Contribution to ray-effect mitigation in discrete-ordinates based transport solvers

Matteo Falabino

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : I-MEP2 - Ingénierie - Matériaux, Mécanique, Environnement, Energétique, Procédés, Production
Spécialité : MEP : Mécanique des fluides Energétique, Procédés
Unité de recherche : CEA SACLAY, Laboratoire de Logiciels pour la Physique des Réacteurs

Contribution à la mitigation de l'effet de raie dans les solveurs de transport basés sur les ordonnées discrètes

Contribution to ray-effect mitigation in discrete-ordinates based transport solvers

Présentée par :

Matteo FALABINO

Direction de thèse :

Jean-François VIDAL Directeur de Recherche CEA-E6, Université Grenoble Alpes	Directeur de thèse
Daniele Sciannandrone CEA	Co-encadrant de thèse
Emiliano Masiello CEA	Co-encadrant de thèse

Rapporteurs :

Sandra DULLA
PROFESSEUR, Politecnico di Torino
Jean RAGUSA
PROFESSEUR, Texas A&M University

Thèse soutenue publiquement le **13 octobre 2022**, devant le jury composé de :

Jean-François VIDAL INGENIEUR HDR, CEA centre de Cadarache	Directeur de thèse
Grégoire ALLAIRE PROFESSEUR, Ecole Polytechnique de Paris	Examinateur
Adrien BIDAUD PROFESSEUR DES UNIVERSITES, Grenoble INP	Président
Sandra DULLA PROFESSEUR, Politecnico di Torino	Rapporteure
Jean RAGUSA PROFESSEUR, Texas A&M University	Rapporteur
Richard SANCHEZ DIRECTEUR DE RECHERCHE, CEA centre de Paris-Saclay	Examinateur

Invités :

Daniele Sciannandrone
INGENIEUR DOCTEUR, CEA Saclay
Emiliano Masiello
INGENIEUR DOCTEUR, CEA Saclay

Contribution to Ray-effect mitigation in Discrete-Ordinates based Transport Solvers

Matteo FALABINO

Director: Dr. Jean-François VIDAL

Reviewers: Dr. Sandra DULLA
Dr. Jean RAGUSA

Examiners: Dr. Grégoire ALLAIRE
Dr. Adrien BIDAUD
Dr. Richard SANCHEZ

Guests: Dr. Daniele SCIANNANDRONE
Dr. Emiliano MASIELLO

École Doctorale I-MEP2 - Ingénierie - Matériaux, Mécanique, Environnement, Energétique, Procédés, Production
Université Grenoble Alpes

This dissertation is submitted for the degree of
Doctor of Philosophy
13 October 2022

Acknowledgements

Needless to say, my dedication and my commitment alone would not have been enough to achieve my Ph.D. I also owe this personal achievement to the people I have spent these past years with. I want to thank all the members of SERMA, the laboratory of CEA Saclay that has welcomed me. I would like to thank my superiors, Fadhel Malouch and Loic De Carlan, for doing everything possible to put me in the best possible position to work, particularly during the pandemic period.

I thank my supervisors, Daniele Sciannandrone and Emiliano Masiello for their teaching and constant support. Thanks in particular to Daniele, who undoubtedly contributed to making me fall in love with this discipline. Thanks to Jean-Francois Vidal for his advices and supevision. Thank also Andrea Zoia for introducing me to CEA when I was still a student at the Politecnico di Torino.

I thank all the people I have met here in France and who have accompanied me on this journey over the last few years. I thank David and Emeline for sharing this experience with me. I thank Chantal for welcoming me in France when I still was unable to speak French. I thank my flatmates, Manon, Adèle, Thibault, Anaïs, Céline, Aurelie, Julien, Marie, Chloé, Théo, Delphine for brightening my days. I thank the small all-Italian community of SERMA, Andrea, Vito, Marco, Alberto, Francesco, Paolo for always making me feel at home. I would also like to thank also my Italian friends, Viola, Eros, Tony and Diego, who made me feel like I have never been far away.

Last but not least, I want to thank my family. Thank to my Mum and Dad for being my first supporters. A special thanks goes to Carola, my girlfriend, for always being by my side and sharing this experience with me.

Abstract

Over the years, a growing interest in the application of deterministic solvers to the calculation of radiation shielding problems has been observed. Deterministic solvers can provide approximate estimations with good accuracy and reasonable calculation times. However, the use of appropriate discretization methods is necessary. In fact, solvers based on the Discrete Ordinates (S_N) approximation, commonly employed in reactor core calculations, are often ineffective in representing particle streaming, thus yielding to results affected by the so-called ray-effect. The aim of this thesis is to develop numerical methods for the calculation of radiation shielding problems, capable of better representing streaming effects. This research has led to the development of a First Collision Source (FCS) method compatible with domain decomposition. Four different strategies for the integration of the flux emitted by point sources have been investigated. It has been shown how the FCS method is capable of providing results with good accuracy for problems of different sizes and degrees of heterogeneity. The most accurate results have been observed for the integration strategies based on the Method of Characteristics. In order to reconstruct the flux at the incoming boundaries of the subdomains, the FCS method required the introduction of an additional interpolative approximation based on equivalent cross sections. Tests have shown how the FCS method guarantees a precise error control, allowing to refine the interpolative approximation independently from the flux integration strategy. In order to mitigate the ray-effect generated by scattering sources, the MP_N method has also been developed. This is an angular discretization method based on the integral formulation of the transport equation, and employing a piecewise polynomial angular representation of the fluxes at the interfaces of the cartesian spatial regions. MP_N showed an error convergence about 2.5 times faster than S_N for relatively fine angular discretizations. Finally, the IDT solver, which implements these methods, has been verified on a realistic radiation shielding problem.

Keywords: Radiation shielding calculations, Neutral particles transport physics, Deterministic shielding calculations, Ray-effect mitigation.

Table of contents

List of figures

List of tables

Introduction

In the domain of neutral particle transport, radiation shielding calculations constitute a large class of problems which are of interest for all industries working with radioactive materials. These range from hospitals, performing medical treatments and imaging, to nuclear facilities, where the radiation containment is a key point for safety. Especially, problems of protection against neutron and gamma radiations in nuclear facilities include the evaluation of dose rates, neutron activation of structures, secondary sources of radiation, heating of components and sizing of shieldings. In this context, numerical calculations play a fundamental role allowing to study the physics of the problem prior to the actual irradiation. The main issue of radiation shielding calculations is providing accurate estimations within a reasonable calculation time. In this respect, the transport solvers currently available at CEA have some limitations: the main deterministic solvers have been developed with the aim of solving reactor core calculations, which are very different from radiation shielding problems. As a result, they generally fail in representing streaming effects, thus providing inaccurate estimations. Other solvers based on a Monte Carlo approach, instead, become very slow wherever the desired event, e.g. particles reaching a distant detector, is rare, which is often the case in radiation shielding calculations. Hence, the interest in developing new numerical methods, allowing for fast and accurate radiation shielding calculations.

IDT, [1], is a deterministic 3D Cartesian transport solver. It relies on the multi-group formalism and on the Discrete Ordinates (S_N) approximation and it uses a spatial discretization by diamond difference, nodal method and both step- and linear-short characteristics methods, [2]. IDT allows to solve calculations using the Domain Decomposition Method (DDM), and disposes of several acceleration techniques, such as Coarse Mesh Finite Difference (CMFD), Boundary-Projection Acceleration (BPA) and Diffusion Synthetic Acceleration (DSA), [1-3]. Over the years, IDT has been employed in reactor core applications and, only recently, in radiation shielding calculations. However, at present, IDT still has several limitations that prevent its use in real radiation shielding calculations. The most critical points are: the angular discretisation, which must be able to adequately take into account the flux anisotropy and the possible streaming effects; the scattering anisotropy, which is particularly relevant in the presence of highly forward-peaked scattering, and the high

dimensionality of radiation shielding problems. Among these, the angular discretization occupies a prominent role. In fact, when performing multi-dimensional discrete-ordinates calculations, if the angular refinement is not sufficiently fine, it can lead to solutions affected by numerical distortions, commonly known as the ray-effect.

The Monte Carlo method allows to estimate any physical quantity of interest by sampling a certain number of particle histories, and computing their average score to a given estimator. Tripoli-4®, [4], is the reference Monte Carlo code developed at CEA and, over the years, it has been extensively employed in shielding calculations. In some problems, typically when the detector is very far from the source or in the presence of preferential paths, the event that a particle scores at the detector is quite rare. In such cases, in order to obtain reliable responses in a reasonable computational time, the use of Variance Reduction techniques (VR) is mandatory. VR are often based on the "artificial" modification of the sampling laws to force particles to follow the paths that contribute the most to the desired scores. The Consistent Adjoint-Driven Importance Sampling (CADIS), [5], is one of the most effective VR techniques. The CADIS method consists in using the deterministic adjoint flux as importance function, which is then employed in the definition of the modified sampling laws. Thus, the calculation is divided into two steps: the first consists of a deterministic adjoint calculation aimed at estimating the importance function, while the second is the actual Monte Carlo calculation. The joint use of Tripoli-4 and IDT has already been investigated, and has given satisfactory results in terms of computation time, [6, 7].

Other deterministic solvers, such as ATTILA, [8], originally developed at LANL, and DENOVO, [9], developed at Oak Ridge, have a longer experience in radiation shielding calculations. ATTILA is a S_N solver for neutral and charged particle transport for three dimensional geometries. It uses the discontinuous finite element method on tetrahedral and hexahedral meshes, allowing to adapt to any geometry. The calculation of the first collision source and the last collided flux, acting respectively as a pre- and post-processing phases of the shielding transport calculations, is also available, [10]. DENOVO, [9], instead, is a S_N solver employing a Cartesian mesh with finite difference discretization, discontinuous linear finite elements or short step characteristics. It relies on a domain decomposition using the KBA method, [11], and it is able to perform the calculations of the first collision source. The numerical methods used by these solvers represent the state-of-the-art for deterministic shielding calculations. Besides these, several other strategies have been explored in the literature. The use of adaptive spatial and angular discretizations, for example, has been shown to be effective in reducing the number of degrees of freedom of the calculation, thus the computational cost. Further, strategies for the ray-effect mitigation, such as S_N-P_N equivalent methods, and angular finite elements have also been proposed as alternatives to the more common S_N to better represent streaming effects.

In summary, different methods for the resolution of radiation shielding calculations have been proposed in literature. Monte Carlo based codes are reliable since the quality of the estimation can be improved by simply increasing the number of particle replica, so the computational cost. However, they can be very expensive in large shielding calculations, making the use of VR techniques mandatory. On the other hand, deterministic transport solvers are typically faster, and the accuracy of the estimation is given by the approximations that are involved. However, discretisation methods capable of properly representing streaming effects have to be employed. A further development of IDT, useful in improving its accuracy when solving shielding calculations, leaves room for the opportunity to investigate new numerical methods.

This thesis consists in the research and development, within the IDT deterministic solver, of numerical methods capable of guaranteeing both precision and reasonable calculation times in radiation shielding calculations. In particular, a higher-order First Collision Source (FCS) method and a new angular discretization technique are developed. Then, both methods have been extended to the IDT domain decomposition framework. Finally, the effectiveness of the recent developments has been tested on a realistic shielding calculation.

Chapter 1 is devoted to a summary of the basics on which this thesis lies. In particular, the fundamental concepts of the neutron transport physics and the approximations that are generally done in deterministic transport solvers are presented. The main issues encountered when discretizing the angular variable in the presence of streaming effects are discussed, as well as the possible remedies that have been proposed in literature. Finally, the Domain Decomposition Method (DDM) is presented.

The objective of Chapter 2 is the development of a set of high-order First Collision Source techniques for the calculation of the flux emitted by localised sources, and the assessment of their effectiveness in solving radiation shielding calculations. At first, a brief literature review of the integration techniques for the uncollided flux is presented. The research resulted in the development of four different high-order FCS techniques. These are the Method of Characteristics (MOC), the Trajectory Splitting MOC (TS-MOC), the Monte Carlo MOC (MC-MOC) and the Quadrature Points (QP) method. The different FCS techniques have been analysed and compared in terms of accuracy and computational cost, for a set of benchmark problems. Finally, for each method the merits, the shortcomings, and the possible fields of application have been identified.

Chapter 3 deals with the extension of FCS method to domain decomposition. The difficulties encountered when performing ray tracing in domain decomposition are discussed and analysed. The Equivalent Cross Section approximation, allowing for the evaluation of the flux at the subdomains boundaries is presented, and an effective FCS method is defined.

Then, the scheduling algorithm, managing the communications and the ordering of the subdomains resolution is also defined. Finally, the accuracy and the computational cost of the FCS method in domain decomposition have been assessed.

Chapter 4 focus on the angular discretization of problems presenting streaming effects. At first, the need for a proper angular discretization is discussed, showing how the ray-effect can be also generated by scattering sources, and so be persistent even when using FCS. Then, the MP_N method, an angular discretization technique based on discontinuous polynomial interface fluxes, is presented. After a verification of the results, an extensive comparison between the MP_N and the S_N methods is performed. This includes an assessment of the accuracy and the convergence order of the MP_N method in a series of simple models, with different heterogeneity degrees and different scattering cross sections.

In Chapter 5, the newly developed methods are analysed and verified in a realistic radiation shielding calculation. First, the complete calculation scheme with DDM and FCS is summarized. Then, the parallel efficiency and the convergence properties of the S_N and MP_N methods in domain decomposition are quantified. Finally, the solver is verified by performing a complete calculation with FCS and DDM, comparing the results with a Monte Carlo reference.

A final section is devoted to the conclusions, and to the presentation of some perspectives for future research.

Chapter 1

Background

Most of the radiation shielding calculations can be traced to the problem of predicting a particle distribution over a certain portion of domain. All quantities of interest, such as reaction rates, kerma and dose, can be computed from this distribution. For neutral particles, such as neutrons and photons, the distribution is described by the linear Boltzmann Transport Equation.

This chapter is devoted to the introduction of the Transport Theory, the discipline that studies the Transport Equation, and so the techniques which are currently used to solve it. A specific focus is given on the numerical methods which are currently used in particle streaming calculations. At last, the Domain Decomposition method, allowing to run calculations on parallel distributed architecture is presented. In the present work only neutral particle transport is considered.

1.1 Elements of neutral particle transport physics

In transport theory, the particle distribution is defined over a six-dimensional phase space $(\mathbf{r}, \mathbf{\Omega}, E, t)$ where three dimensions describe the particle location $\mathbf{r}$, other two identify the flight direction $\mathbf{\Omega}$, another specify the particle energy E, and t is the time. The function that is generally used to represent this distribution is the particle current $\psi(\mathbf{r}, \mathbf{\Omega}, E, t)$, also known as particle flux in the neutron transport community. The flux measures the expected total path length travelled by all particles, at location $\mathbf{r}$, crossing a unit surface with orthogonal direction $\mathbf{\Omega}$, energy E, at time t, per unit area, per unit solid angle and per unit energy. Here we are not considering the flux time-dependence since our research deals with stationary problems only.

The notion of flux ψ is particularly convenient since it allows to evaluate any interaction rate, so any physical observable of interest R_i, by simply computing

$$R_i = \iiint_{\mathcal{X}_\mathcal{T}} d\mathbf{r}\ d\mathbf{\Omega}\ dE\ \psi(\mathbf{r}, \mathbf{\Omega}, E)\Sigma_i(\mathbf{r}, E), \tag{1.1}$$

where Σ_i is the macroscopic cross section of i-th interaction (yet to be defined), and $\mathcal{X}_\mathcal{T}$ is the target domain. By target we mean a portion of the phase space, defined as an interval in space, angle and energy, on which R_i is evaluated.

1.1.1 Definition of cross section

Macroscopic cross sections, indicated with Σ_i, express an interaction probability per unit length travelled by particles. These are specific to each interaction type, meaning that each reaction probability is represented by a different cross section. For an isotropic medium, $\Sigma(\mathbf{r}, E)$ depend uniquely on the medium properties and on the incident particle energy E. Macroscopic cross sections are evaluated as

$$\Sigma_i(\mathbf{r}, E) = \sum_j N_j(\mathbf{r})\sigma_{i,j}(\mathbf{r}, E),$$

where i identifies the reaction type, and index j runs over the different atomic species present with atomic concentration N_j. With $\sigma_{i,j}$ we indicate instead the microscopic cross sections, whose values are available in nuclear data libraries. The σ_i are evaluated by combining experimental measurements and the results of physical models. These are stored for a very large number of energy values (up to $\sim 10^5$) in such a way to well approximate the continuous energy dependence.

Among all macroscopic cross sections, the most important ones are:

- the double differential scattering cross section $\Sigma_s(\mathbf{r}, \mathbf{\Omega'} \cdot \mathbf{\Omega}, E' \to E)$, as the probability, per unit path length travelled, that a particle moving in direction $\mathbf{\Omega'}$ with energy E' collide and is scattered along direction $\mathbf{\Omega}$, with energy E. Scattering cross sections account for a large number of reactions, such as elastic and inelastic scattering for neutrons, photoelectric interactions, Compton scattering and pair production for photons;

- the fission cross section $\Sigma_f(\mathbf{r}, E)$, which is the probability, per unit path length travelled, that the particle with energy E is captured by a nucleus and gives a fission reaction. The $\Sigma_f(\mathbf{r}, E)$ is typically multiplied by the fission spectrum $\frac{\chi(E)}{4\pi}$ which gives the averaged distribution of the neutrons emitted by fission;

- the total cross section Σ, which is the probability, per unit path length travelled, that a particle undergoes any interaction.

We remark how the dependence of Σ_s on angles $\mathbf{\Omega}$ and $\mathbf{\Omega'}$ restricts to the cosine between the two (the cosine of the deflection angle) because of the assumption of isotropic medium. This assumption is valid in general but it may not be the case for some specific problems, e.g. photons impinging on a mono-crystalline shield, requiring a proper treatment of the angular dependence. This particular cases are out of the purpose of this work.

1.1.2 The Neutron Transport Equation

The Neutron Transport Equation (NTE) mathematically describes the particle distribution over the spatial, angular and energy domain. This is derived from the Boltzmann equation for gas kinetics under the hypothesis that neutral particles interacts with the medium nuclei uniquely, while collisions between particles are neglected. The steady-state integro-differential formulation of the NTE is

$$\mathcal{L}\psi(\mathbf{r},\mathbf{\Omega},E) = \mathcal{H}\psi(\mathbf{r},\mathbf{\Omega},E) + \mathcal{F}\psi(\mathbf{r},\mathbf{\Omega},E) + Q_{ext}(\mathbf{r},\mathbf{\Omega},E) \quad \text{with } (\mathbf{r},\mathbf{\Omega},E) \in \mathcal{X}, \tag{1.2}$$

where $\mathcal{X} = \{\mathbf{r} \in \mathcal{D}, \mathbf{\Omega} \in \mathcal{S}^2, E \in \mathbb{R}^+\}$ is the phase space, $\mathcal{D}$ is the spatial domain, and $\mathcal{S}^2$ the surface of the unit sphere. $\mathcal{L}$ is the displacement-plus-removal operator, $\mathcal{H}$ is the scattering (or transfer) operator, $\mathcal{F}$ is the fission operator, and Q_{ext} is the external source, [12].

The NTE takes the form of a particle balance on the differential element $d\mathbf{r}d\mathbf{\Omega}dE$ around point $(\mathbf{r},\mathbf{\Omega},E)$ with the first term,

$$\mathcal{L}\psi(\mathbf{r},\mathbf{\Omega},E) = \left[\mathbf{\Omega}\cdot\nabla + \Sigma(\mathbf{r},E)\right]\psi(\mathbf{r},\mathbf{\Omega},E), \tag{1.3}$$

accounting for particles streaming in direction $\mathbf{\Omega}$ or undergoing a collision, and the second,

$$\mathcal{H}\psi(\mathbf{r},\mathbf{\Omega},E) = \iint d\mathbf{\Omega}'dE'\Sigma_s(\mathbf{r},\mathbf{\Omega}'\cdot\mathbf{\Omega},E'\to E)\psi(\mathbf{r},\mathbf{\Omega}',E'), \tag{1.4}$$

representing particles that, undergoing a collision in a point $(\mathbf{r},\mathbf{\Omega}',E')$ of the phase space, are emitted along direction $\mathbf{\Omega}$ at energy E. The fission term,

$$\mathcal{F}\psi(\mathbf{r},\mathbf{\Omega},E) = \frac{1}{4\pi}\sum_{j=1}^{N_f}\chi_j(E)\iint d\mathbf{\Omega}'dE'\nu\Sigma_{f,j}(\mathbf{r},E')\psi(\mathbf{r},\mathbf{\Omega}',E'), \tag{1.5}$$

accounts instead for particles emitted by fission reactions with one of the j-th fissile isotopes, having energy spectrum χ_j. The factor $1/4\pi$ is the probability distribution of the isotropic fission neutron emission, while ν is the average number of neutrons emitted per fission. The fission term, which is typically present in nuclear reactor physics applications, may be null in radiation shielding calculations. In this work we will account for sub-critical systems only, i.e. with multiplication factor

$$k = \frac{\|\mathcal{F}\psi\|}{\|(\mathcal{L}-\mathcal{H})\psi\|} < 1,$$

where k is defined as the eigenvalue with the largest real part of the associated homogeneous equation, [12],

$$(\mathcal{L}-\mathcal{H})\psi(\mathbf{r},\mathbf{\Omega},E) = \frac{1}{k}\mathcal{F}\psi(\mathbf{r},\mathbf{\Omega},E) \quad \text{with } (\mathbf{r},\mathbf{\Omega},E) \in \mathcal{X}, \tag{1.6}$$

The boundary conditions

In order to solve Eq. (1.2), a set of conditions have to be imposed on the domain boundaries. This can be defined in many ways in agreement with the physical and geometrical properties of the problem. Boundary conditions consists in imposing relations for the flux of particles entering the domain,

$$\psi^-(\mathbf{x}) = \beta(\mathbf{x}' \to \mathbf{x})\psi^+(\mathbf{x}') + \psi^-_{inc}(\mathbf{x}) \quad \text{with } \mathbf{x} \in \partial\mathcal{X}^- \text{ and } \mathbf{x}' \in \partial\mathcal{X}^+, \tag{1.7}$$

where $\mathbf{x} = (\mathbf{r}, \mathbf{\Omega}, E)$ is a point in the phase space belonging to $\partial\mathcal{X}^- = \{\mathbf{r} \in \partial\mathcal{D}, \mathbf{\Omega} \in \mathcal{S}^2 | \mathbf{\Omega}\cdot\mathbf{n} < 0, E \in \mathbb{R}^+\}$. The incoming boundary flux ψ^- can be either imposed as part of the problem data, i.e. $\psi^- = \psi^-_{inc}$, or defined by mean of an albedo operator β, acting on the outgoing boundary flux ψ^+, defined over phase space $\partial\mathcal{X}^+ = \{\mathbf{r} \in \partial\mathcal{D}, \mathbf{\Omega} \in \mathcal{S}^2 | \mathbf{\Omega} \cdot \mathbf{n} > 0, E \in \mathbb{R}^+\}$. ∂D and $\mathbf{n}$ denote the domain boundaries and the outgoing normal direction, respectively. Boundary conditions can be set in order to reduce the domain size by exploiting the possible geometrical symmetries of the problem. The most frequently used boundary conditions are the vacuum condition, namely

$$\psi^-(\mathbf{r}, \mathbf{\Omega}, E) = 0,$$

and specular reflection condition, i.e.

$$\psi^-(\mathbf{r}, \mathbf{\Omega}, E) = \psi(\mathbf{r}, \mathbf{\Omega} - 2\mathbf{n}(\mathbf{\Omega} \cdot \mathbf{n}), E).$$

The integral formulation of the Neutron Transport Equation

The integro-differential NTE is not the only mathematical formulation of the neutral particle transport problem. By inverting the displacement-plus-removal operator in Eq. (1.2) along a characteristic line identified by a point $\mathbf{r} \in \mathcal{D}$ and direction $\mathbf{\Omega}$ (see Fig. 1.1), it is possible to derive

$$\begin{aligned} \psi(\mathbf{r} + t\mathbf{\Omega}, \mathbf{\Omega}, E) &= \psi(\mathbf{r}, \mathbf{\Omega}, E) e^{-\int_0^t dt' \ \Sigma(\mathbf{r}+t'\mathbf{\Omega}, E)} \\ &+ \int_0^t dt' \ Q(\mathbf{r} + t'\mathbf{\Omega}, \mathbf{\Omega}, E) e^{-\int_{t'}^t dt'' \ \Sigma(\mathbf{r}+t''\mathbf{\Omega}, E)} \quad \text{with } \mathbf{t} \in \mathbb{R}^+, \end{aligned} \tag{1.8}$$

which takes the name of integral formulation of the Neutron Transport Equation, [12]. The equation describes the flux in each point belonging to the trajectory, $\mathbf{r} + t\mathbf{\Omega}$, as the sum of two contributions. The first one accounts for particles incoming from a generic point $\mathbf{r}$ in which the flux is known. The second contribution accounts instead for particles that are emitted along the characteristic line $\mathbf{r}' = \mathbf{r} + t'\mathbf{\Omega}$ in direction $\mathbf{\Omega}$. In our notation, $t' \in [0, t]$ is the coordinate measuring the distance between two points along the characteristic line, namely $t' = |\mathbf{r} - \mathbf{r}'|$. The term Q is the emission density, defined as the sum of the particle

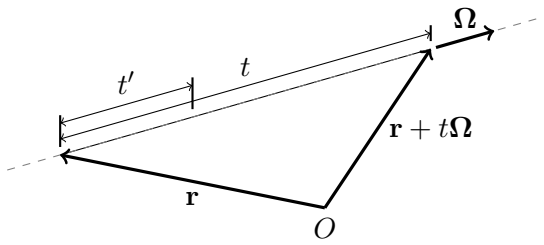


Fig. 1.1 Illustration of a characteristic line.

source terms

$$Q(\mathbf{r},\mathbf{\Omega},E) = \mathcal{H}\psi(\mathbf{r},\mathbf{\Omega},E) + \mathcal{F}\psi(\mathbf{r},\mathbf{\Omega},E) + Q_{ext}. \tag{1.9}$$

Both the incoming flux at $\mathbf{r}$ and the emission density are multiplied by an exponential term which corresponds to the probability for the particle emitted at $\mathbf{r}$ of reaching $\mathbf{r}+t\mathbf{\Omega}$ without undergoing any collision. The integral in the exponent,

$$\tau(\mathbf{r},\mathbf{\Omega},t,E) = \int_0^t dt'\ \Sigma(\mathbf{r}+t'\mathbf{\Omega},E), \tag{1.10}$$

takes the name of optical distance and physically represents a path length, measured in terms of particle mean free paths $1/\Sigma$.

Adjoint Neutron Transport Equation

For certain applications, such as perturbation studies and the speed up of Monte Carlo calculations with deep flux attenuation, it can be useful to solve the adjoint Neutron Transport Equation, [12]. We introduce the inner product

$$(f,g) = \iiint_{\mathcal{X}} d\mathbf{r}d\mathbf{\Omega}dE\ f(\mathbf{r},\mathbf{\Omega},E)g(\mathbf{r},\mathbf{\Omega},E),$$

where f and g are two well-behaved functions, and $\mathcal{B}$ is a generic operator. The adjoint operator $\mathcal{B}^\dagger$ is defined by relation

$$(\mathcal{B}f,g) = \left(f,\mathcal{B}^\dagger g\right). \tag{1.11}$$

We now consider f being the angular flux ψ, and $\mathcal{B}$ as the Boltzmann operator, defined as $\mathcal{B} = \mathcal{L} - \mathcal{H} - \mathcal{F}$. By using Eq. (1.11), we are able to define the adjoint operator, $\mathcal{B}^\dagger$, and the adjoint flux, $\psi^\dagger = g$, as the solution of the adjoint NTE,

$$\begin{aligned}\left[-\mathbf{\Omega}\cdot\nabla + \Sigma(\mathbf{r},E)\right]\psi^\dagger(\mathbf{r},\mathbf{\Omega},E) &= \iint d\mathbf{\Omega}'dE'\ \Sigma_s(\mathbf{r},\mathbf{\Omega}\cdot\mathbf{\Omega}',E\to E')\psi^\dagger(\mathbf{r},\mathbf{\Omega}',E') \\ &+ \sum_{j=1}^{N_f}\nu\Sigma_{f,j}(\mathbf{r},E)\iint d\mathbf{\Omega}'dE'\ \frac{1}{4\pi}\chi_j(E')\psi^\dagger(\mathbf{r},\mathbf{\Omega}',E') + Q^\dagger_{ext} \quad \text{with } \mathbf{x}\in\mathcal{X},\end{aligned} \tag{1.12}$$

with boundary conditions,

$$\psi^{\dagger,+}(\mathbf{x}) = \beta(\mathbf{x}' \to \mathbf{x})\psi^{\dagger,-}(\mathbf{x}') + \psi^{\dagger,+}_{inc}(\mathbf{x}) \quad \text{with } \mathbf{x} \in \partial\mathcal{X}^{+} \text{ and } \mathbf{x}' \in \partial\mathcal{X}^{-}. \tag{1.13}$$

By sobstituting, Eq. (1.11) for the generic functions f and g becomes

$$\left(\mathcal{B}\psi, \psi^{\dagger}\right) = \left(\psi, \mathcal{B}^{\dagger}\psi^{\dagger}\right). \tag{1.14}$$

Moreover, since $\mathcal{B}\psi = Q_{ext}$ and $\mathcal{B}^{\dagger}\psi^{\dagger} = Q^{\dagger}_{ext}$, with $Q^{\dagger}_{ext}$ as the adjoint external source, Eq. (1.14) becomes

$$\left(Q_{ext}, \psi^{\dagger}\right) \;=\; \left(\psi, Q^{\dagger}_{ext}\right). \tag{1.15}$$

The main differences between adjoint Eq. (1.12) and Eq. (1.2) lay in the energetic and angular coupling provided by the scattering and fission operators, and the sign of the streaming operator. In order to appreciate the physical meaning of the adjoint flux, we should first recall the definition of detector response R_i, as presented by Eq. (1.1). The detector response R_i is a physical quantity of interest, defined by the scalar product

$$R_i = (\psi, \Sigma_i) = \iiint_{\mathcal{X}} d\mathbf{r}\; d\mathbf{\Omega}\; dE\; \psi(\mathbf{r}, \mathbf{\Omega}, E)\Sigma_i(\mathbf{r}, E), \tag{1.16}$$

with cross section $\Sigma_i(\mathbf{r}, E)$ being non-zero in a limited portion of the phase space $\mathcal{X}_{\mathcal{T}}$, called detector. If the adjoint NTE is solved for a value of the adjoint external source $Q^{\dagger}_{ext} = \Sigma_i$, we have that

$$R_i = (\psi, \Sigma_i) = \left(Q_{ext}, \psi^{\dagger}\right). \tag{1.17}$$

Because of Eq. (1.17), the adjoint flux can be interpreted as a measure of the particle importance in contributing to a detector response, [12]. The larger the adjoint flux $\psi^{\dagger}(\mathbf{r}, \mathbf{\Omega}, E)$, the larger the expected contribution that a particle in position $\mathbf{r}$, direction $\mathbf{\Omega}$ with energy E, will give to the detector response R_i.

The notion of importance plays a key role in Monte Carlo calculations since it quantifies the potential contribution that a particle may provide to a given score (measuring the detector response). Since the physical meaning of the adjoint flux coincides with the importance, the approximate solution of the adjoint transport problem is of considerable interest for variance reduction techniques in Monte Carlo, [6, 7].

1.2 Solving the Neutron Transport Equation

The main difficulty in solving the NTE is dealing with the high dimensionality of the problem. Hence, two strategies: the Monte Carlo approach allows to integrate the particle flux, or any physical quantity of interest, by computing the score of a certain number of independently-

simulated particle histories. By sampling the particle histories from the continuous probability distributions governing the physics of the problem, no other mathematical approximation is required. Instead, the deterministic approach requires the discretization of the energy, the angular and the spatial variables.

In common practice, energy discretization is performed by means of the so-called multigroup formalism. The angular variable is treated by first approximating the angular coupling within the scattering operator, and then by approximating the flux angular distribution. For this second step, the techniques that are more frequently used are the method of Spherical Harmonics (P_N) and the Discrete Ordinates (S_N) method. As concerns the spatial variable, different discretization methods exist. The spatial discretization should allow for a faithful representation of the medium properties within the model, provide a proper discretized representation of the spatial dependence of the flux, and finally allow for a cost-efficient numerical solution.

In the following sections, we examine these approximations individually, and we present how they are applied to the NTE. Particular attention is paid to the techniques currently used within the IDT solver, which are the starting point of this research.

1.2.1 The multigroup approximation

The multigroup formalism is based on the discretization of the energy domain into a set of energy intervals, called groups. In practical applications, the energy domain may be quite large: it may range from tens of meV to several MeV for neutrons, and from keV up to hundreds of GeV for photons. By convention, groups are numbered starting from high energy values downwards. Each group, here identified by index g, is defined as the energy interval $[E_g, E_{g+1}]$. The approximation consists in integrating the transport equation over each of these energy intervals, and then solving for the group integrated flux

$$\psi_g(\mathbf{r}, \mathbf{\Omega}) = \int_{E_{g+1}}^{E_g} dE \psi(\mathbf{r}, \mathbf{\Omega}, E) \quad g = 1, N_g, \tag{1.18}$$

where N_g is the total number of groups.

The multigroup approximation implies the use of group-constant cross sections, known by the name of multigroup cross sections. These are obtained by imposing conservation relations on the in-group reaction rates. In detail, total multigroup cross section and the differential multigroup scattering cross section are defined respectively by the averages

$$\Sigma_g(\mathbf{r}, \mathbf{\Omega}) = \frac{\int_{E_{g+1}}^{E_g} dE \ \psi(\mathbf{r}, \mathbf{\Omega}, E) \Sigma(\mathbf{r}, E)}{\int_{E_{g+1}}^{E_g} dE \ \psi(\mathbf{r}, \mathbf{\Omega}, E)}, \tag{1.19}$$

$$\Sigma_{g,g'}(\mathbf{r},\mathbf{\Omega},\mathbf{\Omega}') = \frac{\int_{E_{g+1}}^{E_g} dE \int_{E_{g'+1}}^{E_{g'}} dE' \; \psi(\mathbf{r},\mathbf{\Omega}',E')\Sigma_s(\mathbf{r},\mathbf{\Omega}'\cdot\mathbf{\Omega},E'\to E)}{\int_{E_{g'+1}}^{E_{g'}} dE' \; \psi(\mathbf{r},\mathbf{\Omega}',E')}, \tag{1.20}$$

where the angular flux plays the role of integral weight function. In the same way, the group-constant fission cross section is defined as

$$\nu\Sigma_{f,j,g}(\mathbf{r}) = \frac{\int_{E_{g+1}}^{E_g} dE \; \Phi(\mathbf{r},E)\nu\Sigma_{f,j}(\mathbf{r},E)}{\int_{E_{g+1}}^{E_g} dE \; \Phi(\mathbf{r},E)}, \quad j=1,\ldots,N_f, \tag{1.21}$$

for each of the N_f fissile isotopes, with Φ the scalar flux,

$$\Phi(\mathbf{r},E) = \int_{4\pi} d\mathbf{\Omega}\psi(\mathbf{r},\mathbf{\Omega},E).$$

The evaluation of the multigroup cross sections leads to two main issues: at first, the integrals can not be evaluated exactly since this would imply the knowledge of the flux itself, which is the unknown. Secondly, the use of the exact formulation of the multigroup cross sections would introduce an undesired angular dependence in the cross sections.

In practice, Eqs (1.19) and (1.20) are not evaluated analytically but approximately. The angular dependence is typically neglected by using the scalar flux as weight function, rather than the angular flux. If cross sections presents mild variations within the group, the space dependence is also neglected, and the flux is replaced by a properly chosen energy spectrum. For heavy and intermediary isotopes having resonances, however, one has to use a problem dependent flux which is obtained via approximations using the self-shielding method. A brief introduction to self-shielding is proposed in the following.

As a matter of fact, the error involved by the approximation is greater the larger the amplitude of the energy intervals. At the same time, the greater the number of groups, the higher the calculation cost. The number of groups is therefore a free parameter that is chosen by establishing a compromise between computational cost and accuracy.

The self-shielding formalism

The multi-group cross sections are evaluated through nuclear data libraries processing systems such as NJOY, [13]. These operate on the microscopic cross sections, rather than macroscopic. Macroscopic cross sections are typically evaluated by the deterministic solver, for each media, only when the energy discretization have been already applied. Each of the multigroup microscopic cross section,

$$\sigma_{i,j,g}(\mathbf{r}) = \frac{\int_{E_{g+1}}^{E_g} dE \; \Phi(\mathbf{r},E)\sigma_{i,j}(E)}{\int_{E_{g+1}}^{E_g} dE \; \Phi(\mathbf{r},E)}, \tag{1.22}$$

with the j-index denoting the isotope and i the reaction type, is evaluated by neglecting the spatial dependence and approximating the flux with an appropriate energy spectrum. Neutron libraries are generated using a Maxwellian spectrum for thermal groups, a $1/E$ spectrum for slowing-down regimes, and a fission spectrum for fast groups. For photons, instead, a $1/E$ energy spectrum is commonly used, [14].

Equation (1.22) is evaluated using as $\sigma_{i,j}$ the pointwise cross sections (PENDF), also evaluated by NJOY, through a series of calculations including interpolations of ENDF (Evaluated Nuclear Data File) libraries, [15], and Doppler broadening at the given temperature.

The calculation of the multigroup cross sections is a relevant issue for reactor calculations due to the characteristics of the neutron-matter cross sections, and in particular due to the presence of resonance isotopes such as ^{238}U. The effect of resonances is visible as a peaked increase in the microscopic cross section, even of several orders of magnitude, around a given energy value. For the same interval, a simultaneous sinking of the neutron flux can be observed. This physical phenomenon takes the name of self-shielding and the resultant reaction rates exhibit a much milder variation with respect to the punctual cross section values. Self-shielding is typically accounted by modern deterministic codes using appropriated energy meshes, and performing a pre-treatment of cross sections of resonant isotopes by mean of ad-hoc methods able to properly model resonances, [16, 17].

Resolution scheme of the multigroup Neutron Transport Equation

Multigroup approximation transforms the NTE, Eq. (1.2) into a set of mono-kinetic equations,

$$\mathcal{L}_g\psi_g = \sum_{g'=1}^{N_g} \mathcal{H}_{g,g'}\psi_{g'} + \sum_{g'=1}^{N_g} \mathcal{F}_{g,g'}\psi_{g'} + Q_{ext,g} \quad g = 1, N_g, \tag{1.23}$$

coupled together by the transfer and by the fission operator. $\mathcal{L}_g\psi_g$, $\mathcal{H}_{g,g'}\psi_{g'}$ and $Q_{ext,g}$ are the displacement-plus-removal term for the group g,

$$\mathcal{L}_g\psi_g = (\mathbf{\Omega}\cdot\nabla + \Sigma_g(\mathbf{r}))\psi_g(\mathbf{r},\mathbf{\Omega}), \tag{1.24}$$

the transfer term from group g' to g,

$$\mathcal{H}_{g,g'}\psi_{g'} = \int_{4\pi} d\mathbf{\Omega}'\Sigma_{g,g'}(\mathbf{r},\mathbf{\Omega}'\cdot\mathbf{\Omega})\psi_{g'}(\mathbf{r},\mathbf{\Omega}'), \tag{1.25}$$

and the group constant source

$$Q_{ext,g} = \int_{E_{g+1}}^{E_g} dE\ Q_{ext}(\mathbf{r},\mathbf{\Omega},E), \tag{1.26}$$

respectively. $\mathcal{F}_{g,g'}\psi_{g'}$ is the fission operator

$$\mathcal{F}_{g,g'}\psi_{g'} = \frac{1}{4\pi}\sum_{j=1}^{N_f}\chi_{j,g}\int d\mathbf{\Omega}'\nu\Sigma_{f,j,g'}(\mathbf{r})\psi_{g'}(\mathbf{r},\mathbf{\Omega}'), \tag{1.27}$$

giving the fission neutrons appearing in group g from fissions occurring in group g', with $\chi_{j,g} = \int_{E_{g+1}}^{E_g} dE\ \chi_j(E)$.

Since too large to be directly inverted, the system of equations is solved by iterating on the source terms. In the presence of fissile isotopes, a first iterative scheme, called outer iterations, is employed in order to converge the solution on the fission source. Each iteration (o),

$$\mathcal{L}_g\psi_g^{(o+1)} - \sum_{g'}\mathcal{H}_{g,g'}\psi_{g'}^{(o+1)} = \sum_{g'}\mathcal{F}_{g,g'}\psi_{g'}^{(o)} + Q_{ext,g} \quad g = 1, N_g. \tag{1.28}$$

consists in solving a whole multigroup calculation for a fixed fission source. The fission source is then updated, for each energy group g, at the end of each iteration.

Within each outer iteration, a second iterative scheme acting on the scattering source is used. This is conventionally made of two levels: the first, known as thermal iterations, perform Gauss-Seidel iterations over the scattering source from group-to-group. The second level instead, known as inner iterations, consists of fixed-point iterations on the self-scattering source. Whenever neutral particle transport takes place in a pure slowing down regime, this scattering matrix is lower triangular. That is, the only groups that contribute to the scattering source of the g group are the higher energy groups $g' < g$. In the pure slowing down regime, the system of equations, Eq. (1.23), is solved exactly by addressing equations in order, starting from the group with the highest energy downwards, and progressively updating the scattering source of the lower energy groups. If up-scattering is present, as usually is the case for neutrons at low energies, particles colliding in group g' may contribute to the source term of higher energy groups $g < g'$. This results in a transfer matrix presenting non-zero terms on the above-diagonal part. The dependence given by up-scattering is solved by thermal iterations using a Gauss-Seidel scheme. Hence, per-group equations are solved one after the other, like for the pure slowing down regime, and keeping constant the up-scattering source. Once the iteration is completed, the up-scattering source is updated and the process is repeated starting from the highest energy group affected by up-scattering. By rewriting Eq. (1.23) indicating with (t) the iterative index of the thermal iterations, the process can be described as

$$(\mathcal{L}_g - \mathcal{H}_{g,g})\psi_g^{(t+1)} = \sum_{g'<g}\mathcal{H}_{g,g'}\psi_{g'}^{(t+1)} + \sum_{g'>g}\mathcal{H}_{g,g'}\psi_{g'}^{(t)} + Q_{ext+f,g}^{(o+1)} \quad g = 1, N_g. \tag{1.29}$$

where the first two terms on the RHS represent the contribution of down-scattering and up-scattering of the transfer operator respectively, and

$$Q_{ext+f,g}^{(o+1)} = \sum_{g'} \mathcal{F}_{g,g'} \psi_{g'}^{(o)} + Q_{ext,g}.$$

While solving Eq. (1.29) for a fixed scattering source from other groups, the $\mathcal{L}_g - \mathcal{H}_{g,g}$ operator still have to be inverted. This is done solving the mono-kinetic transport problem,

$$\mathcal{L}_g \psi_g^{(i+1)} = \mathcal{H}_{g,g} \psi_g^{(i)} + Q_g^{(t+1)} \quad g = 1, N_g, \tag{1.30}$$

by updating, at each inner iteration (i), the self-scattering source $\mathcal{H}_{g,g}\psi_g^{(i)}$. Here, (i) is the inner iterations index while $Q_g^{(t+1)}$ is defined by the sum of the scattering source from other groups plus fission and external source,

$$Q_g^{(t+1)} = \sum_{g'<g} \mathcal{H}_{g,g'} \psi_{g'}^{(t+1)} + \sum_{g'>g} \mathcal{H}_{g,g'} \psi_{g'}^{(t)} + \sum_{g'} \mathcal{F}_{g,g'} \psi_{g'}^{(o)} + Q_{ext,g}. \tag{1.31}$$

Furthermore, inner iterations are also responsible of updating the incoming fluxes at the domain boundaries by applying boundary conditions, as in Eq. (1.7).

In conclusion, the multigroup approximation transforms the continuous energy transport equation into a set of mono-kinetic equations solved, at each iteration, for a fixed source term. From now on, the group index g will be dropped unless necessary.

1.2.2 The treatment of the transfer operator

The treatment of the angular dependence of the transfer operator is generally carried out under the hypothesis of isotropic media. That is to say that the scattering cross sections is assumed varying with the deflection cosine $\hat{\mu} = \mathbf{\Omega'} \cdot \mathbf{\Omega}$, rather than with directions $\mathbf{\Omega'}$ and $\mathbf{\Omega}$ separately. The dependence over $\hat{\mu}$ is generally expressed using a series expansion over orthogonal Legende polynomials, [12]. The scattering cross section is rewritten as

$$\Sigma_s(\hat{\mu}, \mathbf{r}) \approx \frac{1}{2} \sum_k P_k(\hat{\mu}) \Sigma_{s,k}(\mathbf{r}), \tag{1.32}$$

where P_k is the Legendre polynomial of order k, and $\Sigma_{s,k}$ are the moments of the expansion multiplied by the normalization coefficient $(2k+1)$, i.e.

$$\Sigma_{s,k}(\mathbf{r}) = (2k+1) \int_{-1}^{1} d\hat{\mu} P_k(\hat{\mu}) \Sigma_s(\mathbf{r}, \hat{\mu}). \tag{1.33}$$

In practice, the summation over moments k is truncated at order K, taking the name of anisotropy order of the scattering operator.

By introducing the addition theorem,

$$P_k(\mathbf{\Omega}' \cdot \mathbf{\Omega}) = \frac{1}{2\pi} \sum_{l=-k}^{k} A_{k,l}(\mathbf{\Omega}) A_{k,l}(\mathbf{\Omega}'), \tag{1.34}$$

Legendre polynomials can be rewritten as the sum of spherical harmonics functions. In this document we consider as harmonic functions the Schmidt's real-valued spherical harmonics,

$$A_{k,l}(\mathbf{\Omega}) = \begin{cases} (-1)^l \sqrt{\frac{2(k-|l|)!}{(k+|l|)!}} P_k^{|l|}(\mu) sin(|l|\varphi) & \text{if } l < 0, \\ P_k^0(\mu) & \text{if } l = 0, \\ (-1)^l \sqrt{\frac{2(k-l)!}{(k+l)!}} P_k^l(\mu) cos(l\varphi) & \text{if } l > 0, \end{cases} \tag{1.35}$$

where θ and φ are the polar and the azimuthal components of direction $\mathbf{\Omega}$, $\mu = cos(\theta)$ is the polar cosine, and P_k^l are the associated Legendre functions, defined by

$$P_k^l(\mu) = \frac{(-1)^l}{2^k k!} (1-\mu^2)^{l/2} \frac{d^{k+l}}{d\mu^{k+l}} ((\mu^2-1)^k). \tag{1.36}$$

By substituting the cross section expansion of Eq. (1.32), and the addition theorem, Eq. (1.34), into Eq. (1.25), the scattering kernel (for any couple of group indexes g and g') can be rewritten as

$$\mathcal{H}\psi = \sum_{k=0}^{K} \Sigma_{s,k}(\mathbf{r}) \sum_{l=-k}^{k} A_{k,l}(\mathbf{\Omega}) \phi_{k,l}(\mathbf{r}). \tag{1.37}$$

where $\phi_{k,l}$ are the angular moments of the flux, defined by the integral

$$\phi_{k,l}(\mathbf{r}) = \Big(A_{k,l}, \psi\Big) = \frac{1}{4\pi} \int_{4\pi} d\mathbf{\Omega} A_{k,l}(\mathbf{\Omega}) \psi(\mathbf{r}, \mathbf{\Omega}). \tag{1.38}$$

Schmidt's Spherical harmonics are a family of not-normalized orthogonal functions. They constitute a basis for the angular variable $\mathbf{\Omega}$ over the unit sphere $\mathcal{S}^2 = \{(\theta, \varphi) \mid \theta \in [0, \pi],\ \varphi \in [0, 2\pi]\}$, and they obey to relation

$$\frac{1}{4\pi} \int_{4\pi} d\mathbf{\Omega} A_{k,l}(\mathbf{\Omega}) A_{k',l'}(\mathbf{\Omega}) = \frac{1}{2k+1} \delta_{k,k'} \delta_{l,l'}. \tag{1.39}$$

An almost isotropic scattering such as the one occurring between neutrons and heavy nuclei can be described using a low order expansion in Legendre polynomials (low K value). On the other hand, interactions between neutrons and light nuclei (e.g. hydrogen), as well as many photons-matter interactions, are characterized by highly anisotropic scattering, thus requiring a high order of anisotropy K to be properly represented. Since the deflection angle and the energy variation are often linked in the scattering law (i.e. in elastic scattering), the scattering anisotropy may be also influenced by the energy discretization. In general, the finer

the groups, the higher the degree of anisotropy K required, [18]. As K is increased, also the number required of flux angular moments is increased. In particular, for three-dimensional geometries, the number of spherical harmonics involved in the flux expansion is $H = (K+1)^2$.

In the following, wherever possible, we will index spherical harmonics functions (and so also the related moments) using the more synthetic single-index notation $\langle A_h \rangle_{h=1,H}$ instead of notation $\langle A_{k,l} \rangle_{k=0,K \ l=-k,k}$.

