{\Omega})\Phi^{\dagger}(\mathbf{r}, E, \mathbf{\Omega})\, d\mathbf{\Omega}}{\int_{4\pi} \Phi(\mathbf{r}, E, \mathbf{\Omega})\, d\mathbf{\Omega}} \tag{5.18}$$

This relation was derived from the hope of taking the information of the angular flux from deterministic methods and incorporating it into the importance map. This relation is exactly what our score does, except that we score this quantity in Monte Carlo simulations. The authors report that taking this quantity as an importance map yields better MC accelerations than the classical CADIS methodology (for some cases only).

5.8 Conclusions

In this chapter, we presented how an adjoint estimator can be derived for different Monte Carlo strategies. Namely, when following analog transport, importance sampling or AMS. We detailed the implementation in TRIPOLI-4® and verified it against two simple multi-group problems. In both cases, the adjoint score succeeds in estimating the adjoint flux in some regions of the phase space. These regions depend on the variance reduction method used to reach the detector and on the importance map fed to the method. However, we did not verify our estimator in the case where anisotropy is dominant. An angular discretisation should be investigated, despite the large memory footprint that would result. The idea is now to see if this adjoint score can accelerate the variance reduction methods present in TRIPOLI-4®.

Chapter 6

Using the adjoint score as an importance map

Contents

6.1 Two-step scheme **90**
6.1.1 Strong attenuation 90
6.1.2 Coupled Neutron-Gamma problem 91
6.1.3 Conclusions 94
6.2 Fixed-step alternating scheme **96**
6.2.1 Strong attenuation 97
6.2.2 Neutron Gamma 99
6.2.3 Conclusions 102
6.3 Bootstrap scheme **102**
6.3.1 Strong attenuation 104
6.3.2 Neutron Gamma 106
6.3.3 Capturing a variance jump? 106
6.3.4 Conclusions 111
6.4 A few words on the parallelisation of these strategies. **111**
6.5 Conclusions **113**

Now that we are capable of scoring the adjoint flux on a mesh, we would like to evaluate the accelerations obtained by re-using the adjoint score as an importance map for AMS and ET. In order to do so, one must first decide about a strategy for updating the importance map with the results of the adjoint score. This chapter proposes three different strategies. The first strategy consists in scoring the importance once only in a first step and then reintegrating it into a second-step Monte Carlo simulation. The second strategy is an attempt to quantitatively evaluate the convergence of the adjoint score. Finally, the third strategy - to which we will refer to as bootstrapping - consists in beginning the forward simulation with an initial importance map and then replacing this importance map by the result of a backward phase during which the adjoint score is computed.

6.1 Two-step scheme

In this section we are interested in verifying whether the adjoint score can improve the figure of merit of AMS or ET. In order to do so, we perform a two-step calculation. The first step is an *exploration phase*: the adjoint response is scored on a user-defined discretised phase space with the methods implemented in the last chapter. Because we know AMS is more robust compared to ET, we use AMS to perform this exploration phase. Depending on the complexity of the problem, and on the difficulty to generate an importance map, we can choose between three possible initialisations for the importance map: (a) a purely geometrical importance map, (b) INIPOND and (c) IDT.

After a *reasonable* simulation time, the score is dumped into an importance map and a second independent simulation follows. For now, the definition of "reasonable" is left to the discretion of the user. This step will be referred to as the *exploitation phase*. Once the target response of the second step converges to a pre-defined tolerance (usually about 10%), the calculation is stopped. Figure 6.1 illustrates the scheme.

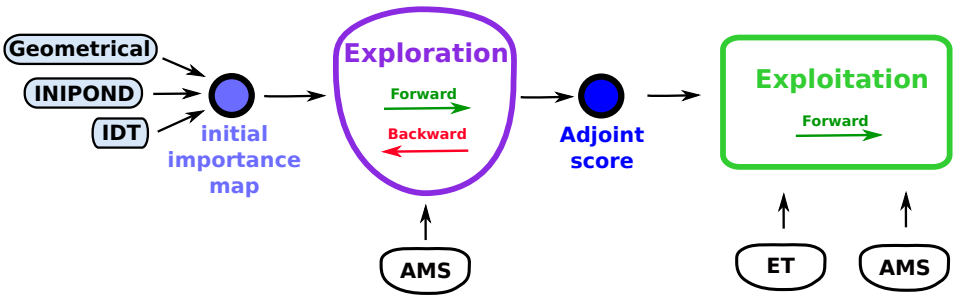


Figure 6.1. Two-step adjoint response update scheme.

6.1.1 Strong attenuation

We begin by looking at the strong attenuation benchmark [Appendix A.5].

First step: scoring the adjoint flux

AMS is run with the importance map produced by IDT with cross sections from APOLLO3®. After running 24 hour on 24 threads [1], the calculation is stopped and the resulting score

[1] Intel Xeon CPU E5-2620 0 @ 2.00GHz, operating system Centos 7.

is dumped in an importance map. Figure 6.2a shows the importance map produced with IDT (cross sections from APOLLO3®) and Figure 6.2b the scored one. Overall, and qualitatively, the two importance maps relay the same message: at a given position, a fast particle will contribute more to the detector than a thermal one. Also, trajectories thermalising near the source are never duplicated by AMS, which results in very poor convergence of the adjoint response for thermal groups farther than 1 m from the detector.

In order to be more quantitative, Figure 6.2c shows the ratio between the two importance maps. A few energy groups were selected for the figure to be readable. The IDT and scored importance maps are seen to deviate from each other and reach a maximum ratio of 20 near the direct source. The disagreement is due to an inaccurate solution of the adjoint transport equation by IDT. There are several possible factors that may contribute:

1. the cross sections used by IDT were not condensed on a sufficiently fine energy grid.
2. the angular discretisation (P5) of the transport calculation was not sufficiently fine. This may especially be the case in water, where high-energy neutron scattering is strongly anisotropic.
3. the spatial discretisation is not sufficiently fine.

Second step: using the adjoint flux as an importance map

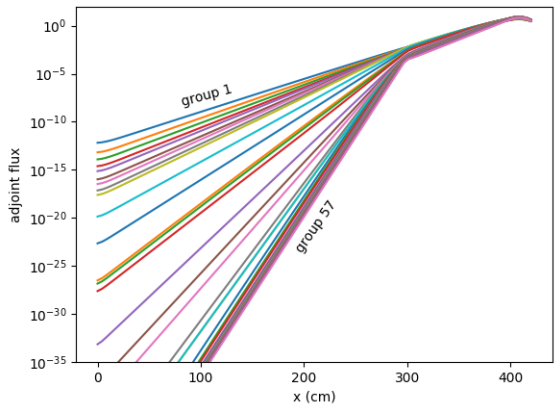
Two subsequent calculations follow where the scored importance map shown in Figure 6.2b is used with (a) AMS and (b) ET. Results are shown in Table 6.1 along with those already presented in Chapter 4, Table 4.1.

AMS is run with 10% re-sampled particles per iteration. All simulations, apart from ET with IDT and cross sections from TRIPOLI4 as already discussed in chapter 4, are in agreement with the reference. AMS yields a figure of merit of the same order of magnitude as those obtained with IDT (cross sections from TRIPOLI-4®), or IDT (cross sections from APOLLO3®). On the other hand, the FOM obtained with the Exponential Transform using the adjoint flux as an importance map is twice as large as the Exponential Transform used with IDT (cross sections from APOLLO3®).

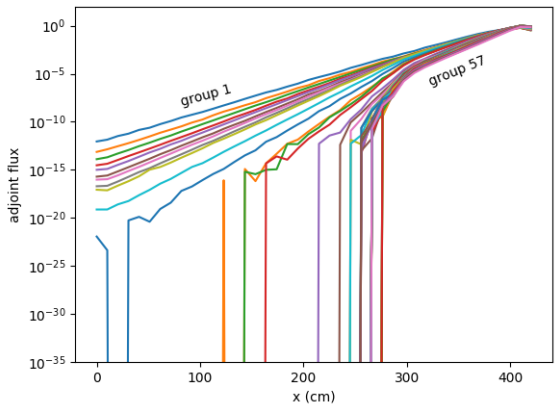
The main difficulty of this benchmark is to transport fast neutrons near the concrete part, and then thermalize. We saw that a factor of 2 was gained with the scored importance map compared to the deterministic one. We do not consider that this is a significant difference. It is surprising that importance maps with ratio as large as 20 may yield FOMs that are relatively close to each other. Let us now turn to a more challenging type of shielding problem.

6.1.2 Coupled Neutron-Gamma problem

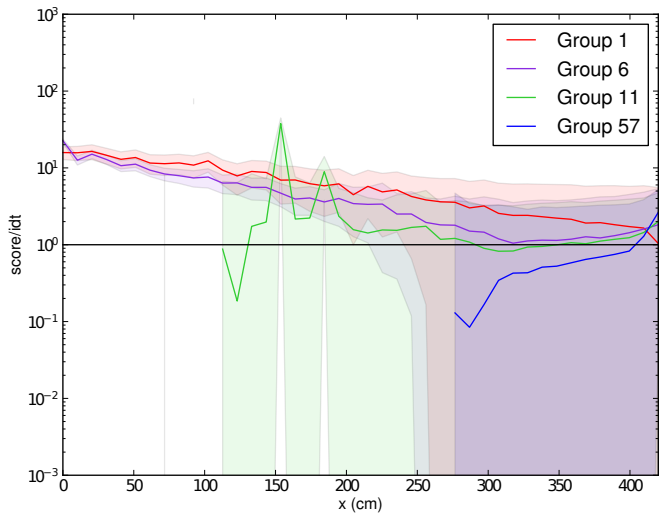
For that, we focus on a problem where neutrons and gammas propagate. A full description of the benchmark is presented [Appendix A.6]. This benchmark was initially studied with TRIPOLI-4® by Petit, Lee, and Diop (2014). The main reason to study this problem is that the current IDT coupling does not handle neutron-gamma problems. So for this problem, there is a real need for reference importance maps to be generated. Figure 6.3 illustrates the problem geometry. The benchmark consists of a neutron source,



(a) IDT (cross sections from APOLLO3®)



(b) Adjoint score



(c) Ratio of the adjoint score over IDT (cross sections from APOLLO3®), for four energy groups. Discrepancies of the order of a factor of 10 are observed for fast groups. Thermal groups are consistent given the statistical uncertainty.

Figure 6.2. Comparison between adjoint fluxes computed by IDT and scored with AMS.

Importance map (properties)	**Mean**($n/cm^3/s$)	**error**(%)	**time**(s)	**FOM** (arbitrary units)
Adaptive Multilevel Splitting				
INIPOND (Manual mode)	2.58×10^{-15}	9.90	1.67×10^5	9
IDT (cross sections from TRIPOLI4)	2.61×10^{-15}	9.83	1.20×10^5	13
IDT (cross sections from APOLLO3)	2.78×10^{-15}	7.11	1.59×10^5	16
Adjoint score (scored with AMS)	2.66×10^{-15}	9.98	1.08×10^5	13
Exponential Transform				
INIPOND (Manual mode)	2.55×10^{-15}	6.51	9.41×10^4	39
IDT (cross sections from TRIPOLI4)	2.04×10^{-15}	6.60	2.39×10^5	23
IDT (cross sections from APOLLO3)	2.81×10^{-15}	0.82	3.27×10^3	576
Adjoint score (scored with AMS)	2.77×10^{-15}	0.52	4.33×10^3	111313

Table 6.1. Results for the strong attenuation problem. We present the integrated mean response in the detector, as well as the estimate of its standard error after a certain simulation time. The figures of merit can be compared for both the AMS and the ET method used with the four importance maps : INIPOND in manual mode, IDT with cross-sections condensed with TRIPOLI-4®, IDT with cross sections from APOLLO3® with anisotropy order 5 and the adjoint response scored with AMS.

and the sought response is a gamma dose. For the sake of simplicity, however, we focus on the flux.

First step: scoring the adjoint flux

We discretise the problem into six energy groups for both neutrons and gammas: [20 MeV, 1 MeV, 100 keV, 10 keV, 1 keV, 0.625 eV, 1E−5 eV]. Photons have an energy cut at 0.1 keV. The adjoint score is collected during an AMS simulation with a geometrical importance for both neutrons and gammas. This geometrical function is the inverse of the spatial distance between the center of the detector and the particle position. Figure 6.4 shows the adjoint score for neutrons and photons for the four highest energy groups of each species.

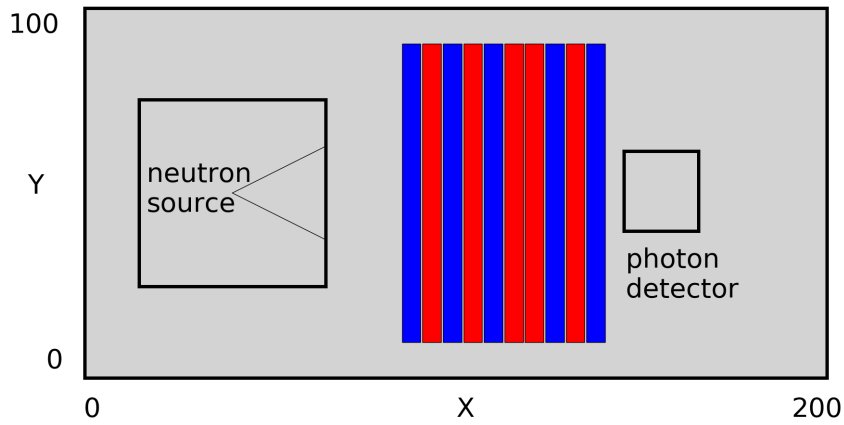


Figure 6.3. Geometry of the coupled neutron gamma problem.

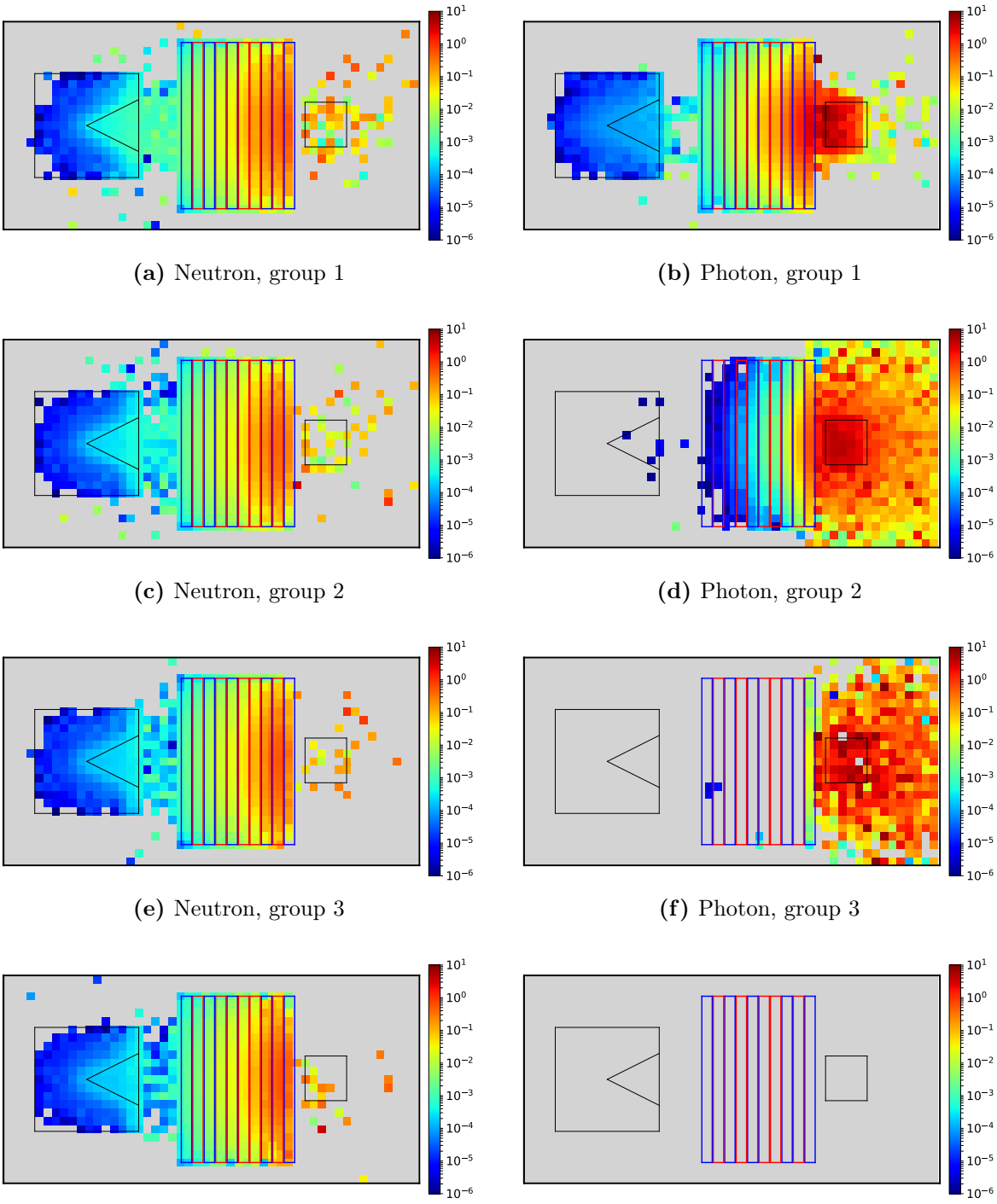
Second step: using the adjoint flux as an importance map

In order to test the scored importance map in this setting, we perform four simulations. Table 6.2 collects the results for these simulations. The first one is an analog simulation. It is used as a reference for the FOM calculations. Two simulations are performed with AMS, one with a spatial importance map (the inverse of the spatial distance between the center of the detector and the particle position) , and one with the adjoint score. Finally, a simulation is run with ET and the adjoint score. ET is not run with geometrical functions because it mainly looks at the gradient of the importance function. If this is set arbitrarily, ET would either split or roulette as much as it can.

Both AMS simulations converge to the same target value. The spatial importance map performs 30 times faster than the analog simulation, whereas the adjoint score makes AMS converge 130 times faster than the analog simulation. We consider this to be a strong acceleration compared to what we can currently achieve with INIPOND (Louvin, 2017). ET seems to slightly underestimate the final result.

6.1.3 Conclusions

In this section, we showed that the adjoint score can indeed accelerate AMS or ET. For the strong attenuation problem, the adjoint score yields acceleration factors comparable to the ones produced by the IDT coupling. In spite of the low convergence of the adjoint score in some rarely explored regions (see Figure 6.4), this adjoint flux is still a good alternative because both ET and AMS converge when it is taken as an importance map. However, this scheme requires the user to fix the simulation duration for the first phase. This could be tricky as it is hard to tell when the convergence is adequate for the score



(a) Neutron, group 1

(b) Photon, group 1

(c) Neutron, group 2

(d) Photon, group 2

(e) Neutron, group 3

(f) Photon, group 3

(g) Neutron, group 4

(h) Photon, group 4

Figure 6.4. Adjoint score for the neutron-gamma coupled benchmark.