By defining with $\mathcal{A}_H$ the space spanned by spherical harmonics of order smaller or equal than K,

$$\mathcal{A}_H = span\{A_h(\mathbf{\Omega}), h \leq H(K)\},$$

and $\mathbf{A}_H$ the column vector containing the first H spherical harmonics, namely $\mathbf{A}_H = \{A_h(\mathbf{\Omega})\}_{h=1,H}$, the scattering operator of Eq. (1.37), can be rewritten as

$$\mathcal{H}\psi = \sum_{h=1}^{H} \Sigma_{s,k(h)}(\mathbf{r}) A_h(\mathbf{\Omega}) \phi_h = \mathbf{A}_H(\mathbf{\Omega}) \cdot \overline{\overline{\mathbf{\Sigma}}}_{s,H} \left(\mathbf{A}_H, \psi\right). \tag{1.40}$$

The projection operator $(\mathbf{A}_H, \ \cdot)$, defined as in Eq. (1.38) but for vector $\mathbf{A}_H$, converts the continuous angular flux into angular flux moments,

$$\boldsymbol{\phi}_H(\mathbf{r}) = (\mathbf{A}_H, \psi) = \frac{1}{4\pi} \int_{4\pi} d\mathbf{\Omega} \ \mathbf{A}_H(\mathbf{\Omega}) \psi(\mathbf{r}, \mathbf{\Omega}), \tag{1.41}$$

with $\boldsymbol{\phi}_H$ as the vector $\boldsymbol{\phi}_H = \{\phi_h(\mathbf{r})\}_{h=1,H}$.
Matrix $\overline{\overline{\mathbf{\Sigma}}}_{s,H} = diag\{\Sigma_{s,k(h)}, h = 1, H\}$ represent the discretized scattering operator relating angular flux moments to the angular moments of the scattering source, according to

$$\mathbf{q}_H(\mathbf{r}) = \overline{\overline{\mathbf{\Sigma}}}_{s,H}(\mathbf{r}) \boldsymbol{\phi}(\mathbf{r}), \tag{1.42}$$

with $\mathbf{q}_H = \{q_h(\mathbf{r})\}_{h=1,H}$ as the vector containing the moments of the scattering source, represented on the spherical harmonics space $\mathcal{A}_H$, namely $\mathbf{q}_h = (A_h, q)$. $\overline{\overline{\mathbf{\Sigma}}}_{s,H}$ is a diagonal matrix of dimension $H \times H$ containing the moments of the scattering cross section $\Sigma_{s,k(h)}$, numbered according to the h index. At last, operator $\mathbf{A}_H(\mathbf{\Omega}) \cdot$ converts the angular moments of the scattering source, defined over the finite-dimensional space $\mathcal{A}_H$, into the angular continuous scattering source q, i.e.

$$q(\mathbf{r}, \mathbf{\Omega}) = \mathbf{A}_H(\mathbf{\Omega}) \cdot \mathbf{q}_H(\mathbf{r}). \tag{1.43}$$

We remark that no normalization coefficient is required in Eq. (1.43) since it is already included in the definition of the scattering cross section moments, Eq. (1.33).

By substituting Eq. (1.40) into Eq. (1.30), and dropping the group index g and the iterative index (i), the mono-kinetic transport equation can be rewritten as

$$\boldsymbol{\Omega} \cdot \nabla \psi(\mathbf{r}, \boldsymbol{\Omega}) + \Sigma(\mathbf{r}) \psi(\mathbf{r}, \boldsymbol{\Omega}) = \sum_{h=1}^{H} \Sigma_{s,h}(\mathbf{r}) A_h(\boldsymbol{\Omega}) \phi_h(\mathbf{r}) + Q(\mathbf{r}, \boldsymbol{\Omega}), \tag{1.44}$$

where Q is a generic source term for the given source iteration. The introduction of expansion Eq. (1.32) has therefore led to a discretization of the transfer kernel. As shown, this has introduced a representation of the scattering source over the finite-dimensional space $\mathcal{A}_H$, limited by the order of anisotropy K. At the same time, however, no approximation has been introduced on the angular flux, which still has to be discretized, [19].

The transport correction

In the field of photon transport, forward-peaked scattering phenomena are quite common, [20]. In such cases, photons experience minimal deflection at each interaction, resulting in a distribution of the deflection cosine $\hat{\mu} = \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}'$ very peaked around 1. The standard Legendre Polynomials expansion of the scattering cross section is ineffective in representing such anisotropic interactions: the required anisotropy order would be unacceptably high, so leading to unreasonable computational costs. A common remedy is the use of the so-called transport correction, [21, 22]. This method consists in approximating the scattering cross section by the addition of a delta-scattering term. In practice, Σ_s is rewritten as

$$\Sigma_s(\boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') \approx \frac{1}{4\pi} \mathbf{A}_H(\boldsymbol{\Omega}) \overline{\overline{\boldsymbol{\Sigma}}}^c_{s,H} \mathbf{A}_H(\boldsymbol{\Omega}') + \Sigma_c \delta(1 - \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}') \tag{1.45}$$

where H is a reasonably low order of the spherical harmonics expansion, $\overline{\overline{\boldsymbol{\Sigma}}}^c_{s,H}$ is the corrected scattering matrix, having the corrected moments $\Sigma^c_{s,k(h)} = \Sigma_{s,k(h)} - \Sigma_c$ on its diagonal, and finally $\Sigma_c = \Sigma^c_{s,k(h+1)}$. The second term on the RHS of Eq. (1.45) is the delta-scattering term. In order to properly account for particles scattered forward, the angular discretization technique employed in the calculation must be able to properly integrate delta-scattering. In other word, the discretization should be able to account for a delta-scattering term, [20],

$$(\mathcal{H}_\delta \phi)(\boldsymbol{\Omega}) = \Sigma_c \psi(\boldsymbol{\Omega}). \tag{1.46}$$

Transport correction has to be intended as a method for accounting for the forward-peaked component of the scattering law within a standard scattering kernel discretized by Legendre polynomials. However, for certain highly forward peaked scattering problems, such as particle beam simulations, it may be preferable to represent the scattering law by a Fokker-Plank kernel, [23], rather that using the standard Legendre expansion.

1.2.3 Angular discretization of the mono-kinetic transport equation

Traditionally, the techniques used for the angular discretization of Eq. (1.44) are two: the Spherical Harmonics method (P_N), and the Discrete Ordinates method (S_N). We will now provide a brief presentation of the two, omitting for a moment the spatial variable.

Angular discretization by Spherical Harmonics: the P_N method

In the P_N method the angular flux is represented using the spherical harmonics expansion,

$$\psi(\mathbf{\Omega}) = \sum_h a_h A_h(\mathbf{\Omega})\phi_h = \mathbf{A}\cdot\boldsymbol{\phi}, \tag{1.47}$$

where a_h are the normalization coefficients $a_h = 2k+1$, with $k = k(h)$, and $\mathbf{A}(\mathbf{\Omega}) = \{a_h A_h(\mathbf{\Omega})\}$. By substituting Eq. (1.47) and the matrix formulation of Eq. (1.40) into Eq. (1.44), and projecting the equation over the finite-dimensional basis $\mathcal{A}_H$, we obtain a system of equation for the flux angular moments $\boldsymbol{\phi}$, namely

$$(\mathbf{A}_H, \mathbf{A\Omega})\cdot\nabla\boldsymbol{\phi} + \Sigma(\mathbf{A}_H, \mathbf{A})\boldsymbol{\phi} = (\mathbf{A}_H, \mathbf{A}_H)\overline{\overline{\mathbf{\Sigma}}}_{s,H}\boldsymbol{\phi} + (\mathbf{A}_H, Q)\,. \tag{1.48}$$

Because of the orthogonality of the spherical harmonic basis, $(\mathbf{A}_H, \mathbf{A}_H)$ is an $H\times H$ diagonal matrix. In the same way also matrix $(\mathbf{A}_H, \mathbf{A})$ displays the same diagonal pattern. The second term on the RHS of Eq. (1.48) contains the projections of the source term over spherical harmonic basis $\mathcal{A}_H$. The matrix $(\mathbf{A}_H, \mathbf{A\Omega})$ at the first term of the LHS describes the coupling between the different equations of the system. By employing spherical harmonic properties, $\mathbf{A\Omega}$ can be expressed as a linear combination of spherical harmonics of different orders. In particular, $A_{k,l}\mathbf{\Omega}$ can be written in terms of spherical harmonics of order $k-1$, k, and $k+1$. Therefore, matrix $(\mathbf{A}_H, \mathbf{A\Omega})$ describes a non-diagonal coupling between different order equations. The system of equations is closed by setting to zero the high-order flux angular moments for $k > K$.

The solution provided by the P_N method is continuous in angle and its quality is given by how well the space $\mathcal{A}_H$ is able to represent the effective angular distribution of the solution. Because of the sparsity pattern of matrix $(\mathbf{A}_H, \mathbf{A\Omega})$, the P_N method becomes particularly expensive for highly anisotropic problems.

Angular discretization by Discrete Ordinates: the S_N method

The Discrete Ordinates method (S_N) consists in approximating the angular flux $\psi \sim \psi(\mathbf{\Omega}_d)$ along a finite set of directions $S_N = \{\mathbf{\Omega}_d\}_{d=1,D}$, and in numerically computing the integral for the angular flux moments, defined in Eq. (1.38). The integral is solved using a numerical

quadrature $\{\mathbf{\Omega}_d, w_d\}_{d=1,D}$, namely

$$\phi_h = \frac{1}{4\pi}\int_{4\pi} d\mathbf{\Omega} A_h(\mathbf{\Omega})\psi(\mathbf{\Omega}) \approx \sum_{d=1}^{D} w_d A_h(\mathbf{\Omega}_d)\psi(\mathbf{\Omega}_d), \tag{1.49}$$

where w_d are the quadrature weights, while the fluxes $\psi_d = \psi(\mathbf{\Omega}_d)$ are computed solving Eq. (1.44) along the discrete set of directions. Discrete Ordinates is a collocation method allowing to rewrite Eq. (1.44) as a system of equations,

$$\mathbf{\Omega}_d \cdot \nabla\psi_d + \Sigma\psi_d = \mathbf{A}_H(\mathbf{\Omega}_d) \cdot \overline{\overline{\mathbf{\Sigma}}}_{s,H}\boldsymbol{\phi}_H + Q_d \quad \text{for } d = 1, \ldots, D, \tag{1.50}$$

coupled together by the flux angular moments $\boldsymbol{\phi}_H$, evaluated as in Eq. (1.49). $Q_d = Q(\mathbf{\Omega}_d)$ is the source term (defined in Eq. (1.31)) evaluated along the discrete set of directions S_N.

By grouping together the different equations of Eq. (1.50), we can write

$$\overline{\overline{\mathbf{L}}}_{SN}\boldsymbol{\psi}_D = \overline{\overline{\mathbf{M}}}_H\overline{\overline{\mathbf{\Sigma}}}_{s,H}\overline{\overline{\mathbf{D}}}_H\boldsymbol{\psi}_D + \mathbf{Q}_D, \tag{1.51}$$

where $\boldsymbol{\psi}_D = \{\psi_d\}_{d=1,D}$, $\mathbf{Q}_D = \{Q_d\}_{d=1,D}$, and $\overline{\overline{\mathbf{L}}}_{SN}$ is a diagonal matrix containing terms $\mathbf{\Omega}_d \cdot \nabla + \Sigma$, representing the S_N-discretized form of the streaming-plus-removal operator. $\overline{\overline{\mathbf{M}}}_H$ and $\overline{\overline{\mathbf{D}}}_H$ are known with the name of moment-to-discrete and discrete-to moment matrix respectively. The first,

$$\overline{\overline{\mathbf{M}}}_H = matrix\{\mathbf{A}_h(\mathbf{\Omega}_d), h = 1, \ldots, H, d = 1, \ldots, D\},$$

transforms the scattering source defined over the spherical harmonic space $\mathcal{A}_H$ into the space of discrete direction S_N. The second, instead, transforms the discrete angular flux $\boldsymbol{\psi}_D$, into the flux angular moments $\boldsymbol{\phi}_H$. The discrete-to-moment matrix is defined as

$$\overline{\overline{\mathbf{D}}}_H = \overline{\overline{\mathbf{M}}}_H^T\overline{\overline{\mathbf{W}}},$$

where $\overline{\overline{\mathbf{M}}}_H^T$ is the transpose of the moment-to-discrete matrix, and $\overline{\overline{\mathbf{W}}}$ is the $D \times D$ matrix having on the diagonal the quadrature weights w_d. Quadrature formulae are typically chosen in such a way that the largest number of angular flux moments ϕ_h is integrated exactly. Furthermore, the set of discrete directions has to be invariant under the geometrical transformations associated to the boundary conditions (see Section 1.1.2).

We notice how the S_N method diagonalizes the streaming-plus-removal operator, allowing to invert $\mathbf{\Omega}_d \cdot \nabla + \Sigma$ for each direction $\mathbf{\Omega}_d$ independently. This gives the method a considerable effectiveness in terms of computational cost. Furthermore, D, i.e. the number of equations in Eq. (1.50), is not strictly related to the order of the spherical harmonics expansion H, and so to the anisotropy order K. The number of directions D is typically chosen so that the angular quadrature is able to integrate all flux angular moments ϕ_h up to the order H. On

the other hand, an increase in the number of spherical harmonics H simply affects the size of the matrices involved in the scattering operator. IDT, like most transport solvers, relies on the S_N method.

1.2.4 The spatial discretization

By introducing the Legendre polynomial expansion of the scattering operator, and using the Discrete Ordinates approximation, the mono-kinetic transport problem, defined in Eqs. (1.30) and (1.31), can be rewritten as

$$(\mathbf{\Omega}_d \cdot \nabla + \Sigma(\mathbf{r}))\psi_d^{(i+1)}(\mathbf{r}) = q_d^{(i+1)}(\mathbf{r}) \quad \text{for } d = 1, \dots, D. \tag{1.52}$$

At each iteration $(i+1)$, the source in direction $\mathbf{\Omega}_d$, i.e. $q_d^{(i+1)}$, is updated according to

$$q_d^{(i+1)}(\mathbf{r}) = \mathbf{A}_H(\mathbf{\Omega}_d) \cdot \overline{\overline{\mathbf{\Sigma}}}_{s,H}(\mathbf{r})\boldsymbol{\phi}_H^{(i)}(\mathbf{r}) + Q_d^{(t+1)}(\mathbf{r}) \quad \text{for } d = 1, \dots, D, \tag{1.53}$$

with $Q_d^{(t+1)}$ as the source contribution coming from other groups, and the flux angular moments

$$\boldsymbol{\phi}_H^{(i)}(\mathbf{r}) = \sum_{d=1}^{D} w_d \mathbf{A}_h(\mathbf{\Omega}_d)\psi_d^{(i)}(\mathbf{r}), \tag{1.54}$$

updated using the flux of previous iteration (i).

In order to solve Eq. (1.52), the introduction of a discretization of the spatial variable is required. Many techniques are available and have been employed over the years, [24]. These include for example Finite Differences, Finite Elements, and the Method of Characteristics. The IDT solver makes use of the spatial discretization by the short-characteristics method. This is based on the integral formulation of the transport equation and relies on a linear polynomial representation of the boundary fluxes and of the source term. The method solves for the flux within regions by integrating along characteristic lines, [25]. A detailed description of short-characteristics method is provided in the following section.

Introduction to the short-characteristic method

In IDT, the spatial variable is discretized using Cartesian conformal grids and assuming the medium properties to be constant within each spatial region. Let now consider a generic Cartesian region $\mathcal{T} \in \mathbb{R}^3$, defined as

$$\mathcal{T} \equiv \mathcal{X}^{\mathcal{T}} \times \mathcal{Y}^{\mathcal{T}} \times \mathcal{Z}^{\mathcal{T}} \tag{1.55}$$

with

$$\begin{aligned} \mathcal{X}^{\mathcal{T}} &\equiv \{x \in \mathbb{R} \mid x_1 \leq x \leq x_2\}^{\mathcal{T}}, \\ \mathcal{Y}^{\mathcal{T}} &\equiv \{y \in \mathbb{R} \mid y_1 \leq y \leq y_2\}^{\mathcal{T}}, \\ \mathcal{Z}^{\mathcal{T}} &\equiv \{z \in \mathbb{R} \mid z_1 \leq z \leq z_2\}^{\mathcal{T}}, \end{aligned} \tag{1.56}$$

where the intervals $[x_1, x_2]^{\mathcal{T}}$, $[y_1, y_2]^{\mathcal{T}}$, $[z_1, z_2]^{\mathcal{T}}$ describe the region extension along the XYZ axes. We indicate by $\Gamma \equiv \partial\mathcal{T}$ the region boundaries, and by (x_m, y_m, z_m) the centroid of the mesh region. We define Γ^i as one of the six faces of boundary $\Gamma = \bigcup_{i=1,6} \Gamma^i$. For convenience, however, we will not address boundary faces with index i, but we will rather use indices s' and s to respectively distinguish between "entering" sides

$$\Gamma^{s'} \in \Gamma_d^- = \Gamma^-(\mathbf{\Omega}_d) = \{\mathbf{r} \in \Gamma |\ \mathbf{n}(\mathbf{r}) \cdot \mathbf{\Omega}_d < 0\},$$

and "exiting" sides

$$\Gamma^{s} \in \Gamma_d^+ = \Gamma^+(\mathbf{\Omega}_d) = \{\mathbf{r} \in \Gamma |\ \mathbf{n}(\mathbf{r}) \cdot \mathbf{\Omega}_d > 0\},$$

for each discrete direction $\mathbf{\Omega}_d$. Here, with $\mathbf{n}$ we indicates the outgoing normal direction. Further, according to the homogeneous assumption, we have $\Sigma(\mathbf{r}) = \Sigma$ for $\mathbf{r} \in \mathcal{T}$.

In IDT, the one-group streaming-plus-removal operator $\mathcal{L}_d = \mathbf{\Omega}_d \cdot \nabla + \Sigma(\mathbf{r})$ is locally inverted within each mesh region using the short-characteristics method, based on the integral form

$$\begin{aligned} \psi_d(\mathbf{r}^{s'} + t\mathbf{\Omega}_d) &= \psi_d(\mathbf{r}^{s'})e^{-\Sigma t} + \int_0^t q_d(\mathbf{r}^{s'} + t'\mathbf{\Omega}_d)e^{-\Sigma(t-t')}dt', \\ &= \psi_d(\mathbf{r}^{s'})e^{-\Sigma t} + \mathcal{K}q_d(t, \mathbf{r}^{s'}, \mathbf{\Omega}_d), \end{aligned} \tag{1.57}$$

where $\mathcal{K}$ is the convolution operator $\mathcal{K}(t, \mathbf{r}^{s'}, \mathbf{\Omega}) = \int_0^t dt'\, e^{-\Sigma(t-t')}$ (see Section 1.1.2). Eq. (1.57) gives the analytical angular flux for any point $\mathbf{r} = \mathbf{r}^{s'} + t\mathbf{\Omega}_d$ along a local trajectory crossing region $\mathcal{T}$. The symbol $\mathbf{r}^{s'}$ indicates the intersection point of the trajectory with the entering side $\Gamma^{s'}$ of boundary Γ.

Also, from Eq. (1.57) we can obtain the outgoing flux for any point $\mathbf{r}^s = \mathbf{r}^{s'} + L\mathbf{\Omega}_d$ on the outgoing side Γ^s as

$$\psi_d(\mathbf{r}^s) = \psi_d(\mathbf{r}^{s'})e^{-\Sigma L} + \mathcal{K}q_d(L; \mathbf{r}^{s'}, \mathbf{\Omega}_d), \tag{1.58}$$

where L is the length of the chord delimited by the two surfaces $\Gamma^{s'}$ and Γ^s, namely $L = \left|\mathbf{r}^s - \mathbf{r}^{s'}\right|$. Figure 1.2 provides a graphical representation of the reference system along the characteristic line $\mathbf{r} = \mathbf{r}^{s'} + t\mathbf{\Omega}_d$.

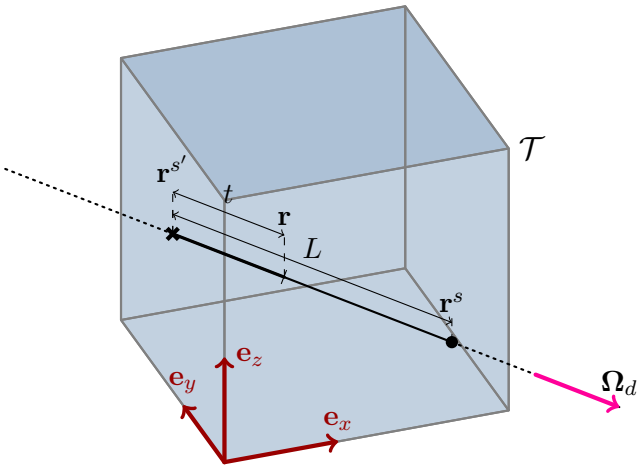


Fig. 1.2 Illustration of characteristic line coordinate system.

According to the short-characteristics approximation, the source q_d and the incoming interface fluxes $\psi_d^{s'}$ are defined as lying on the finite-dimensional spaces $\mathcal{P}_M = \{P_m(\mathbf{r}), m = 1, \ldots, M\}$ and $\mathcal{P}_K^s = \{P_k^s(\mathbf{r}), k = 1, \ldots, K\}$, respectively. The ranks of the bases are respectively $M = 4$ and $K = 3$ for the linear spatial order, and $M = 1$ and $K = 1$ for the constant one. $\mathcal{P}_M$ and $\mathcal{P}_K^s$ are the spaces spanned by the local orthonormal Legendre bases (written up to the linear order), defined as

$$\mathbf{P}(\mathbf{r}) = \{P_m(\mathbf{r})\}_{m=1,4} = \left[1,\ \frac{2(x-x_m)}{(x_2-x_1)},\ \frac{2(y-y_m)}{(y_2-y_1)},\ \frac{2(z-z_m)}{(z_2-z_1)}\right]^T, \tag{1.59}$$

and

$$\mathbf{P}^s(\mathbf{r}) = \{P_k^s(\mathbf{r})\}_{k=1,3} = \begin{cases} \left[1,\ \frac{2(y-y_m)}{(y_2-y_1)},\ \frac{2(z-z_m)}{(z_2-z_1)}\right]^T & \text{for } \mathbf{n}_s = \pm\mathbf{e}_x, \\ \left[1,\ \frac{2(x-x_m)}{(x_2-x_1)},\ \frac{2(z-z_m)}{(z_2-z_1)}\right]^T & \text{for } \mathbf{n}_s = \pm\mathbf{e}_y, \\ \left[1,\ \frac{2(x-x_m)}{(x_2-x_1)},\ \frac{2(y-y_m)}{(y_2-y_1)}\right]^T & \text{for } \mathbf{n}_s = \pm\mathbf{e}_z, \end{cases} \tag{1.60}$$

respectively. The approximation allows to express the source and the incoming flux with the local linear expansions,

$$q_d(\mathbf{r}) = \sum_{m=1,M} \alpha_m P_m(\mathbf{r}) q_{d,m} \quad \text{for } \mathbf{r} \in \mathcal{T}, \tag{1.61}$$

$$\psi_d^{s'}(\mathbf{r}) = \sum_{k=1,K} \beta_k^{s'} P_k^{s'}(\mathbf{r}) \psi_{d,k}^{s'} \quad \text{for } \mathbf{r} \in \Gamma^{s'}, \tag{1.62}$$

where α_m and $\beta_k^{s'}$ are normalization coefficients. The spatial moments in Eq. (1.61) and (1.62) are defined by the volume projections of the source,

$$q_{d,m} = (P_m,\ q_d) = \frac{1}{V} \int_{\mathcal{T}} d\mathbf{r}\ P_m(\mathbf{r}) q_d(\mathbf{r}), \tag{1.63}$$

and the surface projection of the angular flux,

$$\psi_{d,k}^{s'} = \left\langle P_k^{s'},\ \psi_d \right\rangle^{s'} = \frac{\mu_d^{s'}}{A^{s'}} \int_{\Gamma^{s'}} d\mathbf{r}\ P_k^{s'}(\mathbf{r}) \psi_d^{s'}(\mathbf{r}), \tag{1.64}$$

with $\mu_d^{s'} = |\mathbf{\Omega}_d \cdot \mathbf{n}_{s'}|$, $\mathbf{n}_{s'}$ the outgoing normal of face s', and the volume and surface measures defined as

$$\begin{aligned} V &= \int_{\mathcal{T}} d\mathbf{r} = (x_2 - x_1)(y_2 - y_1)(z_2 - z_1), \text{ and} \\ A^{s'} &= \int_{\Gamma^{s'}} d\mathbf{r} = \begin{cases} (y_2 - y_1)(z_2 - z_1) \text{ for } \mathbf{n}^{s'} = \pm \mathbf{e}_x, \\ (x_2 - x_1)(z_2 - z_1) \text{ for } \mathbf{n}^{s'} = \pm \mathbf{e}_y, \\ (x_2 - x_1)(y_2 - y_1) \text{ for } \mathbf{n}^{s'} = \pm \mathbf{e}_z. \end{cases} \end{aligned}$$

Derivation of the short-characteristic method

The short-characteristic method, relies on two equations: the volume-balance equation, solving for the volume-integrated spatial moments of the flux,

$$\boldsymbol{\psi}_d = (\mathbf{P},\ \psi_d) = \frac{1}{V} \int_{\mathcal{T}} d\mathbf{r}\ \mathbf{P}(\mathbf{r}) \psi_d(\mathbf{r}), \tag{1.65}$$

and the transmission equation, solving for the spatial moments of the flux integrated on the region interfaces,

$$\boldsymbol{\psi}_d^s = \langle \mathbf{P}^s,\ \psi_d \rangle^s = \frac{\mu_d^s}{A^s} \int_{\Gamma^s} d\mathbf{r}\ \mathbf{P}^s(\mathbf{r}) \psi_d^s(\mathbf{r}). \tag{1.66}$$

These are obtained by substituting the source and the interface-flux expansions Eq. (1.61) and Eq. (1.62) into the integral transport equations Eq. (1.57) and Eq. (1.58). Then, the flux along the trajectory, described by Eq. (1.57), and the flux in the outgoing point $\mathbf{r}^{s'}$, given by Eq. (1.58), are projected using Eq. (1.65) and (1.66), respectively, to obtain

$$\boldsymbol{\psi}_d = \overline{\overline{\mathbf{C}}}_d \mathbf{q}_d + \sum_{s' \in \Gamma^-(\mathbf{\Omega}_d)} \overline{\overline{\mathbf{I}}}_d^{s'} \boldsymbol{\psi}_d^{s'}, \tag{1.67}$$

$$\boldsymbol{\psi}_d^s = \overline{\overline{\mathbf{E}}}_d^s \mathbf{q}_d + \sum_{s' \in \Gamma^-(\mathbf{\Omega}_d)} \overline{\overline{\mathbf{T}}}_d^{s,s'} \boldsymbol{\psi}_d^{s'} \quad \text{for } s \in \Gamma^+(\mathbf{\Omega_d}). \tag{1.68}$$

with $\boldsymbol{\psi}_d = \{\psi_{d,m}\}_{m=1,M}$ as the vectors of the spatial moments of flux within the region, $\boldsymbol{\psi}_d^{s'} = \{\psi_{d,k}^{s'}\}_{k=1,K}$ and $\boldsymbol{\psi}_d^s = \{\psi_{d,k}^s\}_{k=1,K}$ as the vectors of the spatial moments of flux on the incoming and outgoing boundaries respectively. $\mathbf{q}_d = \{q_{d,m}\}_{m=1,M}$ is the vector of the

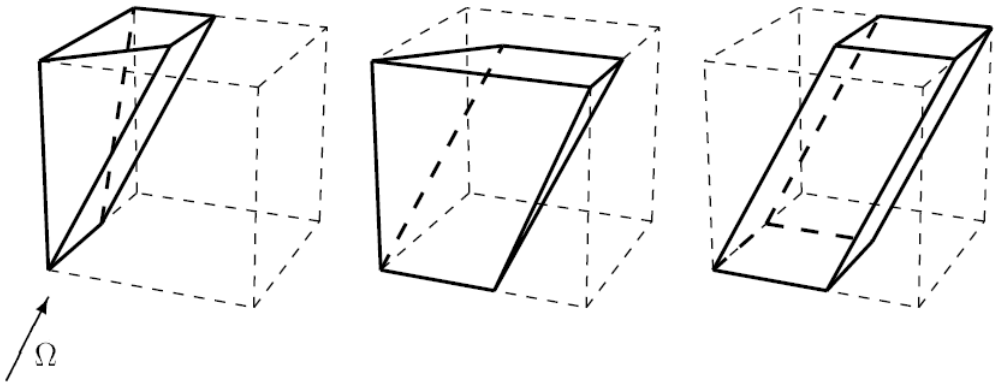


Fig. 1.3 Illustration of the region volume partitioning in the integration of the short-characteristics coefficients. Image taken from [26].

source spatial moments, expressed as

$$\mathbf{q}_d = \sum_{h=1}^{H} A_h(\mathbf{\Omega}_d)\mathbf{q}_h + \mathbf{Q}_d \tag{1.69}$$

$$\text{with } \mathbf{q}_h = \Sigma_{s,h}\phi_h. \tag{1.70}$$

The four matrices: the collision, $\overline{\overline{\mathbf{C}}}_d$, the incoming, $\overline{\overline{\mathbf{I}}}_d^{s'}$, the escape, $\overline{\overline{\mathbf{E}}}_d^{s}$, and the transmission, $\overline{\overline{\mathbf{T}}}_d^{s,s'}$, are computed analytically as

$$\begin{aligned}
\overline{\overline{\mathbf{C}}}(\mathbf{\Omega}_d) &= \overline{\overline{\mathbf{C}}}_d = (\mathbf{P},\ \mathcal{K}\mathbf{P}) = \frac{1}{V}\int_{\mathcal{T}} d\mathbf{r}\ \mathbf{P}(\mathbf{r}) \otimes \int_0^t e^{-\Sigma(t-t')}\mathbf{P}(\mathbf{r}^{s'} + t'\mathbf{\Omega}_d)\ dt', \\
\overline{\overline{\mathbf{I}}}^{s'}(\mathbf{\Omega}_d) &= \overline{\overline{\mathbf{I}}}_d^{s'} = (\mathbf{P},\ e^{-\Sigma t}\mathbf{P}^{s'}) = \frac{1}{V}\int_{\mathcal{T}} d\mathbf{r}\ \mathbf{P}(\mathbf{r}) \otimes e^{-\Sigma t}\ \mathbf{P}^{s'}(\mathbf{r}^{s'}), \\
\overline{\overline{\mathbf{E}}}^{s}(\mathbf{\Omega}_d) &= \overline{\overline{\mathbf{E}}}_d^{s} = \langle \mathbf{P}^s,\ \mathcal{K}\mathbf{P}\rangle^s = \frac{\mu_d^s}{A^s}\int_{\Gamma^s} d\mathbf{r}^s\ \mathbf{P}^s(\mathbf{r}^s) \otimes \int_0^{L(\mathbf{r}^s,\mathbf{r}^{s'})} e^{-\Sigma[L(\mathbf{r}^s,\mathbf{r}^{s'})-t']}\mathbf{P}(\mathbf{r}^{s'} + t'\mathbf{\Omega}_d)\ dt', \\
\overline{\overline{\mathbf{T}}}^{s,s'}(\mathbf{\Omega}_d) &= \overline{\overline{\mathbf{T}}}_d^{s,s'} = \left\langle \mathbf{P}^s,\ e^{-\Sigma t}\mathbf{P}^{s'}\right\rangle^s = \frac{\mu_d^s}{A^s}\int_{\Gamma^s} d\mathbf{r}^s\ \mathbf{P}^s(\mathbf{r}^s) \otimes e^{-\Sigma L(\mathbf{r}^s,\mathbf{r}^{s'})}\mathbf{P}^{s'}(\mathbf{r}^{s'}),
\end{aligned}$$

with $\otimes$ being the tensor product.

Figure 1.3 provides a graphical illustration of the coupling described in short-characteristics matrices. The three images depict respectively the contribution of particles incoming from the left, front and bottom boundaries, along a prescribed direction $\mathbf{\Omega}$, and contributing to the top boundary flux. Within each subregion, delimited by solid-lines, characteristic lines (generated by points on the incoming boundary and tracked along the direction $\mathbf{\Omega}$) are limited by a single couple of interfaces (s, s'). The short-characteristics matrices are obtained performing integrals on each of this subregions individually, [2, 26].

The short-characteristics and discrete ordinates methods lead to a response-matrix formalism describing the particle transfer between spatial regions for each direction of the angular quadrature. The approximation applies locally to each spatial region meaning that particle transport across the domain is not exact, unlike MOC, but it is exact only within regions.

In the following we will refer to the short-characteristic method using the linear spatial bases as *linear-characteristics* (LC), while with *step-characteristics* (SC) we will denote the short-characteristic method with the constant spatial bases.

The linear-characteristics method does not ensure the positivity of the solution, which is instead guaranteed for step-characteristics. However, linear-characteristics have been proved to reproduce streaming effects, such as rays propagation, with a good accuracy and displaying a low numerical diffusion along the directions perpendicular to the propagation one, [27]. The main drawback of the short-characteristics method, like for many response-matrix-based methods, stays in the need of storing in memory the coefficient matrices in order to avoid their calculation at each application. This generally results into an undesired memory consumption.

We have presented the short-characteristics method under the assumption of constant medium properties within each Cartesian region. In IDT, however, this hypothesis can be relaxed by representing heterogeneous media at the interior of each region. This is particularly useful since it allows to solve heterogeneous problems while using reasonably coarse meshes. Since this mode has not been employed in the course of our research, we refer the reader to readings devoted to this topic, [1, 28, 29].

Short-characteristics solution algorithm

Short-characteristics, together with the discrete ordinates method, allows to transform Eq. (1.52) into a linear system of equations whose matrix is block-diagonal, where each block is lower-triangular. Each matrix block describes the coupling between the equations relating different spatial regions for the same discrete direction. The triangular pattern, instead, is obtained by Gaussian elimination, numbering regions following the particle propagation front. At each inner iteration, the scattering source is everywhere known (from the previous iteration) while the incoming fluxes at boundaries are given by boundary conditions. For homogeneous boundary conditions, where incoming flux depends on the outgoing (such as reflection), the incoming flux is given by the last estimate of the outgoing flux. By solving for the first region (encountered by the propagation front), the outgoing interface fluxes are obtained and used to feed regions downstream. By sequentially repeating this process following the propagation front, all regions of the domain are solved. This procedure is generally known as sweep.

Once the angular fluxes are solved for all regions, and for all directions of the angular quadrature, the flux moments are evaluated as in Eq. (1.54). As depicted in Algorithm 1,

the process is repeated by updating the scattering source (see Eq. (1.53)) and boundary conditions at each interaction.

Algorithm 1: The S_N algorithm

```
begin
    do
        Set iteration (i + 1);
        Initialise φ_H^(i+1) ⟵ 0 ;
        Update q_H^(i+1) from φ_H^(i) by Eq. (1.70);
        Update ψ^{s',(i+1)} with boundary conditions: ψ^{s',(i+1)} ⟵ β^{s',s} ψ^{s,(i)};
        forall Ω_d in S_N angular quadrature do
            Compute q_d^(i+1) along Ω_d by Eq. (1.69);
            Solve along Ω_d: ψ_d^(i+1), ψ_d^{s,(i+1)} ⟵ SWEEP(q_d^(i+1), ψ_d^{s',(i+1)});
        end
        Compute φ_H^(i+1) by Eq. (1.54);
    while ε(φ_H^(i+1), φ_H^(i)) ≥ τ;
end
```

The iterative process is repeated up to when a convergence condition is verified. This consists in computing at each iteration the relative error L_∞-norm

$$\epsilon = \max_{\substack{r=1,N_r \\ h=1,H \\ m=1,M}} \left\{ \frac{|\phi_{r,h,m}^{(i+1)} - \phi_{r,h,m}^{(i)}|}{max\{\phi_{r,1,1}^{(i+1)}, |\phi_{r,h,m}^{(i+1)}|\}} \right\}, \tag{1.71}$$

with N_r as the number of spatial regions, and comparing this with the user defined tolerance τ. When the condition $\epsilon < \tau$ is verified, inner iterations end.

1.3 On the angular discretization in particle streaming problems

Standard reactor core calculations and radiation shielding studies may present very different requirements in terms of angular discretization. The first are characterized by optically thick and highly diffusive media. The scattering source is evenly distributed over the domain, and the angular flux generally exhibits smooth variations with the angular variable. In such problems, even a coarse angular quadrature is able to integrate the angular moments with an acceptable accuracy.

In contrast, shielding calculations deal with a larger variability of media, and also with more penetrating particles, such as photons. If the medium is little diffusive (the scattering cross section is much smaller than the total cross section) and the source is defined over a limited portion of the domain, the flux at a given point outside the source will be very

anisotropic. The integration of an anisotropic flux requires quadrature rules much finer than the one used in reactor core calculations. Consider, for example, a problem in which one point of the domain is experiencing a flux that is peaked in one direction. In order to integrate this profile it would be useful to have a large number of quadrature directions in the surroundings of this direction. If the angular quadrature is made of evenly spaced directions, in order to achieve the desired "density" of directions in the aforementioned surround, the order of the angular quadrature will have to be increased.

In radiation shielding problems, the use of an inadequate refinement of the discrete ordinates may dramatically affect the accuracy of the calculation. In particular, it may give rise to the so-called ray-effect, [30, 31]. Ray-effect consists in a series of numerical artifacts visible as distortions of the spatial distribution of the scalar flux (as well as of the flux angular moments). It is typical of multidimensional calculations while it is not experienced in stationary 1-D problems. The ray-effect distortions are due to the inability of the angular quadrature to represent the effective particle streaming. Even if the flux is solved exactly along the directions of the angular quadrature, these directions may not be representative of the actual directions along which particles streams. A solution affected by ray-effect will manifest regions where the flux is particularly picked (where the quadrature directions match the effective particle streaming directions), at the expense of others where the flux is unphysically depressed (streaming directions are not part of the discrete ordinate space). By simulating a single emitting source region in an infinite and purely absorbing domain, the average flux computed with S_N would appear as a set of concentric rays centered in the source, hence the name of this phenomenon.

The inability to reconstruct angular moments of the flux may result in large underestimation of the flux at the detectors (i.e. measurement points) located in the ray-effect 'shaded' regions. In the radiation shielding field, underestimating detector responses such as a radiation doses, for safety reasons, is not acceptable. Hence the importance of an appropriate angular discretization, and of the use of techniques for the mitigation of ray-effect, [32].

The ray-effect is favoured in problems with poorly scattering media or dominated by forward-peaked scattering (problems which presents most anisotropic fluxes). For purely geometrical reasons, the refinement of the spatial discretization can also contribute to a sharpening of the angular distribution of the flux in the single regions. In addition, the numerical diffusion introduced by the spatial discretization may accidentally provide a mitigation of the ray-effect, smoothing out peaks and sinkings. The presence of spatially localized sources, which are present in many radiation protection calculations, is also a factor favouring ray-effect. The most pathological case for ray-effect is represented by the calculation of a point source in vacuum, [33].

The reason of the ray-effect is commonly attributed to the lack of invariance with respect to arbitrary rotations of the S_N discretization, [19, 34]. Since the method of Spherical Harmonics (P_N) verifies this property, its solutions are not affected by ray-effect. Despite this,

the representation of strongly anisotropic fluxes is equally problematic as it would require very high-order flux expansions. Even a P_9 discretization (size of the spherical harmonic base $H = 100$) may be unable to represent the effective angular dependence of very peaked fluxes. Furthermore, the P_N equations has a non-diagonal structure, so requiring higher computational costs for the inversion of the streaming operator. This is why, in practice, the Discrete Ordinates method is still preferred. An example of solutions affected by ray-effect will be provided in Chapter 4.

1.3.1 Discrete Ordinates with improved angular quadratures

The refinement of angular quadrature typically yields to smoothed ray-effect distortions and so to a mitigation of the ray-effect. However, this approach may be inefficient since ray-effect may persist even for extremely high numbers of directions, [31]. For the specific case of a spherical source in a homogeneous infinite domain, and for any value of the scattering cross section strictly smaller than the total cross section, it has been also shown how the error introduced by the S_N approximation becomes unbounded as the distance to the source tends to infinity, [35].

Despite this, several authors have successfully developed new types of angular quadratures in an effort to mitigate the ray-effect and increase the accuracy of the S_N angular discretization.

Probably the most promising approach is given by the use of adaptive quadratures, i.e., those that present a larger number of directions where they are most needed, so as to better represent particles streaming, [36]. A notable advance in this regard is given by the development of LDFE quadratures by Jarrell and Marvin, [37, 38]. These quadratures are constructed from an octahedral support, successively partitioning its faces into sub-triangles, and then projecting them onto the unit sphere. The definition of an interpolative basis in each sub-triangle allows the definition of weights and directions, as well as an error estimator useful in assessing whether or not the quadrature needs to be refined. The main advantages of LDFE quadratures are the very high order of convergence (4th order when homogeneously discretized) and its hierarchical nature, useful in adaptive angular refinements. Jarrell also demonstrated the possibility of using different orders of the quadrature in different spatial regions, [37].

Another noteworthy strategy is given by the Icosahedral (IC) quadratures proposed by Ahrens, [39]. These, being generated from the icosahedron, yield the angular quadrature invariant to the largest possible set of rotations. Nevertheless, the icosahedron symmetries are not necessarily the ones on the problem to be solved (in general reflections, 90° rotations), and so, IC quadratures may cause problems in the exact treatment of boundary conditions. It has been shown how the accuracy of IC quadratures is significantly higher than the one of standard quadrature rules, e.g. level symmetric. While not directly addressing the problem of ray-effect, it has been also shown how IC quadratures are able to produce solutions less

affected by ray-effect than other quadratures due to the better distributions of nodes on the sphere, [40].

1.3.2 The P_N-equivalent S_N methods

One of the first approaches to ray-effect mitigation relies on the S_N-P_N equivalence. Known that the P_N approximation is invariant by rotation, this technique operates by modifying the discrete-ordinates equations in order to obtain solutions that obey to P_N equations. This approach relies on the addition of an artificial source term within the S_N equations, which is solved iteratively together with the scattering source. Many examples of this technique can be found in literature, [31, 41-44]. A recent research by R. Sanchez has identified a common mathematical framework showing how these techniques differ by the definition of the artificial source, [19]. The method effectively removes the ray-effect, but, as a drawback, it drastically affects the convergence speed of source iterations, [34].

It is worth mentioning that the so obtained numerical solutions will exhibit the same limits of the P_N method, i.e. the spherical harmonic basis is unable to represent very anisotropic angular fluxes. Therefore, even a solution obtained by means of the P_N-equivalent S_N method could be affected by not negligible errors.

1.3.3 Angular discretization by the Finite Elements method

A vast number of projective methods have also been proposed as alternatives to the more common S_N and P_N methods. Examples include angular finite elements and wavelets. Similarly to the P_N method, angular finite elements are based on the representation of the flux on a finite-dimensional space, but rather than using a set of functions defined on the entire unit sphere, they use partial-range trial functions. These representations allow for sets of equations coupled by the streaming operator, but simpler to inverse than P_N equations, [20]. In addition, projective methods such as finite elements and wavelets lend themselves well for generating adaptive refinements, [45-47].

Angular finite elements have been applied to both the integro-differential formulation and the second order even-parity form of the transport equation. We do not cover this second example but provide some references for the curious readers, [48-50].

The discretization of the integro-differential transport equation by means of continuous finite elements yields a set of S_N-like equations coupled for a fictitious source term that must be solved iteratively, [34]. If, on the other hand, the flux is approximated on a discontinuous trial space, the equations will present a coupling which is local to each angular element, [34, 51]. The equations will be described by a block diagonal coupling, and they would be solvable by means of a conventional sweep algorithm. Compared to S_N though, each block will have a larger dimension, as it will describe the coupling between different degrees of freedom of the single element. Discontinuous trial functions facilitate the generation of hierarchical meshes

and local angular refinements, [45]. Furthermore, they allow to generate simple mappings between meshes having different degrees of refinement, [37]. In general, continuous finite elements are more effective in mitigating the ray-effect with respect to discontinuous ones, [34], but less capable of providing accurate representations of steep flux variations in the angular variable.

Morel *et al.* have performed a comparison of various ray-effect mitigation methods for a series of 2-D test problems, [34]. In particular they compared a P_N-equivalent S_N method, an angular continuous finite element method and an angular discontinuous finite element method. The authors observed how these methods give different levels of ray-effect mitigation. They further noticed how the effectiveness of these methods is highly problem dependent, and unsatisfactory, as the problem resemble the particle propagation from a line source in a vacuum (source which presents a spatial singularity in 2-D geometries), [34].

1.3.4 Point-like sources treatment and First Collision Source method

We have seen how the point-like source propagation in a vacuum (equivalent to the line source problem discussed by Morel *et al.* for 2-D geometries, [34]) is the most pathological case in terms of ray-effect for S_N simulations. The reason of this is to be found in the mathematical nature of the flux emitted by a point-like source.

Point-like sources

The flux of particles emitted in a purely absorbing medium, by an isotropic point source $Q(\mathbf{r}) = Q\delta(\mathbf{r} - \mathbf{r}_p)$ located at $\mathbf{r}_p$ is expressed by the Green function,

$$\psi^u(\mathbf{r}, \mathbf{\Omega}) = Q \frac{1}{|\mathbf{r} - \mathbf{r}_p|^2} e^{-\tau(\mathbf{r}, \mathbf{r}_p)} \delta_2 \left(\mathbf{\Omega} \cdot \hat{\mathbf{e}}_{r_p} \right), \tag{1.72}$$

where Q is the source intensity, measuring the number of particles emitted per second and steradian, $\hat{e}_{r_p}$ is the unit vector defined as $\hat{e}_{r_p} = \frac{\mathbf{r} - \mathbf{r}_p}{|\mathbf{r} - \mathbf{r}_p|}$, and δ_2 is the Placzek's delta function. The Placzek's delta is defined on the surface of the unit sphere such that for a regular test function $f(\mathbf{\Omega})$ we have that

$$\int d\mathbf{\Omega} f(\mathbf{\Omega}) \delta_2(\mathbf{\Omega} \cdot \mathbf{\Omega}_0) = f(\mathbf{\Omega}_0).$$

Finally, τ is the optical thickness between $\mathbf{r}_p$, and position $\mathbf{r}$, defined in Eq. (1.10), [52]. The term $1/|\mathbf{r} - \mathbf{r}_p|^2$ represents the flux geometrical attenuation, while the exponential term is the optical attenuation, accounting for particles that have not collided between the emission point, $\mathbf{r}_p$, and point $\mathbf{r}$. The flux energy-dependence (here omitted) resides entirely in the source density and in the optical attenuation. Eq. (1.72), also known with the name of point-kernel, presents two singularities: the first is a spatial singularity, and it is related to

the $1/\left|\mathbf{r}-\mathbf{r}_p\right|^2$ term, while the second is a directional singularity, meaning that particles are directed only along the direction joining the source to the measurement point $\mathbf{r}$, [52].

First Collision Source method

In order to reduce the ray-effect and, at the same time, solve for the flux emitted by localized sources, First Collision Source (FCS) methods are generally employed, [31]. FCS methods solve the transport problem, namely Eq. (1.2), by introducing a first-order expansion in Neumann series of the flux. Thus, the flux is rewritten as the sum

$$\psi = \psi^u + \psi^c. \tag{1.73}$$

where the first term is the uncollided flux, i.e. the flux of particle emitted by the external source, and that has not undergone any collision. The second term of Eq. (1.73), is instead the flux of particles having undergone at least one collision. The uncollided flux, ψ^u, is defined by

$$\psi^u = \mathcal{L}^{-1} Q_{ext}, \tag{1.74}$$

while the collided flux component is obtained by solving

$$\mathcal{L}\psi^c = \mathcal{H}\psi^c + \mathcal{H}\psi^u. \tag{1.75}$$

Equation (1.75) is retrieved by substituting the flux expansion, Eq. (1.73), into the transport equation, Eq. (1.2), and then simplifying the expression using the definition, Eq. (1.74). Equation (1.75) describes a classical transport problem whose external source is given by the so-called first collision source,

$$Q_{FCS} = \mathcal{H}\psi^u. \tag{1.76}$$

The idea of the FCS method is to evaluate the two flux contributions, collided and uncollided, in two successive steps, and using different discretization techniques for the streaming-plus-removal operator. The first step, namely the evaluation of uncollided flux, is solved using an ad-hoc discretized operator $\mathcal{L}_{FCS}$ in Eq. (1.74). $\mathcal{L}_{FCS}$ has to yield solutions not affected (or at least less) by ray-effect, and, in the presence of point-like sources, it must be able to properly handle the spatial singularity of Q_{ext}. The second step, i.e. the calculation of the collided component, is solved by mean of a standard transport calculation fed by the spatially distributed source Q_{FCS}. Finally, the flux solution is retrieved by summing the two components according to Eq. (1.73).

Since Q_{FCS} is more spatially distributed that Q_{ext}, the numerical solution of Eq. (1.75) is much less affected by ray-effect than the one of the original problem,

$$\mathcal{L}\psi = \mathcal{H}\psi + Q_{ext}.$$

Therefore, the discrete-ordinates calculation of the collided flux, Eq. (1.75), requires a much coarser angular discretization than the original transport calculation. In this sense, FCS can be seen as a pre-treatment method able to transform the transport problem, i.e. the propagation of Q_{ext}, into one which is cheaper to solve.

At the same time, FCS offers the possibility of solving the uncollided flux contribution, which is the component most effected by ray-effect (because of the spatially localised nature of Q_{ext}), by an ad-hoc streaming-plus-removal operator, able to completely remove ray-effect from this component. Furthermore, if Q_{ext} is a point-like source, FCS allows to properly account for the source singularities.

Examples of FCS methods can be found in several general-purpose deterministic transport codes. These techniques generally differ from each other by the discretization of streaming-plus-removal operator $\mathcal{L}_{FCS}$ for the uncollided flux calculation.

A mock-up version of the IDT solver already has a module capable of performing FCS calculations. This is based on the Method of Characteristics in a spherical coordinates system, originally developed by I. Zmijarevic and D. Sciannandrone, [53]. The MOC discretization is effective in evaluating the uncollided flux emitted from point sources with a good precision. However, this is limited to the constant spatial order, i.e. it is able to provide only the average uncollided flux in each region. Further details of this method will be provided in Chapter 2, while extending MOC for the evaluation of higher order moments.

First Collision Source method features and variants

In general, only peaked and localised sources are used to feed the uncollided flux calculation. If spatially distributed are also present, they can be aggregated to the first collision source and solved directly in the second step calculation. The common practice is to use an $\mathcal{L}_{FCS}$ discretized operator for the calculation of point sources, and then modelling volume sources as an ensemble of point sources. In practice, Q_{ext} is defined as the sum of the different point sources Q_p, i.e.

$$Q_{ext}(\mathbf{r}) = \sum_{p=1,N_p} Q_p \; \delta(\mathbf{r} - \mathbf{r}_p) \tag{1.77}$$

with N_p as the number of source points, Q_p the source intensity, and $\mathbf{r}_p$ its spatial locations. This method is simple but can be problematic in the presence of optically thick source regions, [54]. In such a case, the greatest contribution to the uncollided flux is given by particles emitted by the outer layer of these regions. In this regard, particular care must be taken when decomposing optically thick regions into point sources.

The main limit of FCS is the inability to completely remove numerical errors related to ray-effect since acting on the uncollided flux uniquely. Ray-effect may still be generated by the angular discretization involved in the collided flux calculation. In addition, it can be demonstrated that the analytical formulation of Q_{FCS} may also be singular, [52]. Several authors have proposed techniques derived from the standard First Collision Source method in order to tackle the ray-effect associated to n-th collided sources. For example, the Adaptive Collision Source method proposed by W.J. Walters and A. Haghighat uses a n-th order Neumann expansion of the flux and solves for each order varying adaptively the discrete-ordinates quadrature order, [55]. X. Wang *et al.* has similarly tackled the problem by selectively applying a ray-tracing operator to portions of the n-th collided source, [56, 57]. Their approach is similar to a nested FCS method, where portion of the n-th order scattering sources are removed from the discrete ordinates calculations and solved separately using a fine discretization of the streaming-plus-removal operator. According to the authors, the method aims to mitigate ray-effect where this is needed the most, i.e. in flux contribution associated to particle migrations from peaked scattering sources along a several order scattering expansion.

1.4 The Domain Decomposition Method for the resolution of the Neutron Transport Equation on distributed parallel architectures

The growing demand for computational resources has led over the years to the development of increasingly powerful hardwares. However, because of technological limitations, the evolution of computers has progressively moved towards "machines" with distributed (and not necessarily uniform) resources, both in terms of memory and computing power. Given these progresses, softwares initially developed for serial calculation became ineffective on these modern architectures. In order to take full advantage of the resources offered by new architectures, algorithms had to adapt. It became necessary to resort to algorithms, sometimes less effective in serial calculations, but with a greater parallel efficiency. In particular, a common approach is to use Domain Decomposition Methods (DDM). These methods rewrite the original transport problem in terms of a set of sub-problems coupled together by setting the continuity of the interface angular flux. Each sub-problem typically deals with a portion of the original phase space. The advantage are twofold: on the one hand DDM allows to solve calculations in parallel, by simply running sub-problems on multiple calculation unit. On the other hand, it allows to partition the memory demand, so as to meet the distributed architecture of the memory resources.

In this section, we want to provide a brief introduction to modern computational architectures and parallelism models in shared and distributed memory. We will also present

Domain Decomposition Methods and, in particular, the domain decomposition employed by IDT. The essence of this section is taken from the Ph.D. thesis of R. Lenain, [58].

1.4.1 Brief introduction to the modern parallelism models

Modern computing architectures are equipped with one or more processing units (CPU or processors), and a primary storage unit, namely the memory (RAM). Each processor is made of several calculation units, i.e. the cores, which are capable of independently executing the operations required by an algorithm accessing the same memory. An algorithm running on multiple cores and accessing to the same memory is said to make use of shared-memory parallelism. This consists in decomposing the algorithm operations into tasks, and solving them with independent logical computing units, called threads. By executing threads on different cores, the algorithm can be solved in parallel. Since threads access, and possibly modify, the same memory in parallel, particular care must be given to ensure the coherence of the memory throughout the calculation. Concurrent access to the memory may lead to the so-called race conditions. These are avoided by explicitly setting synchronization points, where threads wait for each other before proceeding the calculation. CPUs make use of a fast-access memory, called cache, which acts as a local copy of RAM data. Cache commonly has a hierarchic structure, typically divided in two or three levels. Lower-level caches have smaller capacity but fast access time with respect to higher-level caches. If the data on which the processor is to operate fit into the cache memory, cores can access data more easily and with a reduced number of cycles. Algorithms that perform a large number of operations on the same portion of memory, fully allocated in the cache, will have to perform few copies of data between cache and RAM during execution. The resulting increase of the calculation speed is commonly referred to as the cache effect.

Ordinary computers, that fit this description, have two main technological limitations. The first concerns the size of the memory, which generally ranges from tens to hundreds of GB. The second is instead related to the number of cores per processor, which, in normal CPUs can range up to tens of units. In order to employ larger amount of resources, both in terms of memory and in terms of computing power, the use of cluster-type architectures (or distributed-memory architectures) is mandatory. These architectures consist of a number of distinct computers (nodes), each operating on its own memory unit, and connected together by an interconnection network. Software capable of operating on these architectures make use of distributed-memory parallelism. This model of parallelism is based on the execution of a number of processes on the nodes of the cluster machine. These processes operate on a private memory, without having access to the private memory of the other processes. Synchronizations and data communications are carried out by means of an exchange of messages, which are explicitly described by the programmer. Several modern computational codes, make use of a hybrid shared-memory/distributed-memory parallelism model. In

this way, they can benefit of the larger amount of memory offered by distributed-memory architectures, as well as the multithread parallelism.

The performances of parallel calculations

By solving a calculation with N processes, it is reasonable to expect the computation time to be reduced by a factor of N with respect to the time of serial computation (i.e., solving the calculation with a single process). In practice, however, this is not the case in general. Parallel algorithms tend to be less effective than the ideal scaling because of the overhead of communications, and the idle time of processes. This latter comes from two factors:

- the non-uniform load of the processes which determines the minimal time required to solve the problem in ideal conditions. In fact, even if all processes can be started at the same time, and with an unlimited quantity of resources, the time required for the parallel execution is determined by the slowest process;
- the need of synchronization between processes which is related to the very nature of the algorithm. Some processes require information that is available only after the execution of another process.

In order to quantify the gain introduced by parallelism we define the speedup as the ratio of the computation time of the serial calculation (on one single process) and the computation time of the parallel one, namely

$$S(N) = \frac{t(1)}{t(N)}. \tag{1.78}$$

We also define the parallel efficiency of the algorithm as the ratio between the speedup at a certain number of processes, and the number of processes itself,

$$E(N) = \frac{S(N)}{N}. \tag{1.79}$$

The parallel efficiency can be interpreted as theratio between the ideal computation time in parallel, $t(1)/N$, and the actual computation time, $t(N)$. In most cases, because of the aforementioned overheads, the efficiency is smaller than 1 but, in some situations, it may be even higher. This happens because of the non linearity of the memory access time with respect to the size of the problem.

According to the Amdahl's law, the speedup of the calculation is limited by the portion of the calculation that can not be solved in parallel. Amdahl's law expresses the theoretical speedup as

$$S_A(N) = \frac{1}{(1-p) + \frac{p}{N}} \tag{1.80}$$

where p is the portion of computation time that can be solved in parallel, and N is the number of cores. For $N \to \infty$, the speedup is limited to $1/(1-p)$, and the minimum calculation time

is given by the portion of the calculation that has to be performed in serial, i.e. $(1-p)\,T$, with T the total computing time of the serial calculation. It follows that, in order to maximise the parallel efficiency, one has to design an algorithm yielding a value p as close as possible to 1.

In the field of parallel computing, speedup and parallel efficiency are used to measure the scalability of an algorithm. That is, they are useful to analyse its behaviour as the number of processes employed varies. The scalability tests are mainly two. The weak scaling test consists of proportionally increasing the problem size along with the number of processes. This allows to evaluate to what extent the algorithm is able to solve problems of increasing size. The strong scaling test, on the other hand, consists of solving the same problem with an increasing number of processors. So, it allows to estimate how much a calculation can be speeded up by means of parallelism.

1.4.2 Parallel calculation of the transport equation

The main idea of Domain Decomposition Methods is to partition the phase space into smaller subdomains, so as to decompose the calculation into a number of smaller calculations. In general, this decomposition is done by partitioning the spatial domain but there exist also examples where DDM is performed on the energy and angular domains, [59]. DDM allows to exploit parallelism to reduce the calculation time, and, most importantly, to partition the memory demand. By running each subdomain on a different process, the large amount of memory required to solve the original problem will be spread over the different processes. However, depending on the algorithm, the coupling between the subdomains requires additional data to be stored and communicated among subdomains.

In discrete-ordinates transport methods, most of the research has been focused on the parallelization of the so-called sweep algorithm, which consists in solving the transport equation for each region of the spatial mesh, for a fixed direction, starting from the problem incoming boundaries, and proceeding following the propagation front until the outgoing boundary is met. This procedure is done for all the angles in the discrete ordinate formulation (see Section 1.2.4). Because of the nature of the transport sweep, downstream regions need to wait the upstream regions to be solved before being executed. If two regions are solved by different nodes, an additional cost is required to communicate the information between the two nodes. As for the load of each process, this is proportional to the number of regions to be solved. The optimal parallel algorithm consists in finding a partitioning of the spatial mesh so that the number of regions is evenly distributed among processes, at the same time guaranteeing a minimal number of communications between processes, and by providing a distribution/ordering of the tasks among the processes (i.e., a scheduling) in order to reduce idle times. A general solution to this problem is not trivial. For conformal Cartesian meshes, this problem has been largely analysed and a first solution was proposed by Koch, Baker and

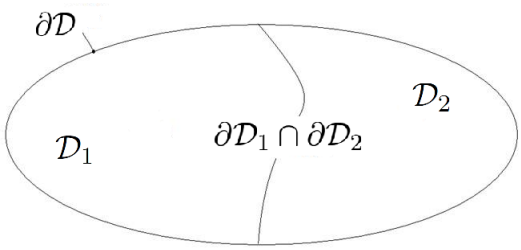


Fig. 1.4 Illustration of the spatial domain decomposition.

Alcouffe (KBA method), [11, 60]. The method has been then improved by Adams, which has designed an optimal scheduling of spatial-angular tasks that maximizes the parallel efficiency, thus reducing at most the idle time, [61].

Other authors have tackled the problem by releasing the condition of exact inversion of the streaming-plus-removal operator in order to eliminate the need of the scheduling. The main idea is to artificially decouple boundary fluxes between subdomains in order to solve a local transport problem for each node with unconverged boundary conditions. The advantage of this approach is that all subdomains can be solved in parallel without any concern about the ordering. This leaves more room for the optimization of the partitioning in order to minimize communications and distribute the load evenly. Such decoupling, of course, requires an iterative solution to converge the actual boundary fluxes. In some cases, this iterative solution may also turn out to be very slow. The use of suitable preconditioners (also known as acceleration methods) tends to stabilize the iterative algorithm, [1, 62].

A variant of this last DDM strategy is obtained by using a decomposition into spatial-angular subdomains, [63]. This approach allows for a finer granularity of parallelism and, if well calibrated, can help in generating a more uniform load distribution among processes. However, by using an angular decomposition, the algorithm requires an additional communication in order to compute the flux moments, and to update the scattering source.

In the following, we will focus on the Domain Decomposition Method employed in IDT.

1.4.3 The Domain Decomposition Method in the IDT solver

IDT makes use of a spatial domain decomposition without overlapping, [1, 58, 64-66], that is, the spatial domain $\mathcal{D}$ is partitioned into N_i box-shaped subdomains $\mathcal{D}_i$ (see Figure 1.4), such that

$$\bigcup_{i=1,N_i} \mathcal{D}_i = \mathcal{D} \text{ and } \mathcal{D}_{i'} \cap \mathcal{D}_i = \emptyset \text{ if } i' \neq i. \tag{1.81}$$

We identify with $\mathcal{X}_i$ the phase space restricted to subdomain $\mathcal{D}_i$, and with $\partial\mathcal{X}_i^+$ and $\partial\mathcal{X}_i^-$

the outgoing and incoming boundaries, i.e.

$$\begin{aligned} \mathcal{X}_i &= \{\mathbf{r} \in \mathcal{D}_i, \mathbf{\Omega} \in \mathcal{S}^2, E \in \mathbb{R}^+\}, \text{ and} \\ \partial\mathcal{X}_i^+ &= \{\mathbf{r} \in \partial\mathcal{D}_i, \mathbf{\Omega} \in \mathcal{S}^2 | \mathbf{\Omega} \cdot \mathbf{n}_i > 0, E \in \mathbb{R}^+\}, \\ \partial\mathcal{X}_i^- &= \{\mathbf{r} \in \partial\mathcal{D}_i, \mathbf{\Omega} \in \mathcal{S}^2 | \mathbf{\Omega} \cdot \mathbf{n}_i < 0, E \in \mathbb{R}^+\}, \end{aligned}$$

with $\mathbf{n}_i$ the outgoing normal direction to the subdomain boundaries.

Domain decomposition (DDM) allows us to solve the original problem,

$$\begin{cases} \mathcal{B}\psi(\mathbf{x}) = Q_{ext}(\mathbf{x}) & \text{with } \mathbf{x} \in \mathcal{X}, \\ \psi(\mathbf{x}) = \psi^-(\mathbf{x}) & \text{for } \mathbf{x} \in \partial\mathcal{X}^-, \end{cases} \tag{1.82}$$

by rewriting it in the form of a set of smaller problems for the N_i subdomains coupled together by the boundary conditions,

$$\begin{cases} \mathcal{B}_i\psi_i(\mathbf{x}) = Q_i(\mathbf{x}) & \text{with } \mathbf{x} \in \mathcal{X}_i, \\ \psi_i(\mathbf{x}) = \psi_i^-(\mathbf{x}) & \text{for } \mathbf{x} \in \partial\mathcal{X}_i^-. \end{cases} \tag{1.83}$$

$\mathcal{B}_i$ denotes a generic transport operator. ψ_i is the angular flux, restricted to the subdomain $\mathcal{D}_i$, namely

$$\psi(\mathbf{x}) = \sum_{i=1,N_i} \chi_i(\mathbf{r})\psi_i(\mathbf{x}), \quad \text{with} \quad \chi_i(\mathbf{r}) = \begin{cases} 1 & \mathbf{r} \in \mathcal{D}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{1.84}$$

Q_i is the source term in subdomain $\mathcal{D}_i$, accounting for the external source and the fission source. ψ_i^-, instead, is the incoming flux at the subdomain boundaries, given either by the outgoing flux of neighbouring subdomains or by boundary conditions (defined on the boundaries of domain $\mathcal{D}$). The solution of each of the N_i sub-problems described in Eq. (1.83) identifies a distinct calculation. In IDT, domain decomposition is applied on the entire multigroup calculation, that is, IDT solves for each of the subdomains for all energy groups. The transport operator $\mathcal{B}_i$ in Eq. (1.83) is defined as $\mathcal{B}_i = (\mathcal{L} - \mathcal{H})_i$. Thus, the calculation of each of the N_i sub-problems involves a local inversion of the entire multigroup transport operator, by means of internal and thermal iterations.

The treatment of subdomain dependencies

In IDT, the dependence between different tasks (caused by the coupling between fluxes at the interfaces between subdomains) is solved by means of an additive Schwarz algorithm, [67, 68], which coincides with a Block-Jacobi iterative scheme. Namely, by introducing the index

of external iterations (e) , the problem in Eq. (1.83) is rewritten as

$$\begin{cases} \mathcal{B}_i \psi_i^{(e+1)}(\mathbf{x}) = Q_i(\mathbf{x}) & \text{with } \mathbf{x} \in \mathcal{X}_i, \\ \psi_i^{(e+1)}(\mathbf{x}) = \psi_i^{-,(e+1)}(\mathbf{x}) & \text{for } \mathbf{x} \in \partial\mathcal{X}_i^-, \end{cases} \tag{1.85}$$

with incoming flux obtained either by the outgoing flux from adjacent subdomains or by the boundary conditions of the global problem. For homogeneous boundary conditions relating edges of the same subdomain, the flux at the interfaces is updated along the spatial and angular transport during the subdomain calculation, namely

$$\psi_i^{-,(e+1)}(\mathbf{x}) = \int_{\partial\mathcal{X}_i^+} dx' \ \beta(\mathbf{x}' \to \mathbf{x}) \psi_i^{+,(e+1)}(\mathbf{x}'). \tag{1.86}$$

At the interface between neighbouring subdomains, and when the boundary conditions relate interface fluxes of different subdomains (e.g. translation boundary conditions), the flux is obtained by the previous external iteration:

$$\psi_i^{-,(e+1)}(\mathbf{x}) = \begin{cases} \psi_{i'}^{+,(e)}(\mathbf{x}) & \text{for } \mathbf{x} \in \partial\mathcal{X}_i^- \cap \partial\mathcal{X}_{i'}^+, \\ \int_{\partial\mathcal{X}_{i'}^+} dx' \ \beta(\mathbf{x}' \to \mathbf{x}) \psi_{i'}^{+,(e)}(\mathbf{x}'), & \text{otherwise.} \end{cases} \tag{1.87}$$

When an external source is imposed on a boundary, the incoming flux is defined as

$$\psi_i^{-,(e+1)}(\mathbf{x}) = \psi_{inc}^-(\mathbf{x}) \quad \text{with } \mathbf{x} \in \partial\mathcal{X}_i^-. \tag{1.88}$$

At each iteration $(e+1)$, each subdomain is solved independently for a fixed source and incoming boundary fluxes (except for the interfaces affected by the homogeneous boundary conditions described in Eq. (1.86)). Boundary fluxes are updated with fluxes of the previous iteration (e), and no dependence is present between subdomains within the current iteration. At the end of each iteration, the boundary fluxes are exchanged between subdomains and the process is repeated until the flux, Eq. (1.84), converges on the solution of the original problem, Eq. (1.82).

The term Q_i is defined as the external multigroup source $Q_{i,ext}$. In the presence of fissions, Q_i assumes the iterative index $(e+1)$ as it accounts also for the fission source term. In detail, we have that $Q_i^{(e+1)} = Q_{i,ext} + \mathcal{F}\psi_i^{(e)}$, where the fission source is updated using fluxes of the previous iteration. Therefore, external iterations will have to ensure both the convergence on the fission source and of the boundary fluxes.

The Parallel Multigroup Block Jabobi iterative scheme in IDT

The iterative scheme is named Parallel Multigroup Block Jabobi (PMBJ) since each block consists of a multigroup calculation. Being independent to each other, the blocks, i.e. the subdomain calculations, are solved in parallel on several processes.

Since the conformity of the spatial mesh is ensured between subdomains, the exchange of the boundary fluxes can take place in an exact manner, i.e., without introducing any additional approximation. The communication simply consists in the transmission of the spatial moments of the angular flux at the interfaces of the regions coincident with the edges of the subdomains, for each direction and for each energy group.

Iterations are repeated until convergence is reached. In practice, at each iteration the algorithm estimates the value of the error on the partial currents at the subdomain interfaces,

$$\epsilon_{\psi}^{(e+1)} = \max_{\substack{s\in\partial\mathcal{D}_i \\ i=1,N_i \\ g=1,G}} \left| 1 - \frac{\int_{\Gamma_i^s} d\mathbf{r} \int_{2\pi^-} d\mathbf{\Omega}\ |\mathbf{\Omega}\cdot\mathbf{n}_i^s| \psi_{i,g}^{-,(e+1)}(\mathbf{r},\mathbf{\Omega})}{\int_{\Gamma_i^s} d\mathbf{r} \int_{2\pi^-} d\mathbf{\Omega}\ |\mathbf{\Omega}\cdot\mathbf{n}_i^s| \psi_{i,g}^{-,(e)}(\mathbf{r},\mathbf{\Omega})} \right|, \tag{1.89}$$

and check that this is less than a certain user-defined tolerance. Here, Γ_i^s stays for the interface s of the subdomain $\mathcal{D}_i$ boundaries. In the presence of fissions, an error estimation on the multiplication coefficient and on the fission source is added to this convergence test.

The major drawback of the PMBJ algorithm is the possible deterioration of the convergence properties of the iterative scheme, which may lead to instabilities. In particular, by breaking the inversion of the streaming-plus-removal operator over multiple blocks (subdomains), the particle transmission will be delayed across iterations. At the start, since the incoming fluxes are initially far from the converged value, the solution of each block will converge on an incorrect result. In order to propagate a certain information across all subdomains, a minimum number of external iterations is required. The solution used in IDT to stabilize the convergence of the PMBJ is to use a global Coarse Mesh Finite Difference (CMFD) operator as preconditioner, [1]. At each external Jacobi iteration, the CMFD is solved for the whole problem to obtain a correction for the boundary and volume fluxes. Other preconditioners are available in IDT such as the Boundary Projection Acceleration (BPA), [3], and the Diffusion Synthetic Acceleration (DSA), [2], but they have never been applied to DDM for stabilizing PMBJ. Moreover, this scheme has only been tested for reactor core applications.

The PMBJ algorithm is an important example of coarse grain parallelism as it maximizes the number of operations solved by the single independent calculation, and minimizes the number of exchanges and synchronizations between subdomains. Domain decomposition via PMBJ has a very high computational efficiency. Tests of weak and strong scaling on shared-memory architectures have shown parallel efficiencies above unity. In fact, for small subdomain sizes, the memory required to solve the calculation may fit entirely within the

CPU cache memory. It has been observed that, in such cases, the faster memory access allowed by cache effects traduces into parallel performances up to 120%, [58].

1.4.4 The IDT solver and the PMBJ implementation

In IDT, subdomains are grouped into lists, and distributed over the different nodes of the cluster. Each node is thus in charge of solving a list of subdomains. Synchronizations between nodes occur at the beginning of each external iteration thanks to MPI communications. Within each iteration, each node solves the tasks (subdomains calculations) in parallel running on the CPU cores. Their execution is managed and synchronized using OpenMP directives. IDT, therefore, makes use of a mixed distributed-shared memory parallelism.

The discussion so far has presented the general resolution scheme employed by IDT. However, although not employed in this document, it is worth to mention how IDT can also employ hybrid Gauss-Seidel resolution schemes. While subdomains on different nodes are still solved block-wise (that is, using the incoming fluxes of the previous external iteration), the solution of subdomains within the same node can be done by solving them in a given order, and immediately updating the incoming fluxes of the neighbours. These schemes aim to reduce the transmission delay implied by PMBJ. Examples are the KBA-like (the ordering of the subdomains is given by a propagation front), and the red-and-black scheme presented in [58].

1.5 Summary

In this background chapter, we introduced the transport theory and the deterministic approximations that are generally used to solve transport calculations. Further, we presented the main issues encountered when solving radiation shielding problems, in particular for S_N-based solvers, such as the need of an accurate angular discretization and a proper representation of the angular anisotropy of the flux and of the scattering. We have also seen how, in order to address large size problems, a large amount of computing resources is required. The access to distributed computational resources is therefore essential.

The deterministic IDT solver has been extensively employed in reactor core calculations, thus with a major focus on diffusive problems with optically thick media and evenly distributed sources. However, the presence of an effective domain decomposition makes IDT an ideal candidate for solving radiation shielding calculations. The goal of this thesis is to research and develop numerical methods for solving shielding calculations with higher accuracy, and within a reasonable computational time.

Preliminary calculations, [69], as well as the extensive bibliography on this topic, [31], have highlighted the importance of First Collision Source methods. Moreover, FCS alone is not enough to completely remove ray-effect since it deals with uncollided particles only,

which may not be the only source of ray-effect. In the first part of this research, we addressed the topic of the FCS calculation methods. Secondly, we developed an alternative angular discretization method in order to reduce ray-effect arising from the collided particles. These methods have been developed within a DDM context. The final part of this manuscript is dedicated to the application of the new methods to a more realistic calculation.

Chapter 2

Development of a fine transport operator for the treatment of point-like sources

The solution of transport problems where particles are emitted from an external point-like source are in general problematic for common transport solvers. This is the case for two reasons. In the first place, the spatial and angular discretization methods may be unable to properly treat the singularities of the point-kernel. Secondly, the solution would be drastically affected by ray-effect. Hence the importance of properly treating point-like sources using ad-hoc discretization, i.e. by the First Collision Source (FCS) method, [70].

The objective of the research presented in this chapter is the design of an ad-hoc discretization of the transport operator for the calculation of point-like sources by the First Collision Source within the IDT solver. In this respect, we have reviewed the techniques that have been presented in literature. Noticing the number of viable techniques, we explored some of them. We first designed a FCS method based on the Method of Characteristics (MOC) and relying on two different angular discretizations. By observing the limits of MOC, i.e the non uniform trajectory-regions intersection density over the domain, and the need of evaluating spherical-harmonics-dependent trajectory weight, we proposed two alternative methods: the trajectory-splitting MOC (TS-MOC), and the Monte Carlo MOC (MC-MOC). Then, we also tested a method based on quadrature points (QP) which is a quite common approach in literature. Next, we analysed the integration accuracy and computational cost of these techniques on a set of benchmark problems. Finally, we verified them on the Kobayashi benchmark problem.

2.1 Analysis of the available First Collision Source methods

In Section 1.3.4 we discussed the importance of dealing with localized sources thanks to First Collision Source methods. This has to rely on an ad-hoc spatial-angular discretization of the streaming-plus-removal operator so as to remove the ray-effect from the uncollided flux, and yield an accurate estimate of the first collision source.

In practice, the operator is applied individually to each point source, and, for each of them, it performs the calculation of the uncollided flux over the entire phase space. Sources whose volume is not negligible are generally decomposed into point sources and then solved separately by FCS. Since the computational cost of FCS scales linearly with the number of sources, the decomposition of three-dimensional volume sources into points could be expensive. This is why the application of FCS is in general restricted to the calculation of the most peaked components of source.

Brief review of FCS methods

In literature, various strategies for the calculation of the uncollided flux have been proposed. One of the first FCS methods is the semi-analytic technique developed by Alcouffe *et al.* within the 2-D transport code TWODANT, [71]. The method uses a ray-tracing technique for evaluating the flux attenuation between the source and the vertices of each region of the spatial mesh. By means of semi-analytical relations, it evaluates the currents at the interfaces of each spatial region and finally, it calculates the region average fluxes by solving an angle-integrated balance equation. The same code presents also a second FCS method based on a stochastic approach, [33]. This second algorithm relies on a Monte Carlo estimator for integrating the region average uncollided fluxes. The method operates the ray tracing along randomly sampled trajectories crossing the entire domain. Each of them is generated by sampling both the source point and the propagation direction. A different FCS strategy has been implemented in the transport code ATTILA, [72] : it makes use of a 3-D quadrature formula for the computation of the angular moments of the uncollided flux in each spatial region. More precisely, a ray-tracing strategy is used for the evaluation of the local value of the uncollided flux on the quadrature points. Then, the code uses a linear mapping for reconstructing the flux moments on the nodes of the tetrahedral finite-element mesh used by the code. A similar approach consists in using spatial (2D or 3D) quadrature rules for directly integrating the uncollided flux on the spatial mesh regions. This has been applied to transport codes such as DENOVO, [9], and Rattlesnake, [73]. In a similar manner, FNSUNCL3, the FCS algorithm in TORT, [74], uses a single point quadrature (constant approximation), or rather quadrature rules with points uniformly distributed over the spatial regions. The code JSNT-S also relies on an analogous strategy, but this time reconstructing the region average uncollided flux using the point values of the uncollided flux, evaluated on the corners of the spatial mesh, [75].

The IDT solver, [53], presents a different technique based on the Method of Characteristics (MOC) in spherical coordinates. This method relies on an angular quadrature for generating a set of trajectories passing from the point source. The spatial integration is performed by using the spatial discretization resulting from the intersection of these trajectories with the calculation mesh. Another application of MOC can be found in the FCS developed by Hanuš *et al.* for spatially distributed sources, [54]. Their approach is based on a double-MOC integration: the first one, similarly to IDT, allows for uncollided flux spatial integration, while the second one is used for accounting the spatial dimensionality of the emitting source and its optical thickness. The author reports how this strategy can be better suited than others for optically thick source media. In such a case, the most relevant contribution to the uncollided flux all over the domain derives from the superficial portions of source regions. The standard approach would require a very fine point source decomposition in order to account for this effect. In this respect, the treatment of volume sources through double ray tracing can be particularly useful.

Summary

In summary, we notice how different FCS approaches are available. By considering their features, we may group them into two families: the one based on a MOC integration, and the ones relying on quadrature rules or semi-analytical relations for the uncollided flux reconstruction in domain regions. The methods of the first group aim to integrate more than one spatial region with the same trajectory. The seconds, instead, integrate the uncollided flux on each spatial region independently, requiring to trace, for each region, an ad-hoc set of trajectories. Intuitively, the first approach should be preferred for unstructured meshes. This is because in these cases ray tracing may be expensive, and one may be interested in limiting the number of trajectories, [54]. Methods of the second group may be useful instead for monitoring the uncollided flux estimation precision on each region separately. Furthermore, this gives room for methods performing an adaptive refinement. At present state, a MOC-based FCS has been already implemented in a mock-up version of IDT, but this is limited to the constant spatial order. As first step, we want to extend the MOC method up to the linear order. However, in order to define the first collision source, we have preliminarily to derive a discretized formulation of the uncollided flux.

2.2 Derivation of the expression for the uncollided flux emitted by point-wise sources

As mentioned, the principle of FCS is to estimate the uncollided flux, as in Eq. (1.74), and then feed Eq. (1.75) with the first collision source, defined in Eq. (1.76). Since the scattering operator $\mathcal{H}$ is approximated by the well-known Legendre polynomials expansion

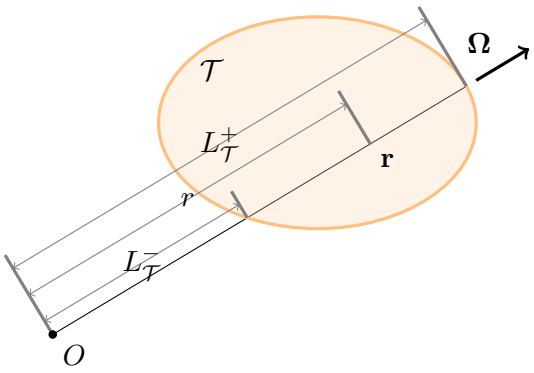


Fig. 2.1 Illustration of the characteristic line coordinate system with point source located in the axes origin.

(see Section 1.2.2), the computation of Q_{FCS} entails the calculation of uncollided flux angular moments $\phi^u_h(\mathbf{r})$ defined as

$$\phi^u_h(\mathbf{r}) = \frac{1}{4\pi}\int_{4\pi} d\mathbf{\Omega}\ A_h(\mathbf{\Omega})\psi^u(\mathbf{r},\mathbf{\Omega}) \quad \text{for } h = 1,\ldots,H. \tag{2.1}$$

Since the spatial discretization in IDT is inherited from the short characteristics method up to the linear order (see Section 1.2.4), it is also necessary to compute higher-order spatial moments by projecting Eq. (2.1) onto the constant- and linear-order polynomials in each target region $\mathcal{T}$. The spatial-angular moments $\phi^{u,\mathcal{T}}_{m,h}$ are thus defined by the equation

$$\begin{aligned} \phi^{u,\mathcal{T}}_{m,h} &= \frac{1}{4\pi V}\int_{\mathcal{T}} d\mathbf{r}\ P_m(\mathbf{r})\int_{4\pi} d\mathbf{\Omega}\ A_h(\mathbf{\Omega})\psi^u(\mathbf{r},\mathbf{\Omega}), \\ \text{for } h &= 1,\ldots,H \text{ and } m = 1,\ldots,M, \end{aligned} \tag{2.2}$$

where $\mathcal{T} \in \mathbb{R}^3$ is a target Cartesian region of volume V, and $M = 4$ (corresponding to the linear spatial order). The spatial-angular discretization implies the representation of the uncollided flux over the finite dimensional basis

$$\mathcal{A}_H \times \mathcal{P}_M = span\left\{A_h(\mathbf{\Omega})P_m(\mathbf{r}), h = 1,\ldots,H, m = 1,\ldots,M\right\},\ with\ \mathbf{\Omega} \in 4\pi, \mathbf{r} \in \mathcal{T}$$

i.e, the uncollided flux can be expressed with expansion

$$\psi^u(\mathbf{r},\mathbf{\Omega}) = \sum_{m=1}^{M}\alpha_m\sum_{h=1}^{H} a_h\psi^{u,\mathcal{T}}_{m,h}P_m(\mathbf{r})A_h(\mathbf{\Omega}), \tag{2.3}$$

where α_m and a_h are normalization coefficients of the basis (defined as in the expansions in Eqs.(1.47) and (1.61)), and $\psi^{u,\mathcal{T}}_{m,h}$ are the coefficients of the expansion.

In the propagation of an isotropic point-like source, the source intensity is described by a Dirac δ-function in space. Thus, by locating the origin of the reference system in the point

source, such that

$$\mathbf{r} = r\mathbf{\Omega}, \tag{2.4}$$

Eq. (1.72) becomes

$$\psi^u(\mathbf{r}, \mathbf{\Omega}) = Q\ \frac{1}{r^2} e^{-\tau(\mathbf{r})} \delta_2\left(\mathbf{\Omega} \cdot \hat{\mathbf{e}}_r\right), \tag{2.5}$$

where r is the length of vector $\mathbf{r}$, and $\hat{\mathbf{e}}_r = \mathbf{r}/r$ is the unit vector in direction $\mathbf{r}$ (see Figure 2.1).

Since the spatial integral appearing in Eq. (2.2) is generally evaluated numerically, we have to take into account the errors introduced by the numerical integration, which especially affect the orthogonal properties of the spatial-angular basis. This is done by substituting the analytic formulation of the uncollided flux, Eq. (2.5), on the LHS of Eq. (2.3), to obtain

$$\sum_{m=1}^{M} \alpha_m \sum_{h=1}^{H} a_h \psi_{m,h}^{u,\mathcal{T}} P_m(\mathbf{r}) A_h(\mathbf{\Omega}) \approx Q\ \frac{1}{r^2} e^{-\tau(\mathbf{r})} \delta_2\left(\mathbf{\Omega} \cdot \hat{\mathbf{e}}_r\right), \tag{2.6}$$

where we have inverted the order of the LHS and the RHS of the final equation. Next, the spatial-angular flux components of expansion Eq. (2.3) are obtained by solving the matrix form of the latter equation,

$$\overline{\overline{\mathbf{M}}}\mathbf{\Psi} = \mathbf{\Phi}, \tag{2.7}$$

obtained by projecting numerically Eq. (2.6) onto the spatial angular basis $\mathcal{A}_H \times \mathcal{P}_M$.

In particular,

- $\mathbf{\Psi} = \{\psi_{m,h}^{u,\mathcal{T}}\}_{\substack{m=1,M \\ h=1,H}}$ is the vector containing the components $\psi_{m,h}^{u,\mathcal{T}}$ of the spatial-angular expansion defined in Eq. (2.3).
- $\mathbf{\Phi} = \{\phi_{m,h}^{u,\mathcal{T},*}\}_{\substack{m=1,M \\ h=1,H}}$ is the vector of the numerical spatial-angular moments $\phi_{m,h}^{u,\mathcal{T},*}$, formally computed according to Eq. (2.2), but with numerical spatial integration,

$$\phi_{m,h}^{u,\mathcal{T},*} = Q \int_{4\pi} d\mathbf{\Omega} \int_{\mathcal{T}}^{*} d\mathbf{r}\ P_m(\mathbf{r}) A_h\left(\mathbf{\Omega}\right) \mathcal{K}(\mathbf{r})\ \delta_2\left(\mathbf{\Omega} \cdot \hat{\mathbf{e}}_r\right) \tag{2.8}$$

  with superscript '*' indicating the approximated integrals. The kernel $\mathcal{K}(\mathbf{r})$ is defined as

$$\mathcal{K}(\mathbf{r}) = \frac{1}{r^2} \exp\left[-\int_0^r dr'\ \Sigma\left(r'\hat{\mathbf{e}}_r\right)\right] = \frac{e^{-\tau(\mathbf{r})}}{r^2}. \tag{2.9}$$

  By analytically solving the angular integral of Eq. (2.8), the expression becomes

$$\phi_{m,h}^{u,\mathcal{T},*} = Q \int_{\mathcal{T}}^{*} d\mathbf{r}\ P_m(\mathbf{r}) A_h\left(\hat{\mathbf{e}}_r\right) \mathcal{K}(\mathbf{r}). \tag{2.10}$$

- $\overline{\overline{\mathbf{M}}}$ is the mass matrix having elements defined as

$$M_{(m,h)(m',h')} = \alpha_h \alpha_{h'} a_m a_{m'} \int_{4\pi} d\mathbf{\Omega}\ A_h(\mathbf{\Omega}) A_{h'}(\mathbf{\Omega}) \int_{\mathcal{T}}^{*} d\mathbf{r}\ P_m(\mathbf{r}) P_{m'}(\mathbf{r}). \tag{2.11}$$

  As in Eq. (2.8), the angular integral in Eq. (2.11) is solved analytically, leading to

$$M_{(m,h)(m',h')} = \delta_{h,h'} a_m a_{m'} \int_{\mathcal{T}}^{*} d\mathbf{r} P_m(\mathbf{r}) P_{m'}(\mathbf{r}), \tag{2.12}$$

  where $\delta_{h,h'}$ is the Kronecker delta. Therefore, the orthonormality of the spherical harmonics is respected. However, this is not the case for the spatial basis in general. The mass matrix is thus block-diagonal. This largely simplifies the resolution of Eq. (2.7) since each $M \times M$ block of the spatial components can be inverted separately.

The uncollided flux is therefore computed by inverting Eq. (2.7) and evaluating numerically Eqs. (2.10) and (2.12). Hence, different FCS methods may differ by the way these integrals are approximated. Moreover, we notice how, in case of multigroup applications, the uncollided flux may be evaluated independently in each energy group. In IDT, the calculation is performed simultaneously for all groups. For simplicity, our research is restricted to the propagation of isotropic point sources, so the angular dependence of the source term Q is neglected. Even though isotropic sources are the most common in shielding calculations, the further extension to anisotropic sources does not involve any relevant complexity. Lastly, we consider FCS methods accounting for vacuum boundary conditions only. The treatment of geometrical boundary conditions is for the moment shelved (see Section 1.1.2).

Calculation of the adjoint uncollided flux

Before proceeding further in the derivation of the MOC First Collision Source we want to spend a few words on the application of FCS to adjoint calculations. We have seen in Section 1.1.2 how the main difference between the direct and adjoint calculation lies in the sign of the displacement operator (see Eq. (1.12)). The same change of sign is present also in the Adjoint point-kernel,

$$\psi^{\dagger,u}(\mathbf{r}, \mathbf{\Omega}) = Q^{\dagger}\ \frac{1}{r^2} e^{-\tau(\mathbf{r})} \delta_2\left(-\mathbf{\Omega} \cdot \hat{\mathbf{e}}_r\right), \tag{2.13}$$

which physically represents the importance of particles, without collisions, for a given isotropic response function $Q^{\dagger}$, defined on a point centred with the origin of the reference frame. In particular, the adjoint uncollided flux $\psi^{\dagger,u}(\mathbf{r}, \mathbf{\Omega})$ is non-zero only for particle directions pointing towards the origin.

By following the same procedure used to derive the spatial-angular moments of the uncollided flux, we obtain the formulation for the moments of the adjoint uncollided flux, i.e.

$$\phi_{m,h}^{\dagger,u,\mathcal{T},*} = Q^{\dagger} \int_{\mathcal{T}}^{*} d\mathbf{r}\ P_m(\mathbf{r}) A_h\left(-\hat{\mathbf{e}}_r\right) \mathcal{K}(\mathbf{r}). \tag{2.14}$$

where the only difference with respect to Eq. (2.10) lies in the sign of the propagation direction feeding the spherical harmonics functions. Since no relevant modification of the standard FCS method is required in order to solve adjoint FCS calculations, from now on we will deal with techniques for the integration of the "direct" uncollided flux only.

2.3 Design of the First Collision Source method based on the Method of Characteristics

The Method of Characteristics (MOC) performs the integration of Eq. (2.10) thanks to a coordinate transformation to the spherical reference frame centered in the point source, depicted in Figure 2.1. The change of variable allows to rewrite the volume integral over region $\mathcal{T}$ as

$$\int_{\mathcal{T}} d\mathbf{r} = \int_{4\pi} d\mathbf{\Omega} \int_{L_{\mathcal{T}}^{-}(\mathbf{\Omega})}^{L_{\mathcal{T}}^{+}(\mathbf{\Omega})} dr\ r^2, \tag{2.15}$$

i.e., as an angular and a line integral along the radial direction $\hat{\mathbf{e}}_r$. In the last equation $L_{\mathcal{T}}^{\pm}$ are the distances from the source of the exiting/entering points of trajectory in the target region. The angular variable then is discretized into solid angles according to a given angular quadrature formula. For each solid angle, a continuous trajectory is then tracked starting from the source and crossing the whole domain. The direction of the trajectory is taken as the center of the solid angle. The chords generated by the intersection of these trajectories with the spatial mesh entail a trajectory-based discretization of the geometry. Each mesh region is so decomposed into a set of homogeneous spherical-shaped slices (cones) having the spherical incoming/outgoing transverse surfaces as bases, and the chord-length as radial height (see Figure 2.2). By using such representation, we implicitly neglect the angular dependence of the chord length within the cone. The resulting numerical form of the MOC projection is

$$\int_{\mathcal{T}}^{*} d\mathbf{r}\Big|_{MOC} = \sum_{i \cap \mathcal{T}} \int_{\mathcal{S}_i^2} d\mathbf{\Omega} \int_{L_i^{-}}^{L_i^{+}} dr\ r^2, \tag{2.16}$$

where $i \cap \mathcal{T}$ indicates the set of chords intersecting the target $\mathcal{T}$, and $L_i^{\pm}$ are the distances between the source and exiting/entering points of the target region measured along chord i. The symbol $\mathcal{S}_i^2$ is the solid angle associated to the chord i, while $\mathbf{\Omega}_i$ indicates the direction of the chord, corresponding to the reference direction associated to the solid angle $\mathcal{S}_i^2$. Since each trajectory is continuous, all the chords of a trajectory share the same solid angle and reference direction. Namely,

$$\mathcal{S}_i^2 \equiv \mathcal{S}_t^2 \qquad \text{for } i \in I_t \tag{2.17}$$

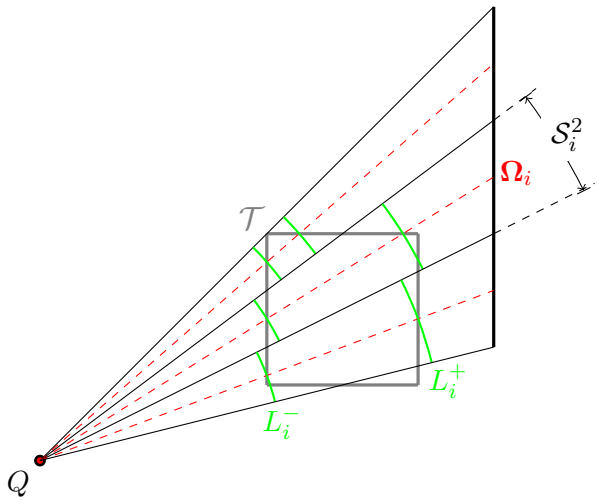


Fig. 2.2 Illustration of the MOC integration.

where t is the index of the continuous trajectory, and I_t is the set of chords belonging to trajectory t. By using the MOC numerical projection in Eq. (2.16), Eq. (2.10) becomes:

$$\phi_{m,h}^{\mathcal{T},*} = Q \sum_{i \cap \mathcal{T}} \int_{\mathcal{S}_i^2} d\mathbf{\Omega} A_h(\mathbf{\Omega}) \int_{L_i^-}^{L_i^+} dr\ r^2 P_m(r\mathbf{\Omega})\ \frac{e^{-\tau(r)}}{r^2}. \tag{2.18}$$

Here we explicitly show the elimination of $1/r^2$ factor to underline the capability of the MOC to remove the singularity of the point-kernel. In the following, we use the coordinate $r' = r - L_i^-$ to derive more convenient formulations for a numerical implementation. Remark that in Eq. (2.18), the exponential attenuation within each solid angle is assumed to be constant along the spherical perpendicular surface of the cone. As for the basis functions, $P_m(r\mathbf{\Omega})$, we rewrite Eq. (1.59) in terms of $\mathbf{\Omega}$ and r:

$$P_m(r\mathbf{\Omega}) = \begin{cases} 1, & m = 1 \\ a_{0,m} + r\ a_{1,m}\mathbf{n}_m \cdot \mathbf{\Omega}, & m = 2,3,4 \end{cases} \tag{2.19}$$

where $a_{0,m}$ and $a_{1,m}$ are coefficients of the m-th components of the spatial base in the target region and $\mathbf{\Omega} \cdot \mathbf{n}_m$ are director cosines. In Eq. (2.19), these coefficients are defined as

$$\begin{aligned} \{a_{0,m}\}_{m=2,3,4} &= \left[\tfrac{-2x_m}{(x_2-x_1)},\ \tfrac{-2y_m}{(y_2-y_1)},\ \tfrac{-2z_m}{(z_2-z_1)}\right]^T, \text{ and} \\ \{a_{1,m}\}_{m=2,3,4} &= \left[\tfrac{2}{(x_2-x_1)},\ \tfrac{2}{(y_2-y_1)},\ \tfrac{2}{(z_2-z_1)}\right]^T, \end{aligned} \tag{2.20}$$

where the region coordinates are measured with respect to the source-centered reference frame.

By substituting Eq. (2.19) into Eq. (2.18), we obtain the formulation

$$\phi_{m,h}^{\mathcal{T},*} = \begin{cases} \sum_{i\cap\mathcal{T}} L_i \psi_i^- \beta_{0,i} \Gamma_{0,h,i}, & \text{for } m = 1 \\ \sum_{i\cap\mathcal{T}} L_i \psi_i^- \left(a_{0,m}\beta_{0,i}\Gamma_{0,h,i} + a_{1,m}\beta_{1,i}\mathbf{n}_m \cdot \mathbf{\Gamma}_{1,h,i} \right), & \text{for } m = 2,3,4 \end{cases} \tag{2.21}$$

where $L_i = L_i^+ - L_i^-$ is the chord length and ψ_i^- is the "incoming flux" in region $\mathcal{T}$, measured along the i-th trajectory. For MOC, this is equal to

$$\psi_i^- = Q\ e^{-\tau(L_i^-)}. \tag{2.22}$$

In Eq. (2.21), the Γ coefficients are the angular weights that take into account the integrals over the solid angle. The coefficient

$$\Gamma_{0,h,i} = \int_{\mathcal{S}_i^2} d\mathbf{\Omega}\ A_h(\mathbf{\Omega}), \tag{2.23}$$

and the vector

$$\mathbf{\Gamma}_{1,h,i} = \int_{\mathcal{S}_i^2} d\mathbf{\Omega}\ \mathbf{\Omega} A_h(\mathbf{\Omega}), \tag{2.24}$$

support the constant and linear spatial contributions, respectively. These terms are strictly related to the solid angles shape, and therefore to the angular discretization yet to be defined. The integration techniques for integrals in Eqs. (2.23) and (2.24) will be presented subsequently in Section 2.3.1. As for the β coefficients, they account for the spatial contribution along the i-th trajectory and are respectively defined as

$$L_i\beta_{0,i} = \int_0^{L_i} dr'\ e^{-\Sigma_i r'} = L_i \left(\frac{1 - e^{-\tau_i}}{\tau_i} \right), \tag{2.25}$$

and

$$L_i\beta_{1,i} = \int_0^{L_i} dr'\ r'\ e^{-\Sigma_i r'} = L_i^2 \left(\frac{\beta_{0,i} - e^{-\tau_i}}{\tau_i} \right). \tag{2.26}$$

In these equations, we have used Σ_i for the total cross-section of the intersected region, and $\tau_i = \Sigma_i L_i$ for the optical thickness.