Method	Map	Mean	$\sigma(\%)$	time(s)	$\frac{\text{FOM}}{\text{FOM}_{Analog}}$
Analog	-	3.74×10^4	2.55	4.95×10^4	1.00
AMS	spatial	3.78×10^4	0.42	6.36×10^4	28.52
	adjoint score	3.77×10^4	0.197	6.36×10^4	130.28
ET	adjoint score	3.65×10^4	0.308	6.36×10^4	53.23

Table 6.2. FOM comparisons for the coupled neutron-gamma setup. We report the mean and standard error for the sought response (gamma flux in a cylinder), as well as the CPU time, for four different calculation methods. FOMs are normalised to the analog calculation.

to be re-used as an importance map. Also, FOMs do not include the CPU time for the first pass. Scoring the adjoint flux has a cost that should be minimised.

6.2 Fixed-step alternating scheme

We propose a strategy that periodically assesses the efficiency of the current adjoint score. Following the principles of reinforcement learning, we alternate between fixed-length phases of exploration and fixed-length phases of exploitation. During the exploration phase, the adjoint score is collected as in the previous strategy; during the exploitation phase, an *instantaneous* FOM_{inst} is estimated. This FOM_{inst} is estimated by measuring the execution time of the exploitation phase $T_{explore}$ and estimating the variance from the spread among the batches that form the exploitation phase ($\sigma^2_{explore}$).

$$FOM_{inst} = \frac{1}{\sigma^2_{explore} T_{explore}} \tag{6.1}$$

The aim of this strategy is to give periodic updates on the state of convergence of the importance map. Two indicators will be looked at: the actual value of the response score and the instantaneous figure of merit. This scheme is illustrated in Figure 6.5.

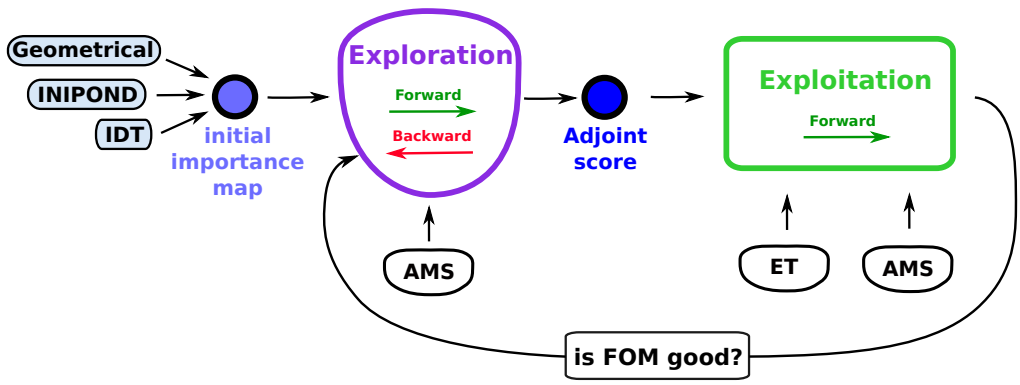


Figure 6.5. Fixed-step alternating adjoint response update scheme.

6.2.1 Strong attenuation

We illustrate the alternating scheme on the strong attenuation benchmark.

Initial importance map: IDT with cross sections produced by APOLLO3®

We start by looking at the behaviour of the scheme when the initial importance map is already a good guess of the adjoint flux. We choose to run 2000 batches with the following parameters:

- **Exploration phase**: 5 batches of AMS with IDT and cross sections produced by APOLLO3®.
- **Exploitation phase**: 5 batches of ET with the importance map being updated from the exploration phase.

This means that we have simulated 400 phases (200 of exploration and 200 of exploitation).

Figure 6.6a shows the evolution of the response estimate in each batch. Here and in the following, the exploration phase will be assigned the violet color, whereas the exploitation phase will be assigned the green one. We observe that the adjoint score rapidly yields an importance map that enables the ET to produce correct estimates during the exploitation phase. Moreover, the behaviour of the exploitation phases seems quite stable over the iterations, which suggests that exploration could be stopped early without jeopardizing the convergence.

Figure 6.6b shows the evolution of the instantaneous Figure of Merit against the number of simulated phases. We observe that the FOM of the ET converges to a higher value than AMS. This is in agreement with results of Table 6.1.

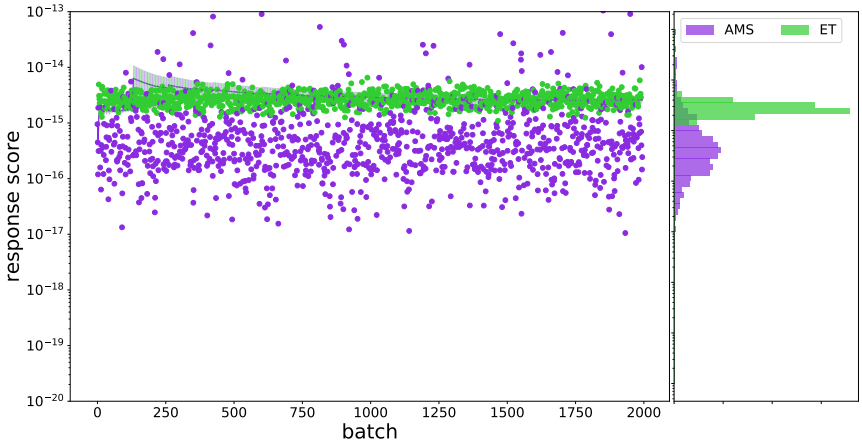
One may object that the importance map used for the exploration phase (IDT with cross sections from APOLLO3®) is already very close to the optimal one. In this scenario, there would be no real need to score another importance map to use with ET. Nevertheless, it is remarkable that AMS is able to transfer the nub of the importance map from the exploration to the exploitation phase in such a short time. We now study a setting where the initial importance map lacks vital information on the energy, i.e. an geometrical importance map.

Initial importance map: geometrical

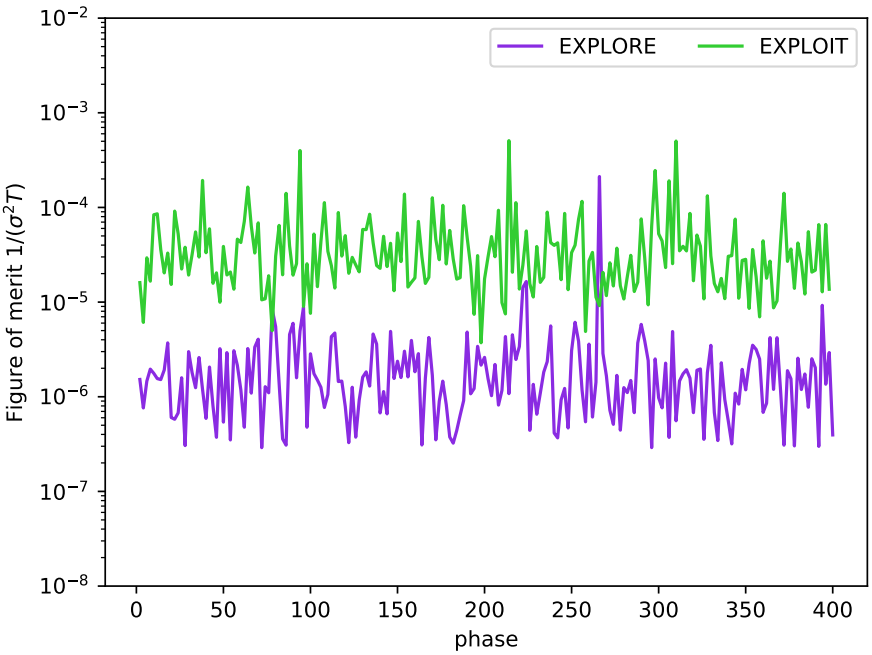
We perform the same simulation scheme, except that the initial importance map is now taken as purely geometric. Namely, the inverse of the distance to the detector. We choose to run 2000 batches with the following parameters:

- **Exploration phase**: 5 batches of AMS with a geometrical importance map
- **Exploitation phase**: 5 batches of ET with the importance map being updated from the exploration phase.

Again, this means that we have simulated 400 phases (200 of exploration and 200 of exploitation).



(a) Evolution of the response as a function of the number of simulated batches. The exploitation phase converges very quickly. This means that AMS recovered the information from the initial importance map to the adjoint score during the exploration phase very quickly.



(b) Evolution of the figure of merit as a function of the phase. The figure of merit of the exploitation phase converges to a higher value than the exploration phase.

Figure 6.6. Fixed-step alternating scheme for the strong attenuation benchmark. Each exploration phase lasts 5 batches and is performed with AMS running with the importance map computed with the cross sections coming from APOLLO3®. Each exploitation phase lasts five batches and is performed with ET.

Figure 6.7a shows the evolution of the response score at each batch. We observe that the AMS spread during the exploration phase is much larger than what it was with IDT (cross sections from APOLLO3®) as an importance map (compare Figure 6.7a with Figure 6.6a). We also observe that the exploitation phase undergoes an ascendant convergence of the response score, starting from batches with zero contributions, then converging to a plateau approximately at batch 800.

Figure 6.7b shows the evolution of the estimate of the instantaneous Figure of Merit against the number of phases. We observe that the FOM of the exploration phase is stable whereas the exploitation phase's FOM converges after 200 phases, which corresponds to 1000 batches. This returns to wait 5 h on a single processor(Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz, 64 Go RAM). However, this takes into account the update time of the importance map at the end of each exploration phase. It is still appreciable compared to the time it takes for the deterministic solutions to be found. IDT (cross sections from TRIPOLI-4®) converged in 17 min, whereas IDT (cross sections from APOLLO3®) (in 25 min). But the time for the cross sections condensation (for IDT with cross sections produced by APOLLO3®) in the direct calculation is not taken into account here. It should logically double.

6.2.2 Neutron Gamma

We perform the same study with the coupled neutron-gamma problem. We start with a spatial importance: because it is hard to have a good a priori approximation of the importance, we decide to initialise the problem with a simple geometrical importance map.

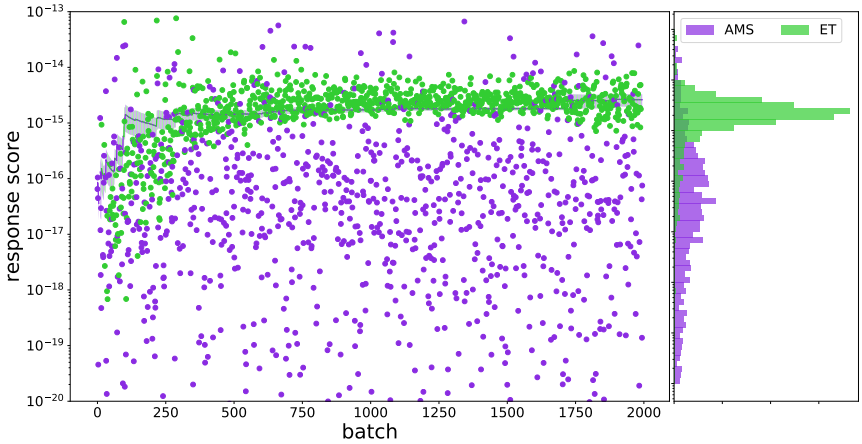
Explore with AMS and exploit with AMS

Let us start looking at how AMS reacts to its own scored importance map. We choose to run 5000 batches with the following parameters:

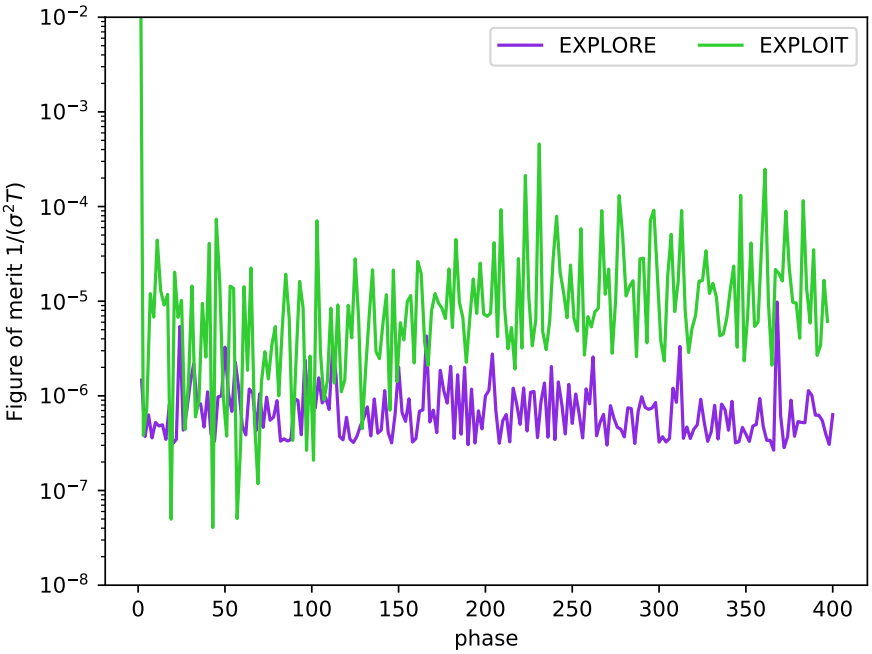
- **Exploration phase**: 10 batches of AMS with a geometrical importance map.
- **Exploitation phase**: 10 batches of AMS with the importance map being updated from the exploration phase.

This means that we have simulated 500 phases (250 of exploration and 250 of exploitation). Figure 6.8a shows the evolution of the response as a function of the number of batches. We observe that the responses estimated with the exploiting phase start with the same spread as the exploration phase, but gradually, converge to a smaller spread after approximately 2500 batches. This corresponds to 250 exploration phases.

Figure 6.8b shows the evolution of the instantaneous FOM as a function of the phase number. We observe that the FOM of the exploration varies around a constant value, whereas the FOM of the exploitation phase starts at the same value, but increases before stabilizing around phase 150. At this point, we consider that the importance map scored by the exploration phase has converged *in the sense of AMS*. The ratio between the FOM of the final state and the FOM of the exploration phase is 2.

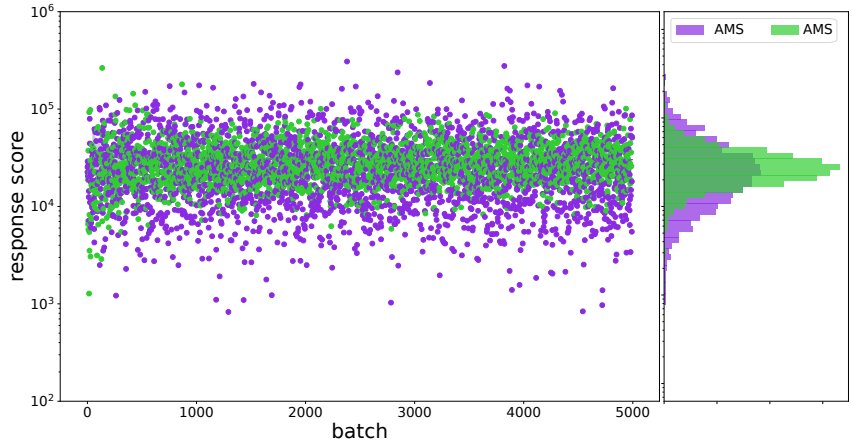


(a) Evolution of the response as a function of the number of simulated batches.The exploitation phase yields a very large variance of the detector response. The exploitation phase converges after around 1000 batches.

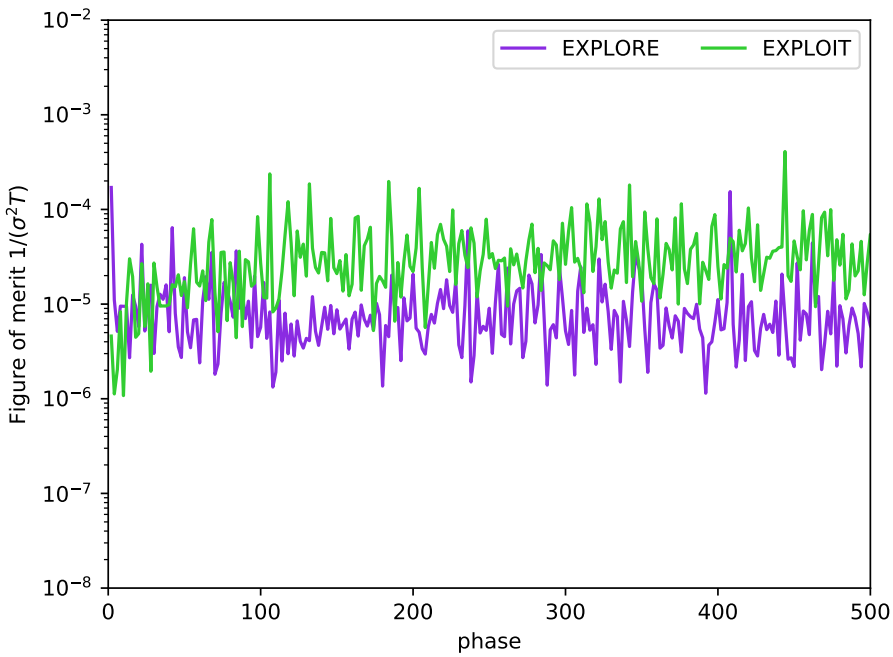


(b) Evolution of the figure of merit as a function of the number of phases. The figure of merit of the exploitation phase increases as the exploration phase gathers more information on the adjoint score. It converges after around 200 phases.

Figure 6.7. Fixed-step alternating scheme for the strong attenuation benchmark. Each exploration phase lasts 5 batches and is performed with AMS running with geometrical importance map (inverse of distance to detector). The exploitation phase lasts five batches and is performed with ET.



(a) Evolution of the response as a function of the number of simulated batches. The exploitation phase starts with values of the response spreaded with the same variance of the exploration phase. After a while, this spread decreases and converges to a lower level.



(b) Evolution of the figure of merit as a function of the phase. The figure of merit of the exploitation converges towards an asymptotical value that is higher than the one reached by the exploration phase.

Figure 6.8. Fixed-step alternating scheme for the neutron-gamma benchmark. Each exploration phase lasts 10 batches and is performed with AMS running with a geometrical importance map(inverse of distance to detector). The exploitation phase lasts 10 batches and is performed with AMS.

Explore with AMS and exploit with ET

What happens now when ET is used for the exploitation phase?
We choose to run more than 30000 batches with the following parameters:

- **Exploration phase**: 10 batches of AMS with a geometrical importance map.
- **Exploitation phase**: 10 batches of ET with the importance map being updated from the exploration phase.

This means that we have simulated more than 3000 phases (1500 of exploration and 1500 of exploitation).