Equations (2.21) and (2.22) allow for the so-called trajectory sweep. The line integrals are evaluated sequentially by following the trajectory direction from the source to the domain boundaries and by adding their contributions in each intersected region. The incoming flux for each chord i is taken as the exiting flux of the previous chord i',

$$\psi_i^- = \psi_{i'}^+ = \psi_{i'}^- e^{-\tau_{i'}}, \tag{2.27}$$

while, for the first chord, $\psi_i^- = Q$. Note that Γ weights in Eqs. (2.23) and (2.24) do not depend on the target region, but only on the solid angle associated to the chord. Since all chords belonging to a trajectory share the same solid angle, Γ weights are computed before sweeping each trajectory. The flux coefficients $\psi_{m,h}^{u,\mathcal{T}}$ are then computed with Eq. (2.7) by inverting the mass matrix. The MOC mass matrix is evaluated consistently with the method, i.e. by using MOC numerical integration, Eq. (2.16), to evaluate Eq. (2.12). The formulation of the mass matrix discretized with MOC is described in Appendix A.

2.3.1 Definition of the angular discretization and of Γ coefficients integration strategy

In the previous section, we have used an angular quadrature to discretize the unit sphere and to derive equations for the MOC. We required the unit sphere to be subdivided into spherical surfaces having solid angle $\mathcal{S}_t^2$, with t being the index of the trajectory. To each surface we also associate a reference direction, $\mathbf{\Omega}_t$, used to track the trajectory. In order to obtain the desired accuracy for the spatial integration throughout the whole geometry, the discretization of the unit sphere must be such that all spatial regions are crossed by a sufficient number of trajectories. In the present implementation of MOC, we have used two kinds of quadrature formulae:

Boundary quadrature (BQ) : built by projecting the domain boundaries onto the unit sphere centered on the source;

Uniform quadrature (UQ) : product-type quadrature built by splitting the azimuthal and polar angles into a given number of subdivisions.

For a given quadrature, we also need to compute the Γ weights of Eqs. (2.23) and (2.24). In both cases, we use a numerical integration, leading to:

$$\Gamma_{0,h,t} \approx \sum_k \omega_{t,k}\ A_h(\mathbf{\Omega}_{t,k}), \tag{2.28}$$

$$\mathbf{\Gamma}_{1,h,t} \approx \sum_k \omega_{t,k} \mathbf{\Omega}_{t,k} A_h(\mathbf{\Omega}_{t,k}), \tag{2.29}$$

with $\{\omega_{t,k}, \mathbf{\Omega}_{t,k}\}$ representing the weights and the nodes of the quadrature set. Here, due to Eq. (2.17), with the index t we are regrouping all chords i belonging to the same trajectory, so sharing the same solid angle. Please note that the angular quadrature employed by MOC, used to partition the unit sphere into solid angles, is distinct from the quadrature employed for the integration of weights Γ. The details of these two quadratures are discussed in the following sections.

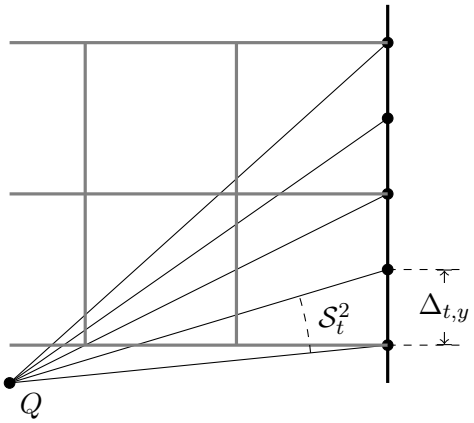


Fig. 2.3 Illustration of the MOC method with the BQ quadrature.

Boundary quadrature (BQ)

The boundary quadrature is obtained by building a 2D Cartesian mesh on the boundaries of the domain, and by projecting it on the unit sphere centered in the point-source. The 2D-Cartesian boundary mesh is obtained by fixing a uniform step, and adjusted in order to guarantee a user-defined minimum number of trajectories per boundary surface of the calculation mesh. By connecting the source point to the nodes of this 2D Cartesian mesh, we identify a set of oblique rectangular pyramids having as bases the Cartesian boundary mesh and as apex the source point. The solid angle associated to each pyramid is the projection of the base of the pyramid on the unit sphere built around the source. The reference direction of the trajectory is obtained by connecting the centroid of the base of the pyramid with the source point. Figure 2.3 depicts in 2D the angular discretization strategy, where the "pyramids" take the shape of a triangle.

The trajectory weights Γ are integrated numerically by subdividing the solid angle $\mathcal{S}_t^2$ into smaller solid angles, $\mathcal{S}_{t,k}^2$, such that $\mathcal{S}_t^2 \equiv \cup_{k \subset t} \mathcal{S}_{t,k}^2$. A step approximation is used for the spherical harmonics and the angular cosine within $\mathcal{S}_{t,k}^2$. The weights in Eq. (2.28) and (2.29) are thus the measures of each solid angle $\mathcal{S}_{t,k}^2$, while the quadrature points are the direction pointing towards their centroids.

As an example, we consider a boundary-surface mesh located on one of the top/bottom side of the XYZ domain. We define z_b as the distance of such boundary from the source point, which is the center of the coordinates system. We consider the generic solid angle defined by the pyramid having the apex in the source point and the base defined by four points

$$\mathbf{r}_{t,1} \equiv (x_{t,1}, y_{t,1}, z_b), \qquad \mathbf{r}_{t,2} \equiv (x_{t,2}, y_{t,1}, z_b),$$
$$\mathbf{r}_{t,3} \equiv (x_{t,1}, y_{t,2}, z_b), \qquad \mathbf{r}_{t,4} \equiv (x_{t,2}, y_{t,2}, z_b),$$

as the vertex of its base. The intervals $[x_{t,1}, x_{t,2}]$ and $[y_{t,1}, y_{t,2}]$ are subdivided in K equally-spaced steps,

$$\Delta_{t,x} = x_{t,2} - x_{t,1}, \quad \delta_{t,x} = \frac{\Delta_{t,x}}{K},$$
$$\Delta_{t,y} = y_{t,2} - y_{t,1}, \quad \delta_{t,y} = \frac{\Delta_{t,y}}{K},$$

where K is a user-defined parameter. Each solid angle $\mathcal{S}^2_{t,k}$ is then evaluated on the sub-pyramid having as base the rectangle defined by the points

$$\begin{aligned}
\mathbf{r}_{t,k,1} &\equiv (x_{t,1} + (i-1)\delta_{t,x},\ y_{t,1} + (j-1)\delta_{t,y},\ z_b) \\
\mathbf{r}_{t,k,2} &\equiv (x_{t,1} + i\delta_{t,x},\ y_{t,1} + (j-1)\delta_{t,y},\ z_b), \\
\mathbf{r}_{t,k,3} &\equiv (x_{t,1} + (i-1)\delta_{t,x},\ y_{t,1} + j\delta_{t,y},\ z_b), \\
\mathbf{r}_{t,k,4} &\equiv (x_{t,1} + i\delta_{t,x},\ y_{t,1} + j\delta_{t,y},\ z_b), \\
\text{for } i,j &= 1 \dots K, \text{ and } k = 1 \dots K \times K.
\end{aligned}$$

The magnitude of solid angle $\mathcal{S}^2_{t,k}$, here indicated with $\Delta\Omega_{t,k}$, is computed using expression

$$\begin{aligned}
\Delta\Omega_{t,k} &= \gamma_z(\mathbf{r}_{t,k,1}) + \gamma_z(\mathbf{r}_{t,k,2}) - \gamma_z(\mathbf{r}_{t,k,3}) - \gamma_z(\mathbf{r}_{t,k,4}), \\
\text{for } k &= 1 \dots K \times K,
\end{aligned} \tag{2.30}$$

with function $\gamma_z(\mathbf{r})$ as

$$\gamma_z(\mathbf{r}) = tan^{-1}\left(\frac{xy}{z\sqrt{x^2+y^2+z^2}}\right). \tag{2.31}$$

In Eqs. (2.28) and (2.29), $\omega_{t,k} = \Delta\Omega_{t,k}$, and the value of $\mathbf{\Omega}_{t,k}$ is chosen as the direction defined by the source-point pointing to the centroid of the base $(\mathbf{r}_{t,k,1}, \mathbf{r}_{t,k,2}, \mathbf{r}_{t,k,3}, \mathbf{r}_{t,k,4})$. In the same manner the integrals over the solid angles defined on the x and y boundary surfaces are computed. It can be shown that Eqs. (2.30) and (2.31) are equivalent to the formulation obtained in [76]. Because of the nature of Eq. (2.31), the evaluation of the amplitude of square-based solid angles implies a not negligible number of floating-point operations. By considering also the number of solid angle evaluations (K^2 times the number of MOC trajectories), we may expect the BQ quadrature to be relatively expensive in terms of computational cost. In the light of these considerations, we have designed an alternative angular quadrature, entailing a computationally lighter solid angle evaluation.

Uniform quadrature (UQ)

The uniform quadrature (UQ) is based on a uniform 2D discretization of the unit sphere. By taking the angular coordinates $\mathbf{\Omega} = (\varphi, \theta)$, with φ as the azimuthal angle in XY, θ as the polar angle with respect to the Z-axis, and $\mu = \cos\theta$ as the polar cosine, the interval $[0, 2\pi] \times [0, \pi]$ is subdivided in an $N_\varphi \times N_\theta$ uniform grid. The resulting solid angles, their measures and their associated reference directions are respectively

$$\begin{aligned}
\mathcal{S}_t^2 &= \left[\frac{2(i_\varphi - 1)\pi}{N_\varphi}, \frac{2i_\varphi\pi}{N_\varphi}\right] \times \left[\cos\left(\frac{i_\theta\pi}{N_\theta}\right), \cos\left(\frac{(i_\theta - 1)}{N_\theta}\pi\right)\right], && (2.32)\\
\Delta\Omega_t &= \int_{\Delta\mathbf{\Omega}_t} d\mathbf{\Omega} = \frac{2\pi}{N_\varphi}\Delta\mu_{i_\theta}\ , && (2.33)\\
\mathbf{\Omega}_t &= \left(\frac{(2i_\varphi - 1)\pi}{N_\varphi}, \frac{(2i_\theta - 1)\pi}{2N_\theta}\right) && (2.34)\\
\text{for } i_\varphi &= 1, \ldots, N_\varphi\ ,\ i_\theta = 1, \ldots, N_\theta,\\
t &= 1, \ldots, N_\varphi N_\theta,
\end{aligned}$$

with $\Delta\mu_{i_\theta} = 2\sin\left(\frac{(2i_\theta - 1)\pi}{2N_\theta}\right)\sin\left(\frac{\pi}{2N_\theta}\right)$. The angular integrals over solid angle $\mathcal{S}_t^2$ are computed by building a product-type $K \times K$ quadrature formula within $\mathcal{S}_t^2$, with a uniform quadrature for the azimuthal variable φ, and a Gauss-Legendre quadrature for the polar cosine μ. The resulting angular weights and directions used in Eqs. (2.28) and (2.29) are :

$$\begin{aligned}
\omega_{t,k} &= \Delta\varphi_{t,i} w_{t,j} = \frac{2\pi}{N_\varphi K} w_{t,j} && (2.35)\\
\mathbf{\Omega}_{t,k} &= (\varphi_t^{in} + \frac{(2i - 1)}{2}\Delta\varphi_{t,i},\ cos^{-1}(\mu_{t,j}))\\
\text{for } i, j &= 1, \ldots, K, \text{ and } k = 1, \ldots, K \times K,
\end{aligned}$$

where $\{w_{t,j}, \mu_{t,j}\}$ are the weights and nodes of the K-order Gauss-Legendre quadrature rule, and $\varphi_t^{in} = \frac{2(i_\varphi - 1)\pi}{N_\varphi}$ for each polar angle i_θ. We remark that, while the quadrature used for MOC contains equally distributed polar angles θ, the quadrature employed to integrate Γ weights is built by using a Gauss-Legendre rule for the variable $\mu = \cos(\theta)$. The choice of uniform θs allows for a better distribution of trajectories within the geometry, and in particular near the poles. Whereas, the Gauss-Legendre quadrature on μ allows for a better integration quality within each solid angle.

2.4 Design of the First Collision Source MOC method with trajectory splitting

The tests in [53] have shown results on the effectiveness and the robustness of the MOC. The method effectively removes the spatial singularity of the point-kernel, and its accuracy can be improved by simply refining the angular quadrature, i.e. by increasing the number of trajectories passing through each region. However, by construction, the geometrical factor r^2 affects the distribution of trajectories in the space, leading to trajectories that tend to rarefy as the distance from the source increases. More precisely, by resorting to the cone-discretization shown in Figure 2.2, the cross sectional area associated to the cone with measure of the solid angle $\Delta\Omega_t$ is given by $r^2\Delta\Omega_t$. Thus, the regions close to the source are intersected by an higher number of chords than the peripheral ones. This causes a consequent non-uniform accuracy of the volume integration that tends to decrease in target regions that are far from the source. In other words, given a goal accuracy, the number of trajectories in the regions around the source is oversized, with a consequent increase in calculation times. These observations motivated the development of the Trajectory Splitting MOC (TS-MOC) technique.

The TS-MOC approaches the problem by splitting the trajectories after a certain distance from the source (splitting radius) in order to obtain a more uniform number of trajectories per unit volume. In this way, the method attempts to ensure uniform accuracy throughout the domain. At each splitting radius, the parent trajectory is ended and a set of child trajectories is tracked until the next splitting radius, or until the geometry boundaries (see Figure 2.4 for a graphical representation). As opposed to the MOC algorithm, trajectories in TS-MOC are no more continuous lines starting from the source and ending at the domain boundaries. Instead, the set of trajectories can be described as a tree of sub-trajectories whose roots are the sub-trajectories tracked from the source, and with arborescences constituted by the child trajectories of each root. The depth of the tree depends on the number of splitting radii and, in general, may vary depending on the size of the geometry, the position of the source, and the splitting strategy (see Figure 2.5).

The numerical projection used by TS-MOC is identical to the one used in MOC, i.e. Eq. (2.16). Eq. (2.21) also remains valid, whereas an additional approximation is done to define the incoming flux for each intersection, ψ_i^-. While the incoming flux for MOC is given by Eq. (2.22), which is exact along the trajectory t and leads to Eq. (2.27), in TS-MOC we assume the optical length of the parent trajectory to be representative of the optical length of the child trajectories at the splitting radius. Eq. (2.27) is then replaced by the expression

$$\psi_i^- = \begin{cases} \psi_m^+ & \text{on the splitting radius} \\ \psi_{i'}^+ & \text{elsewhere} \end{cases}, \tag{2.36}$$

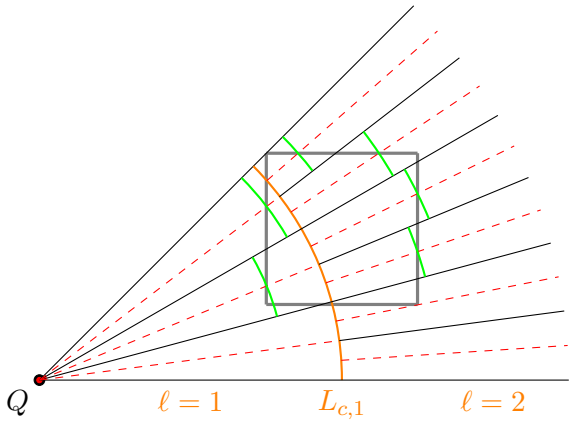


Fig. 2.4 Illustration of the TS-MOC splitting method.

where the subscript m is used to indicate the parent trajectory, while subscript i' indicates the previous intersection. Notice that, since trajectories are no more continuous, the solid angle depends on the sub-trajectory to which the intersection belongs. Therefore, Γ weights in Eqs. (2.23) and (2.24) must be evaluated for each sub-trajectory.

2.4.1 Definition of the TS-MOC splitting strategy

To close the system, it remains to define the splitting strategy, which provides the splitting radii and the solid angles/directions associated to the child trajectories at each splitting radius. Although the TS-MOC can be adapted for several quadrature formulae, in practice, we have applied it to the UQ quadrature (see Section 2.3.1). Therefore, we will derive the conditions for the splitting radii for this particular case. We consider that the first root trajectories are obtained from a $N_\varphi \times N_\theta$ formula, while the measure of the solid angles associated to each root trajectory, $\Delta\Omega_c$, is given by Eq. (2.33). Here, we are using the subscript c to denote the root discretization to stress on the fact that solid angles vary with splitting. The splitting strategy is defined by the following conditions:

1. The number of child trajectories per parent is a user-defined integer constant ν;
2. The maximal cross-sectional area associated to the cone of a sub-trajectory is of the order of a user-defined parameter δ^2;
3. The splitting radii are the same for all sub-trajectories generated by a single root trajectory.

The third condition allows us to denote the splitting radii in terms of the index of the root solid angle, c, and in terms of the depth of the tree, $\ell = 1, \ldots, N_\ell(c)$. Under the above hypotheses, we can define the equation relating the splitting radii, $L_{c,\ell}$, the δ^2 parameter,

the parameter ν and the measure of root solid angle $\Delta\Omega_c$:

$$\frac{\Delta\Omega_c}{\nu^{\ell-1}} L_{c,\ell}^2 = \delta^2 \quad \text{for } \ell = 1, \ldots, N_\ell(c) \text{ and } c = 1, \ldots, N_\varphi N_\theta. \tag{2.37}$$

The LHS of Eq. (2.37) represents the average exiting spherical area associated to a trajectory of level ℓ. Therefore, condition 2 of maximal cross-sectional area of a given cone is not satisfied exactly, but verified only for the average cone of level ℓ and belonging to root c. From Eq. (2.37) we can obtain the expression for the splitting radii

$$L_{c,\ell} = \delta \sqrt{\frac{\nu^{\ell-1}}{\Delta\Omega_c}}. \tag{2.38}$$

Eq. (2.38) is solved $\forall \ell$ such that $L_{c,\ell} \leq L_{\max}$, where $L_{\max}$ is an upper bound, fixed to the diagonal of the XYZ geometry. Because the measure of the solid angle $\Delta\Omega_c$ changes with c (by the polar angle index i_θ), the number of splitting levels $N_\ell(c)$ depends on the root index c. For a given root c, at each level ℓ, the $\nu^{\ell-1}$ new-generated trajectories have an equal total length

$$\begin{aligned} L_{c,\ell,t} &= L_{c,\ell} - L_{c,\ell-1} = \Delta L_{c,\ell}, \\ \text{for } t &= 1, \ldots, \nu^{\ell-1} \text{ and } \ell = 1, \ldots, N_\ell(c) \text{ with } L_{c,0} = 0. \end{aligned}$$

The ray-tracing is thus indexed on a non uniform polar grid, where each ray-tracing spherical region (c, ℓ), defined by

$$\begin{aligned} \Delta L_{c,\ell} \times \Delta\Omega_c &= \left[L_{c,\ell-1}, L_{c,\ell}\right] \times \left[\frac{2(i_\varphi - 1)\pi}{N_\varphi}, \frac{2 i_\varphi \pi}{N_\varphi}\right] \times \left[\cos\left(\frac{i_\theta \pi}{N_\theta}\right), \cos\left(\frac{(i_\theta - 1)}{N_\theta}\pi\right)\right] \\ \text{for } \ell &= 0, 1, \ldots, N_\ell(c), \text{ for } i_\varphi = 1, \ldots, N_\varphi,\ i_\theta = 1, \ldots, N_\theta, \text{ and } c = 1, \ldots, N_\varphi N_\theta, \end{aligned}$$

contains $t = 1, \ldots, \nu^{\ell-1}$ trajectories. At each ℓ generation, the ν trajectories, coming from a parent $m(t)$ of level $\ell - 1$, are uniformly distributed inside the surface area associated to the parent. Because of the uniform distribution, the trajectories directions, namely $\mathbf{\Omega}_{c,\ell,t}$, are defined by the mid points of the regions of a uniform spherical-shaped grid. The grid is made of $\nu_\varphi \times \nu_\theta$ meshes on solid angle, $\mathcal{S}^2_{c,\ell-1,m(t)}$, with integers ν_φ and ν_θ representing the number of trajectories along φ and θ, respectively, and such that $\nu = \nu_\varphi \nu_\theta$. The inner and outer hang-points of trajectories, i.e. $\mathbf{\Omega}_{c,\ell,t} L_{c,\ell-1}$ and $\mathbf{\Omega}_{c,\ell,t} L_{c,\ell}$, are defined by the projection of the mid points of the regions of the local $\nu_\varphi \times \nu_\theta$ grid over the spherical surface $\Delta\Omega_{c,\ell-1,m(t)} L^2_{c,\ell-1}$.

The computation of the chord length is performed by superposing the Cartesian calculation mesh to such 3D ray-tracing spherical grid having $\nu^{\ell-1}$ trajectory per region (c, ℓ).

Algorithm 2: Algorithm of the TS-MOC

```
function TS-MOC solver(set of trajectories, ℓ, c)
    foreach trajectories in the set do
        if trajectory starting point is not in geometry then
            continue;
        end
        Evaluate Γ, Eq. (2.28,2.29);
        ψ⁻ = ψ⁺_{ℓ−1};
        Track the trajectory until the nearest splitting point or domain boundary;
        Evaluate the flux in each intersected mesh region, Eq.(2.21);
        if ℓ < N_l(c) then
            Update ψ⁺_ℓ;
            Split the trajectory and generate a set of child trajectories;
            call TS-MOC solver (set of child trajectories, ℓ + 1, c);
        end
    end
end
```

2.4.2 Definition of the TS-MOC algorithm

As for standard MOC, the TS-MOC requires to sequentially solve the chords belonging to each trajectory because of the continuity condition given by Eq. (2.36). The difference with respect to MOC is that, at splitting radii, a set of ν child trajectories is generated from the same parent. The children depend on the parent trajectory because of Eq. (2.36), which provides the incoming flux $\psi^-_{c,\ell,t}$ required to start the sweep. By repeating this for many levels ℓ, we notice that a large number of fluxes $\psi^+_{c,\ell,t}$, outgoing from parent trajectories have to be stored along with the TS-MOC execution. Consequently, a naive implementation may lead to a memory footprint of order $\mathcal{O}(\nu^{N_\ell})$. To avoid such overhead, we implemented the sweeping strategy displayed in Algorithm 2. The latter consists of a depth first search algorithm, ordering the sub-trajectories in a tree (see Figure 2.5). In this configuration, one can notice that the storage can be reduced to $\mathcal{O}(N_\ell)$. In fact, as the tree is explored in depth, the parent trajectory updates the value of the flux that will be used right after by one of its children. Once the deepest level is reached, the sweep starts back from level $N_{\ell-1}$ going towards the deepest level and so on for the ν children of the same parent. The incoming flux for the ν children is the same given by the shared parent and, once all children are swept, it is no longer needed. The process continues in the same manner for all levels. Therefore, the entering flux can be stored in a single buffer of the size of the largest $N_l(c)$.

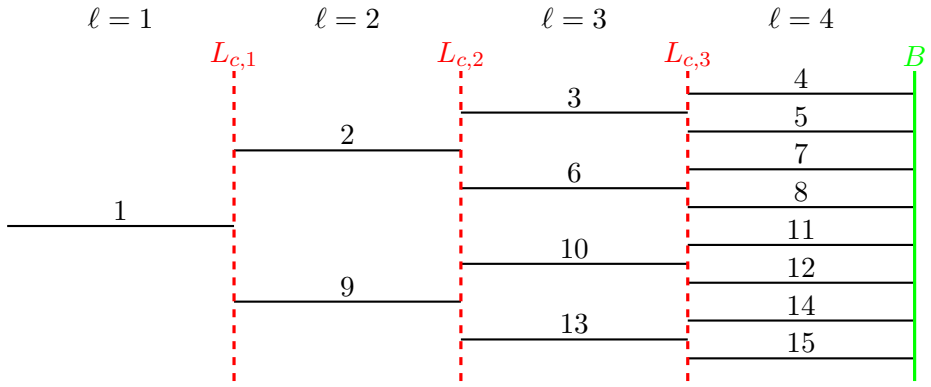


Fig. 2.5 Illustration of the trajectory segments ordering scheme in TS-MOC with ν, number of child trajectories generated per splitting event, equal to 2.

Weight integration technique for TS-MOC

As compared to the MOC, the TS-MOC algorithm is able to reduce the number of trajectories/regions intersections required to achieve the same accuracy. However, because of the splitting of solid angles, the number of Γ weights that need to be evaluated in Eq. (2.21) increases considerably as compared to the MOC. This is because the solid angles $\mathcal{S}^2_{c,\ell,t}$ that define the support of integrals in Eqs. (2.23) and (2.24) change for each child trajectory. Therefore, computing Eqs. (2.23) and (2.24) needs particular care in the TS-MOC implementation.

As already mentioned, the present implementation of TS-MOC is based on the UQ quadrature formula (see Section 2.3.1), for which Γ weights are computed by a product quadrature formula on $\Delta\mathcal{S}^2_{c,\ell,t}$ using uniform steps in the azimuthal angle and a Gauss-Legendre quadrature for the cosine of the polar angle.

Since TS-MOC involves a large variability in the sizes of solid angles, the required order of the quadrature varies depending on the sub-trajectory. To guarantee a given target integration accuracy, we tabulated the order K of the quadrature (see Eq. (2.35)) as a function of the accuracy of the numerical integration of Γ, the dimension of the solid angle, $\Delta\Omega_{c,\ell,t}$, and the maximum order of the spherical harmonics, H. The procedure we used for the evaluation of the quadrature orders, as well as a short version of these tables, are presented in Appendix B.

2.5 Design of the First Collision Source MOC method with a Monte Carlo angular integration

As previously seen, the MOC method, and so also TS-MOC, rely on a quadrature rule for solving the angular integral in Eq. (2.15). This approach inevitably introduces some numerical approximations which are necessary to factorize the angular and the line integration. In addition, the use of an angular quadrature entails the evaluation of coefficients Γ, whose calculation may sometimes be relevant in terms of computational cost. Such drawbacks

are not present if, instead of an angular quadrature, a Monte Carlo integration is used. The approach used by Monte Carlo MOC (MC-MOC) consists in randomly sampling the propagation directions $\mathbf{\Omega}$ and using a Monte-Carlo integration to solve the angular integral in Eq. (2.15). As for the standard MOC, the integration of the point-kernel along direction $\mathbf{\Omega}$, is solved analytically thanks to a trajectory sweep, i.e. successively solving for all intersections along the same trajectory. A suitable formulation for the MC-MOC method can be obtained by rewriting the expression for the uncollided flux moments, namely Eq. (2.10), using the usual transformation into spherical coordinates,

$$\phi_{m,h}^{u,\mathcal{T},*} = Q \int_{4\pi} d\mathbf{\Omega} \int_{L_{\mathcal{T}}^{-}(\mathbf{\Omega})}^{L_{\mathcal{T}}^{+}(\mathbf{\Omega})} dr\ P_m(r\mathbf{\Omega})\ A_h(\mathbf{\Omega}) e^{-\tau(r\mathbf{\Omega})}, \tag{2.39}$$

and then rewriting Eq. (2.39) in form of Monte Carlo integration,

$$E^{\mathcal{T}}[\xi] = \int_{4\pi} d\mathbf{\Omega}\ p(\mathbf{\Omega})\ \xi(\mathbf{\Omega}). \tag{2.40}$$

The moment $\phi_{m,h}^{u,\mathcal{T},*}$ is now expressed by the expected value $E[\xi]$ of the estimator ξ. The probability density function $p(\mathbf{\Omega})$ is the isotropic distribution

$$p(\mathbf{\Omega}) = \frac{1}{4\pi}. \tag{2.41}$$

while the estimator ξ is defined as

$$\xi(\mathbf{\Omega}) = 4\pi Q A_h(\mathbf{\Omega}) \int_{L_{\mathcal{T}}^{-}(\mathbf{\Omega})}^{L_{\mathcal{T}}^{+}(\mathbf{\Omega})} dr\ P_m(r\mathbf{\Omega})\ e^{-\tau(r\mathbf{\Omega})}. \tag{2.42}$$

According to the theory of Monte Carlo integration, the expected value E is estimated by the sample average

$$\phi_{m,h}^{u,\mathcal{T},*} \approx \frac{1}{N_s} \sum_{t \cap \mathcal{T}} \xi(\mathbf{\Omega}_t) \tag{2.43}$$

where $\mathbf{\Omega}_t$ is the sampled direction, and N_s is the sample size. In our case, N_s coincides with the number of sampled trajectories. For each sampled direction, MC-MOC operates a trajectory sweep until the boundary surface and update the estimation of the uncollided flux moments on all intersected target regions. In order to ensure the consistency of the method, the mass matrix is evaluated accordingly. Further details on the Monte Carlo integration can be found in references [77, 78].

The newly presented method is "exact" in a statistical sense. In fact, the variation with respect to $\mathbf{\Omega}$ of the chord lengths and, thus, of the optical thickness of the region, are taken into account without introducing any approximation. On the other hand, MC-MOC, like in general Monte Carlo integrations, could suffer slow convergence. Hence, large numbers

of trajectories, N_s, may be required in order to achieve the prescribed accuracy all over the domain.

In order to increase the Monte Carlo performances, variance reduction methods are generally employed. These methods consist in modifying the probability distribution p in order to increase the number of sampled events (trajectories), scoring on a certain detector tally (spatial region where the uncollided flux is estimated). In MC-MOC, a variance reduction method may increase the number of trajectories directed towards regions having little statistics, so characterized by a poor uncollided flux estimation.

Design of a suitable variance reduction technique for MC-MOC

The variance reduction technique we propose is based on a solid-angle discretization of the unit sphere. This is obtained by subdividing the angular domain $[0, 2\pi] \times [-1, 1]$ in a $N_\varphi \times N_\mu$ uniform grid. The direction sampling is then split in two successive random samplings: with the first, the method samples the solid angle, while with the second, the direction in it. The probability of sampling a solid angle $\mathcal{S}_c^2$, with $c = 1, \ldots, N_\varphi N_\mu$ is then

$$p[\mathcal{S}_c^2] = \frac{\Delta\Omega_c}{4\pi} = \frac{1}{N_\varphi N_\mu}, \tag{2.44}$$

where $\Delta\Omega_c = \frac{4\pi}{N_\varphi N_\mu}$ is the amplitude of solid angle $\mathcal{S}_c^2$. The conditional probability of sampling a trajectory in direction $\mathbf{\Omega}$ within $\mathcal{S}_c^2$ is expressed by

$$p[\mathbf{\Omega}|\mathcal{S}_c^2] = \frac{1}{\Delta\Omega_c}. \tag{2.45}$$

In our biasing scheme the sampling of the direction remains analog (unbiased), while the probability of sampling a certain solid angle has been modified by defining

$$\widetilde{p}[\mathcal{S}_c^2] = p[\mathcal{S}_c^2] W_c, \tag{2.46}$$

the biased probability density function and indicating with W_c the solid angle weight coefficient.

In the strategy we propose, the W_c are associated to an extra score of the MC calculation, which is the outgoing flux from the domain boundaries associated to each solid angle. More precisely, the outgoing flux from the solid angle, $\phi^+_{\partial V,c}$, defined by

$$\phi^+_{\partial V,c} = Q \int_{\mathcal{S}_c^2} d\mathbf{\Omega}\; e^{-\tau(L_{\partial V}(\mathbf{\Omega}))}, \tag{2.47}$$

is estimated by computing the sample average

$$\phi^{+}_{\partial V,c} \approx \frac{4\pi Q}{N_s} \sum_{t \subset \omega_c} e^{-\tau(L_{\partial V}(\mathbf{\Omega}_t))}, \tag{2.48}$$

where ∂V stays for the domain boundary, and N_s is the sample size, corresponding to the number of sampled trajectories. $\tau(L_{\partial V}(\mathbf{\Omega}_t))$ is the total optical path of the trajectory t, with $L_{\partial V}(\mathbf{\Omega}_t)$ the total chord length of the trajectory across the domain. By using the relative standard deviation σ_c of the MC estimation, Eq. (2.48), calculated with

$$\sigma_c = \frac{1}{(N_s)^{\frac{1}{2}} \left|\phi^{+}_{\partial V,c}\right|} \left[\left(\frac{1}{N_s} \sum_{t \subset \omega_c} (4\pi Q)^2 e^{-2\tau(L_{\partial V}(\mathbf{\Omega}_t))} \right) - \left(\phi^{+}_{\partial V,c}\right)^2 \right]^{\frac{1}{2}}, \tag{2.49}$$

we define the weights W_c as

$$W_c = \frac{\sigma_c}{\sum_{c=1}^{N_\varphi N_\mu} \sigma_c}. \tag{2.50}$$

Finally, in order to preserve the unbiased expected value, the uncollided flux estimator (see Eq. (2.42)) is modified as $\widetilde{\xi}(\mathbf{\Omega}) = \xi(\mathbf{\Omega})/W_c$. Since the same trajectories are used to integrate both the uncollided flux over the domain regions and the per-solid angle outgoing fluxes $\phi^{+}_{\partial V,c}$, the standard deviation σ_c will be "in some way" proportional to the error of the uncollided flux estimation in regions illuminated by the same solid angle. Therefore, by defining the weights W_c as in Eq. (2.50), the biasing method will favour the sampling of trajectories directed towards regions characterized by a poor statistics. Even though MC-MOC is not able to address the $1/s^2$ effect (causing the trajectories spreading) tackled by TS-MOC, we expect the variance reduction method to effectively reduce the sample size required to achieve a certain accuracy of the uncollided flux estimation all over the domain.

As concerns the implementation of this method, trajectory simulations are arranged into batches. The user is able to set the size and the number of batches to be simulated. The biasing scheme is said to be adaptive since the biasing coefficient W_c are updated at the end of each batch. This means that the biasing corrections change along the runtime. A potentially smarter implementation would require to associate the solid angle weighting coefficient to the relative standard deviation of the flux moments in each region illuminated by the solid angle itself. This solution could be probably more effective but it would involve further complexities related to the association between target regions and solid angles.

In conclusion, the trajectory calculation with MC-MOC is slightly lighter than that with MOC, since it is not necessary to evaluate the solid angle associated to the trajectory. The MC-MOC does not require the computation of angular weights Γ. It instead makes use of the solid angle intervals just as a support for the biasing scheme.

2.6 Design of the First Collision Source method based on a spatial quadrature

The use of spatial quadrature rules is a common way to perform the numerical integration of Eq. (2.10) in XYZ geometries. This approach has been implemented in several solvers, such as DENOVO, [9], and Rattlesnake, [73]. For the sake of completeness, we have decided to consider also this integration method in our study, and to explore its application in the design of an adaptive integration technique.

The Quadrature Point method (QP) performs the integration of the uncollided flux on each spatial region independently. The integral in Eq. (2.10) belongs to the family of numerical integrations over hyperrectangular n-dimensional regions, therefore it can be numerically approximated by simply applying a product-type quadrature formula, [77]. The uncollided flux moments $\phi_{m,h}^{u,\mathcal{T},*}$ are computed by

$$\phi_{m,h}^{u,\mathcal{T},*} = Q \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{N} w_i w_j w_k \; P_m(\mathbf{r}_{i,j,k}) A_h \left(\hat{\mathbf{e}}_{r_{i,j,k}} \right) \mathcal{K}(\mathbf{r}_{i,j,k}), \tag{2.51}$$

with $\hat{\mathbf{e}}_{r_{i,j,k}} = \frac{\mathbf{r}_{i,j,k}}{r_{i,j,k}}$, w_i, w_j and w_k as the quadrature weights along the x, y and z axis, respectively, and $\mathbf{r}_{i,j,k}$ as the quadrature point defined by $\mathbf{r}_{i,j,k} = (x_i, y_j, z_k)$.

The quadrature rule is constructed by fixing the order of the Gauss-Legendre quadrature to N on the x, the y and the z steps of the target, with the target intervals defined in Eq. (1.56). By relying on Cartesian coordinates, the QP method is not able to eliminate the spatial singularity of the the point-kernel. In order to avoid the evaluation of the uncollided flux on its singular point, the QP method decomposes the spatial region in correspondence of this point. In the target region containing the source, the uncollided flux is therefore computed using a compound quadrature, i.e. summing the contributions evaluated on each sub-region thus generated. The contribution of each sub-region is computed by applying the same quadrature rule. By relying on Gauss-Legendre quadratures, the product quadrature formula guarantees a bounding error of order $O(xyz)^{(2N-1)}$ if the function to be integrated is sufficiently smooth. But, because of the singularity of the point-kernel, Eq. (1.72), the estimation of the error bound is no longer valid for regions close to the source, and for regions that see steep variations of the flux. As concerns the evaluation of the mass matrix, the coefficients in Eq. (2.12) are exactly integrated up to the linear order for every product quadrature with $N \geq 2$.

In order to control the integration accuracy of the uncollided flux, the QP has been implemented in two ways:

- in the *single quadrature mode* (QP-SQ), the quadrature order N is a user-defined parameter;

- in the *automated integration mode* (QP-AI), the accuracy is fixed by the user by setting a parameter, τ, which acts as an error upper bound of the numerical integration.

2.6.1 The single quadrature mode

The QP-SQ method requires the number of quadrature points N to be fixed in each target step. This fixes the order of the integration error residual. No error estimation is performed, the algorithm evaluates the integrand functions on each quadrature point and returns the approximation of the integral. In general, the method is more accurate where the integrand function is sufficiently smooth. The accuracy of the approximation is expected to increase with the order of the quadrature, [77].

2.6.2 The automated integration mode

The QP-AI adaptively refines the quadrature rule in order to provide an estimation of the uncollided flux with an accuracy of the order of the parameter τ (provided that the integrand function is sufficiently smooth). QP-AI repeatedly computes the integral of the uncollided flux by progressively increasing the quality of the numerical approximation, until a fixed accuracy is achieved.

Brief presentation of automatic integration methods

In order to design an automated integration method we have to set two main aspects, i.e. the strategy for the refinement of the quadrature rule, and the error estimation strategy. In general, two types of quadrature refinements are possible: it can be done by either increasing the order of the quadrature (p-refinement), or by splitting the target region in sub-regions (h-refinement) and solving each sub-region using the same quadrature. The QP-AI method uses the second strategy because it is better at representing the discontinuities of the integrand function, [77]. More specifically, QP-AI relies on the algorithm presented by Gander and Gautschi in [79] ; this technique is based on successive region subdivisions. The flux moments $\phi_{m,h}^{u,\mathcal{T},*}$ are evaluated by summing the contributions of different sub-regions, similarly to a compound quadrature rule, [77].

By assuming this refinement strategy, the automated quadrature algorithms can be further distinguished in two types: the ones based on a local estimation of the error, and the ones which rely on a global one, [77, 80]. Most of the algorithms, such as in QUADPACK, [81], and in ADAPT, [82-84], rely on the second solution. The global monitoring of the error, i.e. the error committed on the entire integral to be estimated, allows for more reliable error estimations. The global error on the target region is bounded by the sum of the errors in the different sub-regions which contribute to the integral. This allows to progressively refine the compound rule by monitoring the improvement in terms of accuracy on the error of the whole integral. On the other hand, this approach requires to store the errors and the

approximations of the integrals of all sub-regions. Hence, in 3D geometries this solution may have a large memory footprint.

Methods based on a local monitoring of the error lend well themselves for a recursive implementation. They relies on the independent monitoring of the error on each sub-region, i.e. the further refinement (or not) of the local quadrature for a given sub-region, depends uniquely on the error of the integral restricted to the sub-region. By doing this, the method acts and refines each region independently, without having to store the results of each sub-region. In general, the method requires the error of the estimation of the integral on each sub-region to be smaller then a certain tolerance. This condition clearly does not ensure that the error of the global estimation will have the same accuracy. After considering the strengths and weaknesses of both approaches, we selected a local error estimation for QP-AI, as presented in reference, [79].

A last aspect to cover is the definition of the error estimator in each sub-region. The error, $\epsilon = |\widetilde{I} - I|$, is obtained by comparing the result of the numerical integration, $\widetilde{I}$, with the exact value of the integral, I. This last term, being unknown, is generally replaced by a higher-order numerical approximation of I. A common approach is the use of Gauss-Kronrod quadrature formulae as high-order approximation of the underlying Gauss quadrature, [80, 85]. To this approach, however, we preferred computing the higher-order approximation of the integral using a compound quadrature, obtained by dividing the region in equal parts. In principle, this allows for a more efficient implementation and a smaller number of punctual evaluations of the integrand function.

The QP-AI algorithm

For the sake of simplicity, we will now present the QP-AI method for the integration of a generic function $f(\mathbf{r})$ defined on the target region $\mathcal{T}$. Let $I_{\mathcal{T}}$ be the integral,

$$I_{\mathcal{T}} = \int_{\mathcal{T}} d\mathbf{r}\ f(\mathbf{r}),$$

and $G_N^{(1)}[f;\mathcal{T}]$ the approximation of $I_{\mathcal{T}}$ obtained by applying a N-order product Gauss-Legendre quadrature rule,

$$G_N^{(1)}[f;\mathcal{T}] = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{k=1}^{N} w_i w_j w_k\ f(\mathbf{r}_{i,j,k}). \tag{2.52}$$

The true absolute error, defined by $\epsilon_N = \left|G_N[f;\mathcal{T}] - \mathcal{I}_{\mathcal{T}}\right|$, is approximated by ϵ_N^* as

$$\epsilon_N \sim \epsilon_N^* = \left|G_N[f;\mathcal{T}] - G_N^{(8)}[f;\mathcal{T}]\right|. \tag{2.53}$$

where $G_N^{(8)}[f;\mathcal{T}]$ is a better-quality approximation with respect to $G_N^{(1)}[f;\mathcal{T}]$. $G_N^{(8)}[f;\mathcal{T}]$ is obtained by splitting the target region $\mathcal{T}$ into 8 equally-shaped sub-regions $\{\mathcal{T}_a\}_{a=1,8}$ and summing their contributions obtained by applying the $N-$order Gauss quadrature on each sub-region, i.e.

$$G_N^{(8)}[f;\mathcal{T}] = \sum_{a=1}^{8} G_N^{(1)}[f;\mathcal{T}_a]. \tag{2.54}$$

The sub-regions subdivision is obtained by simply splitting in 2 each side-length of the target. The approximated ϵ_N^*, together with the user-required accuracy τ, defines the stopping criteria of the algorithm.

If this criteria is satisfied, the algorithm returns the estimation $G_N^{(8)}[f;\mathcal{T}]$. If it is not, then the local error $\epsilon_{N,a}^*$,

$$\epsilon_{N,a}^* = \left| G_N^{(1)}[f;\mathcal{T}_a] - G_N^{(8)}[f;\mathcal{T}_a] \right| \quad a = 1,\dots,8 \tag{2.55}$$

is computed and tested on each sub-region $\mathcal{T}_a$. The procedure is recursively repeated until the local errors,

$$\begin{aligned} \epsilon_{N,(a,b,c\dots)}^* &= \left| G_N^{(1)}[f;\mathcal{T}_{a,b,c\dots}] - G_N^{(8)}[f;\mathcal{T}_{a,b,c\dots}] \right|, \\ a,b,c\dots &= 1,\dots,8 \end{aligned} \tag{2.56}$$

satisfy the stopping criterion. Here, the indexes $a,b,c\dots$ stand for successive splittings of the region as: b in a, c in b, and so on.

The stopping criteria proposed by Gander and Gautschi, [79], consists in checking

$$\texttt{if } (is^* + \epsilon_{N,(a,b,c\dots)}^* == is^*) \texttt{ return } G_N^{(8)}[f;\mathcal{T}_{a,b,c\dots}] \tag{2.57}$$

where "==" is the logical equal, and is^* is a broad estimation of $I_{\mathcal{T}}$, namely $G_N^{(1)}[f;\mathcal{T}]$, scaled with the ratio of the user's error bound, τ, and the machine precision, eps,

$$is^* = G_N^{(1)}[T] \cdot \frac{\tau}{eps}, \tag{2.58}$$

(in our case $eps \sim 1 \cdot 10^{-16}$). The value of is^* is fixed once and for all at the beginning of the recursive integration. Condition (2.57) requires the local absolute error in each sub-region to be so smaller than is^*, such as to be not aggregated because of the machine precision. For further details, we address the reader to the original publication of Gander and Gautschi, [79].

The automated integration mode is represented in Algorithm 3. The QP-AI is implemented as a recursive function, here called *QP solver*, that takes the lower order approximation as the input while it gives the best estimate of the integral as the output.

Algorithm 3: Algotithm of the iterative QP method

function *QP solver(*$\mathcal{T}$ *region,* $G_N^{(1)}$*,* is^**)*
  Split $\mathcal{T}$ into 8 equal sub-region $\mathcal{T}_a$;
  $G_N^{(8)} = 0$;
  for $a = 1, 8$ **do**
    Evaluate $G_N^{(1)}[\mathcal{T}_a]$ according to Eq. (2.52);
    $G_N^{(8)} = G_N^{(8)} + G_N^{(1)}[\mathcal{T}_a]$;
  end
  $\epsilon_N^* = G_N^{(8)} - G_N^{(1)}$;
  if $(is^* + \epsilon_N^* == is^*)$ **then**
    return $G_N^{(8)}$;
  else
    $G_N^{(8)} = 0$;
    for $a = 1, 8$ **do**
      $G_N^{(8)}[\mathcal{T}_a]$ = QP solver ($\mathcal{T}_a$ sub-region, $G_N^{(1)}[\mathcal{T}_a]$, is^*);
      $G_N^{(8)} = G_N^{(8)} + G_N^{(8)}[\mathcal{T}_a]$;
    end
    return $G_N^{(8)}$;
  end
end

Finally, one last remark. All automatic integration methods rely on the hypothesis of continuity and smoothness of the integrand function. When this hypothesis is not satisfied, the error estimate is likely to fail in representing the actual error, [77, 80, 85, 86]. Therefore, the error upper bound τ of automated integration QP method should be interpreted as a figure of merit rather than an effective measure of the precision of the numerical approximation.

2.6.3 Considerations over the QP First Collision Source method

The QP method requires to evaluate the flux attenuation along each trajectory, corresponding to the quadrature points in each region. Its evaluation, as compared to the MOC, does not involve any solid angle and any angular integration, but only the calculation of the exponential and geometrical attenuation of the flux. The mean number of floating-point operations per chord is therefore significantly smaller than the one required by MOC. On the other hand, the cost of the algorithm scales with the number of quadrature points used per region, and inevitably with the number of target regions.

Some preliminary tests have shown that the single quadrature mode is sufficient to provide accurate estimations in regions where the flux is more regular. The spatial regions with the highest approximation error are those containing the source itself and those presenting a highly anisotropic incoming flux, like for half-shadowed regions.

Regarding the automatic integration mode, a highly anisotropic flux results in slow convergence in certain regions, incorrect error estimations and large numbers of quadrature points. The automatic integration mode has to achieve a good compromise between numerical accuracy and computational time. In order to avoid an excessive computational cost, the user is allowed to limit the number of successive subdivisions of the target region.

2.7 Comparative analysis of the First Collision Source techniques

In the previous pages, we have designed four different First Collision Source methods for the calculation of point-like sources, and so also localised sources which can be approximated by a set of point sources. FCS methods are capable of providing estimations of the uncollided flux moments, represented according to the spatial-angular basis. However, because of their features, their effectiveness may vary depending of the problem size and material composition. In this section, FCS methods are tested and verified in terms of accuracy and computational performances. We present a first set of benchmarks based on simplified problems consisting in a uniform box with one or more absorber stumbling blocks. The test suite has been called Box Test. With these tests, we aim to asses:

- the error predictability with parameter τ in the automated integration QP method (QP-AI),
- the impact of the numerical integration of the trajectory weights, i.e. the coefficients Γ, on the overall accuracy of the MOC method and, finally,
- the capability of reducing the uncollided flux estimation relative error by increasing the accuracy of the FCS methods.

As a second and final test, the results on the Kobayashi benchmark problem $3i$, [87], are presented and compared with the reference Monte Carlo results.

2.7.1 Analysis of the First Collision Source methods on the Box Test problem

The Box Test is a set of problems reproducing a simple radiation shielding test case with different dimensions and different degrees of spatial heterogeneity of the medium. The problems are generated by repeating a common pattern along the x, y, and z directions. The basic pattern is made of two nested cubes. The outer cube has a side of 10 cm and the inner one has a side equal to 1/3 of the outer. The spatial mesh consists of a uniform $3 \times 3 \times 3$ regions discretization (of each pattern model). The cubes are made of purely absorbing materials. The outer and the inner cubes have respectively a total cross section of 2×10^{-3} cm^{-1} and 2 cm^{-1}. Therefore, a ratio of 1000 is set between the two cross sections.