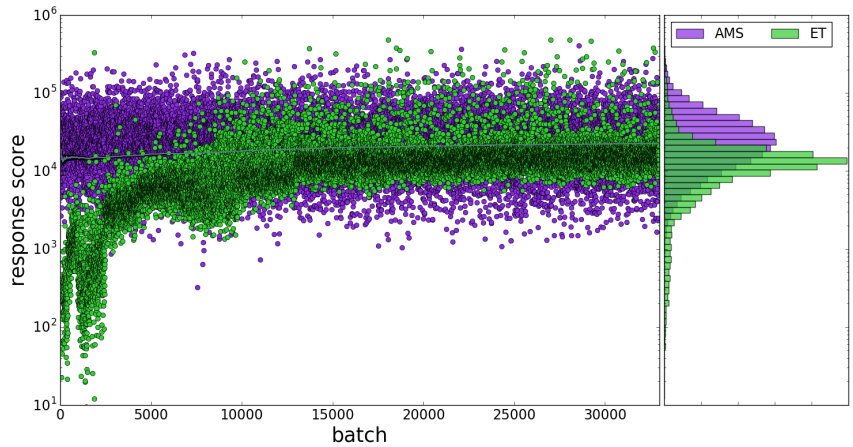
Figure 6.9a shows the evolution of the response score as a function of the batch numbers. We observe that the exploiting phase really struggles to converge, and eventually converges to an underestimated value. Due to the response underestimation, the plot of the figure of merit in Figure 6.9b is not relevant. It is surprising that what was a sufficiently well converged importance map for AMS is not for ET. One explanation may be that the energetic grid on which the adjoint score is computed is too coarse. Also, this could simply come from the fact that ET needs the gradient to be computed properly. However because the gradient of the importance is computed with finite differences with the neighbouring cells, statistically poorly converged regions will produce highly varying gradients.

6.2.3 Conclusions

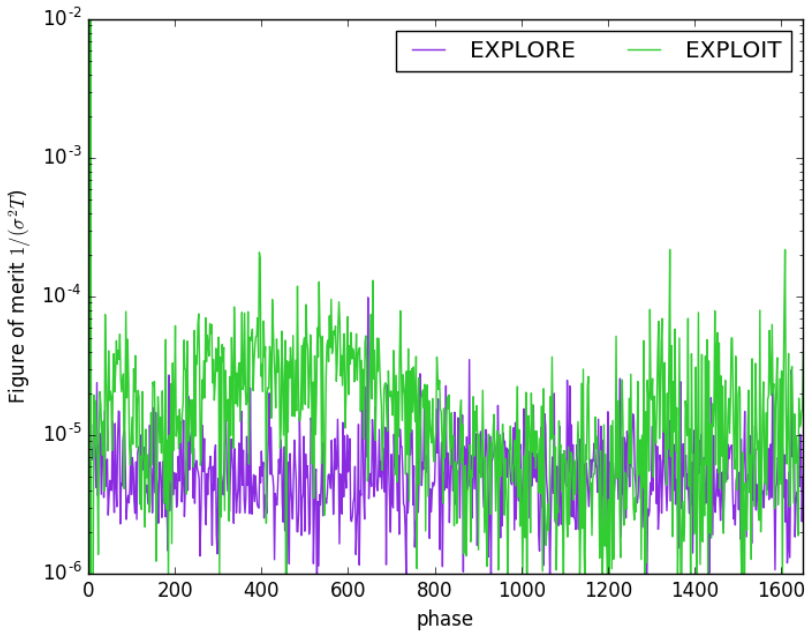
In this section, we explored a strategy that alternates between exploration phases where AMS scores the adjoint flux and exploitation phases where AMS, or ET re-uses it as an importance map. We saw that AMS can transfer the information contained in the initial importance map to the adjoint score very quickly when starting from a well converged initial importance map. When the importance map was only geometrical, AMS took more time to converge, typically a few hundreds of batches. This convergence is also visible on the plots of the instantaneous figure of merit: the exploiting FOM starts from the exploration FOM and gradually increases till converging to a higher value. Also, we saw cases where the exploitation response did not even converge to the exploration response and in such cases the FOM was irrelevant.

6.3 Bootstrap scheme

The previous scheme has shown its limitations. We would like to accelerate the rate of convergence of the adjoint score. For that, we try to make the exploration phase re-use its own adjoint score as an importance map. In this situation, we start with an exploration phase where AMS scores the adjoint flux, then the importance map is updated and the first exploitation phase runs. After this, the same importance map (accumulated over all exploration phases) is used for the next exploration phase. This means that if AMS has not succeeded to reach the detector and to converge the adjoint score satisfyingly, the simulation might stop immediately because ***the initial importance map is completely overridden***. We call this strategy the *bootstrap* scheme, and it is illustrated in Figure 6.10.



(a) Evolution of the response as a function of the number of simulated batches. Each exploitation phase converges very slowly to an asymptotical level that is lower than the one achieved by the exploration phase.



(b) Evolution of the figure of merit as a function of the number of simulated phases. Because the response is not consistent with the exploration phase, this should not be given too much credit.

Figure 6.9. Fixed-step alternating scheme for the neutron-gamma benchmark. Each exploration phase lasts 10 batches and is performed with AMS running with a geometrical importance map(inverse of the distance to the detector). The exploitation phase lasts 10 batches and is performed with ET.

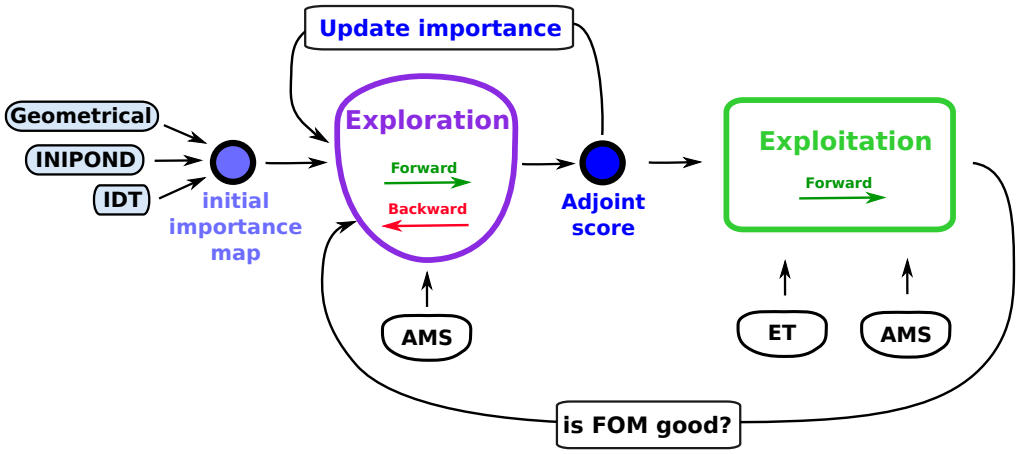


Figure 6.10. Bootstrap scheme

A few questions immediately come to mind: What happens if the importance map is updated too early? Can the exploration phase rely on its own adjoint score? If this succeeds, the exploration phase will use an importance map close to the initial guess. So what if a variance jump occurs? is it taken into account in the adjoint score ? and if so, how will AMS react to such an update in the importance map?

6.3.1 Strong attenuation

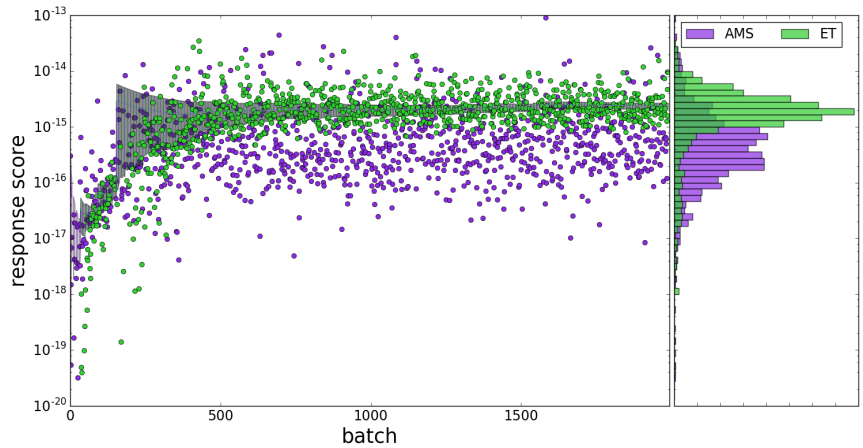
Let us start by looking at the strong attenuation benchmark. We choose to run 2000 batches with the following parameters:

- **First exploration phase**: 5 batches of AMS with a geometrical importance map.
- **Exploitation phase**: 5 batches of ET with the importance map being updated from the exploration phase.
- **updated exploration**: 5 batches of AMS with the importance map being updated from the exploration phase.

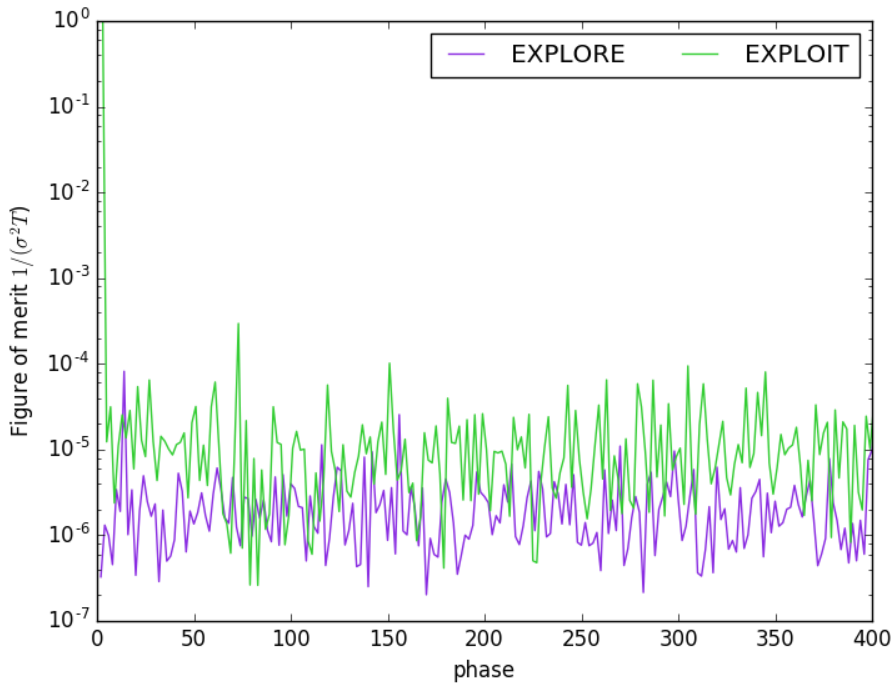
This means that we have simulated 400 phases (200 of exploration and 200 of exploitation).

Figure 6.11a shows the evolution of the mean response as a function of the number of batches simulated. We remark that the spread of AMS is much smaller than for the simple alternation scheme Figure 6.7a. This means that AMS is already getting return on what it scored so far. Let us have a look at the exploitation phase. It converges after 1000 batches just as the previous scheme.

Figure 6.11b shows the evolution of the instantaneous FOM with respect to the number of simulated phases. We observe that the FOM converges faster than the alternate scheme(Figure 6.7b) to its asymptotic value. Although the FOM should not be given any significance before the mean value has converged, it still provides information on the variance of the instantaneous spread of the responses during one phase. During the transitory phase, ET is “dragged up” by AMS with lower variance of variance than the alternating scheme.



(a) Evolution of the response as a function of the number of simulated batches. Each exploitation phase is dragged up by the exploration phase. Both phases converge with a variance lower than in the previous scheme.



(b) Evolution of the figure of merit as a function of the number of phases simulated. The FOM of the exploitation phase converges quickly to its asymptotical value.

Figure 6.11. Bootstrap scheme for strong attenuation benchmark. Each exploration phase lasts 5 batches and is performed with AMS running with a geometrical importance map (inverse of the distance to the detector). The exploitation phase lasts 5 batches and is performed with ET. After the second phase, the exploration phase starts using the scored importance map.

6.3.2 Neutron Gamma

How does the bootstrap scheme apply with the coupled neutron-gamma problem? We choose to run more than 30000 batches with the following parameters:

- **First exploration phase**: 10 batches of AMS with a geometrical importance map.
- **Exploitation phase**: 10 batches of ET with the importance map being updated from the exploration phase.
- **updated exploration**: 10 batches of AMS with the importance map being updated from the exploration phase.

This means that we have more than 3000 phases (1500 of exploration and 1500 of exploitation).

Figure 6.12a shows the evolution of the response score when applying the bootstrap scheme. We remark that the simulation does not stop after the first exploration phase. We can thus infer that the information retrieved by the first exploration phase was sufficient for the next exploration phase to converge. We see that as in the previous case, the spread of the response value is lower than the one obtained with the alternating scheme (see Figure 6.7).

Moreover, the ET seems to converge *much* faster to the correct value than the alternating scheme. For this problem, we saw that AMS performs better than ET with the two-step scheme (see Table 6.2). The bootstrap strategy does help ET converge better than the alternating one.

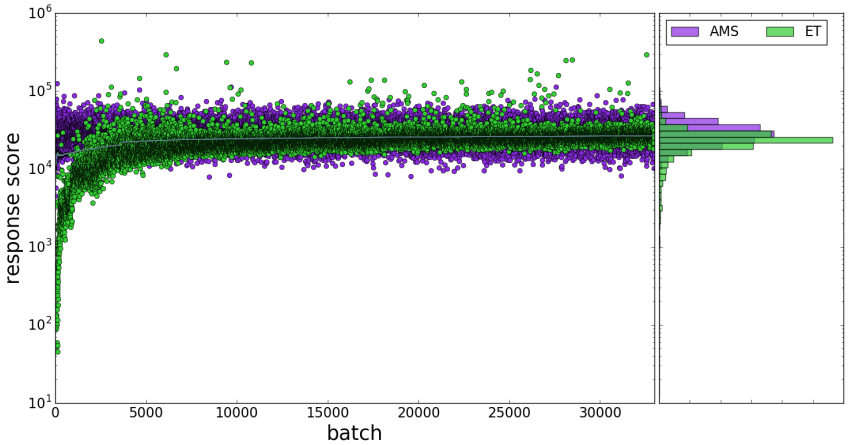
6.3.3 Capturing a variance jump?

Now that we have tested two strategies to monitor the convergence of the adjoint flux, we would like to answer this question: If a rare event occurs during the simulation, and if this event highly contributes to the detector response, would the *alternating* scheme, and the *bootstrap* scheme be able to update the importance map so that at the next iteration, the rate of occurrence of this rare event increases?

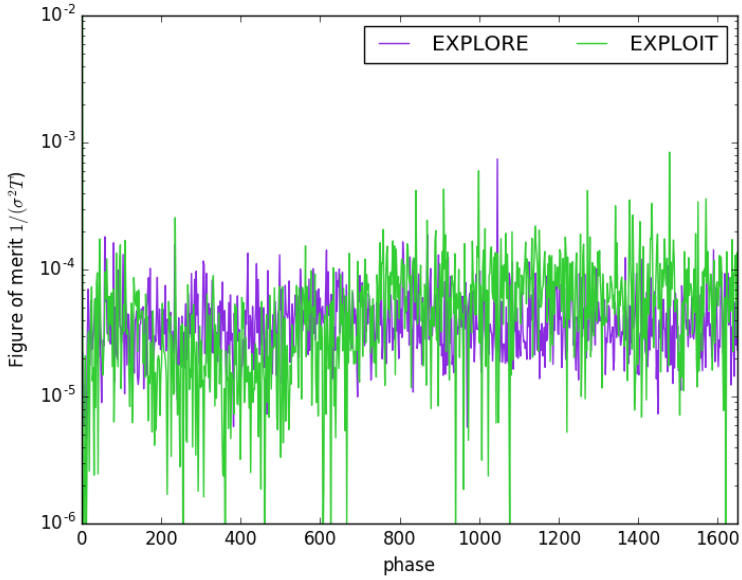
In order to answer this questions, we design a simple, yet illustrative 2D benchmark (see [Appendix A.7]) where variance jumps are easily triggered. The bypass consists in a 80 cm × 80 cm tank filled with helium. In the middle stands a 60 cm × 60 cm block of concrete. A point neutron source is placed in the bottom-left corner, and the detector in which we compute the total neutron flux is placed in the top-right corner. Figure 6.13 is a description of the problem. Leakage boundary conditions are imposed.

In order to provide some intuition on the expected particle trajectories, an IDT simulation was run for one energy group on a 100×100 uniform spatial mesh. The resulting adjoint flux is shown in Figure 6.14a. We see that the main contributions come from bypassing the concrete block through helium.

In order to trigger variance jumps, an ill-designed importance map that leads particles through the concrete block was hypothesized. This importance map is shown in Figure 6.14b



(a) Evolution of the response as a function of the number of simulated batches. Each exploitation phase converges quickly to its asymptotic value



(b) Evolution of the figure of merit as a function of phases.

Figure 6.12. Bootstrap scheme for coupled neutron-gamma problem. Each exploration phase lasts 5 batches and is performed with AMS starting with a geometrical importance map (inverse of the distance to the detector) and continuing with the scored importance map. The exploitation phase lasts 5 batches and is performed with ET.

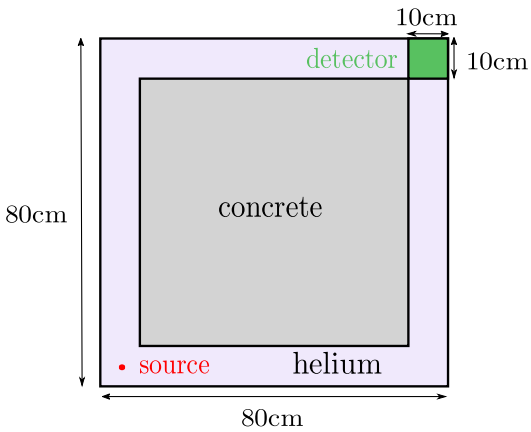
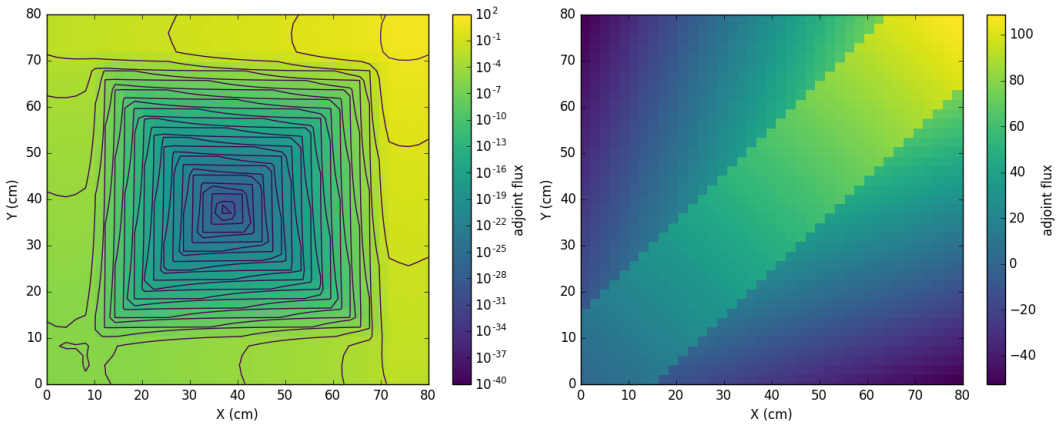


Figure 6.13. Geometry of the theoretical bypass problem.



(a) IDT **(b)** Fake

Figure 6.14. Importance maps for the bypass benchmark.

AMS is run with $N = 100$ particle trajectories per batch. Two strategies are tested and their results are gathered in Figure 6.15: the fixed-step alternating strategy and the adaptive bootstrap one.