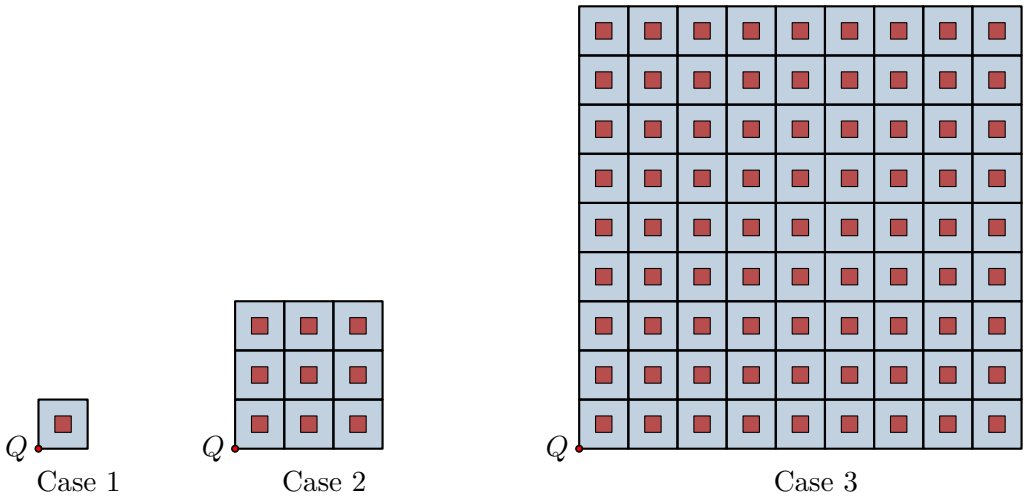


Fig. 2.6 Front view of the Box Test problem, Cases 1, 2, and 3. The regions having a total cross section $\Sigma = 2 \times 10^{-3}$ cm^{-1} are coloured in blue while in red are represented the absorbing cubes having $\Sigma = 2$ cm^{-1}.

By repeating this pattern up to 27 times along each Cartesian axis, we designed 4 different models, described in Table 2.1. The number of blocks (and so also the number of absorbing regions) goes from 1, for Case 1, to 19683, for Case 4.

Case	Patterns per axis	Domain (cm)	Spatial regions
1	1	$[0,\ 10] \times [0,\ 10] \times [0,\ 10]$	27
2	3	$[0,\ 30] \times [0,\ 30] \times [0,\ 30]$	729
3	9	$[0,\ 90] \times [0,\ 90] \times [0,\ 90]$	19683
4	27	$[0, 270] \times [0, 270] \times [0, 270]$	531441

Table 2.1 Box Test model data.

In each problem, the source is located in the south-west-bottom corner. Vacuum boundary conditions are everywhere applied. Figure 2.6 provides a 2-D representation of the first three Box Test cases.

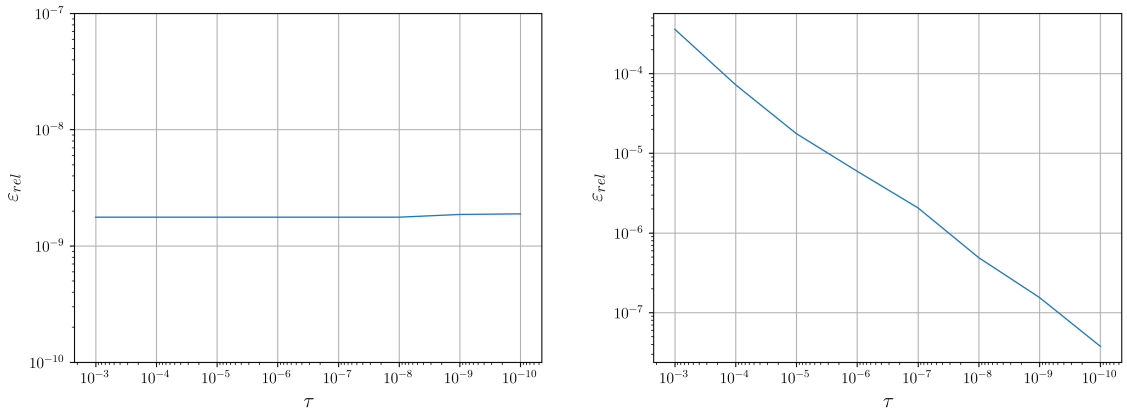
Assessment of error predictability with parameter τ in the automated integration QP method

As first test, we want to verify the capability of the automated-integration QP method (QP-AI) to reduce the uncollided flux estimation error as the τ parameter decreases. Therefore, we are interested in assessing to what extent τ is representative of the actual relative error.

In this context, Case 1, composed by $3 \times 3 \times 3$ target regions, is studied. By using the Cartesian indexes (i_x, i_y, i_z) to specify a target region, we have selected region $(3, 1, 1)$ and $(3, 3, 3)$ for monitoring the particle flux. The first target sees an almost smooth incoming

flux, while the second one sees sharper variations of the flux because of its position shadowed by the absorber region. The study takes the form of a verification test of the QP-AI with respect to the MOC, with the angular quadrature rule based on the boundary surfaces mesh (MOC-BQ). The MOC method is run by setting a very large number of trajectories (about 6×10^{10}, generated using a boundary mesh having regions size equal to $1 \cdot 10^{-8}$ cm^2), whereas the QP method is run using decreasing values of the figure of merit τ. Given the amount of trajectories used in MOC, we are confident that the error committed with respect to the real solution is smaller than the error measured between the solutions obtained with the two FCS methods. The MOC simulation is therefore used as reference. The relative error between two simulations has been defined as

$$\varepsilon_{rel} = \frac{|\phi_{MOC} - \phi_{QP,\tau}|}{\phi_{MOC}}. \tag{2.59}$$



(a) Region $(3, 1, 1)$ (b) Region $(3, 3, 3)$

Fig. 2.7 Relative error trend of the constant spatial-angular component of the uncollided flux. Test on regions $(3, 1, 1)$ and $(3, 3, 3)$ of the $3 \times 3 \times 3$ Case 1 Box Test. Comparison between a very fine MOC simulation with the automated integration QP method using different τ values.

Figure 2.7 shows the results of the convergence test for the constant spatial-angular component of the flux, i.e. the region average scalar flux. The results obtained with the two methods are in agreement. As for region $(3, 1, 1)$, the flat trend at low relative error shows that the QP-AI method can provide a good estimation even for high values of τ. As expected, this result confirms that the QP method performs well for problems in which the flux shape is sufficiently smooth and well behaved. On the other hand, the relative error in the target $(3, 3, 3)$ converges linearly as τ decreases. Moreover, we notice that the true error, ε_{rel}, is larger than the one predicted by the parameter τ. Those simple results show that

both methods are capable to converge towards the same results within a prescribed accuracy. The same agreement has been obtained also for higher-order spatial-angular moments, that are not showed here for brevity. Parameter τ can be therefore adjusted in order to refine the FCS estimation but, as expected, it is not representative of the actual relative error.

Furthermore, results show that both MOC and QP-AI are able to provide results with a significantly high accuracy. Thus, in the following tests, the MOC-BQ will be used systematically as reference.

Assessment on the effect of the Γ weight integration in MOC-based methods

As seen in previous sections, the MOC and TS-MOC methods require to evaluate, for each trajectory, the trajectory weights Γ. These take the form of integrals of spherical harmonics over portions of solid angles. By their nature, these terms are related to the projection operator of the uncollided flux on the spatial-angular moments. It is therefore important to asses how much the accuracy of the Γ integration affects the accuracy of the result. This is even more important in highly anisotropic problems since higher order spherical harmonics involved in the uncollided flux expansion are in principle more difficult to integrate. We remind that the angular weights are specific to the MOC, while they are not required in the QP and MC-MOC methods.

Case 1 is studied by changing the parameters for computing Γ and by monitoring the errors of the uncollided flux components. In this test, a P_9 anisotropy is used, corresponding to an angular flux expansion on a 100 spherical-harmonics basis. The analysis is performed focusing on target regions $(1, 1, 1)$ and $(3, 3, 3)$. On each of them, we have monitored the constant flux component ($h = 1$, $m = 1$) and the higher order flux component identified by the indexes $h = 99$, $m = 3$. This last component corresponds to the linear moment along the Y axis of the angular moment ($k = 9, l = 8$). The analysis consists in monitoring the variation of the relative error of the flux components with respect to the number of chord-lengths. We have performed this study by using:

- the MOC method with the boundary-based quadrature (MOC-BQ). The accuracy of the method has been progressively increased by running simulations with decreasing values of the boundary surfaces steps, i.e. the trajectory spacing $\Delta_{t,x}$ and $\Delta_{t,y}$. For MOC-BQ, the angular weights are evaluated according to Eqs. (2.28) and (2.29) by subdividing each solid angle in $2^N \times 2^N$ smaller solid angles. Using the symbol Γ-N to specify the order of the quadrature for Γ, we have tested from Γ-0, i.e. the step-approximation, up to Γ-3, i.e. a 64 sub-solid angle decomposition;

- the MOC method with the uniform quadrature (MOC-UQ). For the MOC-UQ, the accuracy has been progressively increased by running simulations with increasing number of trajectories. In this case, the integration of the angular weights is performed

by using the product type quadrature rule defined in Section 2.3.1 and denoting with the symbol Γ-N the quadrature order corresponding to $2^N \times 2^N$ quadrature points;

- the TS-MOC method with ν, i.e. the number of trajectories generated per splitting event, equal to 4. The trajectory refinement is operated by progressively reducing the parameter δ^2. The starting quadrature of the TS-MOC has been fixed to the coarser uniform quadrature requiring $4_\varphi \times 2_\theta$ angular directions at the point-source. We run several simulations using tabulated orders for the Γ's quadrature. These tabulated orders have been obtained computing the Γ weights up to P_9 moments and by requiring a decreasing error tolerance for the numerical integrals, i.e. 10^{-1}, 10^{-2} and 10^{-3}. Furthermore, we have also tested a step approximation inside the solid angle associated to the trajectory, here indicated as Γ-0.

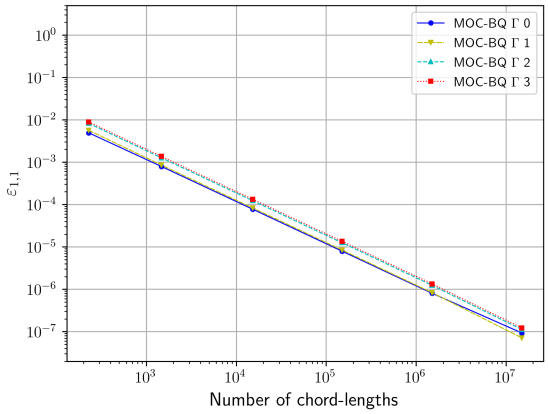
The reference values for the error estimation have been obtained with MOC-BQ, setting the boundary mesh region dimensions $\Delta_{t,x} = \Delta_{t,y} = 0.003$ cm and using the angular quadrature Γ-3.

Figures 2.8, 2.9 and 2.10 show the results obtained with MOC-BQ, MOC-UQ and TS-MOC, respectively.

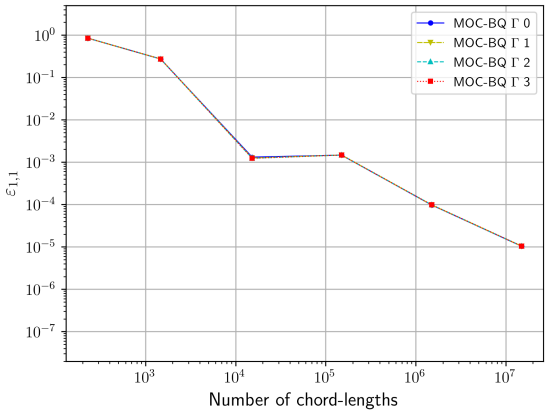
As shown in Figures 2.8-(a)-(b) and 2.9-(a)-(b) relative to MOC-BQ and MOC-UQ, respectively, the accuracy of the scalar flux components is not improved by high-order integration. On the other hand, the accuracy of high-order moments can be enhanced by high-order quadrature, especially in regions close to the source, where the angular integration domain is large, as shown in Figs. 2.8-(c) and 2.9-(c).

On the other hand, the TS-MOC is largely sensible to the variation of the Γ integration order when a coarse solid angle discretization is used at the point-source, as for the cases considered here. This is evident in Figure 2.10 for the higher-order moment evaluated in region $(1, 1, 1)$. Figure 2.10-(c) shows a large improvement of the integration quality when using the tabulated integrations orders rather then Γ-0. This is valid for all values of δ^2. This behaviour is explained by the fact that, before the first splitting radius, the size of the solid angle involved in the angular integration of Γ coincides with the whole quadrant. In these particularly large solid angles, a simple step angular quadrature is not sufficient to correctly integrate higher order harmonics. According to this, the use of tabulated Γ orders is preferable. This allows performing higher-order integrations when the solid angles are large (close to the source) and to progressively coarsen the quadrature rule as the solid angle dimensions shrinks.

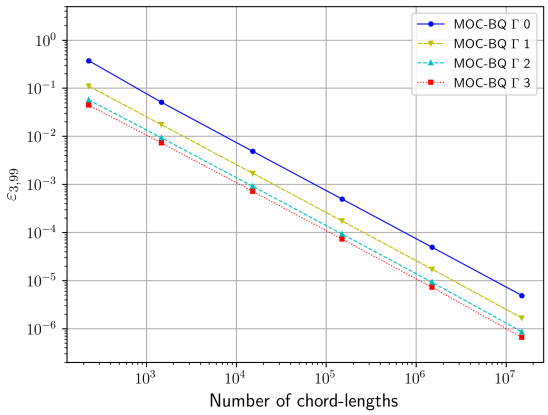
In conclusion, in MOC-BQ and MOC-UQ the sensitivity of the error to the accuracy of Γ is small with respect to the dependence on the MOC discretization refinement (at least for the regions not containing the source point). On the other hand, we observe a larger sensitivity in TS-MOC. Therefore, in the following we will use a step approximation (Γ-0)



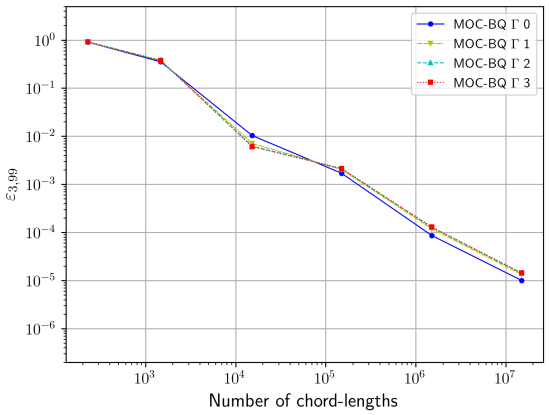
(a) Region $(1,1,1)$, uncollided flux component $h=1, m=1$



(b) Region $(3,3,3)$, uncollided flux component $h=1, m=1$



(c) Region $(1,1,1)$, uncollided flux component $h=99, m=3$



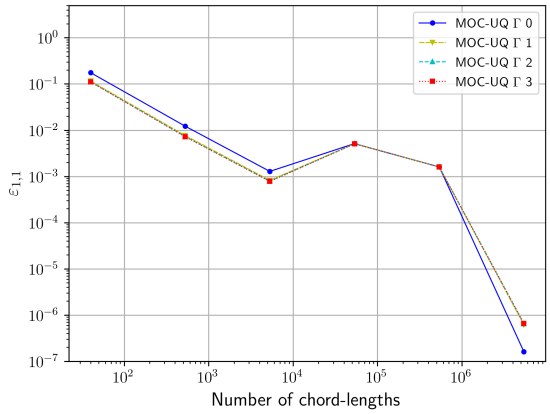
(d) Region $(3,3,3)$, uncollided flux component $h=99, m=3$

Fig. 2.8 Box-Test Case 1. Results of the sensitivity study of the MOC-BQ (Boundary-based quadrature) integration quality with respect to the trajectory weight integration order. The plots shows the trend of the uncollided flux components relative error with respect to the total number of trajectories/spatial regions intersections.

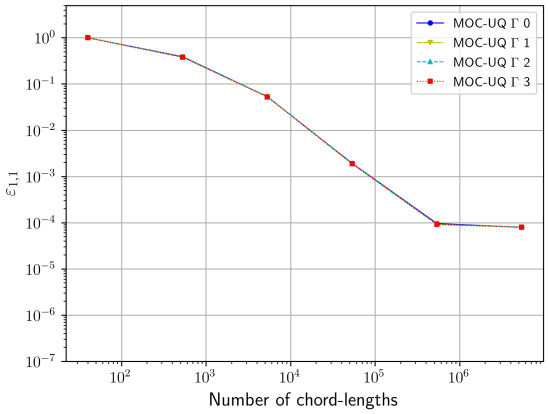
for the weight integration in MOC-BQ and MOC-UQ, and the tabulated integration orders for TS-MOC.

Assessment of the computational performances of FCS methods

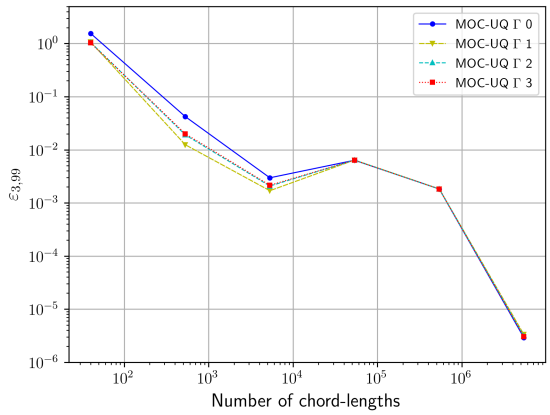
In order to compare the characteristics and accuracy of the FCS methods, the four problems of Table 2.1 have been studied. For each of them, we have solved both the heterogeneous problem, and the corresponding homogeneous one. The homogeneous problem is obtained by setting a total cross section equal to 2×10^{-3} cm^{-1} all over the domain. Each method has been run with increasing accuracy, by incrementing the number of trajectories for the



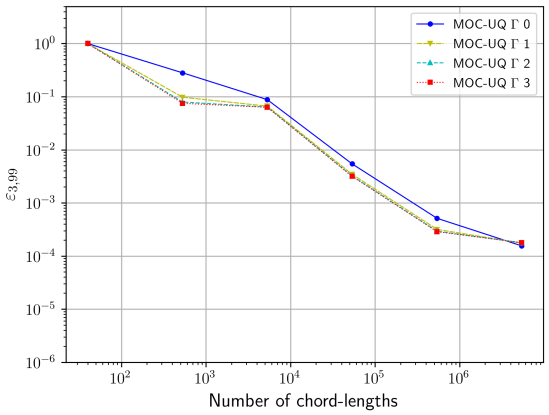
(a) Region $(1,1,1)$, uncollided flux component $h=1, m=1$



(b) Region $(3,3,3)$, uncollided flux component $h=1, m=1$



(c) Region $(1,1,1)$, uncollided flux component $h=99, m=3$

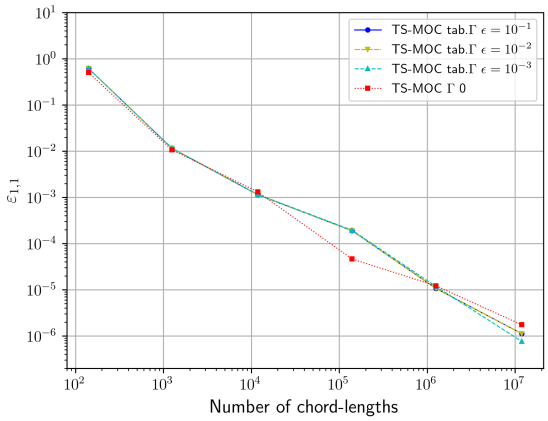


(d) Region $(3,3,3)$, uncollided flux component $h=99, m=3$

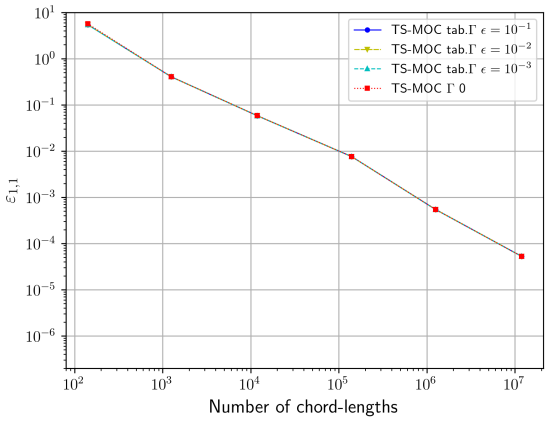
Fig. 2.9 Box-Test Case 1. Results of the sensitivity study of the MOC-UQ (Uniform quadrature) integration quality with respect to the trajectory weight integration order. The plots shows the trend of the relative error of the uncollided flux components with respect to the total number of trajectories/spatial regions intersections.

MOC-based techniques, or the number of quadrature points for the QP method. The angular moments are evaluted up to the P_1 order. The analysis consists in monitoring the relative error of the estimation of the uncollided scalar flux on the whole computational domain. In detail, we have run:

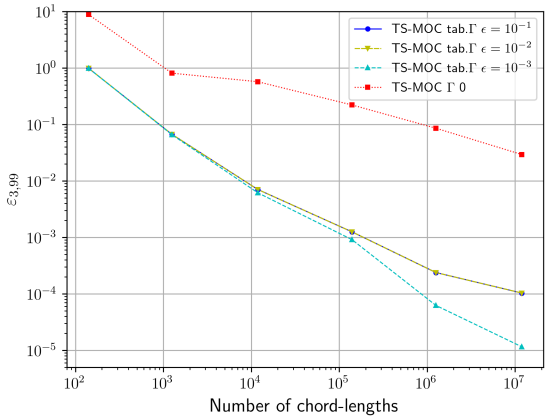
- the MOC-BQ using Γ-0 angular weights. The accuracy is increased by progressively refining the boundary mesh used as support for the solid angle discretization;
- the MOC-UQ using Γ-0. In this case, the number of trajectories is increased in such a way that the number of azimuthal directions doubles the number of the polar directions, i.e. $N_\varphi = 2\ N_\theta$;



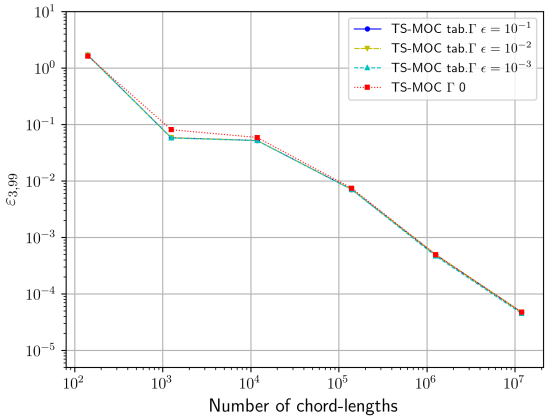
(a) Region $(1,1,1)$, uncollided flux component $h = 1, m = 1$



(b) Region $(3,3,3)$, uncollided flux component $h = 1, m = 1$



(c) Region $(1,1,1)$, uncollided flux component $h = 99, m = 3$



(d) Region $(3,3,3)$, uncollided flux component $h = 99, m = 3$

Fig. 2.10 Box-Test Case 1. Results of the sensitivity study of the TS-MOC integration quality with respect to the trajectory weight integration order. The plots shows the trend of the relative error of uncollided flux components with respect to the total number of trajectories/spatial regions intersections.

- the TS-MOC method with the S_2 initial quadrature $(4_\varphi \times 2_\theta)$ and $\nu = 4$. We used tabulated order of the Γ quadrature obtained by requiring a 10^{-3} error tolerance. The TS-MOC accuracy is increased by reducing the parameter δ^2;
- the MC-MOC method without biasing (MC-MOC-A). The accuracy is increased by increasing the number of sampled trajectories.
- the MC-MOC method with biasing (MC-MOC-B);
- the single quadrature QP method (QP-SQ). The simulations have been run by progressively increasing the order of the Gauss quadrature formula;

- the automated integration QP method (QP-AI). The simulations have been run by progressively decreasing the value of the parameter τ. The order of the Gauss quadrature is set equal to 2 in all cases except Case 4 which uses a 6-order Gauss quadrature.

In order to evaluate the error over the domain, the RMS relative error,

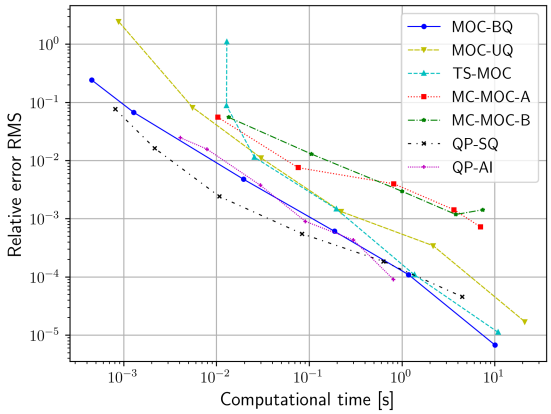
$$RMS = \sqrt{\frac{\sum_{i=1}^{N_r} \left(\frac{\phi^i - \phi^i_{ref}}{\phi^i_{ref}} \right)^2}{N_r}}, \tag{2.60}$$

is monitored for each FCS method over the total number of regions N_r. The reference results are obtained using the MOC-BQ Γ-0 with a trajectory spacing of 0.003 cm for Case 1 and 2, and 0.009 cm for Case 3 and 4. In order to compare FCS methods which are really different by nature, the RMS relative error has been monitored with respect to the computational time. We are aware that the calculation time depends not only on the method but also on the quality of its implementation, nonetheless it turns out to be the only basis of comparison of the various FCS methods. We also remember that the number of trajectories is not an effective term of comparison, since, in our experience, the trajectory tracking is a minor component of the computational cost of FCS in Cartesian geometries. Furthermore, FCS methods carry out for each trajectory different number of operations, thus leading to different computational costs.

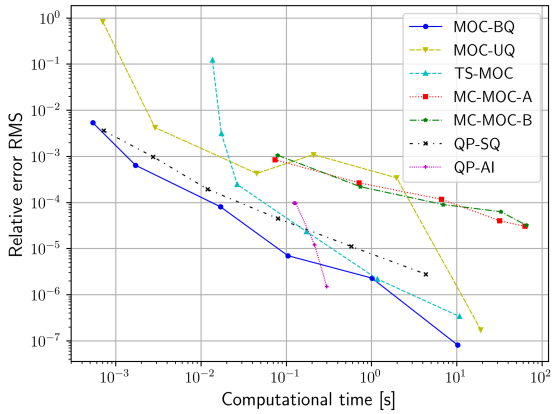
Figures 2.11 and 2.12 show the results of the tests for the 4 cases. Each figure displays the RMS of the uncollided scalar flux distribution in the heterogeneous problem (on the left) and in the associated homogeneous case (on the right), versus the computational time.

Overall, the MOC-BQ, the MOC-UQ and TS-MOC present the best compromise in terms of computational cost and accuracy. In the presented cases, the MOC-BQ is very robust. The boundary quadrature has provided better results than the uniform one, especially for smaller problems. This behaviour is relevant in Cases 1 and 2 but the differences tend to disappear as the dimension of the problem increases. We believe that the superior accuracy of the boundary quadratures is due to the capability to adapt the quadrature to the spatial shape of the domain.

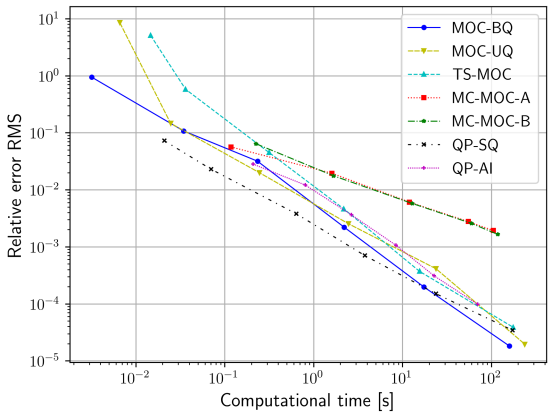
The performances of the TS-MOC are similar to others MOC especially for larger problems, as for Case 3 and 4. As compared to the MOC method, the TS-MOC appears to be less effective for relatively large δ^2 values, i.e. for points of the graph corresponding to smaller computation times, and more effective for smaller δ^2 values. This threshold effect characterizes the simulations of the heterogeneous problem, while it is less pronounced in the associated homogeneous cases. By considering the approximation made on the trajectories splitting levels, and the presence of localized absorbers in the models, it is possible to provide an intuitive explanation for this effect. If the δ^2 parameter (which should be interpreted as the maximum cross sectional area of the trajectory) is too large as compared to the



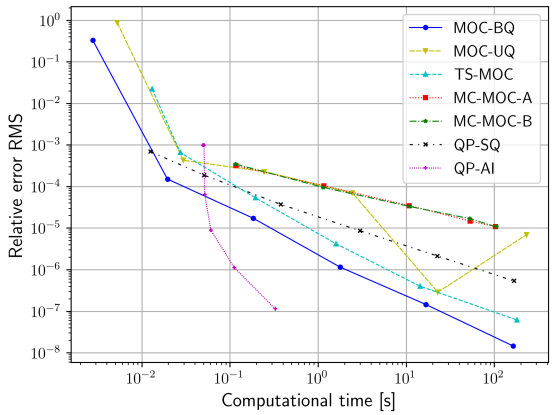
(a) Case 1, uncollided flux constant component estimation



(b) Case 1, associated homogeneous problem, uncollided flux constant component estimation



(c) Case 2, uncollided flux constant component estimation

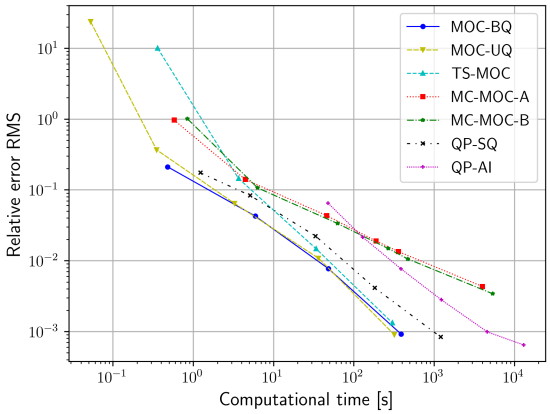


(d) Case 2, associated homogeneous problem, uncollided flux constant component estimation

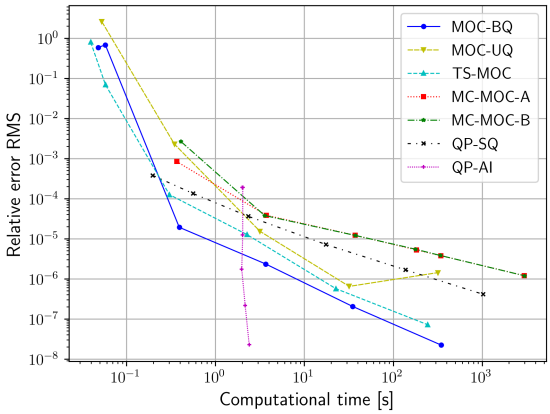
Fig. 2.11 Box Test, Cases 1 and 2. Assessment of the accuracy of integration of the constant component of the uncollided flux. Representation of the RMS of the relative error with respect to the computational time. Heterogeneous problem (left) and homogeneous problem (right)

characteristic size of the absorber, the TS-MOC method is not able to adequately represent the flux anisotropy in the spatial region downstream the absorber, so it produces less accurate results. We notice also that the convergence trend of the TS-MOC presents less oscillations as compared to the MOC-UQ. This behaviour is explained by the fact that the splitting procedure tends to distribute trajectories more evenly in space.

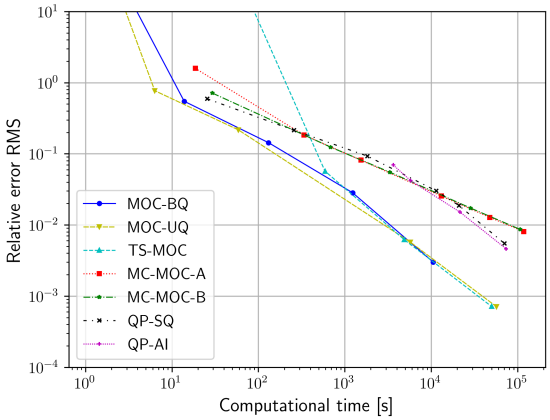
Even though TS-MOC provides a better trajectory distribution all over the domain, the constant flux approximation along the splitting event could have a not-negligible impact on the error. In this sense, the problems proposed here, having strongly heterogeneous patterns, are particularly challenging for the method.



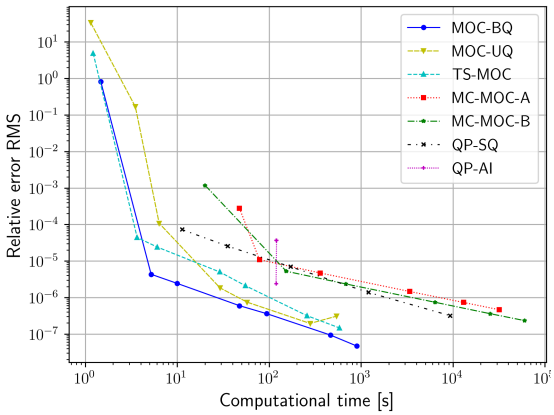
(a) Case 3, uncollided flux constant component estimation



(b) Case 3, associated homogeneous problem, uncollided flux constant component estimation



(c) Case 4, uncollided flux constant component estimation



(d) Case 4, associated homogeneous problem, uncollided flux constant component estimation

Fig. 2.12 Box Test, Cases 3 and 4. Assessment of the accuracy of integration of the constant component of the uncollided flux. Representation of the RMS of the relative error with respect to the computational time. Heterogeneous problem (left) and homogeneous problem (right)

The figures show that the QP method is more effective in problems with small dimensions while it performs worse than MOC in strongly anisotropic and large size problems. This is consistent with the nature of the method: since QP performs a region-by-region integration by tracing trajectories from the source to the integration point, the cost of the algorithm increases more than linearly with the size of the computational mesh. Because of its characteristics, we believe that QP method might be used whenever the user requires an estimate of the uncollided flux on a circumscribed portion of the domain, as a detector.

As concern the MC-MOC, it presents the poorest results with significantly lower performances and a slower error convergence. Furthermore, the current biasing method (MC-MOC-B) appears to be not effective in reducing the computational cost of the simulations.

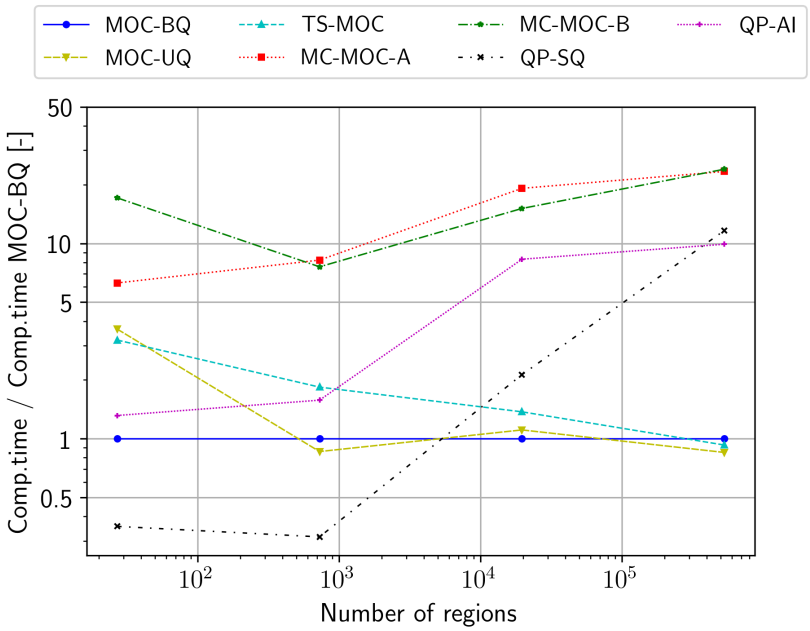


Fig. 2.13 Box Test, comparison of the FCS methods at different problem sizes. Representation of the time necessary for the calculation to achieve a RMS of the relative error equal to 10^{-2}, with respect to the number of spatial regions. The computational time are obtained by mean of interpolations of the results of the simulation of the 4 Box-Test cases. The computational times are expressed as relative ratio with respect to the time spent by the MOC-BQ simulation.

Figure 2.13 proposes an additional comparison between the methods. It represents the time required to achieve an RMS smaller than 10^{-2} for the four problems of Table 2.1. These time values have been obtained by mean of an interpolation of the results displayed in Figures 2.11 and 2.12. The computational time is scaled with respect to the computational time of the MOC-BQ, taken as reference.

We can see that the TS-MOC performs better as the domain size increases since the number of splitted trajectories grows homogeneously in the space. The MOC-UQ method also shows an increase of the performance with the problem size. Finally, the computational cost of the MC-MOC and the QP methods increases with the problem size. This trend is particularly relevant for the QP-SQ method.

2.7.2 Verification of the First Collision Source method on the Kobayashi benchmark problem

The Kobayashi test 3*i*, [87], consists of a Cartesian purely absorbing domain crossed by a void duct. The geometry of the problem is shown in Figure 2.14, while the data related to the media are shown in Table 2.2.

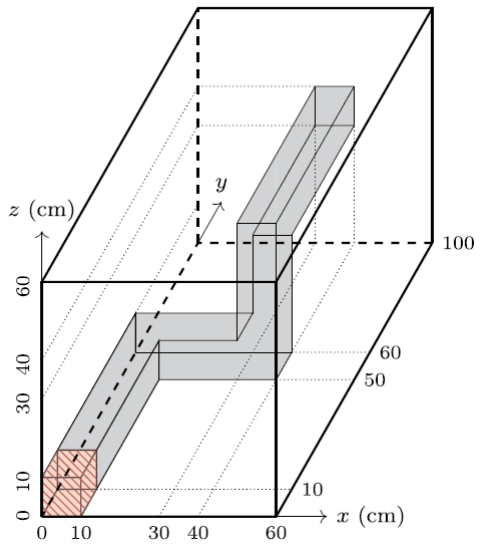


Fig. 2.14 Representation of the Kobayashi benchmark problem 3. Image taken from [54]. The source region is the small red coloured cube in the bottom left corner. The domain is crossed along its length by a duct which is represented by the shaded region.

Region	Σ (cm^{-1})	Source Intensity (cm^{-3}s^{-1})
Source	0.1	1.0
Duct	$1 \cdot 10^{-4}$	0.0
Shield	0.1	0.0

Table 2.2 Kobayashi benchmark problem 3, medium data.

The model has reflective boundary conditions on the surfaces lying on the planes $x = 0$, $y = 0$ and $z = 0$. Vacuum boundary conditions are imposed on the remaining boundaries. An anisotropy order $K = 1$ is used. Since none of the methods is currently able to manage reflections, contributions from reflected particles are included by doubling the size of the source in X, Y and Z so that the entire computational domain is comprised between $[-10, 100] \times [-10, 60] \times [-10, 60]$. The source is thus defined in the volume $[-10, 10] \times [-10, 10] \times [-10, 10]$. It has an intensity of 1 cm^{-3}s^{-1} and it is isotropic. The source volume has been decomposed into $10 \times 10 \times 10$ equally spaced point-sources. The spatial mesh discretization consists of a

volume subdivision into cubical regions with a side length equal to 10 cm. The test consists in the evaluation of the average flux in some specific mesh regions. The reference results are taken from [54]. According to the author, these values are obtained using the Monte Carlo code MCNP with 10^{13} particle histories. The reported statistical error corresponding to 2σ varies in the range 0.01% and 0.04% in the regions of interest.

The Kobayashi test problem has been simulated using:

- the MOC-BQ with Γ-0 weights with a trajectory spacing $\Delta_{t,x} = \Delta_{t,y} = 2$ cm resulting in 1.14×10^8 chord lengths;
- the MOC-UQ with $100_\varphi \times 50_\theta$ directions, providing a total of 8.8×10^7 chord lengths;
- the TS-MOC method with ν equal to 4. We used the tabulated orders for the Γ integration imposing a 10^{-3} error tolerance up to P_1 moments. The splitting radii are defined by setting the reference area δ^2 equal to 4 cm^2 and requiring $20_\varphi \times 10_\theta$ angular directions at the source, resulting in a total of 5.2×10^7 chord lengths;
- the MC-MOC-A with 10^4 trajectories per point source resulting in a total of 4.6×10^7 chord lengths;
- the MC-MOC-B by a biased probability distribution over $16_\varphi \times 8_\theta$ discrete solid angles. We sampled a total of 10 batches of 1000 trajectories generating a number of chords approximatively equal to 4.8×10^7;
- the QP-SQ with Gauss quadrature of order $N = 4$;
- the QP-AI with Gauss quadrature of order $N = 3$ and a τ value equal to 1.

The parameter settings have been chosen in such a way to obtain the results with a comparable computational cost. Each simulation has run for a time that is strictly smaller than 2 minutes on a single thread.

The results of the test are presented in Figure 2.15. The plots represent respectively the region average scalar flux and the ratio between flux and reference on 3 sets of spatial regions. Using Cartesian indexes (i_x, i_y, i_z), those sets are

$$\begin{aligned} A: &\quad (5, i_y, 5), \text{ with } i_y \in [5, 15, 25, 35, 45, 55, 65, 75, 85, 95], \\ B: &\quad (i_x, 55, 5), \text{ with } i_x \in [5, 15, 25, 35, 45, 55], \\ C: &\quad (i_x, 95, 35), \text{ with } i_x \in [5, 15, 25, 35, 45, 55]. \end{aligned}$$

The results show a good agreement between the proposed methods and the reference. All simulations, except the QP method in region (5,5,5), have provided results within 5% of error.

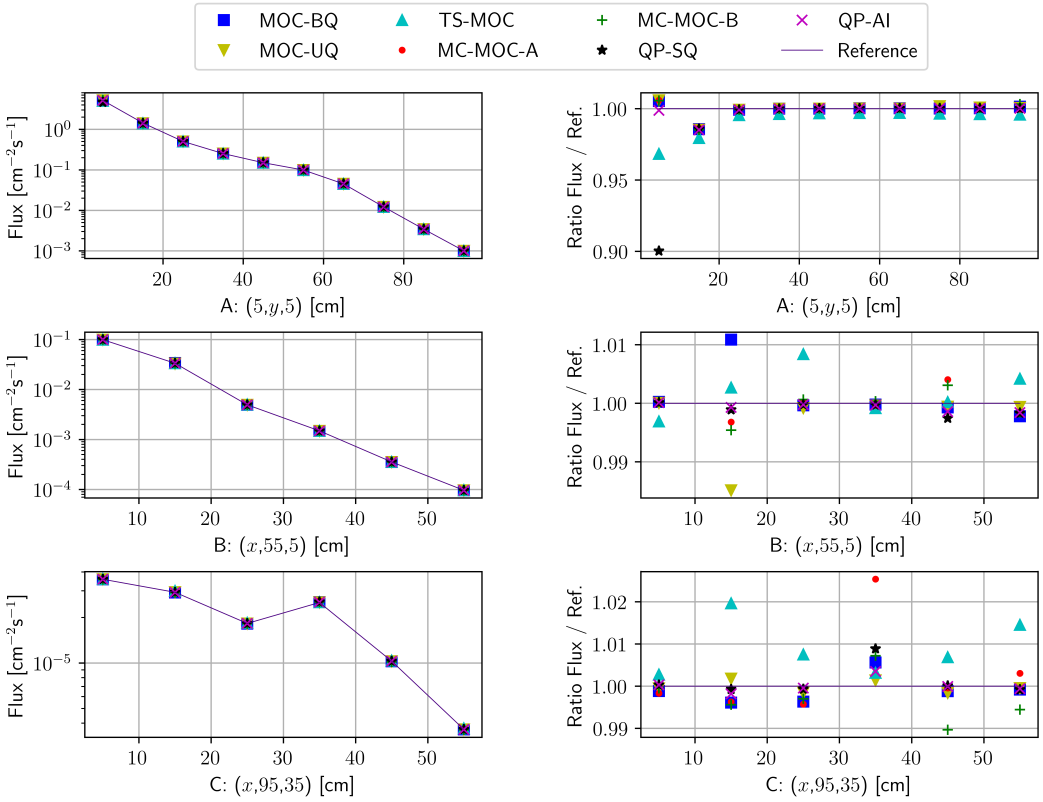


Fig. 2.15 Kobayashi benchmark problem 3i, comparison of the results of the simulations with the reference values. The particle flux constant component value and its ratio with respect to the reference are displayed for sets of spatial regions A, B and C.

2.8 Partial conclusion

First Collision Source methods are known to be indispensable techniques for most shielding calculations dealing with localized sources in solvers based on Discrete Ordinates. Our research has led to the development of four different FCS techniques for point source propagation:

- the MOC, based on the uncollided flux integration by characteristic lines in spherical coordinates. It relies on a fine quadrature rule for the angular discretization. It has been implemented at first in [53], and now it has been extended up to the linear spatial order;
- the TS-MOC, which comes as a variation of MOC. It relies on a trajectory splitting algorithm in order to ensure a more uniform trajectory distribution over the domain, and to reduce the overall computational cost of the MOC algorithm;
- the MC-MOC, which uses a characteristic line integration and a random sampling of the propagation directions;

- the QP method, which preforms a region-by-region integration of the uncollided flux by means of a spatial quadrature. It is able to progressively refine the quadrature formula in order to converge the accuracy of the estimation up to the achievement of an error-related stopping criterion.

Our tests have shown how the QP method with automated integration is capable to refine the estimation quality as parameter τ reduces. We have discussed the sensitivity of MOC-based method with the integration quality of trajectory weights, and then, we have compared the different methods in terms of accuracy and computational cost. We highlighted the robustness and effectiveness of MOC and TS-MOC methods in providing accurate uncollided flux estimations in a limited computing time, and for different problem sizes. TS-MOC appears as a very promising technique able to guarantee an approximatively uniform trajectory-based spatial discretization of the domain. The method has shown an increasing effectiveness as the size of the problem increases. Nevertheless, in the cases we studied, the MOC-BQ method was on average the most effective. QP method showed good performances (in terms of precision, for a given computing time) only for simulations with a limited number of regions, while MC-MOC resulted to be poorly effective. Even if we focused only on isotropic sources, all methods can be easily extended to anisotropic ones. However, particular care is needed in the case of TS-MOC for correctly reconstructing the source along the directions of the child trajectories. Furthermore, we have also shown how the different techniques can be employed for adjoint flux calculations, by simply changing the sign of the propagation direction while evaluating spherical harmonics functions. The extension of the FCS techniques here presented to boundary conditions different from vacuum may be not trivial depending on the method. For example, the introduction of reflective boundary conditions inevitably requires modifications in the trajectory tracking algorithm.

In conclusion, the first collision source method is a very powerful technique, able to mitigate and, sometimes, completely remove the ray-effect produced by the propagation of localized sources in deterministic discrete-ordinates methods. Nevertheless, as observed by some authors, [55], ray-effect may be also produced by n-th collided sources, and so, the FCS method may not be sufficient. Whenever a generic scattering source can be assimilated to a localized source emitting in a poorly diffusive and optically thin medium, a fine angular discretization may be required. In Chapter 4, we will address this issue by the development of an angular discretization, alternative to the Discrete Ordinates method.

Finally, we remark how, in order to solve large size calculations, a large amount of memory and computing power is needed. The Domain Decomposition Method is essential for performing calculations on parallel distributed-memory architectures, but, it is not yet available for FCS. An extension of the FCS method allowing to solve parallel calculations in domain decomposition will be presented in Chapter 3.

Chapter 3

Extension of the First Collision Source method to parallel Domain Decomposition

The memory required by radiation shielding calculations is typically much larger than the one available on today's standard computers. Practical problems we strive to solve require to be discretized using tens of millions of spatial regions and tens of energy groups. For a problem made of 10 million regions, linear spatial approximation, anisotropy order $K = 5$ and 20 energy groups, the mere memory storage of the uncollided flux and the first collision source is of about 460 GB in double precision, without accounting for auxiliary memory required by the solver. So large memory occupations, compared with the few tens of GB available in standard workstations, led us to address FCS methods scalability.

Here, our objective consists in extending the capabilities of the FCS module of IDT in order to solve calculations on parallel distributed-memory machines by the use of DDM. In this chapter we define a discretization method for the displacement-plus-removal operator, allowing to compute the uncollided flux in each subdomain of the domain decomposition. Then, we develop an efficient and scalable algorithm for the solution of the problem. Finally, we analyse the accuracy and the computational cost of the method for different refinement degrees, and for different anisotropy orders.

3.1 Introduction to the Domain Decomposition Method for First Collision Source calculations

In Section 1.4 we have seen how IDT offers the possibility of solving large size problems using the Domain Decomposition Method (DDM). The method divides the spatial domain into a finite number of non-overlapping subdomains, which are then solved as independent

problems coupled together by flux continuity relations, acting as boundary conditions of the subdomains. Subdomains are distributed over the nodes of the cluster computer, and then solved by the cores of each node. The DDM of IDT makes use of a mixed shared-distributed memory parallelism. That is, the same node may be in charge of solving multiple subdomains by distributing them over the available cores.

In order to perform FCS calculations on distributed-memory architectures, FCS has to be modified so as to be compatible with the data structure of the DDM of IDT. That is, the FCS calculation has to be split into a set of sub-problems. Further, since the uncollided flux calculation has generally a lower cost than the collided flux calculation, it is reasonable to assume that the domain decomposition is driven by the collided flux calculation. In this respect, FCS has to inherit the same domain partitioning of the collided-flux calculation, i.e. Eq. (1.81).

Here, we will develop DDM for the MOC method with BQ quadrature (see Section 2.3). However, the approach we are going to present can be easily extended to other FCS techniques, such as TS-MOC and QP.

3.1.1 Definition of the problem

With the same notation introduced in Sections 1.3.4 and 1.4.3, the original problem

$$\begin{cases} \mathcal{L}\psi^u(\mathbf{x}) = Q_{ext}(\mathbf{x}) & \text{for } \mathbf{x} \in \mathcal{X} \\ \psi^u(\mathbf{x}) = 0 & \text{for } \mathbf{x} \in \partial\mathcal{X}^- \end{cases} \tag{3.1}$$

defined in domain $\mathcal{X}$, has to be transformed into a set of sub-problems for the uncollided flux ψ_i^u in the different subdomains $\mathcal{X}_i$ with $i = 1, \ldots, N_i$. Here, $\mathcal{L}$ is the the displacement-plus-removal operator, and Q_{ext} is the external source, measuring the number of particle emitted per second, per steradians and per unit energy.