The evolution of the response score for the fixed-step alternating strategy of Section 6.2 is given in Figure 6.15a. The simulation alternates between:

- **Exploration phase**: 10 batches with AMS and the initial ill-designed importance map.
- **Exploitation phase**: 10 batches with AMS with the importance map being updated from the exploring phase.

During the exploration phase, variance jumps occur with a frequency of approximately 150 batches, with the first one occurring at batch 120. We observe that the variance jump affects the importance map because the exploitation phases converge to higher values after the jump.

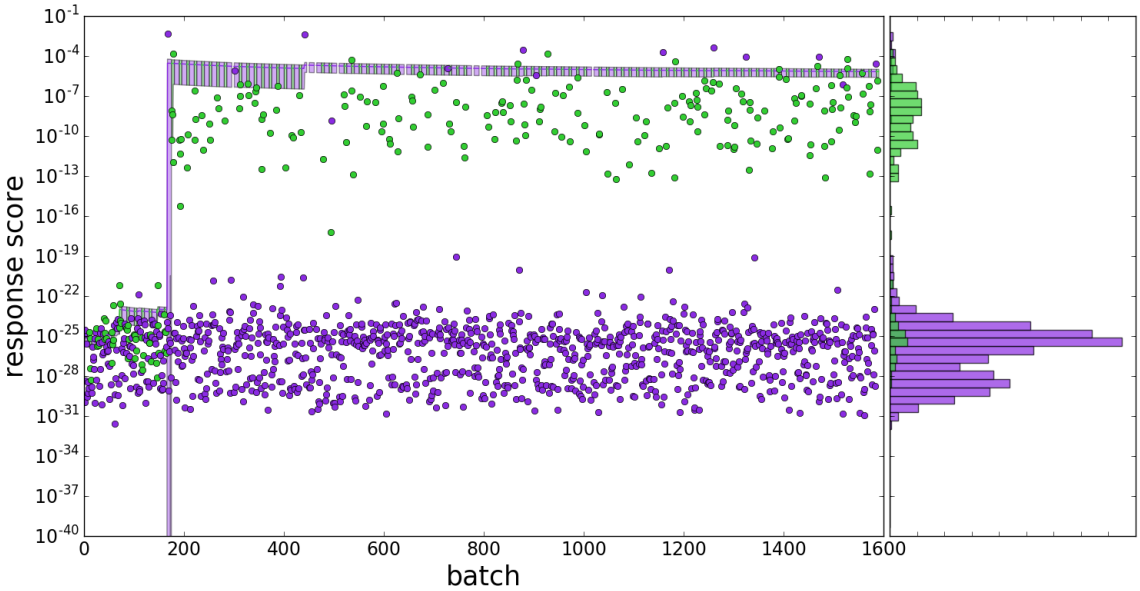
Let us now apply the bootstrap strategy to this problem. The simulation now alternates between:

- **First exploration phase**: 10 batches with AMS and the initial ill-designed importance map.
- **Exploitation phase**: 10 batches with AMS with the importance map being updated from the exploring phase.
- **update exploration phase**: 10 batches with AMS with the importance map being updated from the exploring phase.

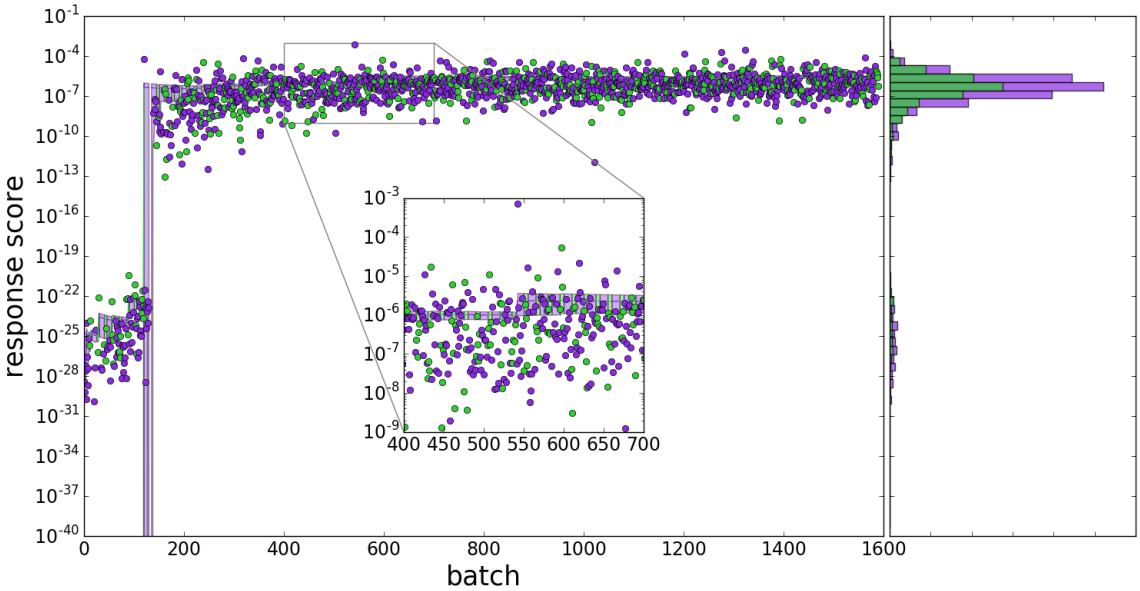
Figure 6.15b shows the response score in function of the number of simulated batches. We remark that the variance jump is taken into account and the exploration phase immediately profits from this gain of information. A second variance jump occurs as shown in the inset of Figure 6.15b.

So how can a single AMS batch, where highly contributing trajectories were sampled, dominate over the first 150 batches where poor trajectories were sampled? Let us answer this question by cutting the simulation into pieces and illustrate the main steps that lead to such convergence (see Figure 6.16).

The importance map is initialised with the ill-designed path importance from Figure 6.14b. The first trajectories sampled with AMS are shown in Figure 6.16a. We observe that trajectories take the path through the concrete as expected. After 115 batches, the importance map has been updated to something that essentially carries the same information as the initial importance map, as shown in Figure 6.16b (i.e. particles are pushed through the concrete). With this importance map, a trajectory finally succeeds to bypass the concrete block through the right bottom-right path at batch 120 (Figure 6.16c). This trajectory triggers a drastic revision of the adjoint score which is updated as shown in Figure 6.16d. This event corresponds to the first variance jump of Figure 6.15b.



(a) Fixed-step alternating scheme convergence for the bypass problem.



(b) Bootstrap scheme convergence for the bypass problem.

Figure 6.15. Convergence of the bypass problem with the fixed-step alternating strategy and the bootstrap strategy

We observe that the cells populated by this trajectory are very sparse. However, this suffices for trajectories to bypass the concrete block after very few phase iterations. The next trajectory bypassing the concrete block thus occurs at batch 138 (Figure 6.16e). We observe densely populated regions that overlap the cells where the adjoint score was updated by the first bypassing trajectory. After 385 batches Figure 6.16f shows that the importance maps is now well converged over the whole path through the bottom-right duct. After 542 batches (Figure 6.16g), a trajectory is sampled through the left bypass duct, thus leading to an update in the scored importance map (Figure 6.16h). This event corresponds to the second variance jump, depicted in the inset of figure Figure 6.15b

Phases after phases, trajectories bypassing through the left channel are sampled more frequently (Figure 6.16i) and eventually lead to a fully converged importance map (Figure 6.16j).

6.3.4 Conclusions

We have seen that updating the importance map for the exploration phase can indeed accelerate the convergence of the adjoint score. The coupled neutron-gamma problem converged much faster than with the simple alternating strategy.

The theoretical bypass model was very useful to illustrate the dynamics of the update policy. We saw that the loops manage to reconstruct a reasonable importance map even if the initial one was inadequate. Of course, the positive outcome is surely facilitated by the relative simplicity of the problem. It is conceivable that the bootstrap scheme of Section 6.3 may not manage to escape from its initial, "metastable" state in harder problems. Nevertheless, it is remarkable that AMS is capable of recovering the correct importance map given very little initial information.

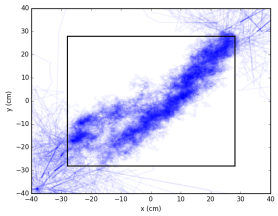
6.4 A few words on the parallelisation of these strategies.

We wish to spend a few words on the potential parallelisation of the strategies describes in Sections 6.1-6.3. The two-step strategy (Section 6.1) is straightforward to parallelize. We first mobilize the maximum resources for the first phase, and then for the second. There is a single meeting point at the end of the first pass, which is unlikely to have a sizeable negative impact on performance.

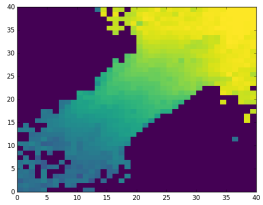
Concerning the other two, we have at least two obvious choices for the parallelisation scheme.

process-based This parallelisation scheme involves having some processes dedicated to exploration and others dedicated to exploitation. The difficulty of this scheme lies in the communication that must happen between the processes. Clearly, the explorer processes must send importance map updates to the exploiting processes and perhaps to the other exploring processes. This looks too difficult.

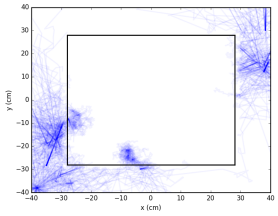
phase-based This parallelisation scheme involves having all processes starting with the exploration phase. At the end of this global phase, processes meet at a point where the importance map is collected from all calculations. Then, it is scattered back



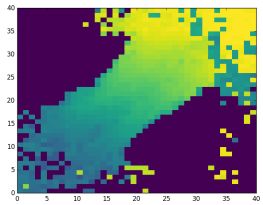
(a) Trajectories at batch 1



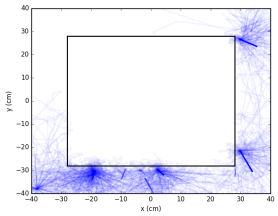
(b) Importance map at batch 115



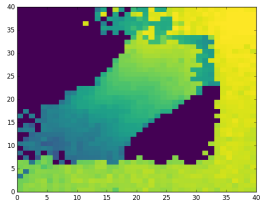
(c) Trajectories at batch 120: a particle takes the right path



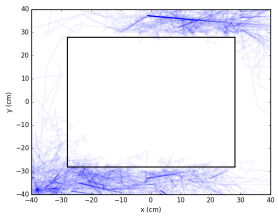
(d) Importance map at batch 130: batch 120 is taken into account



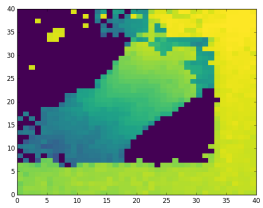
(e) Trajectories at batch 138, particles bypass on the right very easily



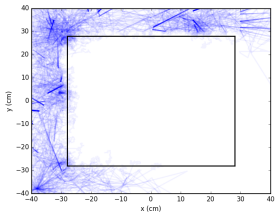
(f) Importance map at batch 385, the right path is well converged



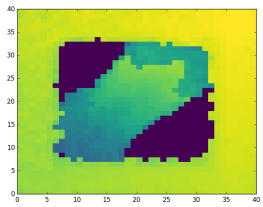
(g) Trajectories at batch 542, particles go through the left duct



(h) Importance map at batch 550, the left trajectories are present in the importance map



(i) Trajectories at batch 1506



(j) Importance map at batch 1990, converged

Figure 6.16. Illustration of the convergence of the bypass problem with the bootstrap scheme. Figures a,c,e,g and i illustrate the trajectories sampled during a given batch. Alternatively, figures b,d,f,h and j show the state of the importance map at a given batch.

to all the processes that can start the exploitation phase. This naive strategy will probably yield unacceptably long blocking waiting times before collecting all the importance maps.

6.5 Conclusions

In this chapter we have seen three strategies to re-integrate the adjoint score into the forward calculation. The first scheme (Section 6.1) is a simple separation of the two phases. The user needs to decide what is a "reasonable" convergence for the importance map. The second strategy (Section 6.2) is inspired by reinforcement learning principles and assesses the convergence of the adjoint score by periodically running an exploitation phase during which an estimation of the FOM is computed. This allowed to show that the adjoint score convergence speed depends on the importance map used for the exploration phase. Finally, a third strategy (Section 6.3) consists in updating the importance map with the adjoint score for the exploration phase as well. For a theoretical problem and a realistic configuration, we saw that this scheme successfully recovers the most important trajectories faster than the fixed-step alternating scheme.

Overall, we could ask ourselves the question: should the importance map and the source used for the exploration phase be different from the one used for the exploitation phase? J. Hoogenboom and Légrády (2005) ask this question of the WWG of MCNP. They speculate on setting the source of the simulation with the WWG different from the source of the real problem we are trying to solve. K.W. Burn (1990) uses the term "spray" to show that particles are emitted at each surface in order to accumulate sufficient statistics of the splitting parameters. We gave a partial answer by looking at the convergence of the adjoint score with different fixed importance maps for the exploration phase (Section 6.1). We think that trajectories sampled during the exploration phase should be close to the ones sampled in the exploitation phase (independently of the variance reduction method).

Chapter 7

Conclusion and Perspectives

Contents

7.1 Findings 116
7.2 Perspectives 117

7.1 Findings

The initial goal of this work was to accelerate Monte Carlo shielding simulations by exploiting the information contained in *particle trajectories*. More precisely, the ambition was to develop a new tool to continuously improve the importance map and accelerate the convergence of the target score.

At takeoff, this work was driven by the existence of a zero-variance scheme for *importance sampling* methods. Because this scheme relies on the solution to the adjoint Boltzmann transport equation, the efforts naturally centered on the estimation of the adjoint flux during the Monte Carlo simulation itself.

This first drove us to implement a *score* on a basic mesh: Particle trajectories are gathered during the simulation and rewinded from their death to their initial emission at the source. Along their path, the contribution of the history to the target response are accumulated. Of course, in most shielding cases, this contribution would have vanished without variance reduction. Concerning the choice of the variance reduction technique, Adaptive Multilevel Splitting was the natural option as it ensured to bring all the particles to the detector.

In order to verify this score, the generation of the adjoint fluxes was automatised thanks to a deterministic code. This resulted in the implementation of the CADIS methodology in TRIPOLI-4® and allowed to quickly test the implementation on various problem types without the need to write input files for the deterministic simulation from scratch. Tests showed that the ET can be greatly accelerated by deterministic solutions of the adjoint Boltzmann equation. However, ray effects were observed, and cross sections were only treated up to a P1 transport correction.

Once the score was verified on simplified multi-group problems, We re-used it as an importance function in Monte Carlo simulations and provided it as input to ET and AMS for several benchmarks. It gave us an overview of how the VR methods behave with the adjoint flux. Gains in the FOM were similar to those found with the deterministic importance map.

It also confirmed that the robustness of AMS comes with a downside: its efficiency is somehow bounded by the space of the importance maps and the best accelerations achieved with AMS are structurally lower than with ET. On the other hand an adjoint score generated with the AMS did produce compelling accelerations with the ET. This reinforced the common belief that if the importance map is *close* to the real adjoint flux, then the simulation is still greatly accelerated. The principal disadvantage is that the time required before reaching this level of convergence could be arbitrarily long and that its diagnostic requires a careful monitoring of the simulation.

We concluded that obtaining information from the simulation was not sufficient and asked the question: is it possible to let the simulation decide when it should stop scoring the adjoint response?

Reinforcement learning algorithms, and especially the balance between *exploration* and *exploitation* invited us to make these terms appear explicitly in Monte Carlo shielding simulations. We started alternating *phases* of exploration where we would score the adjoint

flux, and phases of exploitation where ET (or AMS) would use the current state of the adjoint flux as an importance map. This was the first step towards defining an adaptive Monte Carlo strategy for the simulation of shielding problems. We tested this strategy on two benchmarks, and the coupled neutron-gamma benchmark was clearly converging at a slow speed. The exploration phase could benefit from exploiting what had already been learnt.

Hence, a *bootstrap* scheme was developed, which entirely replaces the initial importance map with the current state of the adjoint flux and it improved the convergence speed. We tried to answer the question: what happens when a rare event appears in the simulation? How will this information affect the scored importance map/adjoint flux? We designed a very simple benchmark for which variance jumps are easily triggered and showed that the adaptive scheme can update the adjoint flux when a rare event occurs. This was of course done for a theoretical model. The question remains whether this would work for a problem where energy plays a more important role. Also, the length of the phases is set arbitrarily by the user. This could lead the bootstrap scheme to fail if no particle reaches the detector during the first exploration phase.

Finally, one question was left unanswered: for the adjoint score to be a good candidate for being an importance map, it had to be well discretised. It is the responsibility of the Monte Carlo users to define the discretisation, even when they have *a priori* very little information on the system. The idea was to propose an alternative *support* for the adjoint score.

A cursory inspection of smoothing algorithms promptly revealed that the conversion of particle trajectories to a training dataset is a non-trivial task. Particular attention must be attached to the actual distribution (density) of the points generated during the Monte Carlo simulation. Moreover, because the number of points that contribute to the estimation of the adjoint flux is extremely high, the prediction of off-line algorithms must be projected on a histogram in order for the off-line algorithms to be even considered. The only on-line method that we tried was neural networks. Their convergence and tuning was somewhat erratic. We finally chose to use robust Kernel Density Estimators to smooth the importance map. On the strong attenuation benchmark, the KDE score converged to a smoothed, but biased shaped which accelerated the speed of convergence of ET, but lowered its asymptotic variance compared to the score on a mesh.

7.2 Perspectives

1. The adjoint score is implemented for AMS, and is robust. It could be interesting to look at extending the adjoint score to ET and compare it to the AMS score.

2. Concerning the bootstrap scheme: so far, the adapted strategy has been to completely overwrite the initial importance map with the scored one, but we could think instead of ways to merge the initial importance map with the scored one. A strategy with more than one importance map in memory would be safe because one could still go back to a previous state of the importance map if the simulation decides that the current one yields bad results.

3. The absence of collision biasing in TRIPOLI-4® clearly widens the gap between

the zero variance trajectory and what TRIPOLI-4® can do. TRIPOLI-4® only implements a sampling-rejection technique. One should consider putting more effort into a proper implementation to see if this can accelerate shielding problems where ET fails.

4. Another perspective is to learn from the simulation in another way than only looking at the importance map. For example by using reinforcement learning strategies to optimise some parameters of the simulation across the phase space. An idea would be to super-impose a mesh on the phase space, learn the value of the action: "activate" or "deactivate" the collision biasing sampling-rejection method of TRIPOLI-4® making decisions on a cell-by-cell basis.

5. In most of the benchmarks, we saw that AMS asymptotically converged to figure of merits lower than what the exponential transform can do. We could imagine a strategy where AMS is only supposed to compute an importance map in support for the exponential transform. This would lead us to ask the question: what is the optimal importance map for AMS that would generate an importance map that is optimal to the exponential transform.

6. The strong attenuation benchmark is difficult to perform with a spatial importance map. Intermediate regions had to be added to propose virtual splitting points. This means that the user actually needs to change the geometry itself to make the problem converge. This is not a good solution for industrial use. We could think about including the *stratified-sampling* method along AMS flights. This would increase the number of splitting candidates along the path of the particle and thus reduce the risk of seeing AMS stop before reaching the target zone.