In this chapter, we consider Q_{ext} to be defined by a set of isotropic point sources as in Eq. (1.77), namely

$$Q_{ext}(\mathbf{x}) = \sum_{p=1,N_p} Q_p(\mathbf{x}) = \sum_{p=1,N_p} Q_p(E)\ \delta(\mathbf{r} - \mathbf{r}_p).$$

Because of the nature of the point-kernel, space and angle are dependent. For any point $\mathbf{r}$, the flux generated by Q_p, is directed along direction $\mathbf{\Omega}_p(\mathbf{r}) = \frac{\mathbf{r}-\mathbf{r}_p}{|\mathbf{r}-\mathbf{r}_p|}$ only. Therefore, the angle is fixed for each point $\mathbf{r}$ by knowing the source point $\mathbf{r}_p$, hence, the angular dependency can be omitted. Similarly, from this point on, we omit also the energy variable because uncollided particles move at constant energy. As a result, we are allowed to abandon the phase space notation $\mathcal{X}$ considering only the spatial domain $\mathcal{D}$.

Given a subdomain $\mathcal{D}_i$, with boundaries $\partial\mathcal{D}_i$, and solving for point source Q_p located in $\mathbf{r}_p$, two situations may arise: the one in which the source point is contained in the subdomain, and the one in which it is external. This allows us to identify two types of problems. If $\mathbf{r}_p \in \mathcal{D}_i$, we have

$$\begin{cases} \mathcal{L}\psi^u_{i,p}(\mathbf{r}) = Q_p(\mathbf{r}) & \text{for } \mathbf{r} \in \mathcal{D}_i \\ \psi^u_{i,p}(\mathbf{r}) = 0 & \text{for } \mathbf{r} \in \partial\mathcal{D}_i^- \ . \end{cases} \tag{3.2}$$

If instead, $\mathbf{r}_p \notin \mathcal{D}_i$,

$$\begin{cases} \mathcal{L}\psi^u_{i,p}(\mathbf{r}) = 0 & \text{for } \mathbf{r} \in \mathcal{D}_i \\ \psi^u_{i,p}(\mathbf{r}) = \psi^{in}_{i,p}(\mathbf{r}) & \text{for } \mathbf{r} \in \partial\mathcal{D}_i^- \ , \end{cases} \tag{3.3}$$

where $\psi^u_{i,p}$ is the flux in subdomain $\mathcal{D}_i$, generated by Q_p. $\psi^{in}_{i,p}$ represents the flux entering $\partial\mathcal{D}_i$, and associated to the emission of Q_p. Because of linearity, the flux in subdomain $\mathcal{D}_i$ is computed by summing the contributions of the different sources, i.e.

$$\psi^u_i = \sum_{p=1,N_p} \psi^u_{i,p}. \tag{3.4}$$

The solution of problems of the form of Eq. (3.2) presents no additional complexity to the standard FCS calculation. We are able to perform this calculation by simply discretizing $\mathcal{L}$ with one of the FCS techniques presented in Chapter 2. On the contrary, this is not the case for problems assimilable to Eq. (3.3). In order to solve Eq. (3.3), we have first to introduce a discrete representation of the uncollided flux at the subdomain interfaces.

3.1.2 Analysis of the problem related to the calculation of the uncollided flux in subdomains not containing the source

Consider problem Eq. (3.3), i.e. the calculation of the uncollided flux in a subdomain $\mathcal{D}_i$, with source Q_p located in $\mathbf{r}_p \notin \mathcal{D}_i$. Given the point source, we are able to distinguish between incoming and outgoing boundaries (with respect to the source) as

$$\begin{aligned} \partial\mathcal{D}^-_{i,p} &= \{\mathbf{r} \in \partial\mathcal{D}_i |\ \mathbf{\Omega}_p(\mathbf{r}) \cdot \mathbf{n} < 0\}, \\ \partial\mathcal{D}^+_{i,p} &= \{\mathbf{r} \in \partial\mathcal{D}_i |\ \mathbf{\Omega}_p(\mathbf{r}) \cdot \mathbf{n} \geq 0\}, \end{aligned}$$

with $\mathbf{n}$ the outgoing normal direction to the subdomain boundary $\partial\mathcal{D}_i$. Clearly, given a different point source, incoming and outgoing boundaries may be different. Then, we identify the incoming and the outgoing boundary fluxes respectively as

$$\psi^-_{i,p} = \psi^u_{i,p}(\mathbf{r}) \quad \text{with } \mathbf{r} \in \partial\mathcal{D}^-_{i,p}, \tag{3.5}$$

$$\psi^+_{i,p} = \psi^u_{i,p}(\mathbf{r}) \quad \text{with } \mathbf{r} \in \partial\mathcal{D}^+_{i,p}. \tag{3.6}$$

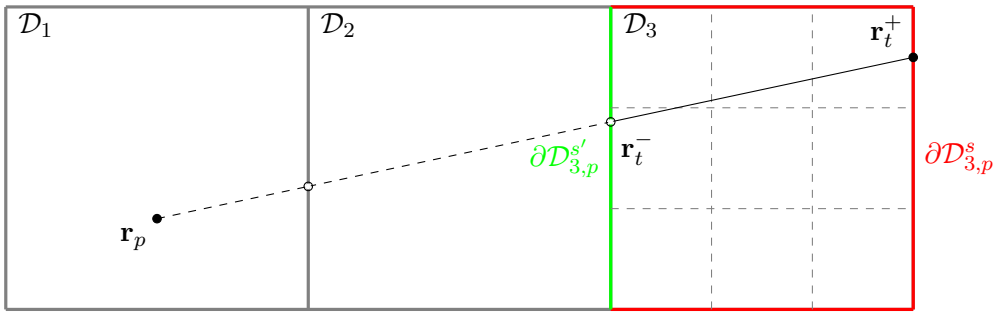


Fig. 3.1 Representation of MOC ray tracing across different subdomains.

Since the subdomains are box-shaped with 6 faces, we denote by $\partial\mathcal{D}_{i,p}^{s'}$ the subdomain faces of the incoming boundary, i.e. $\partial\mathcal{D}_{i,p}^{s'} \in \partial\mathcal{D}_{i,p}^{-}$, with s' as the index of the incoming interface. In the same way, we denote with $\partial\mathcal{D}_{i,p}^{s}$ the interfaces of the outgoing boundaries such that $\partial\mathcal{D}_{i,p}^{s} \in \partial\mathcal{D}_{i,p}^{+}$ with s as the index of the outgoing interface.

Since MOC-BQ relies on ray-tracing with trajectories directed to the domains boundaries, it is useful to introduce a new notation in order to identify the boundary quadratures local to each subdomain. Given an outgoing interface $\partial\mathcal{D}_{i,p}^{s}$, we define with $\mathcal{P}_{i,p}^{s} = \left\{\mathbf{r}_t^+\right\}_{t=1,N_t}$ the set of points of $\partial\mathcal{D}_{i,p}^{s}$ which are used as support for ray tracing. With reference to Section 2.3.1, each point $\mathbf{r}_t^+ \in \mathcal{P}_{i,p}^{s}$ is defined as the centroid of the boundary-mesh region identified by points $\mathbf{r}_{t,1}$, $\mathbf{r}_{t,2}$, $\mathbf{r}_{t,3}$ and $\mathbf{r}_{t,4}$. N_t is instead the number of points in $\mathcal{P}_{i,p}^{s}$. For convenience, we also define $\mathcal{P}_{i,p}^{s'} = \left\{\mathbf{r}'_{t'}\right\}_{t'}$ as the set of points of the boundary quadrature, but generated on the incoming interfaces $\partial\mathcal{D}_{i,p}^{s'}$. $\mathcal{P}_{i,p}^{s'}$ coincides with the set of points $\mathcal{P}_{i',p}^{s}$ of the upstream subdomain $\mathcal{D}_{i'}$.

If the subdomain does not contain the source, the trajectory joining the source with $\mathbf{r}_t^+$ will intersect one of the incoming interfaces in point $\mathbf{r}_t^-$ (see Figure 3.1). By grouping all these points together for each of the incoming interfaces $\partial\mathcal{D}_{i,p}^{s'}$, we define the set

$$\mathcal{Q}_{i,p}^{s'} = \left\{ \mathbf{r}_t^- \in \partial\mathcal{D}_{i,p}^{s'} \mid \frac{\mathbf{r}_t^+ - \mathbf{r}_t^-}{\left|\mathbf{r}_t^+ - \mathbf{r}_t^-\right|} = \mathbf{\Omega}_p(\mathbf{r}_t^+) \right\}.$$

With reference to Fig. 3.1, while solving for $\mathcal{D}_3$, in order to integrate the uncollided flux along trajectory t (generated by point $\mathbf{r}_t^+$) we need to know the incoming flux $\psi_{i,p}^-$ in $\mathbf{r}_t^-$, as indicated by the boundary conditions of problem in Eq. (3.3). This flux, however, is the result of another problem, that is the calculation of the uncollided flux in subdomain $\mathcal{D}_2$. Generalising, in order to solve a subdomain not containing the source, it is first necessary to solve the upstream subdomains. Therefore, the FCS method has to solve subdomains in order, following the propagation front starting from the source. Moreover, the calculation of each subdomain has to provide the uncollided flux in each region of the subdomain, as well as the flux at the outgoing boundaries $\psi_{i,p}^+$, which is required to solve downstream subdomains.

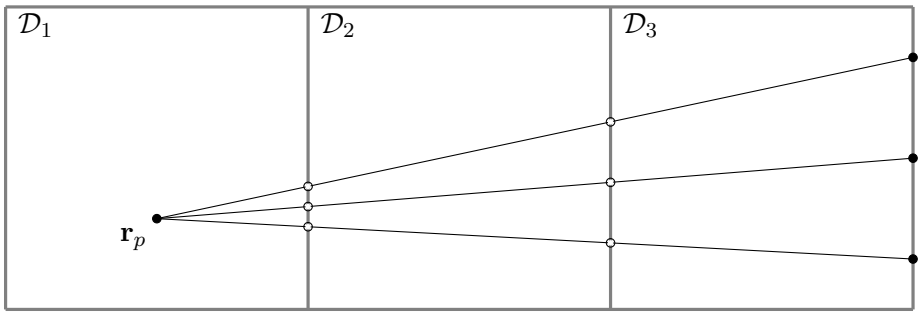


Fig. 3.2 Representation of MOC ray tracing using a unique set of trajectories for all subdomains.

We also remark that, when solving for the flux in $\mathcal{D}_2$ using MOC, point $\mathbf{r}_t^-$ should be part of the boundary quadrature $\mathcal{P}_{2,p}^s$ in order to evaluate $\psi_{i,p}^+$ in $\mathbf{r}_t^-$. Alternatively, an ad-hoc ray tracing has to be preformed to provide those values to the downstream subdomain.

The problem has a straightforward solution, which is the use of a unique set of trajectories discretizing the whole domain. These trajectories can be generated using an unique quadrature defined over the boundaries of the complete domain $\partial\mathcal{D}$ as illustrated in Figure 3.2. In such a case, each point of $\mathcal{Q}_{i,p}^{s'}$ coincides with a point in $\mathcal{P}_{i',p}^s$ of the upstream subdomain $\mathcal{D}_{i'}$. Therefore, the flux in $\mathcal{Q}_{i,p}^{s'}$ is provided by the solution of $\mathcal{D}_{i'}$ using MOC ray tracing. It is easy to see how this approach causes a densification of the trajectories within the subdomain containing the source. In this subdomain, a large number of trajectories has to be traced, thus resulting in a larger computational cost with respect to the solution of more distant subdomains. This problem has already been observed in Chapter 2 when dealing with MOC. Also, this is precisely the reason why we designed the Trajectory Splitting MOC. In the case of domain decomposition, however, an uneven distribution of trajectories across subdomains leads to an imbalance of the workload of each process, causing potential idle times and reduced parallel efficiencies.

3.1.3 Research of a viable solution to trajectory thinning in domain decomposition

Literature on DDM for First Collision Source methods is scarce. However, a notable approach to this issue has been presented by M. Hanuš *et al.*, [54]. The authors propose a MOC-based technique similar to the one of IDT: this is based on the use of a different set of trajectories in each subdomain, as depicted in Fig. 3.3. Trajectories are generated using a BQ quadrature, i.e. tracing from the source towards the points of a boundary mesh $\mathcal{P}_{i,p}^s$ generated on the outgoing boundaries of each subdomain (black dots in Fig. 3.3). Since trajectories do not have support within the subdomains located upstream (with respect to the subdomain to be integrated), Hanuš *et al.* use an interpolative approximation in order to retrieve the flux in the desired points, namely $\mathcal{Q}_{i,p}^{s'}$ (represented by the empty dots in Fig. 3.3).

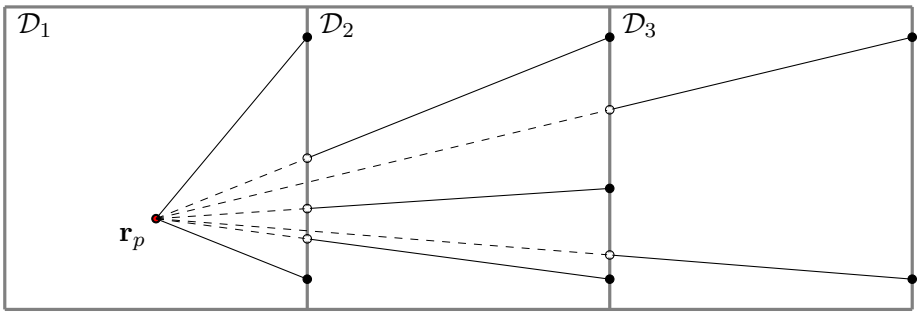


Fig. 3.3 Representation of the trajectory-based discretization using a different boundary quadratures for each subdomain.

The interpolation, however, is not performed directly on the flux but rather on the equivalent cross section Σ_p^{eq} at the boundary. For a generic trajectory generated by tracing from the source point $\mathbf{r}_p$ to point $\mathbf{r}$, the equivalent cross section Σ_p^{eq} in $\mathbf{r}_t^+$ is defined by the relation

$$\Sigma_p^{eq}(\mathbf{r}_t^+) = \frac{\sum_{k=1,N_k} \Sigma_k L_k}{\sum_{k=1,N_k} L_k}, \tag{3.7}$$

where Σ_k and L_k are the total cross section and the chord length in the spatial regions intersected by the trajectory. Numerically, Σ_p^{eq} is the total cross section that an equivalent homogeneous medium should have in order to reproduce the same optical attenuation along the trajectory t, i.e.

$$Q_p \prod_{k=1,N_k} e^{-\Sigma_k L_k} = Q_p e^{-\Sigma_p^{eq} \sum_{k=1,N_k} L_k}. \tag{3.8}$$

For convenience, we will refer to the approximation on Σ_p^{eq} introduced by Hanuš *et al.* as Equivalent Cross Section (EXS) approximation. In the following section we present the EXS approximation, explaining with an example how it is used within the FCS calculation.

3.1.4 Description of FCS method based on the equivalent cross section approximation

On Fig. 3.4, we denote with $\mathcal{D}_1$ the subdomain containing the source, and with $\mathcal{D}_2$ a neighbouring subdomain. The FCS calculation requires to solve subdomains sequentially following the propagation front, therefore, $\mathcal{D}_1$ first and then $\mathcal{D}_2$. We will now denote with

- $\mathcal{P}_{2,p}^s = \{\mathbf{r}_t^+\}$ the set of points on boundary $\partial\mathcal{D}_{2,p}^s$, necessary for both the integration on the flux in $\mathcal{D}_2$, and the calculation of the outgoing flux in $\partial\mathcal{D}_{2,p}^s$;

- $\mathcal{Q}_{2,p}^{s'} = \{\mathbf{r}_t^-\}$ the set of points formed by the intersections of the trajectories, generated using points $\mathcal{P}_{2,p}^s$, with the incoming interface $\partial\mathcal{D}_{2,p}^{s'}$;

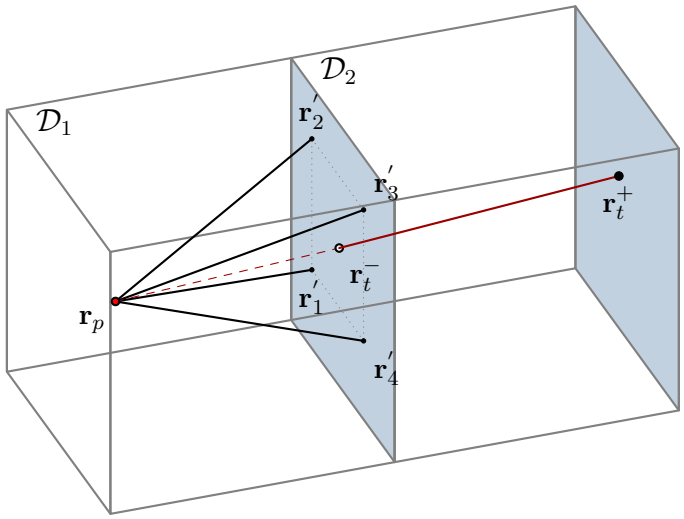


Fig. 3.4 Illustration of the Equivalent Cross Section approximation.

- $\mathcal{P}_{2,p}^{s'} = \{\mathbf{r}_{t'}'\}$ the set of points on the incoming interface $\partial\mathcal{D}_{2,p}^{s'}$, coincident with $\mathcal{P}_{1,p}^{s}$, i.e. the points of the boundary quadrature for the integration of $\mathcal{D}_1$ (in Fig. 3.4, $\mathbf{r}_1'$, $\mathbf{r}_2'$, $\mathbf{r}_3'$ and $\mathbf{r}_4'$ are four points of $\mathcal{P}_{2,p}^{s'}$).

The calculation of $\mathcal{D}_1$ provides both the uncollided flux in the subdomain, and the equivalent cross section Σ_p^{eq} in point $\mathcal{P}_{2,p}^{s'}$. In each point of $\mathcal{P}_{2,p}^{s'}$, Σ_p^{eq} is computed solving Eq. (3.7).

After, the value of the incoming flux in each point $\mathbf{r}_t^- \in \mathcal{Q}_{2,p}^{s'}$ is required when solving for $\mathcal{D}_2$. The method introduced by Hanuš *et al.* retrieves this value by computing

$$\psi_{i,p}^-(\mathbf{r}_t^-) = Q_p \, e^{-\left|\mathbf{r}_t^- - \mathbf{r}_p\right| \Sigma_p^{eq}(\mathbf{r}_t^-)}, \quad \text{for } \mathbf{r}_t^- \in \mathcal{Q}_{2,p}^{s'}, \tag{3.9}$$

and approximating $\Sigma_p^{eq}(\mathbf{r}_t^-)$ using an inverse-distance weighted interpolation between the Σ_p^{eq} evaluated in $\mathcal{P}_{2,p}^{s'}$, [54, 88]. In detail, supposing that $\mathbf{r}_t^-$ falls in the region of the boundary mesh $\mathcal{P}_{2,p}^{s'}$ delimited by $\mathbf{r}_1'$, $\mathbf{r}_2'$, $\mathbf{r}_3'$ and $\mathbf{r}_4'$,

$$\Sigma_p^{eq}(\mathbf{r}_t^-) = \begin{cases} \Sigma_p^{eq}(\mathbf{r}_{t'}') & \text{if } |\mathbf{r}_t^- - \mathbf{r}_{t'}'| = 0 \text{ for } t' = 1,2,3,4 \ , \\ \dfrac{\sum_{t'=1,4} w_{t'} \Sigma_p^{eq}(\mathbf{r}_{t'}')}{\sum_{t'=1,4} w_{t'}} & \text{otherwise,} \end{cases} \tag{3.10}$$

with $w_{t'}$ as the weight function,

$$w_{t'} = \frac{1}{\left|\mathbf{r}_t^- - \mathbf{r}_{t'}'\right|^{\beta}} \quad \text{with } t' = 1,2,3,4 \ , \tag{3.11}$$

and parameter β equal to 2. The finer the boundary mesh, the smaller the distance between the points in $\mathcal{P}_{2,p}^{s'}$, and the greater the approximation accuracy.

While solving for the uncollided flux in $\mathcal{D}_2$, the method also computes Σ_p^{eq} in $\mathbf{r}_t^+ \in \mathcal{P}_{2,p}^s$ as

$$\Sigma_p^{eq}(\mathbf{r}_t^+) = \frac{\Sigma_p^{eq}(\mathbf{r}_t^-)\left|\mathbf{r}_t^- - \mathbf{r}_p\right| + \sum_{k=1,N_k}\Sigma_k L_k}{\left|\mathbf{r}_t^+ - \mathbf{r}_p\right|}. \tag{3.12}$$

Remark that $\Sigma_p^{eq}(\mathbf{r}_t^+)$ is representative of the optical length along the entire path between $\mathbf{r}_p$ and $\mathbf{r}_t^+$, and not of the portion of trajectory crossing one single subdomain.

Eqs. (3.7) and (3.12) are the result of our elaboration of the concepts presented in the original publication of M. Hanuš *et al.*, [54]. Thus, the equations may not be exactly the same as those employed by the authors.

Considerations

Because of the exponential flux attenuation, the flux at the subdomain interfaces varies much more rapidly (in relative terms) than Σ_p^{eq} in the presence of optically thick media ($\Sigma_p^{eq} L > 1$). The interpolation of fluxes presenting very steep variations, rather than Σ_p^{eq}, would result in very large errors. In this respect, the EXS approximation, acting on equivalent cross sections, is expected to be much more accurate, [54].

The EXS approximation provides a synthetic way of representing the entire optical path between the source point and the subdomain of interest. However, in the presence of several subdomains, the EXS approximation operated on a certain subdomain is build on top of the EXS approximations operated in the upstream subdomains. It follows that great care must be taken so as to ensure that these errors do not increase excessively with the number of subdomains.

The method, by using a single boundary mesh supporting the MOC integration and allowing for the evaluation of Σ_p^{eq}, favours the reuse of trajectories. This minimizes the number of trajectories to be traced and, consequently, the computational cost.

In conclusion, the method appears to be effective in rebalancing the workload of the processes, and furthermore, it provides a concise representation of the fluxes at the subdomain interfaces, which is advisable to reduce communication costs in DDM. In our work, we have decided to adopt the EXS approximation illustrated in [54], but we have derived a different method.

3.2 Development of the First Collision Source in domain decomposition

The key idea of the method presented in this work is that the error introduced by the EXS approximation may be difficult to control using the approach presented in Section 3.1.4.

In DDM, the integration error is related to two distinct approximations: the MOC, and the EXS approximation. The former is the error committed while integrating the uncollided flux over a subdomain $\mathcal{D}_i$. The latter, instead, is the error introduced while approximating $\psi^-_{i,p}$ at the incoming boundaries of each subdomain $\mathcal{D}_i$ not containing the source. When performing calculations on large numbers of subdomains, these two contributions may have different weights, hence, the interest of controlling these two sources of error separately.

3.2.1 Definition of the boundary discretization

Our method relies on two different boundary discretizations, one for the MOC integration, and the other for Σ^{eq}_p. The idea is to allow to refine points $\mathcal{P}^{s'}_{i,p}$, used for the interpolations, regardless of the boundary discretization $\mathcal{P}^{s}_{i,p}$ used by MOC. To this extent, we define:

- $\mathcal{P}^{s,M}_{i,p}$ as the set of points generated by the the MOC boundary mesh, for each outgoing interface $\partial\mathcal{D}^{s}_{i,p}$, of each subdomain $\mathcal{D}_i$;
- $\mathcal{Q}^{s',M}_{i,p}$ as the set of points generated by the intersection of the MOC trajectories with $\partial\mathcal{D}^{s'}_{i,p}$, as done in Section 3.1.2.

Points $\mathbf{r}^+_t \in \mathcal{P}^{s,M}_{i,p}$ are the centroids of the square elements of a BQ boundary mesh. The spacing of the mesh can be set differently in each subdomain $\mathcal{D}_i$ (in our tests, however, we always use a equal spacing for all subdomains).

Then, a second boundary mesh for the Σ^{eq}_p evaluation is generated. It is defined on all subdomain boundaries, incoming and outgoing, and using a unique spacing in all subdomains, Δ_{EXS}. This makes the mesh conformal over the interfaces of the entire domain. We denote

- $\mathcal{P}^{s,E}_{i,p}$ as the set of points of the EXS boundary mesh defined on the outgoing boundaries $\partial\mathcal{D}^{s}_{i,p}$;
- $\mathcal{P}^{s',E}_{i,p}$ as the set of points of the EXS boundary mesh on the incoming boundaries $\partial\mathcal{D}^{s'}_{i,p}$, with $\mathcal{P}^{s',E}_{i,p} \equiv \mathcal{P}^{s,E}_{i',p}$ and $\mathcal{D}_{i'}$ being the upstream subdomain;
- $\mathcal{Q}^{s',E}_{i,p}$ as the set of points originated by intersecting $\partial\mathcal{D}^{s'}_{i,p}$ with trajectories generated from $\mathcal{P}^{s,E}_{i,p}$.

Differently by the MOC mesh, the point in $\mathcal{P}^{s,E}_{i,p}$ and $\mathcal{P}^{s',E}_{i,p}$ are defined as the vertices of elements of a BQ mesh. In doing so, each point $\mathbf{r} \in \partial\mathcal{D}^{s'}_{i,p}$ will always be comprised between four points of $\mathcal{P}^{s',E}_{i,p}$, which is ideal for the application of the EXS approximation.

In practise, only points $\mathcal{P}^{s,M}_{i,p}$, $\mathcal{P}^{s,E}_{i,p}$ and $\mathcal{P}^{s',E}_{i,p}$ are actually stored. Points $\mathcal{Q}^{s',M}_{i,p}$ and $\mathcal{Q}^{s',E}_{i,p}$ are instead calculated on-the-fly by tracing backward the trajectories from points $\mathcal{P}^{s,M}_{i,p}$ and $\mathcal{P}^{s,E}_{i,p}$ towards the source point. In the following, we will refer to the boundary mesh generating $\mathcal{P}^{s,E}_{i,p}$ and $\mathcal{P}^{s',E}_{i,p}$ as EXS boundary mesh, and to the one generating points $\mathcal{P}^{s,M}_{i,p}$ as MOC boundary mesh.

3.2.2 Derivation of the equations of the First Collision Source in domain decomposition

By solving separately the calculation of uncollided flux $\psi^u_{i,p}$ and the calculation of outgoing flux $\psi^+_{i,p}$ in the two problems, Eqs. (3.2) and (3.3), we distinguish four different calculations. These are

$$\psi^u_{i,p} = \mathcal{L}^{-1}Q_p, \quad \text{if } \mathbf{r}_p \in \mathcal{D}_i \tag{3.13}$$
$$\psi^u_{i,p} = \mathcal{L}^{-1}\psi^-_{i,p}, \quad \text{if } \mathbf{r}_p \notin \mathcal{D}_i, \tag{3.14}$$

solving for the uncollided flux $\psi^u_{i,p}$, and

$$\psi^+_{i,p} = \mathcal{L}^{-1}Q_p, \quad \text{if } \mathbf{r}_p \in \mathcal{D}_i \tag{3.15}$$
$$\psi^+_{i,p} = \mathcal{L}^{-1}\psi^-_{i,p}, \quad \text{if } \mathbf{r}_p \notin \mathcal{D}_i. \tag{3.16}$$

solving for the outgoing flux $\psi^+_{i,p}$.

In order to solve the four equations, we introduce the discrete representation of the boundary fluxes in points $\mathcal{P}^{s,E}_{i,p}$ and $\mathcal{P}^{s',E}_{i,p}$. The set $\mathcal{P}^{s,M}_{i,p}$ is used instead by the MOC method for the discretization of the uncollided flux. We denote with $\mathbf{\Psi}^u_{i,p}$ the MOC discretized form of $\psi^u_{i,p}$ and, in place of the boundary fluxes, $\psi^-_{i,p}$ and $\psi^+_{i,p}$, we use Σ^{eq}_p evaluated in the incoming and outgoing points, $\mathcal{P}^{s',E}_{i,p}$ and $\mathcal{P}^{s,E}_{i,p}$, respectively. In particular, we define

- $\mathbf{\Sigma}^{eq,+}_{i,p}$, the Σ^{eq}_p evaluated in $\mathcal{P}^{s,E}_{i,p}$, for each of the outgoing interfaces $\partial\mathcal{D}^s_{i,p} \in \partial\mathcal{D}^+_{i,p}$;
- $\mathbf{\Sigma}^{eq,-}_{i,p}$, the Σ^{eq}_p evaluated in $\mathcal{P}^{s',E}_{i,p}$, for each of the incoming interfaces $\partial\mathcal{D}^{s'}_{i,p} \in \partial\mathcal{D}^-_{i,p}$.

We recall that, because of Eq. (3.9), the knowledge of Σ^{eq}_p, source intensity, and the distance from the source, allows for the computation of the uncollided flux in any point of the domain.

Further, by noticing a certain similarity with the balance and transmission equations for the single spatial region, i.e. Eqs. (1.67) and (1.68), we will now denote the discretized (inverted) streaming-plus-removal operator using four different operators, namely the collision, the incoming, the escape and the transmission operators.

By introducing this discretization, Eqs. (3.13) and (3.14) become respectively

$$\mathbf{\Psi}^u_{i,p} = \mathbf{C}^{FCS}_{i,p}Q_p, \quad \text{if } \mathbf{r}_p \in \mathcal{D}_i, \tag{3.17}$$
$$\mathbf{\Psi}^u_{i,p} = \mathbf{I}^{FCS}_{i,p}\mathbf{\Sigma}^{eq,-}_{i,p}, \quad \text{if } \mathbf{r}_p \notin \mathcal{D}_i, \tag{3.18}$$

while Eqs. (3.15) and (3.16) become

$$\mathbf{\Sigma}^{eq,+}_{i,p} = \mathbf{E}^{FCS}_{i,p}Q_p, \quad \text{if } \mathbf{r}_p \in \mathcal{D}_i, \tag{3.19}$$
$$\mathbf{\Sigma}^{eq,+}_{i,p} = \mathbf{T}^{FCS}_{i,p}\mathbf{\Sigma}^{eq,-}_{i,p}, \quad \text{if } \mathbf{r}_p \notin \mathcal{D}_i. \tag{3.20}$$

The system of equations is closed by a flux continuity relation

$$\psi^-_{i,p} = \psi^+_{i',p}, \tag{3.21}$$

for each couple of neighbouring subdomains $\mathcal{D}_i$ and $\mathcal{D}_{i'}$, with $\mathcal{D}_{i'}$ the upstream subdomain. By introducing the discretization of the boundary fluxes, Eq. (3.21) becomes

$$\mathbf{\Sigma}^{eq,-}_{i,p} = \mathbf{\Sigma}^{eq,+}_{i',p}. \tag{3.22}$$

We have that:

- the collision operator $\mathbf{C}^{FCS}_{i,p}$ coincides with the MOC-discretized streaming-plus-removal operator $\mathcal{L}^{-1}_{FCS}$, restricted to subdomain $\mathcal{D}_i$. The calculation of Eq. (3.17) is solved using the MOC-BQ method as presented in Section 2.3, with $\mathcal{P}^{s,M}_{i,p}$ as boundary quadrature;
- the incoming operator $\mathbf{I}^{FCS}_{i,p}$, relates Σ^{eq}_p, defined in points $\mathcal{P}^{s',E}_{i,p}$, with the uncollided flux in the subdomain. The calculation of Eq. (3.18) consists in three steps. For each point $\mathbf{r}^+_t \in \mathcal{P}^{s,M}_{i,p}$, we first trace the trajectory backward in order to compute the entering point $\mathbf{r}^-_t \in \mathcal{Q}^{s',M}_{i,p}$. Then, we compute $\psi^-_{i,p}$ in $\mathbf{r}^-_t$ using Eq. (3.9), and approximating $\Sigma^{eq}_p(\mathbf{r}^-_t)$ using the EXS approximation. In practice, we solve Eq. (3.10), interpolating between $\mathbf{\Sigma}^{eq,-}_{i,p}$. Finally, disposing of $\psi^-_{i,p}(\mathbf{r}^-_t)$, the calculation proceeds as a standard MOC integration with $\mathcal{P}^{s,M}_{i,p}$ as boundary quadrature;
- The escape operator $\mathbf{E}^{FCS}_{i,p}$ relates the source, located in the subdomain, with equivalent cross sections $\mathbf{\Sigma}^{eq,+}_{i,p}$. The calculation in Eq. (3.19) consists in the evaluation of Σ^{eq}_p in each point $\mathbf{r}^+_t \in \mathcal{P}^{s,E}_{i,p}$. This is done by computing Eq. (3.7) along the trajectories generated by joining the source with points $\mathcal{P}^{s,E}_{i,p}$;
- The transmission operator $\mathbf{T}^{FCS}_{i,p}$ relates $\mathbf{\Sigma}^{eq,-}_{i,p}$ with $\mathbf{\Sigma}^{eq,+}_{i,p}$. The calculation of Eq. (3.20) consists of three steps. For each point $\mathbf{r}^+_t \in \mathcal{P}^{s,M}_{i,p}$ we first retrieve point $\mathbf{r}^-_t \in \mathcal{Q}^{s',E}_{i,p}$ by back-tracing from $\mathbf{r}^+_t \in \mathcal{P}^{s,E}_{i,p}$ to the source. Then, we interpolate Σ^{eq}_p in $\mathbf{r}^-_t$ using Eq. (3.10), and $\mathbf{\Sigma}^{eq,-}_{i,p}$. Finally, we compute Σ^{eq}_p in each point of $\mathcal{P}^{s,E}_{i,p}$ solving Eq. (3.12) along the trajectory joining $\mathbf{r}^-_t$ with $\mathbf{r}^+_t$.

Since the calculation of Eqs. (3.18) and (3.20) involves the use of an interpolative approximation, the particle balance in not preserved on the domain. As discussed in Chapter 2, the uncollided flux is computed for all groups simultaneously. This implies that the Σ^{eq}_p have to be stored in all boundary points and for all energy groups. Furthermore, the Σ^{eq}_p are specific to the point source location $\mathbf{r}_p$. Therefore, when solving for different point sources simultaneously, different Σ^{eq}_p have to evaluated for the different point sources. Finally, because of the equations solved, at equal number of points in $\mathcal{P}^{s,E}_{i,p}$ and $\mathcal{P}^{s,M}_{i,p}$, the calculation of Eqs. (3.19) and (3.20) is much cheaper in terms of computational cost than the calculation of Eqs. (3.17) and (3.18).

3.3 Design of an efficient scheduling algorithm for the First Collision Source in Domain Decomposition

By relying on the considerations made in the previous sections, a scheduling algorithm for the FCS in DDM has to be developed. To do so, we define the unit work performed by the process, i.e. the task. Then, we identify the dependencies between the different tasks, and we specify the order in which they are solved.

3.3.1 Definition of the tasks

The way the DDM of IDT is structured, the task coincides with the solution of a transport calculation in a given subdomain. Aiming to use a similar approach for FCS in DDM, we note that, for each subdomain, two calculations are possible. These are the calculation of Σ_p^{eq} at the outgoing boundaries of the subdomain, and the calculation of uncollided flux in the subdomain volume.

This allows us to define two types of tasks. For a given subdomain $\mathcal{D}_i$, and for a given point source Q_p, we define:

- $\text{MOC}_{i,p}$, as the task solving for the uncollided flux in subdomain $\mathcal{D}_i$, emitted by the point source Q_p. If $\mathbf{r}_p \in \mathcal{D}_i$, the task coincide with the solution of Eq. (3.17). Otherwise, it solves Eq. (3.18);

- $\text{EXS}_{i,p}$, as the task solving for the Σ_p^{eq} values of the outgoing boundaries of subdomain $\mathcal{D}_i$, for the given source point Q_p. EXS solves Eq. (3.19), if $\mathbf{r}_p \in \mathcal{D}_i$, and Eq. (3.20), if $\mathbf{r}_p \notin \mathcal{D}_i$.

3.3.2 Definition of the tasks dependencies

By considering the entire calculation in domain decomposition, made of several subdomains $\mathcal{D}_i$, and several sources Q_p, we remark that:

- $\text{MOC}_{i,p}$ and $\text{EXS}_{i,p}$ tasks can be solved in parallel for the same subdomain;

- The calculations of the different point sources are independent of each other. Therefore, tasks solving for different sources can be run in parallel;

- For a given point source, $\text{EXS}_{i,p}$ tasks have to be solved sequentially following the propagation front. For a given subdomain, $\text{EXS}_{i,p}$ and $\text{MOC}_{i,p}$ can be solved only if:

  a) the source is contained in the subdomain;

  b) $\text{EXS}_{i,p}$ has been solved for all upstream subdomains.

3.3.3 Definition of the tasks execution priority

The scheduling of FCS tasks is strongly conditioned by the sequential order in which the subdomain calculations have to be solved. This dependence implies unavoidable idle times. In order to reduce them, the straightforward solution is to schedule multiple source points at the same time. However, processing different source points simultaneously implies the allocation of Σ_p^{eq} for each source point. The memory occupied by the Σ_p^{eq} values, if not bounded, can be a limiting factor for the simulation. In our implementation, the maximum number of sources that can be scheduled at the same time is a user defined parameter. Source points are also ordered, so as to define their execution priority.

The two types of tasks, $\mathrm{MOC}_{i,p}$ and $\mathrm{EXS}_{i,p}$, as already mentioned, may have two very different computational costs. At equal number of trajectories, the $\mathrm{MOC}_{i,p}$ calculation is more expensive than $\mathrm{EXS}_{i,p}$. In order to minimize the idle times, we have designed the algorithm in such a way that all $\mathrm{MOC}_{i,p}$ tasks are solved in parallel for the same point source. In order for the $\mathrm{MOC}_{i,p}$ calculations to take place simultaneously in all subdomains, Σ_p^{eq} values have to be known for all subdomain boundaries. As soon as they are available, the algorithm performs the $\mathrm{MOC}_{i,p}$ calculation on the entire domain. Meanwhile, it solves $\mathrm{EXS}_{i,p}$.

Summarizing,

- the $\mathrm{MOC}_{i,p}$ task is solved only when it is solvable in each subdomain;
- given the first condition, the $\mathrm{MOC}_{i,p}$ task is prioritized over $\mathrm{EXS}_{i,p}$;
- the choice of the point source to be solved takes place according to the absolute numbering of source points.

3.3.4 Design of the tasks scheduling algorithm

We now describe the sheduling algorithm, and how the calculation is solved using a hybrid shared-memory/distributed-memory parallelism model. In Section 1.4.4, while describing the implementation of the DDM of IDT, we have discussed how subdomains are grouped into lists and distributed over the nodes of the cluster. Each node is in charge of solving for all subdomains $\mathcal{D}_i$ belonging to list $L_n^{\mathcal{D}} = \{\mathcal{D}_i\}_{i=1,N_n}$ with N_n the number of subdomains contained in the list. When solving on a shared-memory architecture, all subdomains are arranged in a unique list, with $N_n = N_i$, and N_i the total number of subdomains.

In order to be compliant with the DDM data structure, in FCS, each node has to solve $\mathrm{MOC}_{i,p}$ and $\mathrm{EXS}_{i,p}$ tasks for subdomains $\mathcal{D}_i \in L_n^{\mathcal{D}}$, and all point sources Q_p with $p = 1, \ldots, N_p$. Algorithm 4 displays how the tasks are scheduled in each node when solving FCS.

Algorithm 4: The scheduling algorithm of the FCS in DDM for a process n.

```
   Data: L_n^D, Q_p with p = 1, ..., N_p
 1 begin
 2   do
 3     if ∃Q_p | Σ_p^eq is known for each P_{i,p}^{s',E} with i = 1, ..., N_i then
 4       foreach D_i ∈ L_n^D do (parallel execution)
 5         Solve MOC_{i,p} for Q_p, in subdomain D_i;
 6       end
 7     else
 8       foreach D_i ∈ L_n^D do (parallel execution)
 9         if ∃Q_p | Σ_p^eq is known for P_{i,p}^{s',E} in subdomain D_i then
10           Solve EXS_{i,p} for source Q_p, in subdomain D_i;
11         end
12       end
13     end
14     Synchronization;
15     Solve Eq. (3.22) for all couple of neighbouring subdomains belonging to
         different lists (Exchange Σ_p^eq);
16   while all MOC_{i,p} tasks have not been solved, for all D_i and all Q_p, with
       i = 1, ..., N_i and p = 1, ..., N_p;
17 end
```

The algorithm is structured as an infinite loop, ending when all tasks have been executed. At each loop, using MPI communications, Σ_p^{eq} are exchanged between neighbouring subdomains belonging to different lists.

At each loop, the algorithm first verifies the possibility of solving $\mathrm{MOC}_{i,p}$ calculations simultaneously in every subdomain $\mathcal{D}_i$, for any of the point sources. This coincides to verify that Σ_p^{eq} is known in every point in $\mathcal{P}_{i,p}^{s',E}$, for all subdomains of the domain. If this is possible, the research provides the source Q_p to be solved. Then, the algorithm executes $\mathrm{MOC}_{i,p}$ calculations, solving the different subdomains in parallel on the available CPU cores. At the end of the calculation, the algorithm executes the synchronization with other nodes. We remark that, $\mathrm{MOC}_{i,p}$ are solved in parallel (for the same source point) for all N_i subdomains, and not only for the subdomains in list $L_n^{\mathcal{D}}$.

If the research at line **3** of Algorithm 4 gives a negative result, the algorithm executes a loop over the subdomains of list $L_n^{\mathcal{D}}$. The loop is performed in parallel on the available CPU cores. Then, each core, solving for a different subdomain $\mathcal{D}_i$, performs a second research in order to identify one $\mathrm{EXS}_{i,p}$ task to execute. In detail, the algorithm proceeds considering sources in order, for $p = 1, \ldots, N_p$. For each source, it verifies if Σ_p^{eq} is available in $\mathcal{P}_{i,p}^{s',E}$. If so, $\mathrm{EXS}_{i,p}$ is executed. Otherwise, the algorithm proceed further in the research. If the research at line **9** gives a negative result for all sources, only in this case the core will not perform any calculation. Once the loop over the subdomains ends, the process are synchronized, and Σ_p^{eq}

are exchanged. Remarks that the exchange of Σ_p^{eq} described by Eq. (3.22) interests all couples of neighbouring subdomains. However, only the exchange between subdomains belonging to different lists involves MPI communications.

In order to limit the number of sources treated at the same time, some precautions have been employed. In the actual implementation, the researches at lines **3** and **9** are done on a sub-list of sources whose size is limited by a user-defined parameter. Once a $\mathrm{MOC}_{i,p}$ calculation is completed, source Q_p is removed from the list and a new source is added.

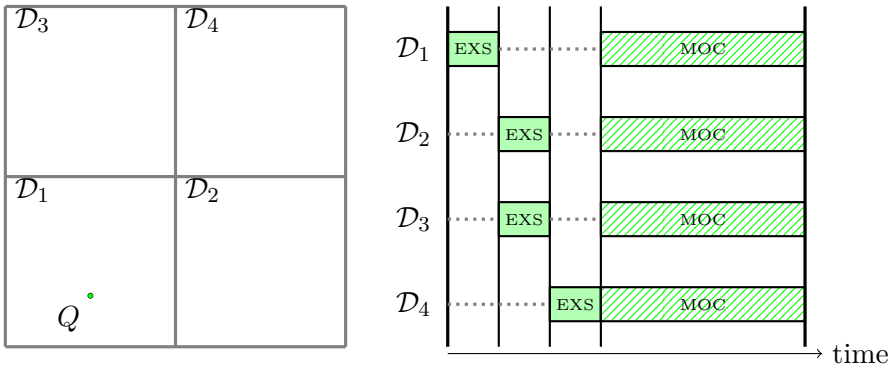
3.3.5 Qualitative analysis of the scheduling algorithm effectiveness

In this section we illustrate, with the help of some examples, how the algorithm works, and how the tasks are actually scheduled. The representations we propose are based on the observation that $\mathrm{EXS}_{i,p}$ is computationally cheaper than $\mathrm{MOC}_{i,p}$ tasks, for an equal refinement of the boundary meshes.

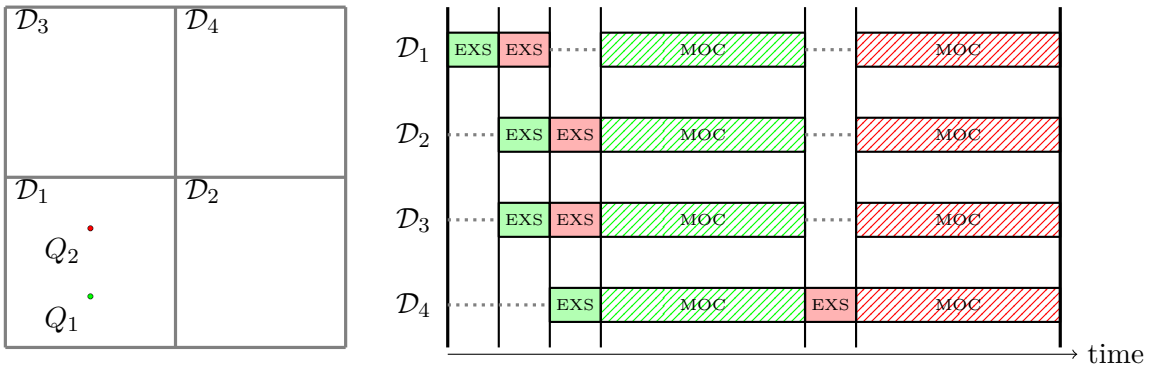
Figure 3.5 shows how the tasks are scheduled in the FCS algorithm. The small filled boxes stand for $\mathrm{EXS}_{i,p}$ tasks while the hatched ones stands for $\mathrm{MOC}_{i,p}$. The box lengths are representative of the time required for their execution. The dotted gray lines denote the idle times, i.e. the time spent by a core without performing any task. Different colors indicate different source points, and the vertical solid lines denote the synchronization points. The three figures represent three possible configurations that can be encountered when performing FCS calculations. The examples are all made of four subdomains. The first one (displayed in Figure 3.5a) contains a single point source located in subdomain $\mathcal{D}_1$. The second (see Figure 3.5b) has two point sources, both contained in the first subdomain. The third contains two sources, one in $\mathcal{D}_1$, and the other in $\mathcal{D}_3$ (see Figure 3.5c).

We can observe how, as the number of sources increases, the number of tasks that are solved at the same time increases as well. This traduces into smaller idle times and, consequently, a better employment of computational resources. In the second model, represented in Figure 3.5b, we see that $\mathrm{MOC}_{i,p}$ tasks are always prioritized over $\mathrm{EXS}_{i,p}$. In fact, $\mathrm{EXS}_{4,2}$ task, computing the equivalent cross section for source Q_2 in $\mathcal{D}_4$, is solved only after the $\mathrm{MOC}_{i,p}$ task has been solved for source Q_1.

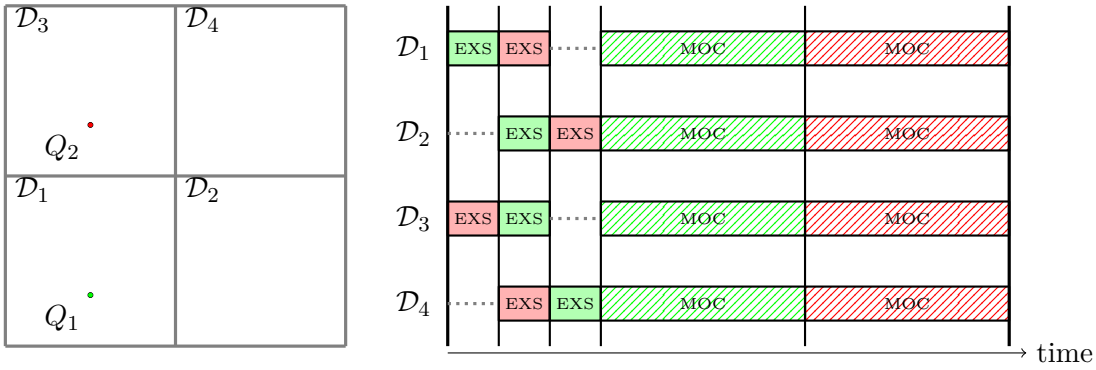
By assuming that the computational times displayed in Figure 3.5 are representative of the actual workload distribution between $\mathrm{MOC}_{i,p}$ and $\mathrm{EXS}_{i,p}$ tasks, we can appreciate the interest of solving all the $\mathrm{MOC}_{i,p}$ tasks at the same time. If this was not the case, the task scheduling diagram for the first model would have been similar to the one displayed in Figure 3.6.



(a) Single source case.



(b) Case with two sources in the same subdomain.



(c) Case with different sources in different subdomains.

Fig. 3.5 Illustration of the task scheduling of FCS in DDM.

3.4 Assessment of the accuracy of the First Collision Source in domain decomposition

The precision of the FCS method in DDM can be adjusted with basically two parameters, i.e. the spacings of the MOC boundary mesh, and the spacings of the EXS boundary mesh. Meanwhile, the accuracy of the results varies with the degree of heterogeneity of the problem

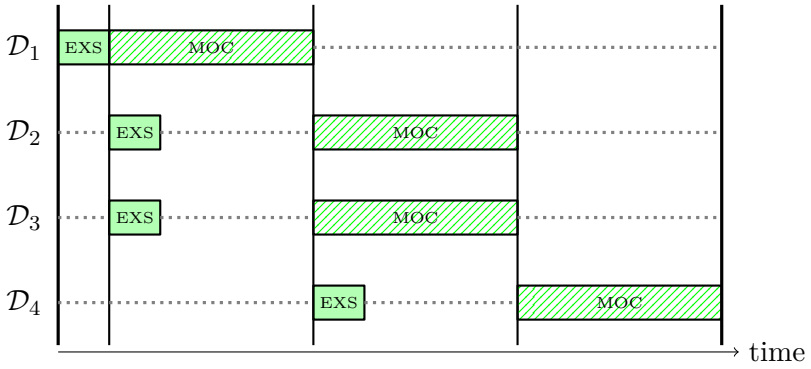


Fig. 3.6 Illustration of an alternative task scheduling strategy for FCS in DDM. Strategy not forcing MOC tasks to be solved simultaneously.

and with the number of subdomains. In this section, we explore these parameters in order to assess the method performances, both in terms of accuracy and calculation time.

We consider for out tests the Case 3 of the Box Test problem, presented in Section 2.7.1. With the first test, we quantify the error introduced by the EXS approximation for different boundary mesh refinements, and for different number of subdomains. Then, we investigate the impact of the medium heterogeneity level on the approximation, and finally we analyse the FCS method capabilities of reducing the computation time at increasing number of cores.

3.4.1 Quantification of the error introduced by the Equivalent Cross Section approximation

Given the repeated-pattern nature of Box Test Case 3, we subsequently decompose the problem geometry into subdomains by cutting each side of the domain into N parts. We study the problem for $N = 1, 2, 3, 4, 5, 6, 7, 8, 9$, i.e. a domain decomposition which ranges from 1 up to 729 distinct subdomains. The source is located in the bottom-left-front corner of the first subdomain.

The aim of the test is to quantify the error related to the EXS approximation only, and so excluding the error introduced by the MOC integration. To this extent, we compare results of simulations having an equal MOC discretization in order to exclude the MOC integration error. Since the decomposition into subdomains induces different trajectory-based discretizations, we had to restrict our analysis to fluxes measured in a limited portion of the domain, preserving the same MOC discretization for different domain decompositions. For geometric reasons, our choice falls on the $3 \times 3 \times 3$ regions cube located in the opposite corner with respect to the source, that is, in the top-right-back corner. It can be seen that, by splitting the domain into $N \times N \times N$ equal cubes (with $N \leq 9$), this zone, being in a corner, will never be subdivided into multiple subdomains (refer to Figure 2.6). Therefore, for an equal spacing of the MOC boundary mesh, even changing the domain decomposition,

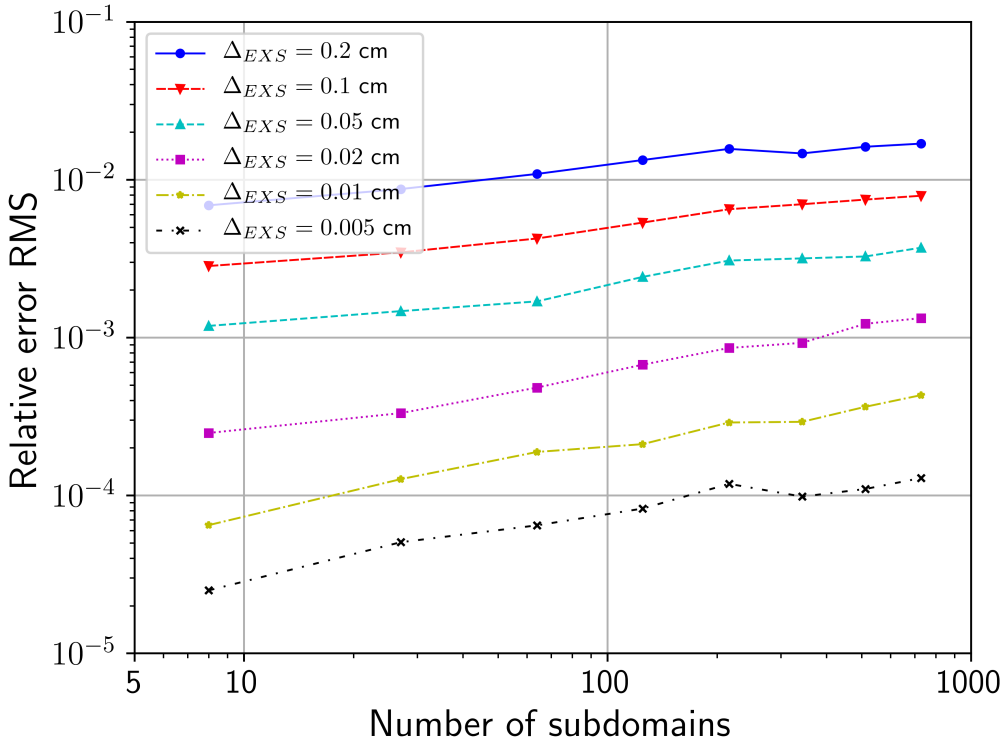


Fig. 3.7 Box Test Case 3, analysis of the EXS approximation accuracy at varying numbers of subdomains, and varying EXS boundary discretizations. Monitoring the relative error RMS introduced by the EXS approximation in the $3 \times 3 \times 3$ cells box located in the top-right-back domain corner. The error RMS is monitored for different domain decompositions and different spacing Δ_{EXS} of the EXS boundary mesh.

it presents the same trajectory discretization, so an equal MOC integration error. All calculations are run using a constant harmonic approximation over the MOC solid angles for the trajectory weights calculation (represented with Γ-0 in Section 2.7.1) and a trajectory spacing of 0.02 cm.

By measuring the uncollided flux in these $3 \times 3 \times 3$ regions we have evaluated the relative error RMS according to Eq. (2.60). The reference result is obtained using the same MOC discretization and without domain decomposition. We estimated the error involved by the MOC integration in the zone of interest to be of the order of 10^{-5}. This estimation has been performed by running the same reference calculation with different spacings for the MOC discretization.

Figure 3.7 displays the relative error RMS for different domain decompositions and different EXS mesh spacings Δ_{EXS}. We notice that the refinement of the EXS boundary mesh has a significant impact on the EXS approximation error. In our test, the error approximatively reduces of an order of magnitude every time the number of trajectories

increases of about a factor 20. The error variation is instead much smoother with respect to the increase of the number of subdomains.

We remark that the purple dotted line is obtained using a spacing Δ_{EXS} equal to the one of the MOC boundary mesh. By comparing MOC and EXS errors for the same spacing, we notice how MOC integration has introduced an error from 1 to 2 orders of magnitude smaller that the one of the EXS approximation. This gives further credit to our choice of separating the evaluation of Σ_p^{eq} from the MOC integration, allowing for a better error control.

3.4.2 Evaluation of the FCS sensitivity with respect to the problem heterogeneity

We want now to explore effectiveness of FCS as the media degree of heterogeneity varies. We consider the Box Test Case 3 subdivided into $9 \times 9 \times 9$ subdomains. Several calculations are performed using different spacing Δ_{EXS}, and progressively varying the media properties of the problem. We recall that the model is constructed by repeating a certain number of times the pattern made by two centred cubes. The total cross section of the outer cube is kept equal to $2 \cdot 10^{-3}$ cm $^{-1}$ while the total cross section of the inner cube is progressively increased. In particular, by defining δ as the ratio of the cross section of the inner cube over the outer one, we repeat the test for $\delta = 2,\ 20,\ 200,\ 2000$.

We monitor the relative error RMS in the top-right-back zone. Its trend with respect to the EXS boundary mesh refinement level is shown in Figure 3.8. We observe that, as the heterogeneity degree of the problem increases, the EXS approximation error also increases. The effect of heterogeneity is particularly relevant, leading to errors up to 3 orders of magnitude larger for case $\delta = 2000$ than in case $\delta = 2$.

3.4.3 Quantification of the FCS parallel performances in domain decomposition

As seen Section 3.3.5, the parallel performance of FCS may depend on many factors. Among them, we recall the number of sources, their distribution through different subdomains, the refinement of MOC and EXS boundary meshes, the number of subdomains, and how they are associated to the different computer nodes. A complete analysis of the computational performances of the method should therefore take place for different values of these parameters.

Here, we propose a concise analysis of the parallel performances of the algorithm. The objective is to observe how, as the number of cores and subdomains varies, the computation times and the speedup change.

The analysis is conduced on the Box Test Case 3 decomposing it into $N \times N \times N$ subdomains, with $N = 1, 2, 3, 4, 5, 6, 7, 8, 9$. The spacing of the MOC boundary mesh is set equal to 0.02 cm. Three different discretizations of the EXS boundary mesh are used: $\Delta_{EXS} = 0.2,\ 0.02,\ 0.002$ cm, i.e. respectively 10 times, equal, and 1 tenth of the MOC trajectory spacing. Simulations are performed on a cluster using AMD EPYC Milan 7763

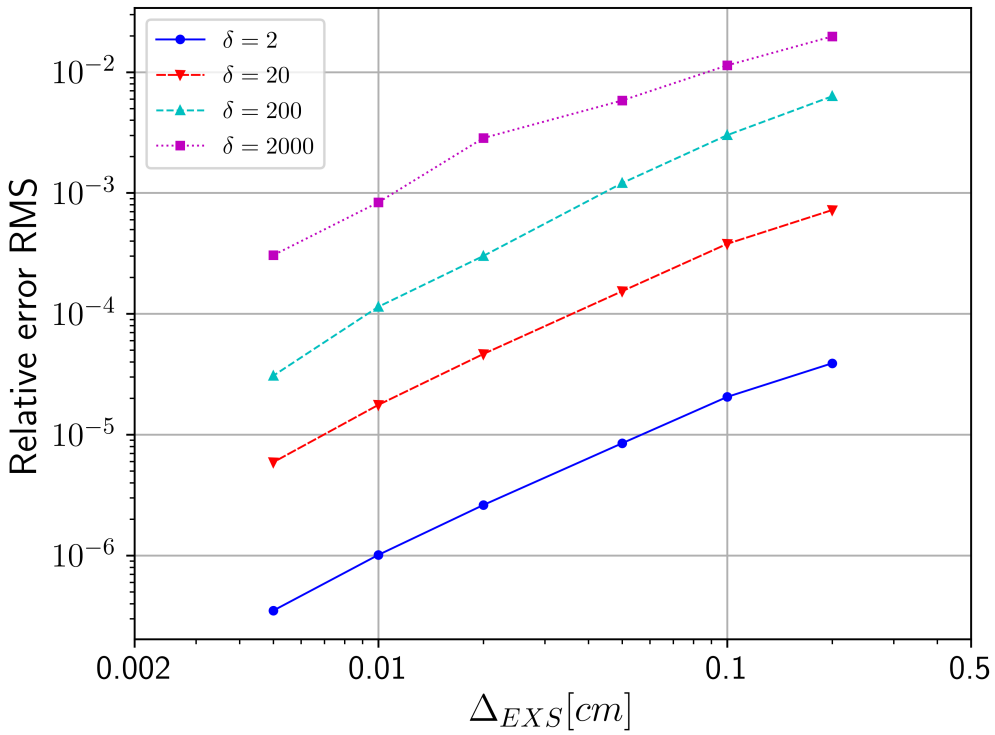


Fig. 3.8 Box Test Case 3, analysis of the EXS approximation accuracy at varying EXS boundary discretizations, and varying heterogeneity degrees. Monitoring the relative error RMS introduced by the EXS approximation in the $3 \times 3 \times 3$ box located in the top-right-back domain corner. Subdomain discretization made of $9 \times 9 \times 9$ subdomains. The error RMS is monitored for different spacing Δ_{EXS} of the EXS boundary mesh, and different values of the ratio δ between the total cross section of inner cubes, and the total cross section of outer ones.

CPUs. The problem is solved using a number of cores equal to the number of subdomains. No more than 24 subdomains are assigned to the same cluster node (a maximum of 24 cores are used in each node).

Figure 3.9 shows the results of the analysis. On the left the computation time is displayed with respect to the number of subdomains. The different contributions to the total computational cost, coming from $\text{MOC}_{i,p}$, $\text{EXS}_{i,p}$, and communications between processes are distinguished. The graphs on the right show instead the speedup factor, evaluated as in Eq. (1.78). The speedup has been computed for the total calculation time, and also for the partial contributions related to the two tasks, namely $\text{MOC}_{i,p}$ and $\text{EXS}_{i,p}$. In order to have a fair estimation of the speedup, the reference calculation solving for the problem not decomposed (1 subdomain, 1 core) has to perform both $\text{EXS}_{i,p}$ and $\text{MOC}_{i,p}$ tasks (no need of the EXS approximation while solving without DDM).

We observe in Figure 3.9a that, running with an EXS mesh coarser than the MOC one, the computational cost of the calculation is almost entirely due to the MOC integration. Since $\text{MOC}_{i,p}$ tasks are solved by the cores all at the same time, and the cost of $\text{EXS}_{i,p}$ is negligible, the algorithm shows a very high speedup factor (see Figure 3.9b).

On the contrary, while using a fine refinement of the EXS boundary mesh, the cost of the entire calculation is dominated by $\text{EXS}_{i,p}$ (Figure 3.9e). Since $\text{EXS}_{i,p}$ tasks are sequentially solved following the propagation front, the speed-up factor associated to $\text{EXS}_{i,p}$ increases mildly with the number of cores. Since the computational cost of simulation is dominated by $\text{EXS}_{i,p}$ calculations, the low speedup of $\text{EXS}_{i,p}$ traduces into low parallel performance of the entire calculation.

The calculation cost due to the MPI communications between nodes is a secondary contribution to the overall cost. It grows as the refinement of the EXS boundary mesh increases, consistently with the increase in the number of points in which Σ_p^{eq} has to be exchanged. The cost of communications remains almost constant as the number of subdomains increases. Other computational costs attributed to the scheduling algorithm are of the order of milliseconds, therefore not represented in Figure 3.9.

In conclusion, the parallel performances of the FCS method in DDM are strictly correlated to the degree of refinement of the EXS boundary mesh.

3.5 Partial Conclusion

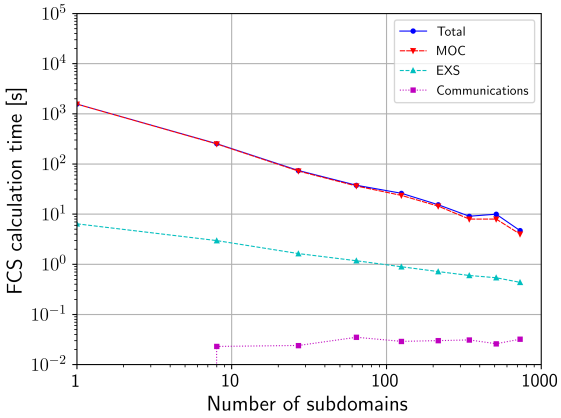
In this chapter we have developed a FCS method allowing for the calculation of the uncollided flux in a spatial domain decomposition framework. This is based on the use of an interpolative approximation (EXS) and equivalent cross sections, in order to evaluate the flux at the subdomains boundaries. The method uses two different boundary meshes, and two different ray tracing techniques: one for the integration of the uncollided flux in the subdomains volumes, and the other for the evaluation of the fluxes at the outgoing interfaces. The FCS method relies on a effective scheduling algorithm allowing to perform calculations on parallel distributed-memory architectures, and enabling to solve for different point sources simultaneously.

Our tests have shown how the method allows for an effective control of the error introduced by the EXS approximation. This error increases with the degree of heterogeneity of the problem, and can be mitigated by refining the EXS approximation, at the expense of a higher computational cost. Tests have revealed how the error associated to the EXS approximation has a reduced sensitivity to the number of subdomains involved in the domain decomposition. Further, the parallel efficiency varies with the refinement of the MOC and the EXS boundary meshes. In particular, solving for a single source point, the speedup factor tends to decrease as the refinement of the EXS boundary mesh gets finer than the refinement of the MOC

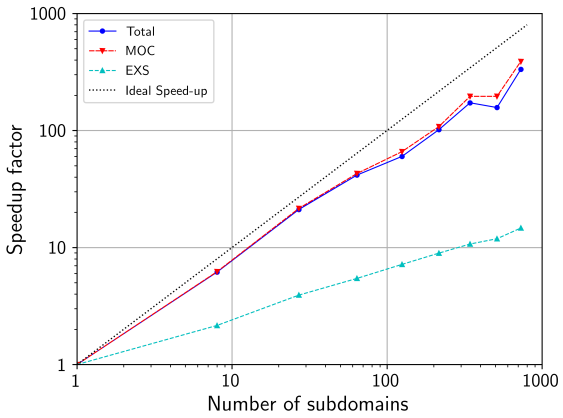
boundary mesh. Nevertheless, it has been qualitatively demonstrated that higher parallel performances are achieved in the presence several point sources.

The FCS method we have developed is based on the MOC-BQ integration technique. However, the method can be adapted to use other integration strategies, such as TS-MOC and QP methods. At present, the FCS method is not able to account for boundary conditions different from vacuum ones. This topic is left to further investigations.

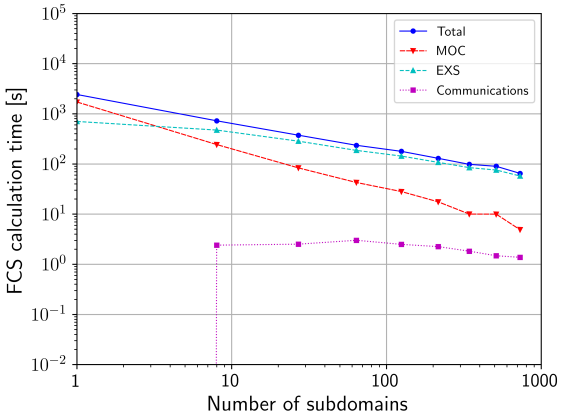
The tests we have performed up to this point are related to the calculation of the uncollided flux only. However, FCS calculations are made of two steps, and both of them have to be solved using DDM. In Chapter 4 we will deal with the angular discretization and the mitigation of the ray-effect in the second step, i.e. calculation of the collided flux. In Chapter 5, instead, we will deal with the complete solution of transport problems in domain decomposition with FCS.



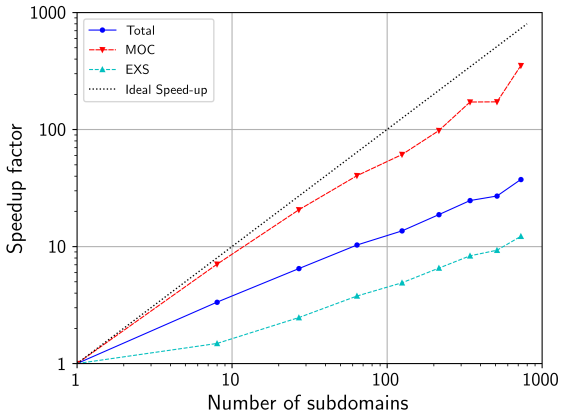
(a) Calculation Time, $\Delta_{EXS} = 0.2$ cm



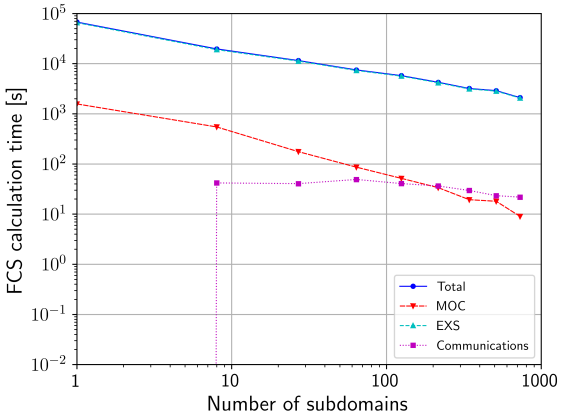
(b) Speedup factor, $\Delta_{EXS} = 0.2$ cm



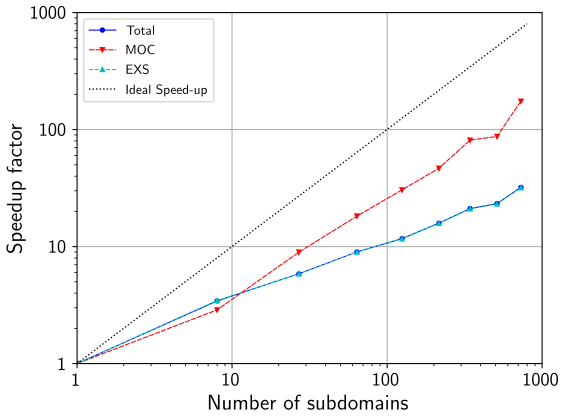
(c) Calculation Time, $\Delta_{EXS} = 0.02$ cm



(d) Speedup factor, $\Delta_{EXS} = 0.02$ cm



(e) Calculation Time, $\Delta_{EXS} = 0.002$ cm



(f) Speedup factor, $\Delta_{EXS} = 0.002$ cm

Fig. 3.9 Box Test Case 3, evaluation of calculation time and speedup factor of the FCS method in DDM. Tests run for different refinements of the EXS boundary mesh. On the left: representation of the calculation time with respect to the number of subdomains. On the right: representation of the speed-up factor with respect to the number of subdomains.

Chapter 4

Development of a new angular discretization method

In physical problems dominated by particle streaming, such as radiation shielding calculations, discrete-ordinates solutions may be affected by the use of an insufficiently refined angular discretization, causing the so-called ray-effect to appear. Even though FCS method is employed, this act on the uncollided component of the angular flux only. Hence, FCS does not provide any ray-effect mitigation on scattering sources. In this chapter we tackle this problem by conceiving an alternative angular discretization method for particle streaming problems, possibly less affected by the ray-effect, and more accurate than Discrete Ordinates in particle streaming calculations.

In the following, we will introduce the problem of lack of angular discretization in shielding calculations with the help of examples. We will present a possible remedy to this, which has led us to the development of the Multi-P_N (MP_N) angular discretization method. This technique is based on the angular expansion of interface fluxes, within the short-characteristic formulation of the transport equation. We will present the MP_N approximation, and we will provide a detailed derivation of the MP_N equations. Then, we will discuss its main features, and describe its implementation in IDT. A comparison between the MP_N and the S_N methods is also provided, together with a series of tests. We will propose a verification of MP_N, together with an evaluation of its capability of mitigating the ray-effect. We will quantify the accuracy of the new angular discretization and estimate the order of convergence of the error, as compared with S_N. Then, we will discuss the computational cost of MP_N, and how it varies depending on the problem characteristics. Finally, we will draw some conclusions.

4.1 Considerations over the application of the IDT solver to radiation shielding calculations

In order to better understand the effect of an insufficient angular refinement in Discrete Ordinates calculations we have considered a problem representative of a radiation shielding study. The model is called Prob500, and is taken from the test suite of the Tripoli-4 Monte Carlo code, [4]. Here we are performing a quite simple qualitative analysis. However, since the problem is quite demanding in terms of computational resources, a more detailed analysis requires the use of Domain Decomposition. Thus, a more extensive study is postponed to Chapter 5.

4.1.1 Illustration of ray-effect via the Prob500 benchmark problem

The Prob500 model, depicted in Figure 4.1, is a parallelepiped domain containing air. A concrete wall is located in the top region (in gray). On the bottom side of the domain, aligned along the X-axis, there are 5 detectors (green) and a Co-60 source (red). Source and detectors are separated by a lead shield (blue), which is meant to avoid photons to directly reach the detector. The Prob500 model presents a reflective boundary condition on the Y-side, while vacuum boundary conditions are elsewhere applied. The problem is solved using macroscopic cross sections generated from the ZZ-KASHIL-E70 nuclear data library provided by NEA, [89], and using the Vitamin-B6 42-groups energy discretization. The maximum scattering anisotropy order is set equal to 3. The source emits photons isotropically with a unit intensity at energies equal to $1,17$ MeV and 1.33 MeV, corresponding to groups 22 and 23 of the multi-group discretization.

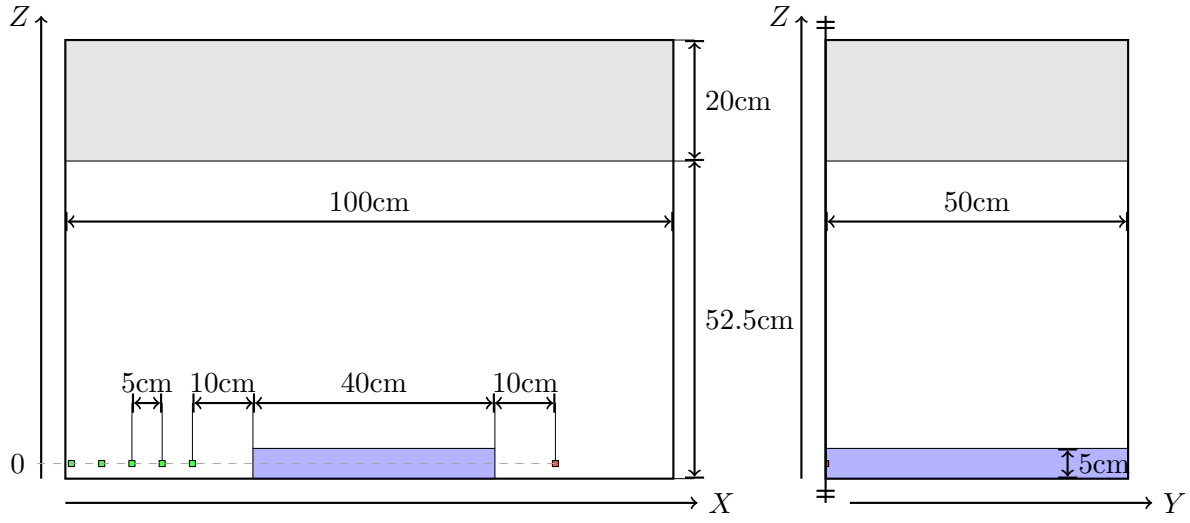
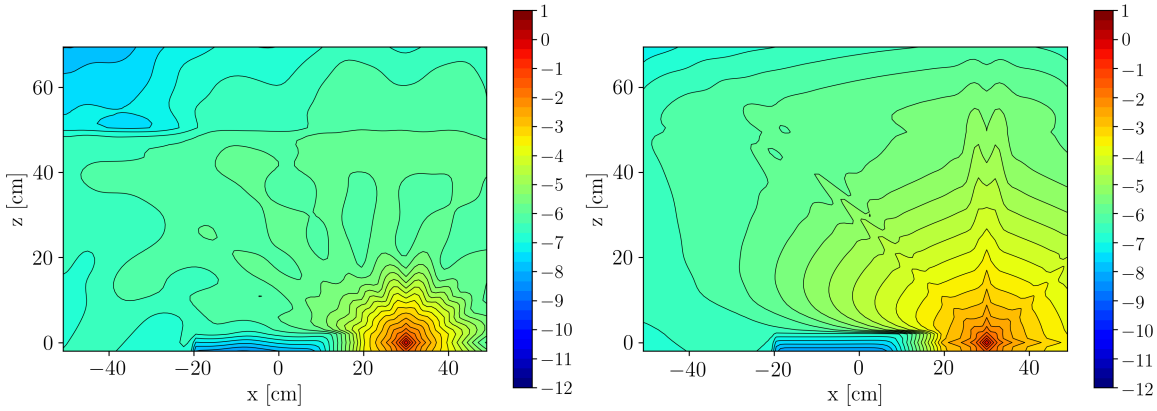


Fig. 4.1 Illustration of the Prob500 problem geometry. Front (left) and lateral (right) views.

In order to stress the importance of ray-effect mitigation we first performed a standard discrete-ordinate calculation using a Chebyshev-Legendre angular quadrature with a triangular arrangement of directions. We solved the problem using an S_8 and an S_{30} quadratures, corresponding to 80 and 960 directions, respectively. Then, we repeated the calculation using the First Collision Source MOC-BQ method and approximating the source using a single source point, centred in the source volume (since the source lies on the symmetry plane, no difficulty is introduced by the presence of reflective boundary conditions). MOC-BQ is run using a Γ-0 weight quadrature and using about 3.2×10^4 trajectories. The calculation for the collided flux is performed using an S_8 angular quadrature.



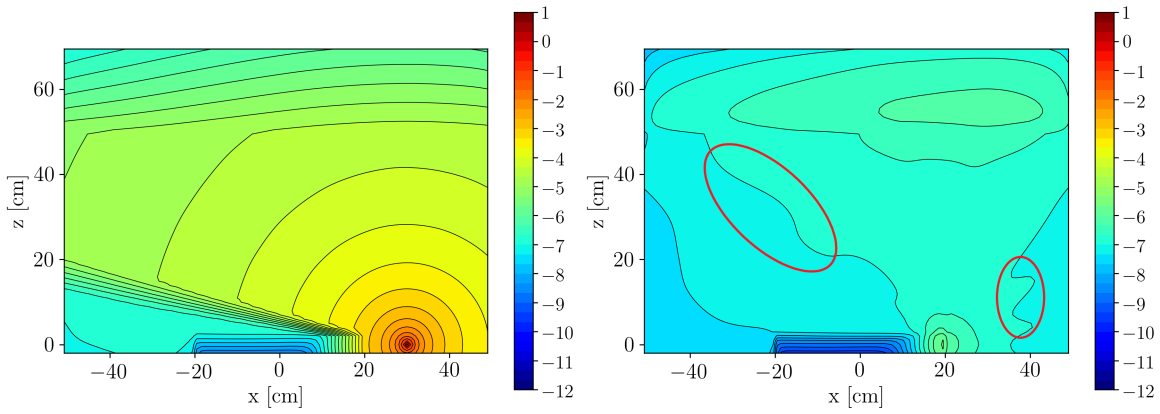
(a) S_8, group 23

(b) S_{30}, group 23

Fig. 4.2 Prob500, representation of ray-effect in S_N calculations. Average scalar flux (log10) over the symmetry plane of the model, in energy group 23. S_N calculation using the S_8 (a) and the S_{30} (b) quadratures.

Figure 4.2 shows the average flux in group 23, corresponding to the energy range 1MeV-800KeV. It represents the flux profile on the symmetry plane of the model, namely the X-Z plane passing through the source point. At this energy, the largest contribution to the flux, in most of the domain, is provided by the uncollided component, i.e. photons which are directly emitted by the source. As expected, ray-effect visibly affects the result of both calculations, causing the distortions of the flux profile. Because of the limited dimension of the source, ray-effect can be hardly mitigated unless a extremely fine quadrature is employed, hence the importance of the First Collision Source method.

Figure 4.3 (a) displays the result of the calculation with FCS and the S8 quadrature. The flux profile appears much smoother in group 23. FCS effectively eliminates the ray-effect generated by photons emitted from the external source. However, we still find traces of ray-effect while looking at lower energies. For example, in group 30 (energy range $400KeV$-$300KeV$), the external source is not emitting. The only contribution to flux comes from particles re-emitted by scattering (see Figure 4.3 (b)). At this energy, photons are scattered



(a) S_8, group 23

(b) S_8, group 30

Fig. 4.3 Prob500, representation of ray-effect in S_N calculations with FCS. Average scalar flux (log10) over the symmetry plane of the model, in energy groups 23 and 30. S_N calculation using the S_8 plus MOC-BQ.

for the most by the upper wall, and by the lead right-surface layer. The resulting flux profile shows some unphysical oscillations (circled in red) which are typical of ray-effect.

We see that, even if dealing with a very simple case and using the FCS method, the ray-effect may still afflict the numerical results of the simulation. As a consequence, the use of an adequate angular discretization is still required.

4.1.2 Definition of a ray-effect mitigation strategy

Although some authors have shown how some angular quadratures are more effective than others in countering ray-effect (see Section 1.3.1), Discrete Ordinates present some intrinsic limits in accounting for particle streaming. Since each problem is different, a single angular quadrature will never be able to integrate all possible angular distributions of the angular flux. In this sense, we believe that the only effective technique for mitigating ray-effect with S_N is the use of adaptive quadrature formulas.

In this research, we opted for a different approach, based on a projective angular discretization. Our idea is to retain the short-characteristics formalism of the transport equation, while enriching the transmission equation by the introduction of an angular expansion of the interface fluxes over a set of partial-range functions. By doing so, we aim to obtain an angular discretization able to represent streaming effects more faithfully, and thus less prone to ray-effects. At the same time, in order to limit the computational cost, we want to avoid system of equations describing a full coupling between all angular degrees of freedom. To this extent, interface fluxes are represented using a piecewise continuous basis. As an advantage, such bases lend themselves to adaptive discretizations much better

than Discrete Ordinates. Moreover, piecewise continuous angular representations are better suited to describe possible discontinuities of the angular flux, [90], with respect to methods expanding the flux over continuous functions, such as P_N.

The angular discretization method we are proposing is called Multi-P_N (MP_N), [91, 92]. It takes its name from the set of equations that it solves, which recalls the one of the DP_N method, [32]. However, the system of equations is solved independently for different portions of the angular domain, and not only for the positive and the negative hemispheres, hence the suffix "Multi". In the following section, we present the method, and we provide a detailed mathematical derivation. However, before going any further, we want to make a premise. The developments presented here have been conceived under the hypothesis of homogeneous material in each region. In spite of this, we do not expect any difficulty in extending MP_N to the heterogeneous Cartesian regions. The MP_N formulation is very similar to the Multiple P_N Surface Integral Transport method designed by J. Stepanek, [93]. The method described by the author is based on an expansion of boundary fluxes over spherical harmonics, and is limited to 2-D geometries. In this respect, the MP_N technique presented here can be seen as an extension of this method to 3-D geometries and using a polynomial angular expansion instead. Furthermore, while noticing a mitigation of the ray-effect, the author focused on the ability of the method to represent the flux angular discontinuities in reactor core calculations. Thus, the method has not been applied to particle streaming calculations. Another technique with the same name has also been designed by Ghazaie *et al.* for the angular discretization of the first-order transport equation [90]. The method, restricted to 2-D geometries, is presented as a generalization of the DP_N method and relies on a piecewise continuous angular expansion of the angular flux.

4.2 Proposition of an angular discretization method based on the piecewise polynomial expansion of the interface fluxes

In the following, we will use symbols $\langle \cdot, \cdot \rangle$ and $(\cdot, \cdot)$ to identify integrations over the boundary surfaces and over the volume of a spatial region, respectively. Blackboard bold (e.g. $\mathbb{A}$) characters are used to distinguish matrices relating together spatial and angular moments from matrices relating moments of a single basis, which are instead identified using the double-barred boldface (e.g. $\overline{\overline{\mathbf{A}}}$) notation.

The first step of the MP_N derivation consists in the definition of a suitable piecewise representation of the interface fluxes over the unit sphere. To this extent, we partition the unit sphere $\mathcal{S}^2 = \{\mathbf{\Omega} \mid \mathbf{\Omega} \in 4\pi\}$ into N_n solid angles, such that

$$\mathcal{S}^2 = \bigcup_{n=1,N_n} \mathcal{S}^2_n, \tag{4.1}$$

and with

$$\Delta\Omega_n = \int_{\mathcal{S}_n^2} d\mathbf{\Omega} \ , \tag{4.2}$$

as the measure of solid angle $\mathcal{S}_n^2$. Then, we define an angular basis $\mathbf{B}^n(\mathbf{\Omega})$ within each solid angle as

$$\mathbf{B}^n(\mathbf{\Omega}) = \left\{B_\ell^n(\mathbf{\Omega})\right\}_{\ell=1,N_\ell} \quad \text{with } \mathbf{\Omega} \in \mathcal{S}_n^2, \tag{4.3}$$

with B_ℓ^n being the element ℓ of the basis, $\ell = 1, \ldots, N_\ell$ as the index of the basis function B_ℓ^n, and N_ℓ as the basis dimension. We refer to $\mathbf{B}^n$ as the local angular basis in order to stress the fact that it is local within the solid angle $\mathcal{S}_n^2$. For the moment, we provide a general derivation of the method by considering $\mathbf{B}_\ell^n$ to be any function. The actual definition will be presented in Section 4.3.

The MP_N approximation consists in representing the interface fluxes by an angular expansion over the space spanned by basis $\mathbf{B}^n$, within each solid angle. Since interface fluxes are already expanded by short-characteristics on the regions surfaces $\mathbf{r} \in \Gamma^{s'}$, the actual interface flux representation becomes

$$\begin{aligned}
\psi(\mathbf{r},\mathbf{\Omega}) &= \psi_n^{s'}(\mathbf{r},\mathbf{\Omega}), \quad \mathbf{\Omega} \in \mathcal{S}_n^2, \ \mathbf{r} \in \Gamma^{s'}, && (4.4)\\
\psi_n^{s'}(\mathbf{r},\mathbf{\Omega}) &= \sum_{k=1,3} \beta_k P_k^{s'}(\mathbf{r}) \sum_{\ell=1,N_\ell} B_\ell^n(\mathbf{\Omega}) \phi_{k,\ell}^{s',n}, && (4.5)\\
\text{for } \mathbf{r} \in \Gamma^{s'} \quad &\text{and} \quad \mathbf{\Omega} \in \mathcal{S}_n^2.
\end{aligned}$$

In contrast to Eq. (1.62), which provides the flux for a single direction, Eq. (4.5) reconstructs the flux over the whole solid angle. Basis functions $B_\ell^n(\mathbf{\Omega})$ are not chosen to be orthogonal to each other, therefore, the N_ℓ angular components associated to a single spatial component k are coupled by

$$\begin{aligned}
\sum_{\ell'=1,N_\ell} \beta_{\ell,\ell'}^{s',n} \boldsymbol{\phi}_{\ell'}^{s',n} &= \left\langle B_\ell^n \mathbf{P}^{s'}, \ \psi \right\rangle_n^{s'} && (4.6)\\
&= \frac{1}{\Delta\Omega_n A_{s'}} \int_{\mathcal{S}_n^2} d\mathbf{\Omega} \ \mu^{s'}(\mathbf{\Omega}) B_\ell^n(\mathbf{\Omega}) \int_{\Gamma^{s'}} d\mathbf{r} \ \mathbf{P}^{s'}(\mathbf{r}) \psi_n^{s'}(\mathbf{r},\mathbf{\Omega}),\\
\text{for } \ell &= 1, \ldots, N_\ell.
\end{aligned}$$

Here, we have used $\boldsymbol{\phi}_\ell^{s',n} = \{\phi_{k,\ell}^{s',n}\}_{k=1,K}$ to indicate the vector of the spatial components associated to the ℓ-th angular component. In Eq. 4.6, the symbol $\langle\cdot,\cdot\rangle_n^{s'}$ stands for the spatial-angular integral over surface $\Gamma^{s'}$ and solid angle $\mathcal{S}_n^2$, whereas $\mu^{s'}$ is the cosine of the surface outgoing normal direction, i.e. $\mu^{s'} = |\mathbf{\Omega} \cdot \mathbf{n}_{s'}|$. Coefficients $\beta_{\ell,\ell'}^{s',n}$ on the LHS of Eq. (4.6), which are defined by

$$\beta_{\ell,\ell'}^{s',n} = \frac{1}{\Delta\Omega_n} \int_{\mathcal{S}_n^2} d\mathbf{\Omega} \ \mu^{s'}(\mathbf{\Omega}) B_\ell^n(\mathbf{\Omega}) B_{\ell'}^n(\mathbf{\Omega}), \tag{4.7}$$

compose the angular mass matrix $\overline{\overline{\mathbf{N}}}_n^{s'} = \{\beta_{\ell,\ell'}^{s',n}\}_{\ell,\ell'=1,N_\ell}$ for indexes (s',n).

4.2.1 Presentation of the MP$_N$ discretization procedure

Since the idea of MP$_N$ is built upon the short-characteristic formalism of IDT (See Section 1.2.4), we consider as starting point for our development the spatially discretized balance and transport equations, Eqs. (1.67) and (1.68). In particular, even though using S$_N$ these are solved for a finite number of direction, Eqs. (1.67) and (1.68) are also valid for any direction $\mathbf{\Omega}$. We can therefore generalize them for a generic direction, namely

$$\boldsymbol{\psi}(\mathbf{\Omega}) = \overline{\overline{\mathbf{C}}}(\mathbf{\Omega})\mathbf{q}(\mathbf{\Omega}) + \sum_{s'\in\Gamma^-(\mathbf{\Omega})} \overline{\overline{\mathbf{I}}}^{s'}(\mathbf{\Omega})\boldsymbol{\psi}^{s'}(\mathbf{\Omega}), \tag{4.8}$$

$$\boldsymbol{\psi}^{s}(\mathbf{\Omega}) = \overline{\overline{\mathbf{E}}}^{s}(\mathbf{\Omega})\mathbf{q}(\mathbf{\Omega}) + \sum_{s'\in\Gamma^-(\mathbf{\Omega})} \overline{\overline{\mathbf{T}}}^{s,s'}(\mathbf{\Omega})\boldsymbol{\psi}^{s'}(\mathbf{\Omega}) \quad \text{for } s\in\Gamma^+(\mathbf{\Omega}). \tag{4.9}$$

In order to discretize the angular variable in Eqs. (4.8) and (4.9), and so derive MP$_N$, we use a standard Galerkin projection technique. The procedure is similar to the one performed for the spatial variable while deriving the short-characteristics equations in Section 1.2.4. It consists in replacing the continuous angular source term $\mathbf{q}$ and interface fluxes $\boldsymbol{\psi}^{s'}$ using finite angular expansions, and then projecting the balance and the transmission on the basis used to expand the source and the interface fluxes, respectively. To do so, we first need to introduce the two bases. For the interface fluxes, the choice falls on the early-defined piecewise continuous angular basis. Rearranging together the spatial moments in Eq. (4.5), the angular expansion of the interface flux can be written as

$$\boldsymbol{\psi}^{s'}(\mathbf{\Omega}) = \boldsymbol{\psi}_n^{s'}(\mathbf{\Omega}) = \sum_{\ell'=1,N_\ell} B_{\ell'}^n(\mathbf{\Omega})\boldsymbol{\phi}_{\ell'}^{s',n} \quad \text{for } \mathbf{\Omega}\in\mathcal{S}_n^2 \text{ and } n=1,\dots,N_n, \tag{4.10}$$

where $\boldsymbol{\phi}_{\ell'}^{s',n} = \{\phi_{k,\ell'}^{s',n}\}_{k=1,K}$. As for the source term representation, for consistency with the scattering kernel, our choice falls on the finite-dimensional spherical harmonic basis $\mathbf{A}_H$, leading to expansion

$$\mathbf{q}(\mathbf{\Omega}) = \sum_{h'=1,H} A_{h'}(\mathbf{\Omega})\mathbf{q}_{h'}, \tag{4.11}$$

where $\mathbf{q}_{h'} = \{q_{h',m}\}_{m=1,M}$ contains the spatial-angular moments of the source, for index m. We remark that while basis $\mathbf{A}_H$ provides a continuous representation of the source over 4π, interface fluxes are defined as piecewise continuous. In other words, each of the basis $\mathbf{B}^n$ has a support which is limited to the solid angle rather than on the unit sphere.

4.2.2 Derivation of the MP$_N$ transmission equation

In order to obtain the MP$_N$ transmission equation, we introduce the angular expansion of the source, Eq. (4.11), and MP$_N$ approximation, Eq. (4.10), into Eq. (4.9), as

$$\begin{aligned} \boldsymbol{\psi}_n^s(\boldsymbol{\Omega}) &= \overline{\overline{\mathbf{E}}}^s(\boldsymbol{\Omega}) \sum_{h'=1,H} A_{h'}(\boldsymbol{\Omega})\mathbf{q}_{h'} + \sum_{s'\in\Gamma^-(\boldsymbol{\Omega})} \overline{\overline{\mathbf{T}}}^{s,s'}(\boldsymbol{\Omega}) \sum_{\ell'=1,N_\ell} B_{\ell'}^n(\boldsymbol{\Omega})\boldsymbol{\phi}_{\ell'}^{s',n} \ . \quad (4.12) \\ \text{for } \boldsymbol{\Omega} &\in \mathcal{S}_n^2 \ \text{ and } n = 1,\ldots,N_n. \end{aligned}$$

Then we project the angularly-continuous transmission equation (4.12) on the local basis $\mathbf{B}^n(\boldsymbol{\Omega})$ over $\mathcal{S}_n^2$, obtaining

$$\begin{aligned} \sum_{\ell'} \beta_{\ell,\ell'}^{s,n}\boldsymbol{\phi}_{\ell'}^{s,n} &= \langle B_\ell^n \ , \ \boldsymbol{\psi}_n^s\rangle_n^s = \frac{1}{\Delta\Omega_n}\int_{\mathcal{S}_n^2} d\boldsymbol{\Omega}\ \mu^s(\boldsymbol{\Omega})B_\ell^n(\boldsymbol{\Omega})\boldsymbol{\psi}_n^s(\boldsymbol{\Omega}) \quad (4.13) \\ &= \sum_{h'=1,H} \mathbb{E}_{n,\ell,h'}^s\mathbf{q}_{h'} + \sum_{s'\in\Gamma_n^-}\sum_{\ell'=1,N_\ell} \mathbb{T}_{n,\ell,\ell'}^{s,s'}\boldsymbol{\phi}_{\ell'}^{s',n}, \\ \text{for } \ell &= 1,\ldots,N_\ell \ \text{ and } n = 1,\ldots,N_n, \end{aligned}$$

with the MP$_N$ escape matrix, defined by the integral

$$\begin{aligned} \mathbb{E}_{n,\ell,h'}^s &= \langle B_\ell^n\ \mathbf{P}^s,\ \mathcal{K}\mathbf{P}A_{h'}\rangle_n^s = \frac{1}{\Delta\Omega_n}\int_{\mathcal{S}_n^2} d\boldsymbol{\Omega}\ \mu^s(\boldsymbol{\Omega})B_\ell^n(\boldsymbol{\Omega})\overline{\overline{\mathbf{E}}}^s(\boldsymbol{\Omega})A_{h'}(\boldsymbol{\Omega}) \quad (4.14) \\ \text{for } \ell &= 1,\ldots,N_\ell \ \text{ and } h' = 1,\ldots,H, \end{aligned}$$

representing the coupling between the spatial-angular moments of the source and the spatial-angular moments of the outgoing flux on surface s, and the transmission matrix,

$$\begin{aligned} \mathbb{T}_{n,\ell,\ell'}^{s,s'} &= \left\langle B_\ell^n\mathbf{P}^s,\ e^{-\Sigma t}\mathbf{P}^{s'}B_{\ell'}^n\right\rangle_n^s = \frac{1}{\Delta\Omega_n}\int_{\mathcal{S}_n^2} d\boldsymbol{\Omega}\ \mu^s(\boldsymbol{\Omega})B_\ell^n(\boldsymbol{\Omega})\overline{\overline{\mathbf{T}}}^{s,s'}(\boldsymbol{\Omega})B_{\ell'}^n(\boldsymbol{\Omega}) \quad (4.15) \\ \text{for } \ell,\ell' &= 1,\ldots,N_\ell \ , \end{aligned}$$

giving the coupling between the spatial-angular moments of the incoming flux on surface s' and the spatial-angular moments of the outgoing flux on surface s.

By collecting together the components of the angular expansion into vectors $\boldsymbol{\phi}_n^s = \{\boldsymbol{\phi}_\ell^{s,n}\}_{\ell=1,N_\ell}$ and $\mathbf{q} = \{\mathbf{q}_h\}_{h=1,H}$, we can rearrange the MP$_N$ transmission equation (4.13) to explicitly solve for all components ℓ in $\boldsymbol{\phi}_n^s$, as

$$\begin{aligned} \boldsymbol{\phi}_n^s &= \mathbb{E}_n^s\mathbf{q} + \sum_{s'\in\Gamma_n^-} \mathbb{T}_n^{s,s'}\boldsymbol{\phi}_n^{s'}, \quad (4.16) \\ \text{for } n &= 1,\ldots,N_n, \ \text{and } s\in\Gamma_n^+, \end{aligned}$$

with

$$\Gamma_n^{\pm} = \left\{ \bigcup \Gamma^{\pm}(\mathbf{\Omega}),\ \forall \mathbf{\Omega} \in \mathcal{S}_n^2 \right\}.$$

In Eq. (4.16), the escape and transmission matrices are normalized by the inverse of the spatial-angular mass matrix, which is defined for each solid angle n and surface s by the tensor product

$$\begin{aligned} \mathbb{N}_n^s &= \overline{\overline{\mathbf{N}}}_n^s \otimes \overline{\overline{\mathbf{1}}}_{K\times K} = \left\{ \beta_{\ell,\ell'}^{s,n}\ \overline{\overline{\mathbf{1}}}_{K\times K} \right\}_{\substack{\ell=1,N_\ell \\ \ell'=1,N_\ell}} \\ &= \begin{bmatrix} \beta_{1,1}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} & \beta_{1,2}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} & \cdots & \cdots & \cdots & \beta_{1,N_\ell}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} \\ \beta_{2,1}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} & \beta_{2,2}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} & \cdots & \cdots & \cdots & \beta_{2,N_\ell}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} \\ \vdots & \vdots & \ddots & & & \vdots \\ \beta_{\ell,1}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} & \beta_{\ell,2}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} & & \beta_{\ell,\ell'}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} & & \beta_{\ell,N_\ell}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} \\ \vdots & \vdots & & & \ddots & \vdots \\ \beta_{N_\ell,1}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} & \beta_{N_\ell,2}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} & & & & \beta_{N_\ell,N_\ell}^{s,n}\overline{\overline{\mathbf{1}}}_{K\times K} \end{bmatrix}, \end{aligned} \tag{4.17}$$

with its inverse,

$$[\mathbb{N}_n^s]^{-1} = [\overline{\overline{\mathbf{N}}}_n^s]^{-1} \otimes \overline{\overline{\mathbf{1}}}_{K\times K},$$

where $\overline{\overline{\mathbf{1}}}_{K\times K}$ is the $K \times K$ identity matrix. The final escape and transmission matrices in Eq. (4.16) are thus defined as

$$\mathbb{E}_n^s = [\mathbb{N}_n^s]^{-1} \left\{ \mathbb{E}_{n,\ell,h'}^s \right\}_{\substack{\ell=1,N_\ell \\ h'=1,H}} \quad \text{and } \mathbb{T}_n^{s,s'} = [\mathbb{N}_n^s]^{-1} \left\{ \mathbb{T}_{n,\ell,\ell'}^{s,s'} \right\}_{\substack{\ell=1,N_\ell \\ \ell'=1,N_\ell}}, \tag{4.18}$$

respectively.