7. Finally, so far the exploration phase and the exploitation have been explicitly separated. But even when AMS is used for the exploration phase, it also contributes to the detector response and is already *exploiting* resources for the computation of the target. Instead of explicitly separating the exploration phase from the exploitation phase, we think of developing an hybrid simulation mode, where ET would be progressively incorporated alongside AMS. This progressiveness could be - for example - controlled by the percentage of flights that should be sampled with ET, or even by learning on a super-imposed mesh the zones where it is worth losing the exploration ability of AMS to get the most from ET's speed of convergence.

Résumé

En radioprotection, on s'intéresse à estimer et réduire l'impact des radiations sur les humains et sur notre environnement. Ce domaine est présent dans de multiples milieux, où l'on souhaite toujours protéger les opérateurs qui sont exposés à des radiations. Prenons d'abord les centrales nucléaires. Dans ces enceintes, le challenge principal est de construire des structures qui résistent à des rayonnements de forte intensité. Dans des réacteurs de recherche, des opérateurs réalisent des expériences lors desquelles ils sont directement exposés à des radiations. Enfin dans le domaine médical, des sources plus faibles sont utilisées pour l'imagerie et des thérapies.

Dans tous ces domaines, il est toujours nécessaire d'estimer avec précision les doses auxquelles nous pourrions être exposéses. Nous sommes donc amenés à trouver la solution à des problème de transport de particules dans l'espace.

Le Service d'Étude et des Réacteurs et de Mathématiques Appliquées (SERMA) au CEA Saclay (France), développe des codes qui simulent le transport des particules et leur interaction avec la matière.

Ces outils sont essentiellement séparés en deux catégories. Les codes *déterministes* implémentent des schémas de calculs pour résoudre l'équation de transport de Boltzmann sur un espace des phases discrétisé. L'autre catégorie est la méthode Monte Carlo. Elle simule le transport des particules de manière la plus précise possible en faisant le moins d'approximation sur la géométrie ou sur la physique du transport des particules. Des trajectoires de particules sont obtenues par tirage de nombre pseudo aléatoire, ce qui donne lieu à des fluctuations. A part ces incertitudes statistiques, les seules approximations se retrouvent dans la description de la géométrie, ainsi que dans les données nucléaires. Les codes Monte Carlo sont donc considérés comme étant la référence pour la simulation de problème 3D complexes, avec potentiellement des subtilités spectrales, ainsi que pour la validation des méthodes déterministes.

Il existe néanmoins un problème de taille lorsque l'on souhaite utiliser ces méthodes Monte Carlo pour la résolution de problème de radioprotection. Par construction, résoudre un problème de radioprotection se résume à simuler des événements rares telles qu'une trajectoire de particule qui se déplace sur plusieurs libre parcours moyens avant de contribuer à sa détection. Simuler une trajectoire selon les densités de probabilités naturelle (simulation dite *analogue*) est donc très coûteux en temps de calcul et en ressources informatiques.

On utilise donc des méthodes mathématiques, dites de réduction de variance, qui permettent de simuler préférentiellement les trajectoires qui contribuent à la réponse recherchée. Ces méthodes ont toutes le même pré requis: elles doivent avoir une information a priori pour quantifier la contribution d'une trajectoire à la réponse. Cette information se présente sous la forme d'une *carte d'importance*. Si cette carte d'importance n'est pas en adéquation avec le système, la méthode de réduction de variance peut avoir du mal à converger, voir ne pas converger du tout.

Un grand nombre de travaux ont déjà été réalisés pour donner une forme adéquat à cette carte d'importance pour accélérer au maximum le calcul. Kahn and Harris (1951) ont montré que sous certaines conditions, il existe une forme optimale de la carte d'importance pour obtenir un jeu à zero variance, c'est à dire l'élimination totale du bruit statistique inhérent au Monte Carlo. Ce résultat a poussé la plupart des travaux à estimer de manière approchée cette fonction à l'aide de méthodes déterministes: la méthode Consistent Adjoint Driven Importance Sampling (CADIS) Le même travail a été réalisé avec des méthodes Monte Carlo T. E. Booth and J. S. Hendricks (1984). L'importance est estimée sur un maillage ou par volume de la géométrie.

Pour résumer, les méthodes qui permettent de produire une carte d'importance à un calcul Monte Carlo sont déjà bien rodées. En revanche, elles sont limitées car elles requièrent l'œil expert d'un utilisateur pour que la simulation converge.

C'est ce qui justifie cette étude qui consiste à générer de manière adaptative et continue des cartes d'importance pour une simulation Monte Carlo.

La première partie de cette thèse introduit les outils mathématiques nécessaires à la formulation du problème (Chapitre 1). Ensuite, une attention est portée sur les deux méthodes de réduction de variance utilisées dans TRIPOLI-4® (Chapitre 2). En l'occurrence la transformée exponentielle, qui est la méthode la plus anciennement implémentée dans TRIPOLI-4® et l'Adaptive Multi-level Splitting, très récemment implémentée par Louvin (2017).

Dans un second temps, les méthodes qui font l'état de l'art de la génération de cartes d'importance sont présentées au Chapitre 3. Étant donnée la forte représentation des méthodes déterministes dans l'état de l'art, il était naturel de commencer le travail de thèse par une vérification de leurs efficacité. Pour cela, un version de développement de TRIPOLI-4® a été couplée avec un solveur déterministe résolvant l'équation de Boltzmann directe et adjointe avec l'approximation multi-groupe (IDT). L'utilisateur peut maintenant, à moindre coût, et surtout de manière semi-automatique générer des cartes d'importance à l'aide d'IDT, un solveur multi groupe de l'équation de Boltzmann directe et adjointe (Chapitre 4). Afin de soulager l'utilisateur de tâches laborieuses, certaines opérations sont effectuées de manière automatique:

- Les sections efficaces sont condensées par INIPOND. Elles sont approchée à l'ordre d'anisotropie P1 grâce à une correction du transport.
- Le maillage fourni par l'utilisateur convertit la géométrie détaillée de TRIPOLI-4® en une géométrie pour IDT. Le mode 1D, 2D ou 3D d'IDT est automatiquement

détecté selon le nombre de mailles dans chacune des directions.

- La source du calcul adjoint est calculée selon la réponse cible que l'utilisateur désigne. (le flux dans un détecteur par exemple.)

IDT renvoi à TRIPOLI-4® un carte d'importance en mémoire, qui est immédiatement réutilisée dans le calcul Monte Carlo. Des tests ont été réalisés sur plusieurs benchmarks avec des types de difficulté variées. Sur des problèmes à forte atténuation, les accélérations de la transformée exponentielle sont significatifs (4 ordres de grandeur par rapport à l'existant). Sur des géométries où le streaming des particules importe plus, de faibles accélérations sont observées avec la transformée exponentielle. Ceci s'explique en partie car le biaisage de la collision n'est pas implémenté dans TRIPOLI-4®.

Dans un troisième temps, le cœur de la thèse a été attaqué: estimer la réponse adjointe avec des trajectoires Monte Carlo venant d'une simulation directe. Le travail a d'abord été réalisé avec le mode analogue de TRIPOLI-4® (sans réduction de variance). Une comparaison à un calcul théorique en milieu infini, homogène et isotrope a mis en évidence que le flux adjoint est bien reconstruit. Cependant, cet estimateur ne suffit pas lorsque les événements auxquelles on s'intéresse sont rares. L'estimateur a donc été étendu à deux méthodes. Celles pour lesquelles le poids des particules varie en fonction du temps, et l'AMS. Etant donnée la robustesse de l'AMS, c'est cet algorithme qui a été choisi préférentiellement pour converger la réponse adjointe. Le travail sur les méthodes déterministe qui avait été réalisé en amont a permis de vérifier que le flux adjoint calculé par l'AMS est bien cohérent avec un flux adjoint de l'état de l'art. Comme attendu, une forte similitude a été observée dans certaines région de l'espace des phase, laissant d'autres régions totalement inexplorées.

Ensuite, il a s'agit de trouver une stratégie de convergence d'un calcul de ratioprotection avec les outils mis en œuvre dans les chapitres précédents. Trois stratégies sont proposées.

- **Stratégie à deux étapes:** La réponse adjointe est calculé lors d'une première simulation pendant une durée fixée par l'utilisateur. La carte d'importance initiale peut être soit géométrique, venant d'INIPOND, ou d'IDT. Un second calcul est ensuite effectué en utilisant la réponse adjoint du premier en tant que carte d'importance.

- **Stratégie alternante:** Cette stratégie est une tentative de donner à l'utilisateur un maximum d'information sur l'état de convergence de la réponse adjointe. Des phases dites d'*exploration* s'alternent avec des phases d'*exploitation*. Lors d'une phase d'exploration (d'une durée qui est fixée par l'utilisateur en nombre de batchs, par exemple 10), une simulation tourne avec la méthode AMS et une carte d'importance initiale choisie par l'utilisateur. Les phases d'exploitation s'alternent en utilisant soit la méthode AMS, soit la transformée exponentielle. Lors de ces phases, la réponse adjointe calculée par les phases d'exploration est convertie en carte d'importance et réutilisée comme telle. Deux estimateurs de convergence sont monitorés: la réponse elle même, qui peut être parfois sous-estimée, et la figure de mérite instantanée. Cette figure de mérite instantanée est calculée sur la dernière phase d'exploitation et non sur l'ensemble des batchs jusqu'alors calculés. Si ces deux critères convergence, l'utilisateur peut stopper la simulation.

- **Stratégie bootstrap:** La dernière stratégie proposée consiste à donner aux phases d'exploration la possibilité de réutiliser la réponse adjointe qu'elle a déjà accumulée. La carte d'importance initiale est alors écrasée par le nouveau flux adjoint lors de la seconde phase d'exploration. La simulation démarre donc d'elle même et oublie aussi tôt les informations a priori donnée par l'utilisateur (la carte d'importance initiale). De large gains en vitesse de convergence ont été observés par rapport à la stratégie alternante. En particulier dans un calcul couplé neutron gammas.

Ces stratégies ont permis de mieux appréhender l'état de convergence de la carte d'importance pour l'utilisateur.

Enfin, une dernière phase de recherche a consisté à voir si les méthodes d'apprentissage automatique pouvaient profiter à approcher la carte d'importance vers une carte d'importance optimale. Les méthodes à estimation de densité par noyaux ont été retenues pour leur robustesse par rapport à des méthodes telles que les réseaux de neurones ou les arbres de décision.

Ce travail se conclut par une présentation de perspectives.

Bibliography

Banerjee, Kaushik and William R Martin (2012). "Kernel Density Estimation Method for Monte Carlo Global Flux Tallies". *Nuclear Science and Engineering* 170, pp. 234–250.

Bell, George I and Samuel Glasstone (1970). *Nuclear Reactor Theory.*

Boireau, G et al. (June 2016). "Online monitoring of the Osiris reactor with the Nucifer neutrino detector". *Phys. Rev. D* 93.11, p. 112006.

Booth, T. E. and J. S. Hendricks (1984). "Importance Estimation in Forward Monte Carlo Calculations". *Nuclear Technology - Fusion* 5.1, pp. 90–100.

Booth, Thomas E (1986). "A Monte Carlo Learning/Biasing Experiment with Intelligent Random Numbers". *Nuclear Science and Engineering* 92, pp. 465–481.

— (2004). *A Transport Process Approach to Understanding Monte Carlo Transport Methods.* Tech. rep. Los Alamos National Laboratory report LA-UR-04-1426.

— (2006). *Genesis of the Weight Window and the Weight Window Generator in MCNP - A Personal History.* Tech. rep. Los Alamos National Laboratory reference LA-UR-06-5807, pp. 1–13.

Booth, Thomas E. (1988). "The Intelligent Random Number Technique in MCNP". *Nuclear Science and Engineering* 100.3, pp. 248–254.

— (2005). "Insights into Stratified Splitting Techniques for Monte Carlo Neutron Transport". *Nuclear Science and Engineering* 151.2, pp. 224–236.

Both, J. P., J.C. Nimal, and T. Vergnaud (1990). "Automated importance generation and biasing techniques for Monte Carlo shielding techniques by the TRIPOLI-3 code". *Progress in Nuclear Energy* 24, pp. 273–281.

Brehier, Charles Edouard et al. (2016). "Unbiasedness of some generalized adaptive multilevel splitting algorithms". *Annals of Applied Probability.*

Brun, E. et al. (2015). "Tripoli-4®, CEA, EDF and AREVA reference Monte Carlo code". *Annals of Nuclear Energy* 82, pp. 151–160.

Burn, K. W. (1992). "Complete optimization of space/energy cell importances with the DSA cell importance model". *Annals of Nuclear Energy* 19.2, pp. 65–98.

Burn, K.W. (1990). "Optimizing cell importances using an extension of DSA - theory, implementation, preliminary results." *Progress in Nuclear Energy* 24, pp. 39–54.

Burn, Kenneth (1997). "A New Weight-Dependent Direct Statistical Approach Model". *Nuclear Science and Engineering* 125, pp. 128–170.

Cérou, Frédéric and Arnaud Guyader (2005). *Adaptive Multilevel Splitting for rare event analysis.* Tech. rep.

Cérou, Frédéric, Arnaud Guyader, et al. (2013). "A multiple replica approach to simulate reactive trajectories". *Journal of Chemical Physics.* arXiv: `1007.4481`.

Chang, Chih-Chung and Chih-Jen Lin (2011). "ajkkjjkjjkjkjkkjkkLIBSVM: A Library for Support Vector Machjkines". *ACM Transactions on Intelligent Systems and Technology* 2.3, 27:1–27:27.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System".

Clark, Francis H. (1966). "The exponential Transform as an Importance-Sampling Device-A Review -". *Nuclear Science Abstracts.*

Coveyou, R R, V R Cain, and K J Yost (1967). "Adjoint and importance in Monte Carlo application". *Nuclear Science and Engineering* 27.2, pp. 219–&.

Cowan, P et al. (2009). "Recent developments to the Monte Carlo code MCBEND". *Nuclear Technology* 168.

Čufar, Aljaž et al. (2017). "The analysis of the external neutron monitor responses in a simplified JET-Like tokamak using ADVANTG". *Fusion Science and Technology* 71.2, pp. 162–176.

Dumonteil, Eric (2009). "On a New Variance Reduction Technique: Neural Network Biasing - A Study of Two Test Cases with the Monte Carlo Code Tripoli4". *Nuclear Technology* 168.3, pp. 793–798.

Evans and Hendricks (1998). *An Enhanced Geometry-Independent Mesh Weight Window Generator for MCNP An Enhanced Geometry-Independent Mesh Weight Window Generator for MCNP*. Tech. rep. Los Alamos National Laboratory reference LA-UR-97-5057.

Franke, Brian C and Ronald P Kensek (2010). "Adaptive Three-Dimensional Monte Carlo Functional-Expansion Tallies". *Nuclear Science and Engineering* 165, pp. 170–179.

Giffard, François-Xavier (2000). "Développements utilisant des méthodes stochastiques et déterministes pour l'analyse de systèmes nucléaires complexe s". PhD thesis.

Griesheimer, David P., William R. Martin, and James Paul Holloway (2005). "Estimation of flux distributions with Monte Carlo functional expansion tallies". *Radiation Protection Dosimetry*.

Hoogenboom (2008). "Zero-Variance Monte Carlo Schemes Revisited". *Nuclear Science and Engineering* 160, pp. 1–22.

Hoogenboom, J.E. and D. Légrády (2005). "A Critical Review of the Weight Window Generator in MCNP". *Proceedings of Monte Carlo 2005 Topical Meeting, Chattanooga, TN on CD-ROM* January, pp. 17–21.

Jia, Yangqing et al. (2014). "Caffe: Convolutional Architecture for Fast Feature Embedding". *Proceedings of the 22Nd ACM International Conference on Multimedia*. MM '14. New York, NY, USA: ACM, pp. 675–678.

Juneja, S. and P. Shahabuddin (Jan. 2006). "Chapter 11 Rare-Event Simulation Techniques: An Introduction and Recent Advances". *Handbooks in Operations Research and Management Science* 13, pp. 291–350.

Kahn, Herman and T.E. Harris (1951). "Estimation of particle transimssion by random sampling". *National Bureau of Standards applied mathematics 12:27–30* 12, pp. 27–30.

Lathrop, K D (1968). "Ray Effects in Discrete Ordinates Equations". *Nuclear Science and Engineering* 32, pp. 357–369.

Leppänen, Jaakko, Tuomas Viitanen, and Olli Hyvönen (2017). "Development of a Variance Reduction Scheme in the Serpent 2 Monte Carlo Code". *M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering.*

Levitt, Leo B (1968). "The Use of Self-Optimized Exponential Biasing in Obtaining Monte Carlo Estimates of Transmission Probabilities The Use of Self-Optimized Exponential Biasing in Obtaining Monte Carlo Estimates of Transmission Probabilities". *Nuclear Science and Engineering* 31, pp. 500–504.

Louvin, Henri (2017). "Development of an adaptive variance reduction technique for Monte Carlo particle transport". PhD thesis. Universite Paris Saclay.

Louvin, Henri, Eric Dumonteil, and Tony Lelièvre (2017). "Three-Dimensional Neutron Streaming Calculations Using Adaptive Multilevel Splitting". *M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering.*

Mosher, S W et al. (2013). *ADVANTG-An Automated Variance Reduction Parameter Generator.* Tech. rep. ORNL/TM- 2013/416, Oak Ridge National Laboratory (2013).

Munk, Madicken et al. (Sept. 2016). *FW/CADIS-Ω: An angle-informed hybrid method for deep-penetration radiation transport.* Tech. rep. University of California, Berkeley, CA.

Peplow, Douglas E (2011). "Monte Carlo shielding analysis with MAVRIC". *Nuclear Technology* 174, pp. 289–313.

Petit, Odile, Yi-Kang Lee, and Cheikh M. Diop (2014). "Variance reduction adjustment in Monte Carlo TRIPOLI-4® neutron gamma coupled calculations". *Progress in Nuclear Science and Technology.*

Schneider, D et al. (2016). "Apollo3®: Cea/Den Deterministic Multi-Purpose Code for Reactor Physics Analysis". *Physor 2016*, pp. 2274–2285.

Sutton, Richard S and Andrew G Barto (1998). *Reinforcement Learning: An Introduction.* 2nd ed. MIT Press.

Sweezy, J. et al. (2005). "Automated variance reduction for MCNP using deterministic methods". *Radiation Protection Dosimetry* 116, pp. 508–412.

Tuffin, Bruno and Pierre L'Ecuyer (2008). "Approximate zero-variance simulation". *Proceedings of the 2008 Winter Simulation Conference*, pp. 170–181.

Wagner, John C (1997). "Acceleration of Monte Carlo shielding calculations with an automated variance reduction technique and parallel processing." PhD thesis. The Pennsyslvania State University.

Williams, M. L. and W. W. Engle (1977). "The Concept of Spatial Channel Theory Applied to Reactor Shielding Analysis". *Nuclear Science and Engineering* 62.1, pp. 92–104.

Zmijarevic, I. and R. Sanchez (2001). "Deterministic solutions for 3D Kobayashi benchmarks". *Progress in Nuclear Energy* 39, pp. 207–221.

Zmijarevic, I. and D. Sciannandrone (2017). "First Collision Source in the IDT Discrete Ordinates Transport Code". *M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering.*

APPENDIX A

BENCHMARKS

This appendix gathers all benchmarks used in this work.

A.1 Sphere 1 group

The sphere problem consists of a homogeneous, infinite medium filled with a material with constant cross sections for one energy group:

$$\Sigma_t = 1\,\mathrm{cm}^{-1}, \Sigma_s = 0.1\,\mathrm{cm}^{-1},$$

A monoenergetic point source is placed at (0,0,0). The detector is the integrated total flux computed in a shell between 5 cm and 5.1 cm. Figure A.1 illustrates the problem geometry.

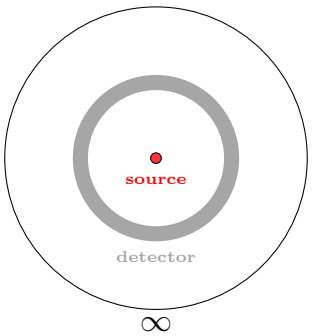


Figure A.1. Geometry of infinite medium with constant cross sections

A.2 Sphere 2 groups

The sphere problem consists of a homogeneous, infinite medium filled with a material with constant cross sections for two energy groups:

$$\Sigma_t^1 = 1\,\mathrm{cm}^{-1}, \Sigma_s^{1\to1} = 0.2\,\mathrm{cm}^{-1}, \Sigma_s^{1\to2} = 0.4\,\mathrm{cm}^{-1}, \Sigma_t^2 = 2\,\mathrm{cm}^{-1}, \Sigma_s^{2\to2} = 0.5\,\mathrm{cm}^{-1}$$

The detector is the integrated total flux computed in a shell between 5 cm and 5.1 cm. The geometry is the same as in Figure A.1.

A.3 Water+Iron

This is a multi-group 2D benchmark. The geometry, illustrated in Figure A.2, is composed of a 70 cm × 70 cm tank of water surrounded by a 30 cm wide bypass filled with iron. Leakage boundary conditions are imposed on the side of the geometry whereas reflective boundary conditions are imposed on the top and on the bottom in the Z-direction with an arbitrary hight of 100 cm Cross sections are multi group and were created by the condensation of point-wise cross sections on the energetic mesh:

$$20\,\text{MeV},\ 2\,\text{MeV},\ 500\,\text{keV},\ 50\,\text{keV},\ 1\,\text{eV},\ 10^{-6}\text{eV}$$

Cross sections are isotropic. The source is uniformly distributed in a 1 cm × 1 cm zone in the bottom left corner in the iron. Its intensity is 0.2 in each group. The detector is the zone contained in a 1 cm × 1 cm corner on the top right corner in the iron and is sensitive to all the energy range - we are interested in the integrated neutron flux. Figure A.2 shows an illustration of the benchmark.

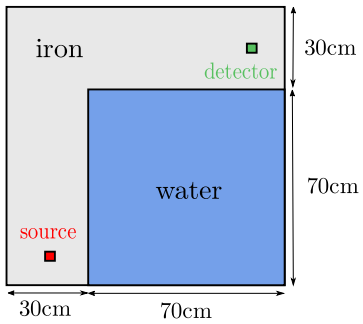


Figure A.2. Water+Iron geometry.

A.4 Bunker

The bunker benchmark is proposed by K. W. Burn (1992). A monodirectional 14 MeV neutron source impinges on a stainless steel sphere. Neutrons are scattered and either go around the wall or penetrate the narrow cylindrical duct in the wall. We are interested in the fast ($\geq$ 9.12 keV) outgoing current through the room exit, in the south-eastern corner. Figure A.3 shows the geometry of the problem.

The energy domain chosen for this benchmark is:
20 MeV 5 MeV 1 MeV 100 keV 9.12 keV 1E−11 MeV

A.5 Strong Attenuation

This benchmark serves to test the capabilities of our algorithms to compute strongly attenuated responses. The mono-directional and monikinetical source of 14 MeV is placed at the begining of 3 m-deep water tank. Figure A.4 illustrates the problem geometry.. The detector is in a concrete wall of 20 cm. Reflection boundary conditions are imposed on all surfaces except on the one behind the detector.

The energetic mesh used for the plots against the energy is (in MeV):
2.0000E+01

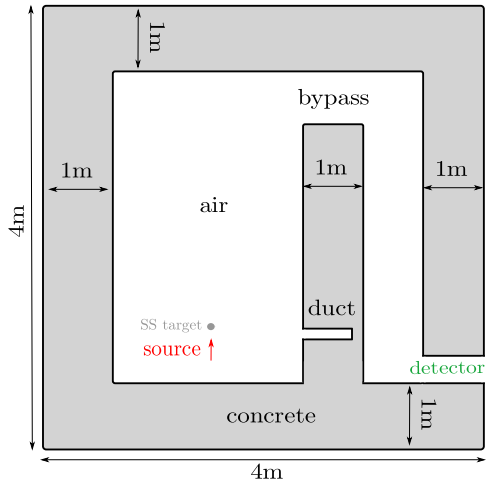


Figure A.3. Bunker geometry.

8.1870E+00
6.7030E+00
5.4880E+00
4.4930E+00
3.6780E+00
3.0110E+00
2.4650E+00
2.0180E+00
1.6520E+00
1.3530E+00
1.1080E+00
9.0710E-01
6.0810E-01
4.0760E-01
2.7320E-01
1.8310E-01
1.2270E-01
8.2290E-02
5.5160E-02
3.6970E-02
2.4780E-02
1.6610E-02
1.1130E-02
7.4650E-03
5.0040E-03
3.3540E-03
2.2480E-03
1.5070E-03
1.0100E-03
6.7720E-04

4.5390E-04
3.0430E-04
2.0390E-04
1.3670E-04
9.1660E-05
6.7900E-05
5.5590E-05
4.5510E-05
3.7260E-05
3.0510E-05
2.4980E-05
2.0450E-05
1.6740E-05
1.3700E-05
1.1220E-05
9.1890E-06
7.5230E-06
6.1600E-06
5.0430E-06
4.1290E-06
3.3800E-06
2.7670E-06
2.3590E-06
2.1290E-06
2.0190E-06
1.9290E-06
1.8390E-06
1.7540E-06
1.6690E-06
1.5900E-06
1.5090E-06
1.4390E-06
1.3700E-06
1.3040E-06
1.2350E-06
1.1690E-06
1.1090E-06
1.0690E-06
1.0340E-06
9.8590E-07
9.2990E-07
8.5990E-07
7.8990E-07
7.0490E-07
6.2490E-07
5.4000E-07

4.8500E-07
4.3290E-07
3.9090E-07
3.5190E-07
3.1440E-07
2.8240E-07
2.4790E-07
2.1990E-07
1.8890E-07
1.6000E-07
1.3400E-07
1.1490E-07
9.4990E-08
7.7000E-08
5.8990E-08
4.2990E-08
2.9990E-08
2.0000E-08
1.4990E-08
1.0000E-08
5.4990E-09
2.9990E-09
1.0000E-11

The energetic mesh used for the importance maps with IDT is (in MeV):
1.96403E+01
1.49182E+01
1.38403E+01
1.16183E+01
1.00000E+01
9.04837E+00
8.18731E+00
7.40818E+00
6.70320E+00
6.06531E+00
4.96585E+00
4.06570E+00
3.32871E+00
2.72532E+00
2.23130E+00
1.90139E+00
1.63654E+00
1.40577E+00
1.33694E+00
1.28696E+00
1.16205E+00

1.05115E+00
9.51120E-01
8.60007E-01
7.06512E-01
5.78443E-01
4.94002E-01
4.56022E-01
4.12502E-01
3.83884E-01
3.20647E-01
2.67827E-01
2.30014E-01
1.95008E-01
1.64999E-01
1.40000E-01
1.22773E-01
1.15624E-01
9.46646E-02
8.22975E-02
6.73795E-02
5.51656E-02
4.99159E-02
4.08677E-02
3.69786E-02
3.34597E-02
2.92810E-02
2.73944E-02
2.61001E-02
2.49991E-02
2.26994E-02
1.85847E-02
1.62005E-02
1.48997E-02
1.36037E-02
2.25243E-05
6.25000E-07
1.10000E-10

A.6 Neutron-gamma problem.

Petit, Lee, and Diop (2014) proposed to revisit a small benchmark to assess the efficiency of TRIPOLI-4® in solving coupled neutron-gamma problems. Figure A.5 illustrates the problem. A point source with a Watt spectrum is placed in a paraffin collimator. The room is filled with air. The detector is cylindrical detector sensitive to the photon dose. Between the source and the detector reside alternating slabs of two materials: polyethylene and stainless steel. Stainless steel slabs should be creating many gammas from the

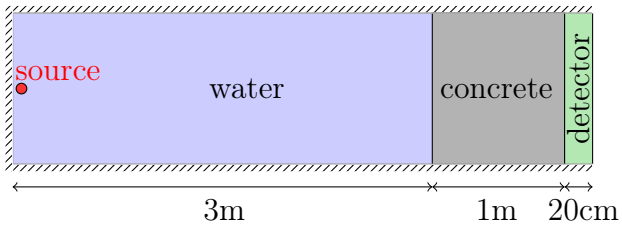


Figure A.4. Geometry for the strong attenuation benchmark. A mono-directional neutron source, sampled on a Watt spectrum is placed in a 3 m-long tank of water. The detector is a 20 cm-wide part of a 120 cm-long concrete wall. Reflections conditions are imposed on all surfaces except for the outer surface of the detector, from which neutrons can leak out.

capture of neutrons. The energetic mesh used for this benchmark is (in MeV):
20. 1. 0.1 0.01 0.001 0.625E-6 1.E-11

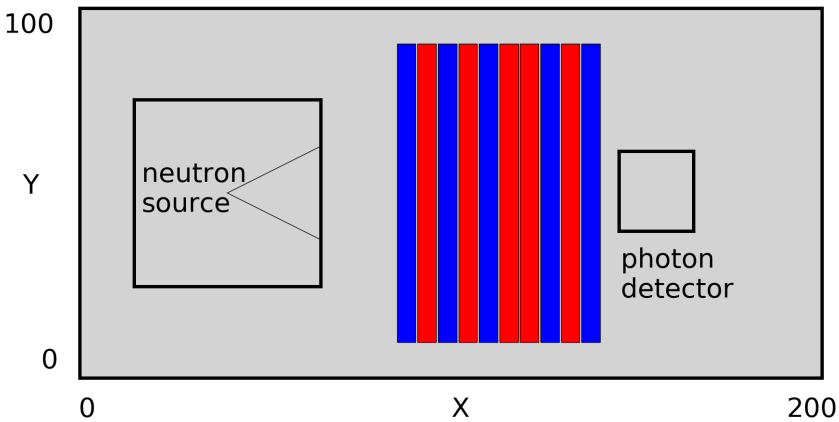


Figure A.5. Geometry of the coupled neutron-gamma problem. Polyethylene slabs are colored in red, whereas stainless steel slabs are in blue. The scale in X and Y are in cm.

A.7 Bypass

The bypass is a 2D theoretical model. It consists in a 80 cm × 80 cm tank filled with helium. In the middle stands a 60 cm × 60 cm block of concrete. A point neutron source at 14 MeV is placed in the bottom-left corner, and the detector in which we compute the total neutron flux is placed in the top-right corner. Figure A.6 is a description of the problem.

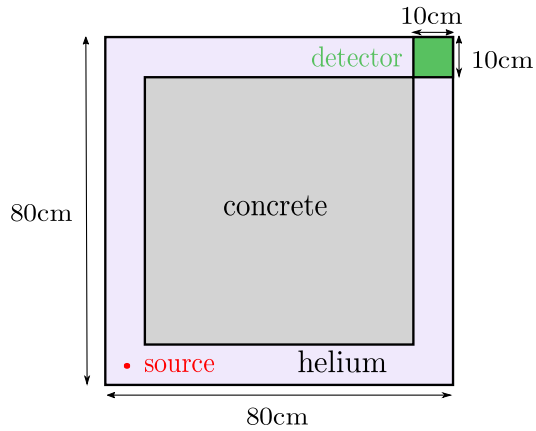
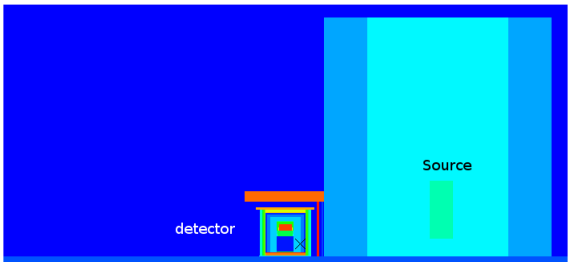


Figure A.6. Geometry of the theoretical bypass problem.

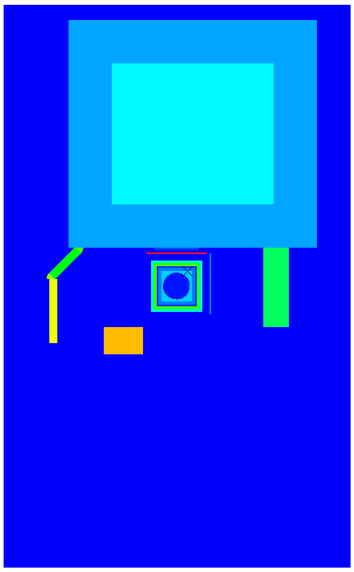
A.8 Nucifer at Osiris

This benchmark proposes to score the fast flux in the Nucifer detector resulting from the sources born in the Osiris reactor.

The sources were collected after a first Monte Carlo criticality simulation. Neutrons leaving the reactor are stored in a file and re-used in a second shielding calculation as a fixed source. The energetic mesh used for this benchmark is (in MeV):
20 12 8 6 4 2 1 1E-11



(a) Horizontal cross section in the middle of the nucifer detector. On the left, the nucifer is placed in a shielding box. On the right, the Osiris reactor.



(b) Vertical cross section in the middle of the nucifer detector. The detector is placed in a room where a few walls surround it. The room is filled with air.

Figure A.7. Nucifer detector next to Osiris

APPENDIX B

CHOICE OF THE SMOOTHING ALGORITHM

Contents

B.1 Requirements **140**
B.2 The dataset **140**
B.2.1 A small condensation 142
B.3 The training phase and performance metrics **142**
B.4 Discussion on models **143**
B.4.1 Some small tests 144
B.5 Conclusion **144**

So what can be done to reduce the variance of the adjoint estimator? In this chapter, we discuss and choose a model to *smooth* the importance map built in Chapter 5. The following chapters should be considered as cursory and exploratory work around the edges of the terrain that was covered by this thesis.

B.1 Requirements

We need a model that *predicts* the importance at any point in the phase space. Because our score is the ratio of two quantities, we choose to treat them independently to minimise the bias of our model. So the problem can be summarised as follows: find two parametric or non-parametric models that can interpolate/smooth the flow of events generated by the direct Monte Carlo simulation. Also, in what follows we shall use the word "training" to refer to the adjustment of whatever parameters/ingredients the model has to reproduce the dataset.

Build $\mathcal{M}_{\chi\chi^\dagger}$ and $\mathcal{M}_\chi$ and estimate the importance by taking their ratio.

$$\hat{I}(P) = \frac{\mathcal{M}_{\chi\chi^\dagger}(P)}{\mathcal{M}_\chi(P)}.$$

Let us enumerate the constraints that we impose on these models. A good model must:

1. predict the importance as quickly as possible. The frequency at which the importance is queried is extremely high as this is done at any collision, emission and source sampling.

2. be able to digest a huge flow of data. Typically, millions of emissions are sampled in a few seconds.

3. be parsimonious memory-wise. The model will be trained with a large number (say N) of events arising from the Monte Carlo simulation. Memory usage by the model must be independent of N or weakly dependent on it. Models with $\mathbb{O}(N)$ memory requirements are not suitable.

4. be trainable in a reasonable amount of time (typically less than a hundred times the time it takes for one batch to be simulated)

5. be robust in the sense that we want some guarantees on the convergence of the model.

6. include rare events during training and not eliminate them from the model because they are outliers.

7. be capable of incremental updates when additional points are available (online)

B.2 The dataset

Recall that our points contributing to the adjoint score are *emission* points. We change our vocabulary from phase space to *feature* space to discuss learning algorithms appropriately. The features considered are chosen depending on the symmetries of the problem

among:

$$x, y, z$$

$$E,$$

$$\Omega_\theta, \Omega_\phi$$

If a model requires some normalisation of the features, we shall do it (for example scale from -1 to 1). We also need to fix our observables, i.e. the quantities that our models will be trained to predict. As explained in Chapter 5, each emission point is associated with the contribution and a statistical weight. The observables are usually referred to as "labels" in the parlance of machine-learning.

The weight distribution depends on the variance reduction method. Because we mainly used AMS to score the adjoint flux, we shall focus on AMS for generating the dataset. AMS produces a specific weight distribution. Let us have a look at the weight distribution for AMS in the simplest case: the mono-energetic infinite, homogeneous sphere presented in [Appendix A.1].

The detector is the shell between $r = 10\,\text{cm}$ and $r = 10.2\,\text{cm}$. In this problem, equally spaced (1 cm) virtual volumes were added. Figure B.1a shows the distribution of the contributons (Williams and Engle, 1977) with AMS. It is quite flat, because it carries the emission density times the adjoint score which both have exponential attenuations. On the other hand, the emission density Figure B.1b is more *complex*. AMS weights span the whole range from the ones at the first iteration to the ones at the last iteration at which particles contribute. Because of the construction of AMS trajectories (Section 5.5), the appearance of the AMS weight distribution is triangular.

Referring to a basic rule of Machine Learning, a model is trained on a "training dataset" and tested on a "test dataset". The error of the model is computed for both phases. During training, the error of the model to reconstruct the training dataset should decrease. The error for the test dataset will also start decreasing, but will eventually increase when the model is overfitting the training dataset. This is a problem that the data scientist should avoid by stopping the training of the model.

This distribution will not be easy to model for two reasons. First, the emission density decreases exponentially, which means that if a certain tolerance is required from the user, any model will eventually cut the values under - let us say 0.0001 - and yield proper convergence only at points where the collision density is "reasonable". This would strongly bias our estimation of the adjoint score across the whole geometry. One way to overcome this problem and make the model treat all regions with equal importance, would be to learn the *logarithm* of the AMS weights. However, the model will then learn the mean of the logarithm of the AMS weights.

We start to understand that the emission density, is in fact composed of two components:

1. the actual density of points in the simulation.

2. the weights associated to these points.

The information on the distribution is *shared* by these two components and the balance depends on: (a) the location in the geometry and (b) on the variance reduction method used. In weight window-based variance reduction methods, more of the information if carried by the particle weights, and the density of simulated points does not vary much across the geometry. A simple histogram easily concatenates these components:

$$\mathbb{E}_{\text{bin}} = \frac{\sum_{P \in T}(w(P) \times \mathbb{1}_{\text{bin}}(P))}{N_{\text{source}}}$$

But is the case for any model? If a model is purely designed for regression, it should not take the density into account at all. So giving raw AMS points to a regression model will trivially fail. One quick fix for AMS emissions is to transfer a part of the information present in the points density towards their weights.

B.2.1 A small condensation

With AMS, we propose to pre-treat the points of the dataset. As we saw in Chapter 5, AMS trajectories are for large part duplicated on other trajectories. This means that many emission points are common to a great number of trajectories. The *condensation* consists in pre-summing the AMS weights on these emission points so that a physical point in the phase space appears only once in the learning dataset instead of appearing ten times with different AMS weights. Such condensation was performed for the mono-energetic sphere. The resulting contributon distribution is shown in Figure B.2a. It does not change a lot, but we were more interested in the emission density Figure B.2b. The AMS weights are now more skewed for a given local region in phase space. This reduction reduces the dataset size and thus the training time!

B.3 The training phase and performance metrics

One way to go would be to simulate - let us say - one batch. Take all the emissions and split them into at least two datasets. The first dataset is used for training the model and the second is used to test it. During the training, the error on the test set is measured. The training is stopped for a given criterion: for example when the test error starts increasing with training iterations. This means that the model starts over-fitting the training dataset.

This procedure is safe, but presents disadvantages in our context. First, it is time consuming. Second, our dataset is very *noisy*. It would not be fair to compute the prediction error of points that are drawn in noise. We would like to build a scheme that *quickly* estimates the *quality* of our model. For that, we take the scheme presented Section 6.2. The models are trained in the exploration phases and the exploitation phase serves as a test. The FOM becomes the performance metric of our models. This allows to always keep an eye on our objective.

B.4 Discussion on models

So what models can we choose? We first review two models that have already been used in Monte Carlo particle transport codes, namely functional expansion tallies and kernel density estimators. We first briefly introduce and discuss popular machine learning models that we considered in our work.

Functional Expansion tallies

Functional expansion tallies (FET) have already been used in Monte Carlo simulations to interpolate tallies when these tallies are not localized (i.e. a fission distribution over all the geometry) (**Griesheimer2005**). They consist in an expansion of the flux on a chosen basis. The coefficients are scored during transport. FET have been used to interpolate thermal neutron distributions (Griesheimer, William R. Martin, and Holloway, 2005). More effort was put in tuning the optimal parameters with an adaptive procedure (Franke and Kensek, 2010).

Kernel Density Estimators

Kernel Density Estimators (KDE) are models that smooth the density of some set of points. The idea is that they squash the mass contained in a very localised point in phase space in its neighbourhood. The main drawback is the difficulty to control the bias they introduce if the parameters are not chosen carefully. Banerjee and William R Martin (2012) use KDE to estimate global flux tallies in Monte Carlo simulations. They explain the choice of bandwidth parameters as well as the care that needs to be taken when dealing with boundary conditions.

Neural Networks

Neural Networks (NN) have gained increasing popularity in the past ten years. Their success in real life applications, such as language processing, image recognition has boosted the amount of research performed round them. They present the advantage to have a fixed structure, and thus respects our condition on the memory footprint. The first proof of concept of their utility in accelerating shielding calculations was made by Dumonteil (2009).

Decision trees

A popular solution nowadays are gradient boosted models such as gradient boosted decision trees (Chen and Guestrin, 2016). They are an iterative procedure that build a series of decision trees. Each successive tree is trained on the error of the previous ones. Although decision trees are not performant for regression tasks, especially when the gradients are important, it can be interesting to see if this method can capture the discontinuities in the phase phase.

Support Vector Machines

The idea of Support Vector Machines (SVM) is to define an hyperplane in the phase space that separates data points by minimizing the margin from it. This is computationally very expensive and the simulation time/memory increase with the number of points that

contribute to the model description.

Who wins?

So how did we choose among all these models? Each of them has advantages and disadvantages.

Most of these models are off-line. They cannot be updated with additional data. This is problematic because many batches will contribute to our estimator. Plus, some of them really scale badly with the number of points in the dataset (SVM for example).

We were thus tempted by neural networks, first because Dumonteil (2009) has already shown "decent" accelerations on theoretical models. Also, because of decision trees' easily tunable parameters, we decided to compare it to neural networks.

B.4.1 Some small tests

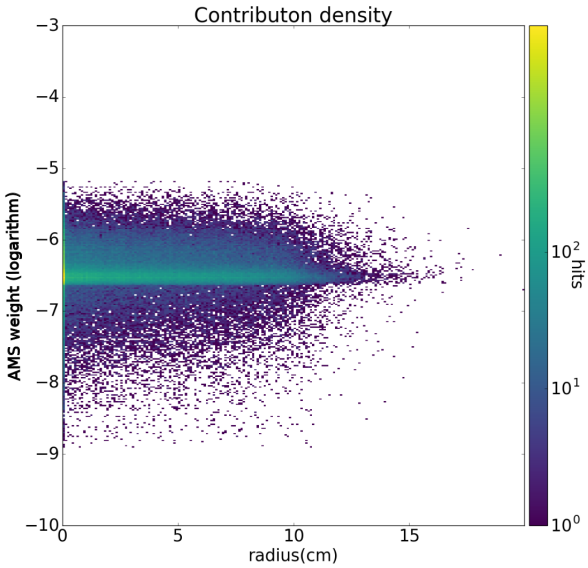
A small library was written to transform the set of emissions per batch into a dataset for machine learning models. Most of the models were simply linked to existing state-of-the-art C++ libraries to test them quickly. Caffe (Jia et al., 2014) was chosen for the neural networks and XGBoost (Chen and Guestrin, 2016) for gradient boosted decision trees. SVM were taken from [Chang and Lin, 2011].

We then tested these models on the emission density of the multi group benchmark A.3. The dataset was condensed with the procedure described in Section B.2.1. The logarithm of the AMS weight was then given as a label to decision trees, or neural networks. Figure B.3 shows the emissions density on the reference histogram (composed of 100×100 mesh cells), then the XGBoost trained on one batch is projected on the same mesh, and finally the neural network. We had difficulties to make the neural network learn abrupt discontinuities as are present in this case. XGBoost did not perform much better capturing these discontinuities. But because the decision trees looked more robust to us, we decided to project the information of one off-line model on a histogram. At each batch, a model is trained and averaged on a histogram. This degenerates the model, but it is necessary. Figure B.4 illustrated the results from this procedure. The first figure is the projection of a single model XGBoost trained on one batch and the second one is the average of 100 XGBoost models. The decision tree is smoothed.

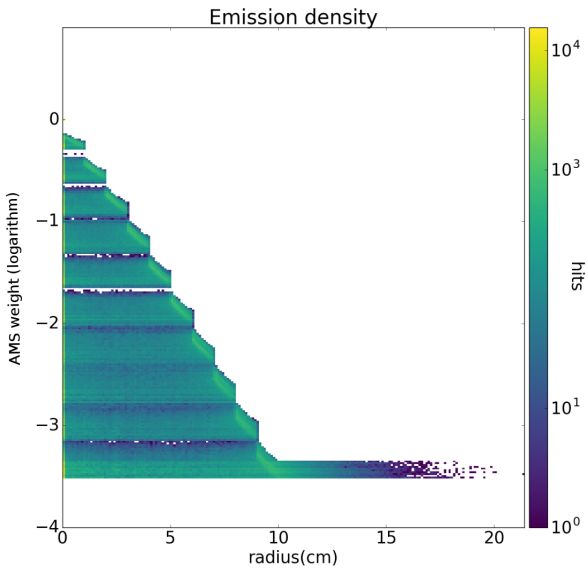
B.5 Conclusion

A small library that gathers machine learning models was written and put in TRIPOLI-4®. It allowed to explore the behaviour of some models. First, the SVM were very heavy in memory and required an unreasonable amount of time to be trained. Second, Neural Networks were hard to train and we concluded that, in spite of their online training capabilities, they are not robust enough for our application. Finally, decision trees were pretty easy to tune and use. This lead us to average offline methods on histograms.

Because we wanted to be as conservative as possible, we conclude by exploring Kernel Density Estimators more thoroughly in the next chapter.

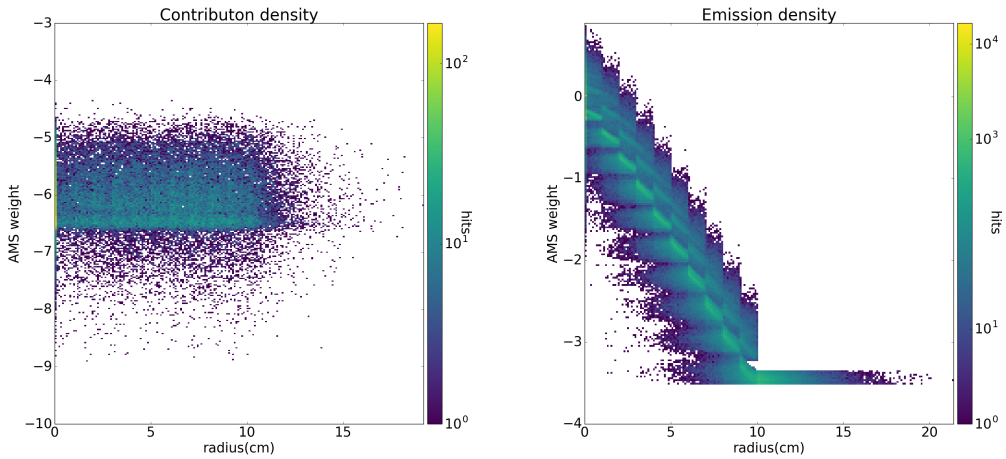


(a) AMS contributon weights distribution



(b) AMS emission weights distribution

Figure B.1. Distribution of points generated by AMS for the adjoint score.



(a) AMS contributon weights distribution **(b)** AMS emission weights distribution

Figure B.2. Distribution of the condensed contribution and emission density.

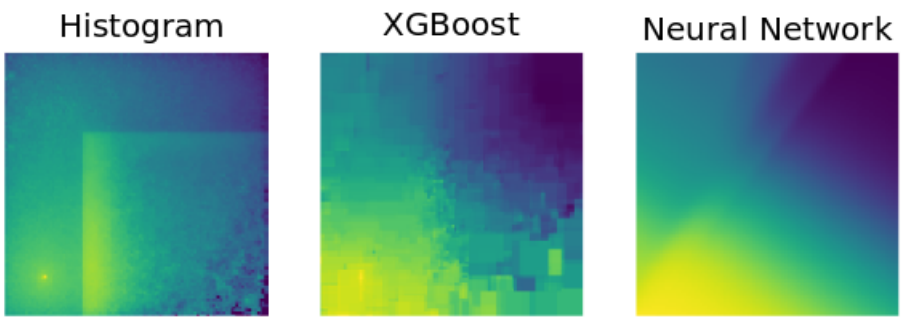


Figure B.3. Comparison of three models for the emission density in the multi group water-iron benchmark. From left to right: emission density projected on a histogram, emission density interpolated during one batch by XGBoost, emission density interpolated with a neural network

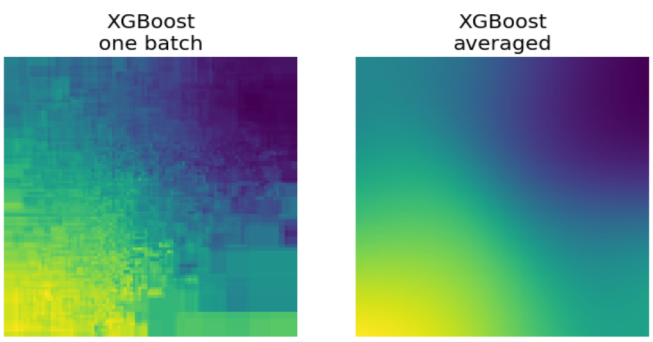


Figure B.4. Illustration of the model projection method. Left: emission density from XGBoost trained on a single batch. Right: Averaged XGBoost models on 100 batches.

APPENDIX C

KERNEL DENSITY ESTIMATORS

Contents

C.1 The model **148**
C.2 Implementation in TRIPOLI-4® **149**
C.2.1 Time monitoring 149
C.3 Acceleration on GPU **151**
C.3.1 Motivation 151
C.3.2 Implementation in TRIPOLI-4 152
C.4 Results **152**
C.4.1 Strong attenuation 152
C.4.2 Neutron-Gamma 152
C.5 Conclusion **154**

The objective is to improve the robustness of our approach. Let us have a look at Kernel Density Estimators. KDE have already been used in MC for nuclear engineering in order to estimate global flux tallies (Banerjee and William R Martin, 2012). A Monte Carlo score on a mesh can be viewed as a probability density function and in this way, be approximated by a KDE. Let us smooth the emission and the contributon density, divide them, and provide the ratio to a variance reduction method (exponential transform or AMS).

C.1 The model

Consider a set of points generated by a Monte Carlo simulation:

$$\{(\mathbf{p}_1, w_1, c_1), \cdots, (\mathbf{p}_n, w_n, c_n)\}$$

where p is the position in phase space, w is the statistical weight assigned to the point, and c is the actual contribution it gave to the response. These points are distributed according to the joint probability density:

$$d(\mathbf{p}, w, c).$$

Intuitively, kernel density estimators *smooth* the density of these sets by redistributing the total mass throughout the entire phase space. The formal KDE estimator writes:

$$\pi_{\text{KDE}}(\mathbf{x}) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{\mathbf{p}\in\mathcal{T}}^{n} \prod_{i=0}^{d} w_i \frac{1}{h_d} K\left(\|\mathbf{x}-\mathbf{p}\|\right) \tag{C.1}$$

Where K is a kernel from $\mathbb{R}^+$ to $\mathbb{R}^+$. h_d is the bandwidth parameter in each direction d. A distance needs to be defined. Banerjee and William R Martin (2012) state that the euclidean distance in phase space should be used. We use the same euclidean distance. Kernels K can be of various form but we restrict ourselves to the use of one of the three following:

$$K_{\text{triangular}} = \prod_{d}^{\text{dim}} (1 - |p_d - x_d|/h_d) \times \mathbb{1}_{|p_d - x_d| < h_d} \tag{C.2}$$

$$K_{\text{tophat}} = \prod_{d}^{\text{dim}} \left(\frac{|p_d - x_d|}{h_d}\right) \times \mathbb{1}_{|p_d - x_d| < h_d} \tag{C.3}$$

$$K_{\text{gaussian}} = \prod_{d}^{\text{dim}} \frac{1}{\sqrt{2\pi}} (e^{-\frac{1}{2}\left(\frac{|p_d - x_d|}{h_d}\right)^2}) \tag{C.4}$$

From the discussion of the previous chapter, it is clear that KDE belong to the family of offline methods. Although a point can be easily added to the KDE support, its size increases with the number of points. We use the same procedure described for offline methods in [Appendix B]. The KDE is thus projected on a mesh at the end of a batch,

such that independent KDE are averaged. Also, we profit from the condensation on the dataset described in the previous chapter.

C.2 Implementation in TRIPOLI-4®

KDE were implemented in TRIPOLI-4®. More precisely, three different kernels were implemented: linear, tophat and gaussian. For the choice of the bandwidths, it is left to the user, who can choose its value in each direction of phase space (except for the angle). Also, because it is hard to define an adequate distance in the energy direction of phase space, we never use the KDE in this direction. The energy is thus treated as in the previous chapters.

Concerning the choice of the bandwidths, there are many references in the literature on the way to choose their optimal value. We started fixing them by empirical considerations. If the bandwidth is too large, the geometry will be smoothed out, and no reliable importance map could arise. Alternatively, if the bandwidth is too small, this reduces to a simple score on a histogram.

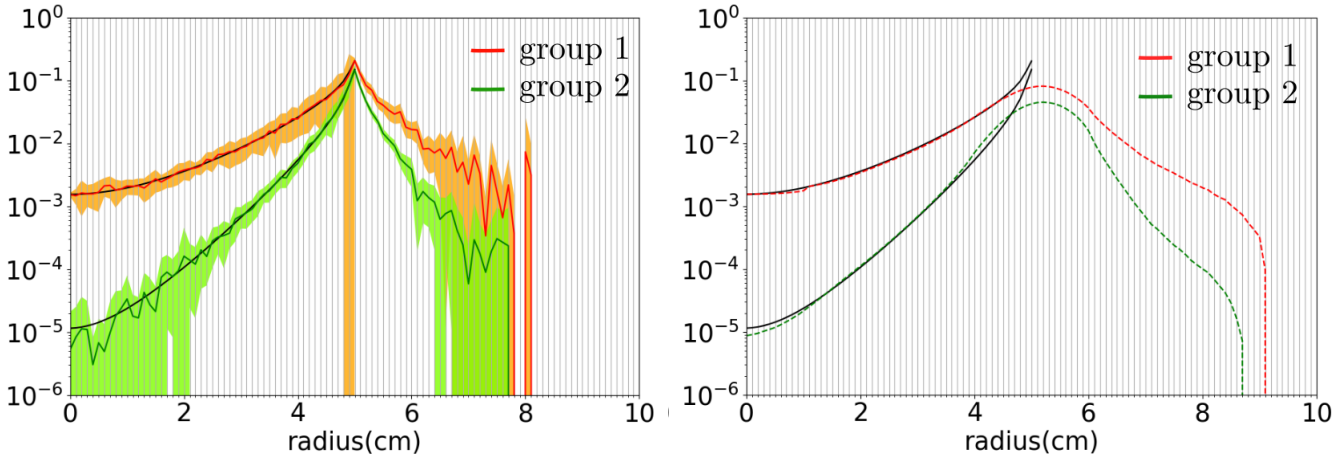
The first version of the KDE was simplistic. Points were accumulated in memory and at the end of the batch, each cell of the histogram asked the KDE for a prediction at its center. This was highly inefficient. The implementation was thus changed and now scores the KDE at the moment when the trajectories are scanned backward: for each emission, the histogram is filled with the contribution of this emission to the KDE prediction.

The algorithm was still extremely slow for large meshes. In order to accelerate the KDE score the following procedure is performed: for each cell of the mesh, we store the index of the neighbouring cells that fall within a range of a certain distance from its center. This maximum distance depends on the kernel used, and is the distance after which the contribution of the kernel to the KDE is negligible. For triangular or tophat kernels, this simply is the bandwidth parameter itself because the contribution vanishes passed this bandwidth. However, gaussian kernels have an infinite range. We choose to truncate the gaussian kernel at three times the bandwidth in order to gain time. Instead of scoring the contribution of a point to the KDE on the entire mesh, only neighbouring cells need to be looked at. This gave significant accelerations to the KDE score.

In order to illustrate the implementation of the KDE, it was tested on the 2 group sphere [Appendix A.2]. Figure C.1 shows two figures. The first is the adjoint score collected after 20 batches with 10000 particles per batch. The second is the KDE score collected for the same amount of particles simulated. The kernel chosen is gaussian with bandwidth 0.5 cm. We see that the variance of the adjoint estimate is indeed reduced in the descending part of the importance. However, there is a large bias near the detector that leads to the question: how strong do we want the smoothing to be if in return, the bias on the adjoint estimate is so high?

C.2.1 Time monitoring

In order to have an idea of how the algorithm scales with the number of cells in a mesh and with the bandwidth, we perform a grid of calculations for the water-iron benchmark. Figure C.2 shows the resulting prediction of the integrated KDE projected on a mesh or a single batch. Figure C.3 shows the time it took for these calculations to



(a) Adjoint score

(b) KDE on adjoint score

Figure C.1. KDE on the sphere 2 groups

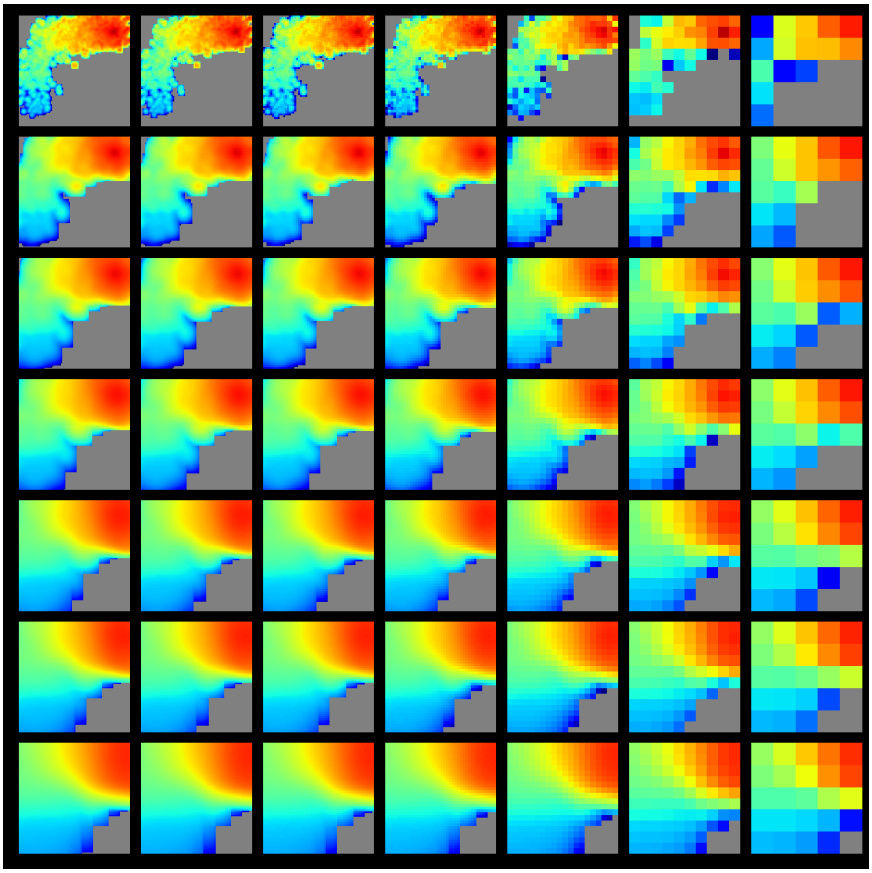


Figure C.2. Grid of KDE prediction for the Water+Iron benchmark. From up to down, the bandwidth in both directions increases and takes the values $h = (1, 3, 4, 5, 7, 8, 10)$. From right to left, the number of cells in each direction of the mesh increases $N = (5, 10, 20, 40, 60, 80, 100)$

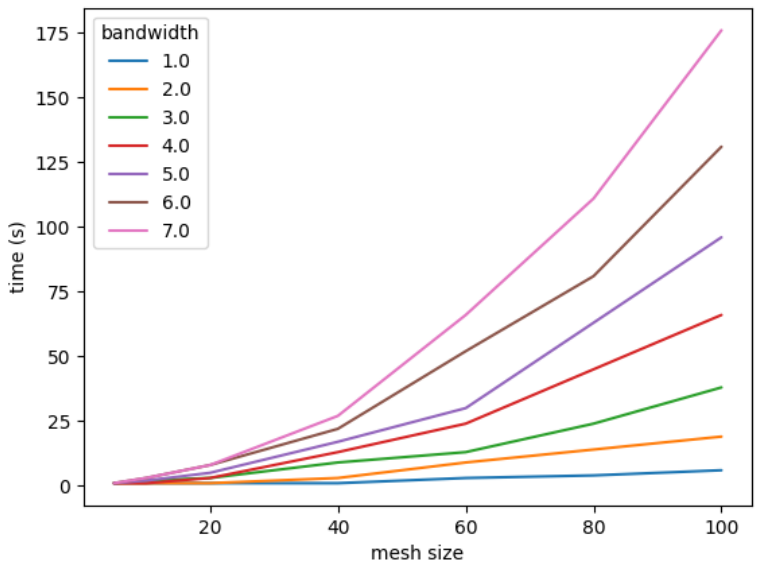


Figure C.3. Simulation time for the KDE on the Water-Iron benchmark. The bandwidth takes values from $h = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$ and the mesh size from $(5, 10, 20, 40, 60, 80, 100)$

finish on a single CPU. We can already imagine that for a 3D problem, with $100 \times 100 \times 30$ mesh cells for ten energy groups, the calculation will be unrealistic.

C.3 Acceleration on GPU

We thus focus our attention on the acceleration of these KDE on GPUs.

C.3.1 Motivation

The complexity of KDE is huge. Let us define: n is the number of emissions, N_m, the number of evaluation points (i.e the number of cells of the mesh) and d the dimension of the problem. Then the complexity of KDE is bounded by:

$$\mathcal{O}_{\text{KDE}}(N_p \times N_m \times d)$$

For large problems, this quickly becomes untreatable in a reasonable period of time. During the last few years, much work has been devoted to the acceleration of algorithms that present a large complexity via accelerators. Among them Graphic Process Units (GPU) gained in popularity. In this work, we perform the computations on two different Nvidia chips. The first one is the *GeForce840M*. The second one is a Nvidia P100, on the calculator Cobalt hosted by the TGCC (CEA).

C.3.2 Implementation in TRIPOLI-4

The developments of the GPU kernels are not made in the main TRIPOLI-4® source folder. This allows to compile TRIPOLI-4® easily with and without GPU support. . Points are stored in a buffer. When this buffer reaches the maximum affordable size that the GPU can manage in its memory, the buffer is sent to the GPU, which computes the results and returns the scores to TRIPOLI-4®.

C.4 Results

So how do KDE perform smoothing the adjoint flux, and being re-used as an importance map? In order to follow the same strategy as in the previous chapter, we begin by looking at the strong attenuation benchmark and then the coupled neutron-photon benchmark.

C.4.1 Strong attenuation

For this benchmark, we take gaussian kernels with a bandwidth of 10 cm. We use the fixed-step alternating scheme of Section 6.2. The exploration phase is run with AMS and the geometrical importance map (inverse of the distance to the detector). The exploitation phase is run with the exponential transform and the KDE adjoint score computed by the exploration phase. Figure C.4 shows the evolution of the response as a function of batches simulated. We observe that the exploitation phase relying on the KDE-smoothed adjoint score converges much faster than the exploitation phase relying in the adjoint score on a histogram. However, the asymptotic spread of the KDE-driven exponential transform is higher than the one with the standard adjoint score. We have gained in convergence speed, but not on the asymptotic variance.

In order to understand where this could come from, let us have a look at the actual KDE prediction for three different energy groups (groups 1, 3 and 20), where the mesh used is the 57 energy groups defined in A.5.

Figure C.5 shows the KDE prediction along with the classical score on the mesh(dotted line). All scores are collected after 1000 batches. We notice that the variance of the KDE score is much lower than the variance of the score on the mesh. However, a real bias is observed compared to the score. Note also the behaviour of group 20. The kernel smooths the value of the importance 1 m farther away than the actual score. This is clearly an under-sampling effect.

C.4.2 Neutron-Gamma

For the neutron gamma problem, the mesh size is 25×50. We use a gaussian kernel. Because scoring the kernel around a point will involve many cells of the mesh(if not all), the GPU version of the KDE was used. Two simulations were run. The first one is the fixed step alternating scheme and the second is the Bootstrap scheme. We perform both simulations because in the previous chapter we saw that they behave very differently.

The response evolution of the alternating scheme is shown Figure C.6. Each exploration phase (AMS+spatial importance map) and each exploitation phase (ET + KDE adjoint score) last ten batches. We observe that the ET is not even starting to converge to the

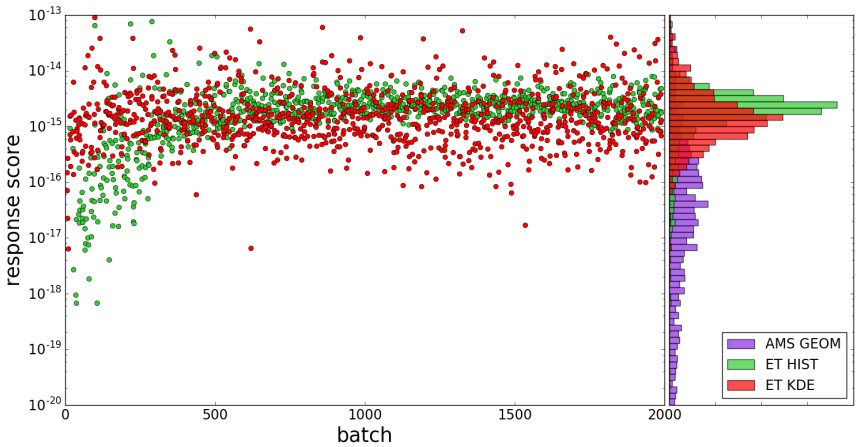


Figure C.4. Score evolution for a fixed-step alternating strategy. Each exploration phase of 10 batches is run with AMS starting from a geometrical initial importance map(the inverse of the distance to the detector.) Each exploitation phase is run with ET. The simulation using the adjoint score smoothed with a KDE is plotted in red and compared to the reference simulation taking the adjoint score on a histogram. The energy groups are listed in A.5

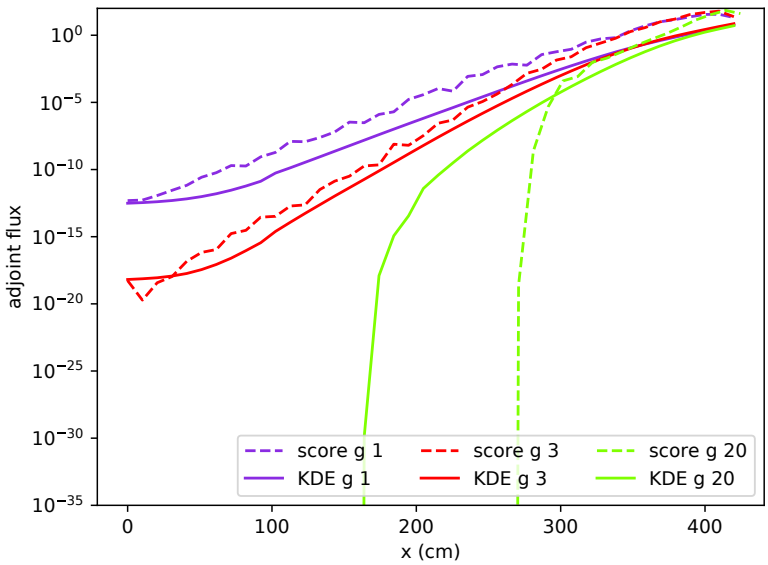


Figure C.5. Prediction of KDE compared to the adjoint score on a histogram after 1000 batches for three energy groups.

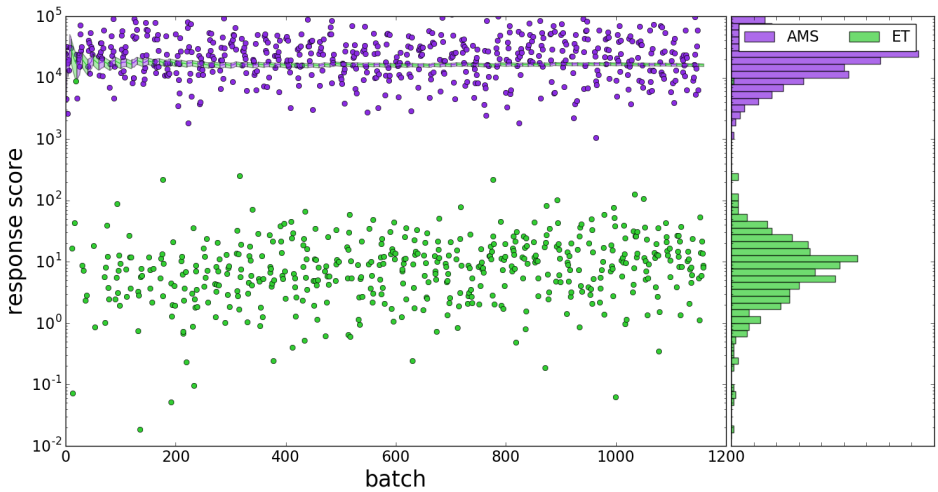


Figure C.6. Fixed step alternating strategy applied to the neutron-photon problem. Each exploration phase is performed with AMS, uses a spatial importance map and lasts 10 batches. Each exploitation phase is done with ET

good response value after 1000 batches. Recall that in the [Appendix B], the bootstrap scheme gave much faster convergence than the alternating scheme. Figure C.7 shows the result of the application of this strategy. Now, AMS also re-uses the KDE adjoint score. We observe that AMS succeeds in converging to the correct response value even with the KDE as an importance map. AMS even converges to a better asymptotic variance than the simple alternating scheme. However, the evolution of the exploitation phase does not seem to converge to the correct expected response. Our explanation is that the KDE has over-smoothed the densities, and killed the geometrical dependence of the problem. In this particular problem, thin slabs alternate. Smoothing the emission density from one slab where photons are preferentially emitted compared to the neighbouring slab seems to deteriorate the estimation of the adjoint score. A more thorough study should be made on the choice of the bandwidth parameter as was made on the mesh.

C.5 Conclusion

In this appendix, we tried to assess the efficiency of smoothing the adjoint score with Kernel Density Estimators. We first explained how the KDE was implemented in TRIPOLI-4®, and described the optimizations made to accelerate the construction of the KDE score. Then we applied the fixed-step alternating scheme to the strong attenuation benchmark. For this benchmark, we saw that the exponential transform with KDE converges faster than with the standard histogram, but with the cost of a higher asymptotic variance. We explained this by looking at the shape of the prediction of the KDE and comparing it with the classical score and observed a strong bias as a side effect of the gained smoothness. For the coupled problem, we did not manage to make the exponential transform converge towards the expected response value. This may be explained

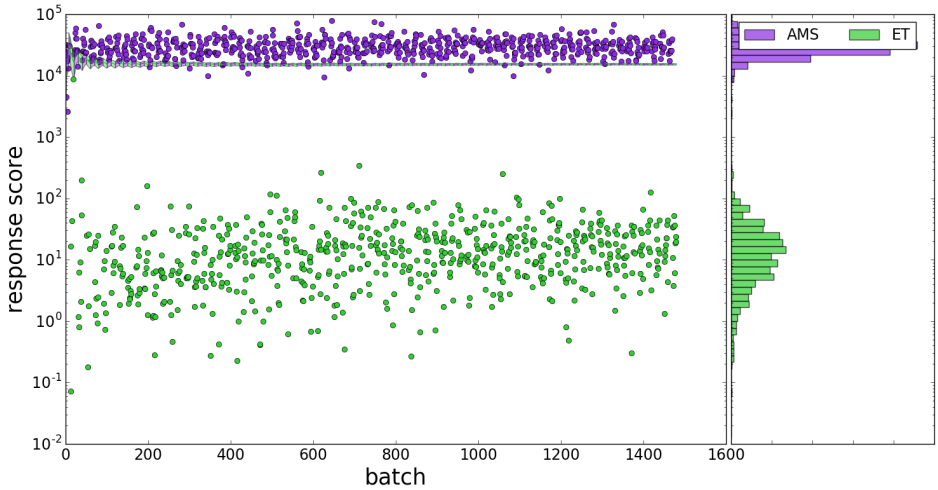


Figure C.7. Bootstrap scheme applied to the neutron-photon problem. Each exploration phase is made with AMS that starts with a spatial importance map. The adjoint KDE score is collected and projected over a mesh. Each exploitation phase alternates with the exponential transform. After the first two phases AMS re-uses the KDE score as an importance map

by the sensitivity of ET to the over-smoothed adjoint score.

université PARIS-SACLAY | ÉCOLE DOCTORALE PHENIICS

Titre : Acceleration de simulations Monte Carlo de transport de particules par génération adaptative de cartes d'importance.

Mots clés : Monte Carlo, radioprotection, réduction de variance, carte d'importance, apprentissage automatique, TRIPOLI-4®

Résumé : Les simulations Monte Carlo de transport de particules sont un outil incontournable pour l'étude de problèmes de radioprotection. Leur utillisation implique l'échantillonnage d'évenements rares grace à des méthodes de réduction de variance qui reposent sur l'estimation de la contribution d'une particule au détecteur. On construit cette estimation sous forme d'une *carte d'importance*.
L'object de cette étude est de proposer une stratégie qui permette de générer de manière adaptative des cartes d'importance durant la simulation Monte Carlo elle-même. Le travail a été réalisé dans le code de tranport des particules TRIPOLI-4®, dévelopé à la Direction de l'Énergie Nucléaire du CEA (Salay, France). Le coeur du travail a consisté à estimer le flux adjoint à partir des trajectoires simulées avec l'*Adaptive Multilevel Splitting*, une méthode de réduction de variance robuste.
Ce dévelopement a été validé à l'aide de l'intégration d'un module déterministe dans TRIPOLI-4®.
Trois stratégies sont proposées pour la réutilisation de ce score en tant que carte d'importance dans la simulation Monte Carlo. Deux d'entre elles proposent d'estimer la convergence du score adjoint lors de phases d'*exploitation*.
Ce travail conclut sur le lissage du score adjoint avec des méthodes d'apprentissage automatique, en se concentrant plus particulièrement sur les estimateurs de densité à noyaux.

Title : Accelerating Monte Carlo particle transport with adaptively generated importance maps.

Keywords : Monte Carlo, shielding, variance reduction, importance map, Machine Learning, TRIPOLI-4®

Abstract : Monte Carlo methods are a reference asset for the study of radiation transport in shielding problems. Their use naturally implies the sampling of rare events and needs to be tackled with variance reduction methods. These methods require the definition of an importance function/map.
The aim of this study is to propose an adaptive strategy for the generation of such importance maps during the Montne Carlo simulation. The work was performed within TRIPOLI-4®, a Monte Carlo transport code developped at the nuclear energy division of CEA in Saclay, France. The core of this PhD thesis is the implementation of a forward-weighted adjoint score that relies on the trajectories sampled with Adaptive Multilevel Splitting, a robust variance reduction method. It was validated with the integration of a deterministic module in TRIPOLI-4®.
Three strategies were proposed for the reintegration of this score as an importance map and accelerations were observed. Two of these strategies assess the convergence of the adjoint score during exploitation phases by evalutating the figure of merit yielded by the use of the current adjoint score. Finally, the smoothing of the importance map with machine learning algorithms concludes this work with a special focus on Kernel Density Estimators.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France