4.2.3 Derivation of the MP$_N$ balance equation

The derivation of the MP$_N$ balance equation entails the projection on the spherical harmonics base $\mathbf{A}_H$ over 4π, so it allows for the direct evaluation of the angular moments of the flux,

$$\boldsymbol{\phi}_h = \frac{1}{4\pi V} \int_{4\pi} d\mathbf{\Omega}\ A_h(\mathbf{\Omega}) \int_{\mathcal{T}} d\mathbf{r}\ \mathbf{P}(\mathbf{r})\psi(\mathbf{r},\mathbf{\Omega}). \tag{4.19}$$

Since the interface flux angular expansion Eq. (4.10) spans direction within a single solid angle only, the evaluation of Eq. (4.19) has to account for the partitioning of the unit sphere. Thus, we rewrite Eq. (4.19) as the sum of contributions evaluated in different solid angles,

$$\boldsymbol{\phi}_h = \sum_{n=1,N_n} \boldsymbol{\phi}_h^n, \tag{4.20}$$

where $\boldsymbol{\phi}_h^n$ is the vector of spatial components of the flux within the volume $\mathcal{T}$ and the solid angle $\mathcal{S}_n^2$, contributing on angular moment h, i.e.

$$\begin{aligned} \boldsymbol{\phi}_h^n &= (A_h\mathbf{P}, \psi)_n \\ &= \frac{1}{4\pi V}\int_{\mathcal{S}_n^2} d\boldsymbol{\Omega}\ A_h(\boldsymbol{\Omega}) \int_{\mathcal{T}} d\mathbf{r}\ \mathbf{P}(\mathbf{r})\psi(\mathbf{r},\boldsymbol{\Omega}). \end{aligned} \tag{4.21}$$

Here symbol $(\cdot,\cdot)_n$ stands for the spatial-angular integration over region $\mathcal{T}$ and solid angle $\mathcal{S}_n^2$. The MP$_N$ equation for $\boldsymbol{\phi}_h^n$ is derived by introducing the MP$_N$ interface flux representation, Eq. (4.10), and the angular expansion of the source term, Eq. (4.11), in Eq. (4.8), thus obtaining

$$\begin{aligned} \boldsymbol{\psi}_n(\boldsymbol{\Omega}) &= \overline{\overline{\mathbf{C}}}(\boldsymbol{\Omega}) \sum_{h'=1,H} A_{h'}(\boldsymbol{\Omega})\mathbf{q}_{h'}(\boldsymbol{\Omega}) + \sum_{s'\in\Gamma^-(\boldsymbol{\Omega})} \overline{\overline{\mathbf{I}}}^{s'}(\boldsymbol{\Omega}) \sum_{\ell'=1,N_\ell} B_{\ell'}^n(\boldsymbol{\Omega})\boldsymbol{\phi}_{\ell'}^{s',n}\ , \\ \text{for } \boldsymbol{\Omega} &\in \mathcal{S}_n^2 \ \text{ and } n = 1,\ldots,N_n. \end{aligned} \tag{4.22}$$

Then, by replacing Eq. (4.22) in Eq. (4.21), one obtains the formulation for the spatial-angular moments within region $\mathcal{T}$ and solid angle $\mathcal{S}_n^2$,

$$\begin{aligned} \boldsymbol{\phi}_h^n &= (A_h,\ \boldsymbol{\psi}_n)_n = \frac{1}{4\pi}\int_{\mathcal{S}_n^2} d\boldsymbol{\Omega}\ A_h(\boldsymbol{\Omega})\boldsymbol{\psi}_n(\boldsymbol{\Omega}) \\ &= \sum_{h'=1,H} \mathbb{C}_{n,h,h'}\mathbf{q}_{h'} + \sum_{s'\in\Gamma_n^-}\sum_{\ell'=1,N_\ell} \mathbb{I}_{n,h,\ell'}^{s'}\ \boldsymbol{\phi}_{\ell'}^{s',n} \\ \text{for } h &= 1,\ldots,H \ \text{ and } n = 1,\ldots,N_n. \end{aligned} \tag{4.23}$$

The resulting formulation is the MP$_N$ balance equation for solid angle $\mathcal{S}_n^2$ and harmonic h. The flux angular moments are evaluated solving Eq. (4.23) for all solid angles, and summing their contributions according to Eq. (4.20). In Eq. (4.23), the MP$_N$ collision matrix, defined by the integral

$$\begin{aligned} \mathbb{C}_{n,h,h'} &= (A_h\mathbf{P},\ \mathcal{K}\mathbf{P}A_{h'})_n = \frac{1}{4\pi}\int_{\mathcal{S}_n^2} d\boldsymbol{\Omega}\ A_h(\boldsymbol{\Omega})\overline{\overline{\mathbf{C}}}(\boldsymbol{\Omega})A_{h'}(\boldsymbol{\Omega}) \\ \text{for } h,h' &= 1,\ldots,H, \end{aligned} \tag{4.24}$$

describes the coupling between the spatial-angular components of the source and the spatial-angular moment of the flux within $\mathcal{T}$ and solid angle $\mathcal{S}_n^2$. Since $\mathbf{q}_h$ does not depend on discrete solid angles, we are able to compute the contribution of collisions to the flux by the global collision matrix $\mathbb{C}_{h,h'}$, which is given by the sum of the partial contributions $\mathbb{C}_{n,h,h'}$,

i.e.

$$\begin{aligned} \mathbb{C}_{h,h'} &= \sum_{n=1,N_n} \mathbb{C}_{n,h,h'} = \frac{1}{4\pi}\int_{4\pi} d\mathbf{\Omega}\ A_h(\mathbf{\Omega})\overline{\overline{\mathbf{C}}}(\mathbf{\Omega})A_{h'}(\mathbf{\Omega}) \\ \text{for } h,h' &= 1,\dots,H. \end{aligned} \tag{4.25}$$

The contribution of the incoming flux on surface s' to the angular-spatial moments of the flux is instead represented by the MP_N incoming matrix, which is defined as

$$\begin{aligned} \mathbb{I}^{s'}_{n,h,\ell'} &= (A_h\mathbf{P},\ e^{-\Sigma t}\mathbf{P}^{s'}B^n_{\ell'})_n = \frac{1}{4\pi}\int_{\mathcal{S}^2_n} d\mathbf{\Omega}\ A_h(\mathbf{\Omega})\overline{\overline{\mathbf{I}}}^{s'}(\mathbf{\Omega})B^n_{\ell'}(\mathbf{\Omega}) \\ \text{for } h &= 1,\dots,H \ \text{ and } \ell' = 1,\dots,N_\ell. \end{aligned} \tag{4.26}$$

As done for (4.16), by collecting the flux angular components $\boldsymbol{\phi}^n_h$ into vector $\boldsymbol{\phi}^n = \{\boldsymbol{\phi}^n_h\}_{h=1,H}$, and defining

$$\mathbb{C}_n = \left\{\mathbb{C}_{n,h,h'}\right\}_{\substack{h=1,H\\h'=1,H}}\ , \ \mathbb{I}^{s'}_n = \left\{\mathbb{I}^{s'}_{n,h,\ell'}\right\}_{\substack{h=1,H\\\ell'=1,N_\ell}}, \tag{4.27}$$

we are able to cast Eq. (4.23) in its final form, solving for all angular components within solid angle $\mathcal{S}^2_n$, thus obtaining

$$\begin{aligned} \boldsymbol{\phi}_n &= \mathbb{C}_n\mathbf{q} + \sum_{s'\in\Gamma^-_n}\mathbb{I}^{s'}_n\boldsymbol{\phi}^{s'}_n. \\ \text{for } n &= 1,\dots,N_n. \end{aligned} \tag{4.28}$$

Finally, by summing the contribution of different solid angles using Eq. (4.20), and factorizing the collision contribution as in Eq. (4.25), we define the MP_N balance equation for the spatial-angular components on 4π,

$$\begin{aligned} \boldsymbol{\phi} &= \sum_{n=1,N_n}\boldsymbol{\phi}_n \\ &= \mathbb{C}\mathbf{q} + \sum_{n=1,N_n}\sum_{s'\in\Gamma^-_n}\mathbb{I}^{s'}_n\boldsymbol{\phi}^{s'}_n \end{aligned} \tag{4.29}$$

with $\mathbb{C} = \sum_{n=1,N_n}\mathbb{C}_n$, and $\boldsymbol{\phi} = \{\boldsymbol{\phi}_h\}_{h=1,H}$.

4.2.4 Consideration over the MP_N method

The MP_N Eqs. (4.16) and (4.28) formally present the same structure of short-characteristics equations using the standard S_N method, i.e. Eqs. (1.68) and (1.67), respectively.

The MP_N transmission couples moments of interface fluxes for the same solid angle $\mathcal{S}^2_n$. Since no coupling is present between different angular elements, the MP_N system of

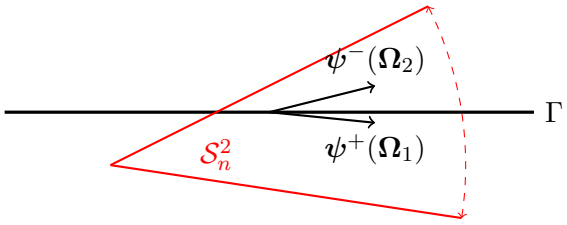


Fig. 4.4 Illustration of the difficulties related to the distinction between incoming and outgoing interfaces in MP_N.

equations retains the block diagonal structure of the short-characteristics streaming-plus-removal operator, so allowing to solve each solid angle independently. However, while in S_N each block can always be rewritten as a lower triangular matrix by numbering spatial regions according to the propagation front, this is not the case for MP_N in general. For certain decompositions of the unit sphere into solid angle, it may happen that certain directions see a region interface as incoming, while others see the same interface as outgoing. An example of this is depicted in Figure 4.4. Here, $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are two different directions belonging to $\mathcal{S}_n^2$, which approach Γ, the interface between two regions, from different sides. In such a case, interface Γ would be both incoming and outgoing at the same time. The resulting coupling between regions would result in a non-triangular system of equations.

This difficulty can be overcome by using planes parallel to region surfaces to section the unit sphere. In IDT, and in Cartesian solvers in general, this can be done by simply constructing a unit sphere decomposition as invariant with respect to reflections on the XYZ planes. As for the usual quadratures in S_N, we have to decompose one single octant into solid angles, and then generate the remaining through reflections with respect the XYZ planes. This construction is doubly useful because it allows also for a straightforward implementation of reflective boundary conditions.

As a consequence of these properties, the inversion of the MP_N operator can be performed using the standard sweeping scheme. Namely, we address each solid angle $\mathcal{S}_n^2$ independently, and, starting from "illuminated" boundaries, we solve the MP_N balance and transmission for all regions following the propagation front. The implementation of the MP_N method does not require to modify the structure of the scattering kernel nor the algorithm of the spatial sweeping of the standard S_N solver, but only acts on the nature of matrices and moments to be solved.

A relevant difference between MP_N and S_N stays in the way angular moments are computed. MP_N method does not rely on any angular quadrature (S_N-like angular discretization with weights and directions) for the integration of the angular moments, but it directly evaluates the contributions of the solid angles to the angular moments. Moreover, there is no need for the source term to be evaluated on a set of prescribed directions, since MP_N

MP_N coefficients		S_N coefficients	
$\mathbb{C}_n$	$[M \cdot N_h] \times [M \cdot N_h]$	$\overline{\overline{\mathbf{C}}}_n$	$M \times M$
$\mathbb{I}_n^{s'}$	$[M \cdot N_h] \times [K \cdot N_\ell]$	$\overline{\overline{\mathbf{I}}}_n^{s'}$	$M \times K$
$\mathbb{E}_n^{s}$	$[K \cdot N_\ell] \times [M \cdot N_h]$	$\overline{\overline{\mathbf{E}}}_n^{s}$	$K \times M$
$\mathbb{T}_n^{s,s'}$	$[K \cdot N_\ell] \times [K \cdot N_\ell]$	$\overline{\overline{\mathbf{T}}}_n^{s,s'}$	$K \times K$

Table 4.1 Comparison of the dimensions of the MP_N and S_N matrices, for a single region, a single pair of surfaces, a single solid angle $\mathcal{S}_n^2$ for the MP_N, and a single direction for the S_N. M is the size of the spatial basis expanding the flux in the region volume, and K is the size of the spatial basis expanding the flux on the region interfaces. N_h and N_ℓ are the number of spherical harmonics expanding the source, and the source of the local angular basis respectively.

acts directly on the component of the source spherical harmonics expansion, as shown in Eq. (4.16) and (4.28).

With MP_N, therefore, the source term is updated by simply computing

$$\mathbf{q}_m^{(i+1)} = \overline{\overline{\mathbf{\Sigma}}}_{s,H} \boldsymbol{\phi}_m^{(i)} + \mathbf{Q}_m \quad \text{for } m = 1, \ldots, M, \tag{4.30}$$

without the need of any moment-to-discrete and discrete-to-moment matrix (see Sections 1.2.3 and 1.2.4), with (i) as the inner iteration index. Here, we are denoting with $\mathbf{q}_m$ and $\boldsymbol{\phi}_m$ the vectors containing the m-th order spatial-angular components of source and flux, respectively. $\mathbf{Q}_m$ is instead the vector containing the m-th order components of the source term, accounting from the external source and particles scattered from other groups.

As concern the memory requirements, we remark that the MP_N needs an increased amount of memory for transport coefficients with respect to S_N. Table 4.1 compares the dimensions of MP_N and S_N coefficient matrices, for a single region $\mathcal{T}$, a single pair of surfaces s and s', for a single solid angle $\mathcal{S}_n^2$ for MP_N, and a single direction $\mathbf{\Omega}_d$ for the S_N. We see that $\mathbb{C}$, $\mathbb{I}$, $\mathbb{E}$ and $\mathbb{T}$ matrices dimensions scale with both the orders of the spatial and the angular bases, while S_N matrices scales with the size of the spatial bases only.

Short-characteristics method with MP_N, exactly like with S_N, is verified to be conservative. This, on the one hand provides robustness to the method, and on the other, allows to relate MP_N matrices using analytical conservation relations. In the event of future storage of coefficients, such relations may be used to retrieve MP_N coefficients from others stored in memory, so as to reduce the MP_N memory footprint. The derivation of a set of conservation relations is presented in Appendix C.

As opposed to S_N, we are not able to evaluate MP_N coefficients analytically. Therefore, the calculation is made using a numerical angular integration. At the present state of the implementation, we use an angular quadrature made of directions and weights $\{\mathbf{\Omega}_{n,i}, w_{n,i}\}_{i=1,N_i}$

in each solid angle. Therefore, we compute MP_N coefficients, as defined in Eqs. (4.14), (4.15), (4.24), (4.26), and the mass matrix elements, Eq. (4.7), by the local quadrature

$$\int_{\mathcal{S}_n^2} d\mathbf{\Omega}\ f(\mathbf{\Omega}) \approx \sum_{i=1,N_i} w_{n,i} f(\mathbf{\Omega}_{n,i}), \tag{4.31}$$

with $f(\mathbf{\Omega})$ as a generic function of the angle.

Finally, MP_N can also be employed in adjoint calculations. The change of sign of the adjoint streaming operator with respect to the direct one (see Section 1.1.2), can be taken into account by simply substituting the spherical harmonics $A_h(\mathbf{\Omega})$ with $A_h(-\mathbf{\Omega})$ within the definitions of the coefficients $\mathbf{C}$, $\mathbf{I}$ and $\mathbf{E}$, in Eqs. (4.24), (4.26) and (4.14), respectively.

4.3 Implementation of the MP_N method in the IDT solver

The MP_N method presents a number of parameters that still have to be defined. These are:

- the number and the shape of the solid angle partitioning of the unit-sphere;
- the definition of the local angular basis $\mathbf{B}^n(\mathbf{\Omega})$;
- the angular quadrature used to compute the MP_N coefficients within each solid angle.

At this stage, our purpose is to explore the MP_N method and to highlight its strengths and weaknesses. Our choices are therefore guided by the criterion of simplicity and effectiveness. In the following, we will present our choices and the solution algorithm of MP_N.

4.3.1 Definition of the sphere partitioning strategy

The MP_N, in principle, is able to account for every type of discretization as far as we are capable of defining a local angular basis in each solid angle. The simplest strategy consist in partitioning the unit sphere using a rectangular latitudinal-longitudinal grid. This type of meshes are known to present clustering effects at the poles (meshes shrink as the polar angle approaches the poles and elements get more densely distributed in these regions), but, on the other hand, allows for a simple definition of local angular basis and angular quadrature.

By using μ as the cosine of the polar angle θ, and φ as the azimuthal angle, we partition the unit sphere

$$\mathcal{S}^2 = \{(\varphi, \mu) : \varphi \in [0, 2\pi],\ \mu \in [-1, 1]\}, \tag{4.32}$$

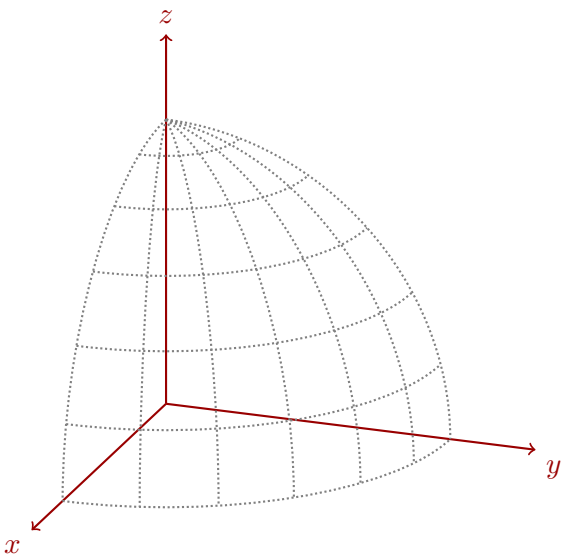


Fig. 4.5 Illustration of the unit sphere decomposition generated using the latitudinal-longitudinal grid. Grid obtained using $N_\varphi = 24$ and $N_\theta = 12$ and partitioning the sphere into 288 solid angles.

using a uniform grid of the two cardinal angles. The result is a $N_\varphi \times N_\theta$ elements grid, where each solid angle is defined by

$$\begin{aligned} \mathcal{S}_n^2 &= \left[\frac{2(i_\varphi - 1)\pi}{N_\varphi}, \frac{2i_\varphi\pi}{N_\varphi}\right] \times \left[\cos\left(\frac{(i_\theta - 1)}{N_\theta}\pi\right), \cos\left(\frac{i_\theta\pi}{N_\theta}\right)\right], \\ \text{for } i_\varphi &= 1, \ldots, N_\varphi \ , \ i_\theta = 1, \ldots, N_\theta, \text{ and} \\ n &= 1, \ldots, (i_\theta - 1)N_\varphi + i_\varphi, \ldots, N_\varphi N_\theta. \end{aligned} \tag{4.33}$$

The resulting discretization, depicted in Figure 4.5, is identical to the Uniform Quadrature defined in Section 2.3.1 for the MOC and the TS-MOC First Collision Source methods. The parameter N_φ and N_θ are chosen such that the solid angle decomposition respects the symmetries of the octant, i.e.

$$N_\varphi = 4a \text{ and } N_\theta = 2b \text{ with } a, b \in \mathbb{N} \setminus \{0\},$$

where $\mathbb{N}^*$ is the ensemble of natural numbers excluded the value 0.

4.3.2 Definition of the local angular basis

Once defined how the unit sphere is partitioned, we have now to define the angular basis $\mathbf{B}^n$ within each solid angle. For consistency with the source term expansion, the first option one may consider is to represent the angular dependence of the interface fluxes using a finite-dimensional spherical harmonics basis within each solid angle. However, this family

of functions are not a basis for a solid angle, as they are not linearly independent. In order to use spherical harmonics, we should select only functions that are linearly independent of each other. At present, we have not explored this solution yet. We opted instead to use a polynomial basis.

Let $\mathcal{S}_n^2$ be a generic solid angle defined by

$$\mathcal{S}_n^2 = \left[\varphi_{n,1},\ \varphi_{n,2}\right] \times \left[\mu_{n,1},\ \mu_{n,2}\right]. \tag{4.34}$$

Within this interval, we define the local angular basis $\mathbf{B}^n$ as

$$\mathbf{B}^n(\hat{\varphi},\hat{\mu}) = \{1,\ \hat{\varphi},\ \hat{\mu},\ \hat{\varphi}\hat{\mu},\ \hat{\varphi}^2,\ \hat{\mu}^2,\ \hat{\varphi}^3,\ \hat{\varphi}^2\hat{\mu},\ \hat{\varphi}\hat{\mu}^2,\ \hat{\mu}^3\}, \tag{4.35}$$

where $\hat{\varphi}$ and $\hat{\mu}$ are the normalized local coordinates such that $(\hat{\varphi},\hat{\mu}) \in [-1,\ 1] \times [-1,\ 1]$ and

$$\hat{\varphi} = -1 + 2\frac{\varphi - \varphi_{n,1}}{\varphi_{n,2} - \varphi_{n,1}}, \tag{4.36}$$

$$\hat{\mu} = -1 + 2\frac{\mu - \mu_{n,1}}{\mu_{n,2} - \mu_{n,1}}. \tag{4.37}$$

Here, basis $\mathbf{B}^n$ spans up to the cubic order. However, in our tests we will consider different subspaces of this basis. In particular, we will look at the constant (piecewise angular step approximation, $N_\ell = 1$), the linear ($N_\ell = 3$), the bilinear ($N_\ell = 4$), the quadratic ($N_\ell = 6$) and the cubic ($N_\ell = 10$) orders.

4.3.3 Definition of the numerical integration strategy for MP_N coefficients

For our implementation, we defined the quadrature set $\{\mathbf{\Omega}_{n,i}, w_{n,i}\}_{i=1,N_i}$ using a Chebyshev-Legendre product quadrature formula, for each of the N_n solid angles. In particular, a uniform quadrature of order $\sqrt{N_i}$ solves the integrals along the interval $[\varphi_{n,1}, \varphi_{n,2}]$, and a Gauss-Legendre quadrature of the same order performs the ones over $[\mu_{n,1}, \mu_{n,2}]$. For each solid angle $\mathcal{S}_n^2$, we use the same quadrature rule to integrate the four MP_N matrices $\mathbb{C}_n$, $\mathbb{I}_n^{s'}$, $\mathbb{E}_n^s$, $\mathbb{T}_n^{s,s'}$, and the mass matrix $\mathbb{N}_n^s$. Other numerical quadratures have not yielded any significant benefit on the MP_N accuracy.

4.3.4 Description of the MP_N algorithm

Algorithm 5 provides a schematic description of the way MP_N equations are solved. We now present how the algorithm works and discuss the main differences with the S_N Algorithm 1.

At each iteration $(i+1)$, the flux is initialized and the source, together with the incoming boundary fluxes, is updated using fluxes of the previous iteration. Then, MP_N solves independently each solid angle. As a first step, it generates the local angular quadrature rule within the solid angle and, for each direction, it evaluates the punctual values assumed by

Algorithm 5: The MP$_N$ algorithm

begin
  do
    Set iteration $(i+1)$;
    Initialise $\boldsymbol{\phi}^{(i+1)} \longleftarrow 0$;
    Update $\mathbf{q}^{(i+1)}$ from $\boldsymbol{\phi}^{(i)}$ by Eq. (4.30);
    Update $\boldsymbol{\phi}^{s',(i+1)}$ with boundary conditions: $\boldsymbol{\phi}^{s',(i+1)} \longleftarrow \beta^{s',s}\boldsymbol{\phi}^{s,(i)}$;
    forall $\mathcal{S}_n^2 \in \mathcal{S}^2$ **do**
      forall $\boldsymbol{\Omega}_{n,i} \in \{\boldsymbol{\Omega}_{n,i}, w_{n,i}\}_{i=1,N_i}$ **do**
        Compute $\overline{\overline{\mathbf{C}}}(\boldsymbol{\Omega}_{n,i})$, $\overline{\overline{\mathbf{I}}}^{s'}(\boldsymbol{\Omega}_{n,i})$, $\overline{\overline{\mathbf{E}}}^{s}(\boldsymbol{\Omega}_{n,i})$, $\overline{\overline{\mathbf{T}}}^{s,s'}(\boldsymbol{\Omega}_{n,i})$;
        Compute $A_h(\boldsymbol{\Omega}_{n,i})$ for $h = 1, \ldots, N_h$;
        Compute $B_\ell^n(\boldsymbol{\Omega}_{n,i})$ for $\ell = 1, \ldots, N_\ell$;
      end
      Compute $\mathbb{N}_n^s$ by Eq. (4.17);
      Compute $\mathbb{C}_n$ by Eq. (4.24);
      Compute $\mathbb{I}_n^{s'}$ by Eq. (4.26);
      Compute $\mathbb{E}_n^s$ by Eq. (4.14) and Eq. (4.18);
      Compute $\mathbb{T}_n^{s,s'}$ by Eq. (4.15) and Eq. (4.18);
      Solve along $\mathcal{S}_n^2$: $\boldsymbol{\phi}_n^{(i+1)}, \boldsymbol{\phi}_n^{s,(i+1)} \longleftarrow$ **SWEEP**$(\mathbf{q}^{(i+1)}, \boldsymbol{\phi}_n^{s',(i+1)})$;
      $\boldsymbol{\phi}^{(i+1)} \longleftarrow \boldsymbol{\phi}^{(i+1)} + \boldsymbol{\phi}_n^{(i+1)}$;
      $\mathbb{C} \longleftarrow \mathbb{C} + \mathbb{C}_n$;
    end
    $\boldsymbol{\phi}^{(i+1)} \longleftarrow \boldsymbol{\phi}^{(i+1)} + \mathbb{C}\mathbf{q}^{(i+1)}$;
  while $\epsilon(\boldsymbol{\phi}^{(i+1)}, \boldsymbol{\phi}^{(i)}) \geq \tau$;
end

the spherical harmonics functions, by the local angular basis, and by the short-characteristics coefficients. Then, MP$_N$ algorithm integrates the coefficients using the local angular quadrature rule, according to Eq. (4.31). As for the escape and the transmission matrices, $\mathbb{E}_n^s$ and $\mathbb{T}_n^{s,s'}$, the mass matrix $\mathbb{A}_n^s$ is also computed, inverted, and then multiplied as in (4.18). Finally, MP$_N$ performs the sweep along the propagation front of the solid angle, i.e. it solves balance and transmission equations, (4.16) and (4.28), in each region staring from the incoming domain boundaries and moving towards the outgoing ones. This provides the values of the contributions from the different solid angles to the flux spatial-angular moments. By summing these contributions, the flux moments are reconstructed, and the process is iterated until a flux convergence criterion, based on error function ϵ (defined in Eq. (1.71)), is verified.

The MP$_N$ algorithm differs from the S$_N$ one by:

- the evaluation of the angular moments of the flux;

- the evaluation of the coefficients;

- the sweep scheme.

As compared to S_N, the source term is not evaluated on a set of discrete directions, and the angular moments of the flux are evaluated without applying any angular quadrature, but rather summing the contributions of the different solid angles.

Since the size of MP_N matrices is significantly greater than the one of the short-characteristic matrices used in S_N, at present, the MP_N matrices are not stored but calculated on-the-fly, solid angle by solid angle. For future developments, we do not exclude the possibility of exploiting symmetries, conservation and reciprocity relations in order to optimize the algorithm memory consumption, and avoid the on-the-fly evaluation.

Finally, in order to reduce the overall computational cost, we opted to modify the standard sweep scheme in order to factorize the collision matrix, as in Eq. (4.29). This allows to avoid multiplying the collision matrix for each solid angle, but applying it only once. In practice, the MP_N sweeps compute only the second term on the RHS of Eq. (4.28), namely the incoming contribution. The collision matrix is accumulated instead, as shown in Eq. (4.25), and applied once and for all, in each spatial region, only at the completion of the sweep on all solid angles. The collision contribution thus calculated is then added to the angular moments of the flux.

4.4 Assessment of the MP_N method capabilities of mitigating the ray-effect in particle propagation problems

We designed MP_N with the intent of enriching the transmission within the short-characteristics method, assuming that this would mitigate the ray-effect, and provide a better representation of the actual particle streaming. To verify the actual qualities of the method, we propose a series of targeted tests.

At first, we want to verify the MP_N capabilities on a problem which is affected by severe ray-effect. Next, we propose a verification of the method on the Kobayashi benchmark. The third and fourth tests are a homogeneous and a heterogeneous problem, respectively. With them, we want to assess the error convergence rate of the MP_N approximation with respect to the degrees of the angular refinement, as compared to S_N. In the fifth test we evaluate the effectiveness and the accuracy of MP_N in diffusive media, for different values of the scattering ratios. Then, we perform an analysis showing the impact of the order of the angular quadrature for the MP_N coefficients on the quality of the flux estimation. Finally, we will draw some considerations regarding the actual computational cost of MP_N, and, in particular, how it varies with the angular refinement, and with the problem anisotropy order.

4.4.1 Evaluation of the ray-effect mitigation capabilities of MP_N

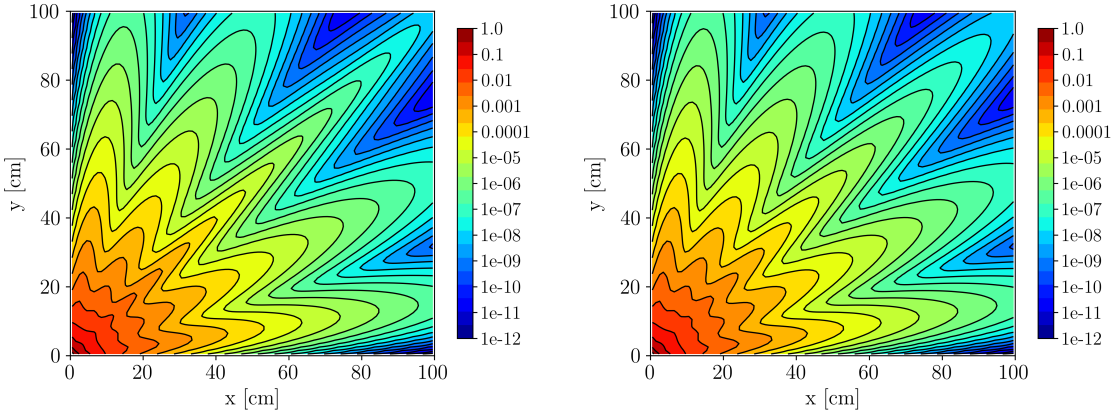
As first test we consider the worst possible case for ray-effect mitigation, that is the particle propagation from a localized source in a non diffusive media. The model is made of a flat parallelepiped of $100 \times 100 \times 1$ cm sides, discretized using a uniform grid composed of $100 \times 100 \times 1$ equal cubic regions. A unit source is located in the bottom left corner region. Reflective boundary conditions are applied on the $Z+$ and $Z-$ boundaries so as to make the 3-D problem equivalent to a 2-D one. Reflections are also applied to the $X-$ and $Y-$ boundaries, while vacuum boundary conditions are applied elsewhere. The medium is purely absorbing and has a total cross section Σ everywhere equal to 0.1 cm^{-1}. We solved the problem using both S_N and the MP_N method with different orders of the local angular basis. S_N is run imposing a product-type Chebyshev-Legendre quadrature rule with 5×5 directions per octant. The MP_N uses instead a 5×5 solid angle discretization in every octant, and a 10×10 angular quadrature per solid angle. Keeping these parameters constant, we tested MP_N with the constant, the linear, the quadratic and the cubic orders of local angular bases. A reference is also provided using S_N with the Chebyshev-Legendre quadrature with 75×75 directions per octant. We run simulations using the short-characteristics spatial discretization with the constant flux approximation.

Ray-effect in step-characteristics method

Here we are showing the results generated using the step-characteristics discretization.

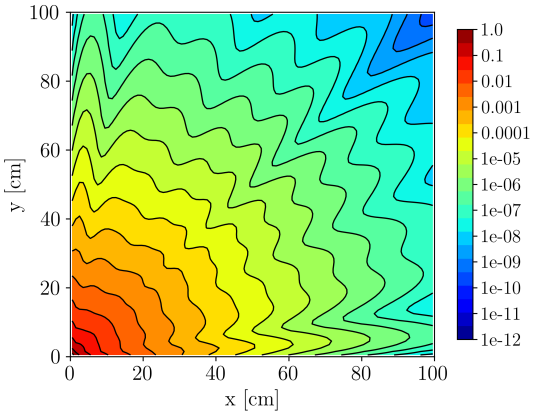
Figure 4.6 provides a graphical representation of the average flux over the whole problem domain. We observe that the 25 angular directions (per octant) used are not sufficient for S_N to represent the effective particle propagation from the small source region in a so refined spatial domain. Figure 4.6 (a) provides an example of ray-effect affecting the S_N calculations. We notice that many solid angles do not show null flux values. This is because the numerical diffusion involved by the short-characteristics spatial discretization helps feeding those regions. In this sense, the spatial discretization already provides a "by-accident" ray-effect mitigation.

The result of MP_N with the constant approximation, displayed in Figure 4.6 (b), shows no mitigation of ray-effect with respect to S_N. The Constant MP_N and S_N transmission are formally identical since one single Angular Degrees of Freedom (ADoF) is solved per sweep. The main difference between the two lies in the fact that MP_N coefficients are integrated over the solid angle, but this seems to have no effect in terms of ray-effect mitigation. With ADoF we intend the number of angular degrees of freedom transmitted per direction/solid angle (i.e 1 for S_N and Constant MP_N, 3 for the Linear MP_N, 4 for the Bilinar MP_N, 6 for the Quadratic MP_N, and 10 for the Cubic MP_N) times the number of directions/solid angles. In Figures 4.6 (c-e) we see how, for a number of solid angles equal to the S_N directions, high-order MP_N shows a mitigated ray-effect. MP_N result approximates better and better

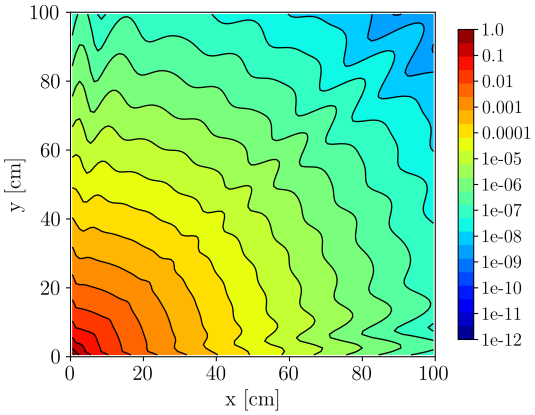


(a) S_N method

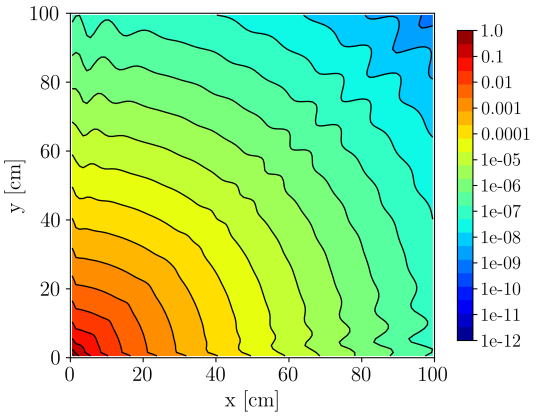
(b) Constant MP_N method



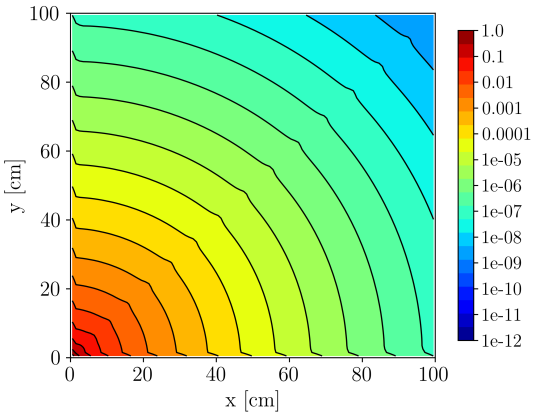
(c) Linear MP_N method



(d) Quadratic MP_N method



(e) Cubic MP_N method

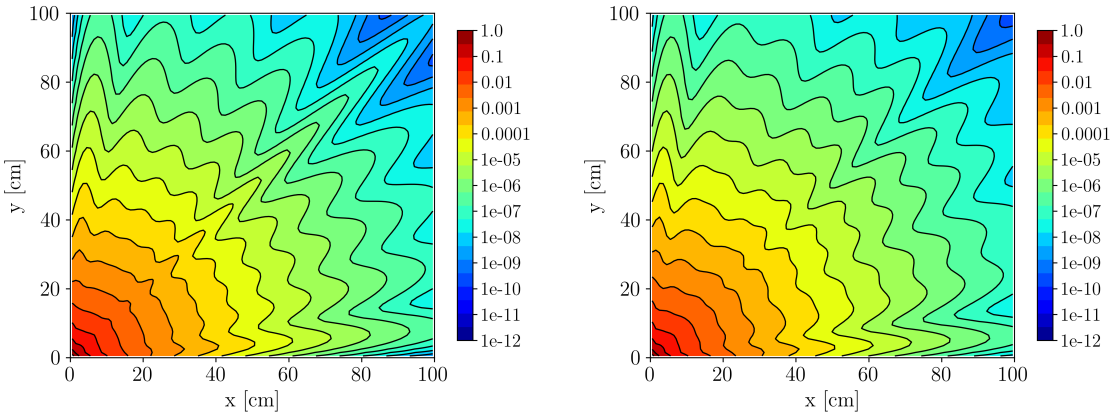


(f) Reference, S_N method

Fig. 4.6 Ray-effect mitigation test, step-characteristics, representation of the average flux intensity for S_N and MP_N, using 5×5 directions/solid angles per octant. The reference result provided by a very fine S_N calculation.

the reference as the order of the local angular basis increases, which is expected, since a higher order approximation is done.

However, a fairer comparison is obtained by running MP_N and S_N for an equal number of ADoF of the transmission equation. Figure 4.7 shows the flux profile obtained with MP_N Linear and a 5×5 solid angle discretization per octant (600 ADoF), as compared to the result of a S_N calculation run with the Chebyshev-Legendre angular quadrature and 9×9 directions per octant (648 ADoF). This time we observe qualitatively similar results for S_N and MP_N. Figure 4.7 (a) and (b) show not-matching distortions but similar amplitudes. A



(a) S_N method (b) Linear MP_N method

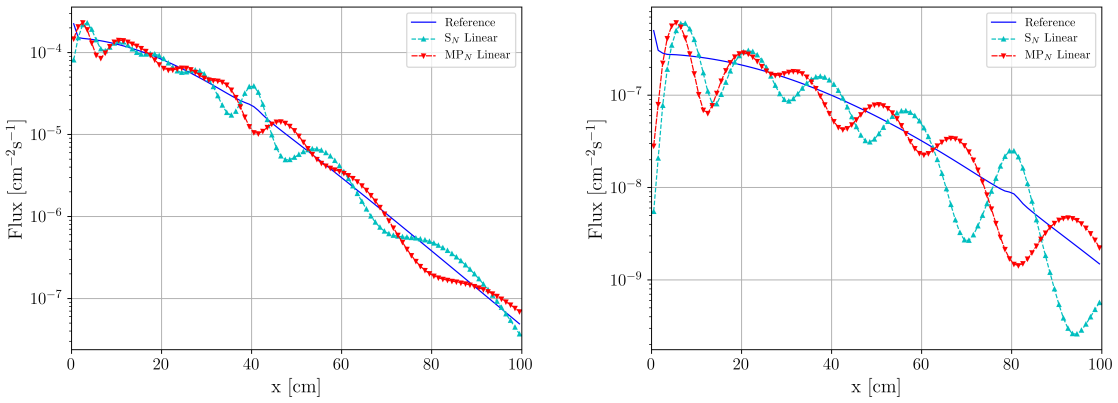
Fig. 4.7 Ray-effect mitigation test, step-characteristics, representation of the average flux intensity using S_N and Linear MP_N, for about the same number of angular degrees of freedom. MP_N run using 5×5 solid angles per octant, and S_N (Chebyshev-Legendre with rectangular arrangement) run using 9×9 directions per octant.

further insight is provided by Figures 4.8 (a) and (b), showing the flux profile along planes $y = 40$ cm and $y = 80$ cm, respectively. The root mean square and the infinity norm of relative error of the average flux along the two planes are reported in Table 4.2.

Table 4.2 Ray-effect mitigation test, step-characteristics, relative error norms of the average flux along planes $y = 40$ cm and $y = 80$ cm.

Plane	RMS [%]		L^∞ [%]	
	S_N	MP_N	S_N	MP_N
40 cm	2.49	4.09	77.93	55.90
80 cm	6.27	5.17	186.06	113.62

We see that both S_N and MP_N profile exhibit fluctuations typical of the ray-effect. The maximum deviation of the flux from the reference is slightly smaller for the MP_N method. However, on average, the two methods are similar both in number of fluctuations and in the magnitude of the deviation from the reference.

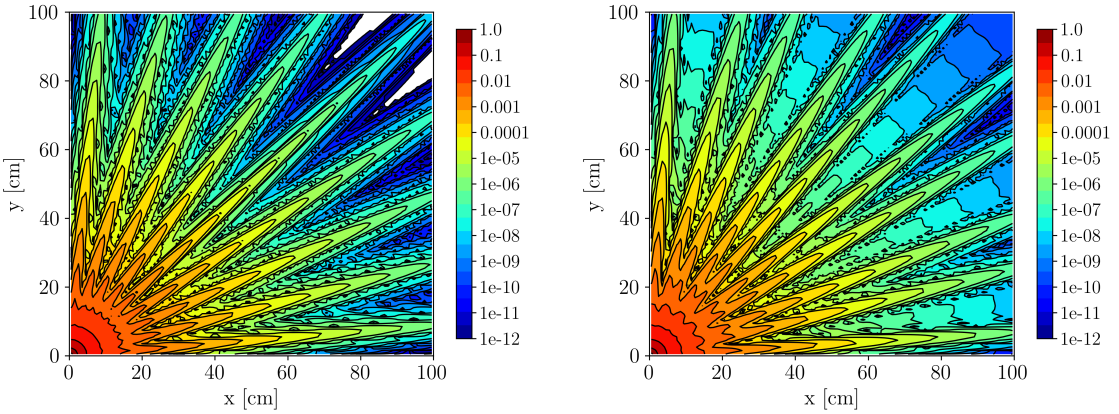


(a) Flux profile $y = 40$ cm (b) Flux profile $y = 80$ cm

Fig. 4.8 Ray-effect mitigation test, step-characteristics, representation of the average flux intensity along planes $y = 40$ cm and $y = 80$ cm. Comparison of S_N and Linear MP_N for about the same number of angular degrees of freedom. MP_N run using 5×5 solid angles per octant, and S_N (Chebyshev-Legendre with rectangular arrangement) run using 9×9 directions per octant.

Ray-effect in linear-characteristics method

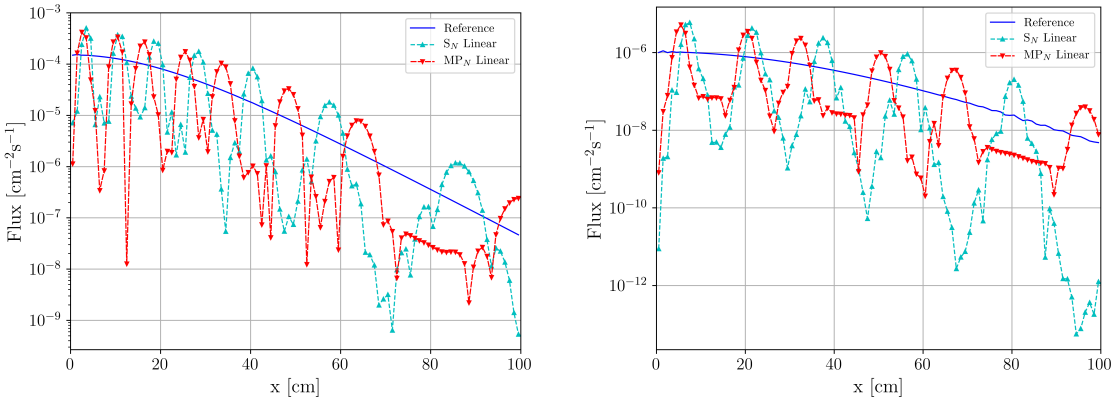
In order to extend our conclusions, we repeated the same test, but this time using linear-characteristics. The scalar flux profiles over the entire domain, obtained with S_N and MP_N, are shown in Figure 4.9. The flux along the 40 cm and the 80 cm planes are shown in Figure 4.10, and analysed in Table 4.3.



(a) S_N method (b) Linear MP_N method

Fig. 4.9 Ray-effect mitigation test, linear-characteristics, representation of the average flux intensity using S_N and Linear MP_N, for about the same number of angular degrees of freedom. MP_N run using 5×5 solid angles per octant, and S_N (Chebyshev-Legendre with rectangular arrangement) run using 9×9 directions per octant.

As expected, linear-characteristics shows a much more pronounced ray-effect than the step- ones. The numerical diffusion that smooth the flux profiles in step-characteristics results, thus artificially mitigating ray-effect distortions, is now strongly reduced. We observe that the S$_N$ flux profile shows peaks at the angular quadrature directions, and sinking in between (see Figure 4.9). Linear MP$_N$, instead, shows two peaks for each solid angle, and flux sinking between each peak.



(a) Flux profile $y = 40$ cm

(b) Flux profile $y = 80$ cm

Fig. 4.10 Ray-effect mitigation test, linear-characteristics, representation of the average flux intensity along planes $y = 40$ cm and $y = 80$ cm. Comparison of S$_N$ and Linear MP$_N$ for about the same number of angular degrees of freedom. MP$_N$ run using 5×5 solid angles per octant, and S$_N$ (Chebyshev-Legendre with rectangular arrangement) run using 9×9 directions per octant.

Table 4.3 Ray-effect mitigation test, linear-characteristics, relative error norms of the average flux along planes $y = 40$ cm and $y = 80$ cm.

Plane	RMS [%]		L$^\infty$ [%]	
	S$_N$	MP$_N$	S$_N$	MP$_N$
40 cm	9.89	41.64	541.61	416.36
80 cm	10.00	6.31	728.12	566.58

Looking at the errors in the Table 4.3, it seems that the conclusions drawn for step-characteristics are perfectly extensible for the linear-characteristics as well. MP$_N$, therefore, when compared to S$_N$ for an equal number of ADoF and a relatively coarse angular discretization, seems unable to mitigate the ray-effect.

4.4.2 Verification of the MP$_N$ results

In order to ensure the exactness of the MP$_N$ results, we consider once again the Kobayashi benchmark problem 3, [87], previously presented in Section 2.7.2. This time, however, we

look at both case $3i$, and $3ii$ of the problem. The two share the same geometry but, while in $3i$ the medium is purely absorbing, in model $3ii$ it is mildly diffusive. Scattering is isotropic ($K = 0$) and the scattering cross sections are everywhere defined as one half of the total cross section. Media properties are shown in Table 4.4.

Region	Source Intensity $[\text{cm}^{-3}\text{s}^{-1}]$	Σ $[\text{cm}^{-1}]$	$\Sigma_s[3i]$ $[\text{cm}^{-1}]$	$\Sigma_s[3ii]$ $[\text{cm}^{-1}]$
Source	1.0	0.1	0.0	$5 \cdot 10^{-2}$
Duct	0.0	$1 \cdot 10^{-4}$	0.0	$5 \cdot 10^{-5}$
Shield	0.0	0.1	0.0	$5 \cdot 10^{-2}$

Table 4.4 Kobayashi benchmark problem $3i$ and $3ii$, medium data.

With respect to Section 2.7.2, reference results are taken from the publication of K.Kobayashi *et al.*, [87], which provides the flux in the points of sets A, B and C, rather then the average flux in the regions in which these are contained. The reference results provided in the article, [87], are obtained for both cases $3i$ and $3ii$ using the GMVP Monte Carlo Code, [94]. Results are provided with a relative standard deviation strictly lower than 0.8%.

We solved the problem using linear-characteristics and a constant step size equal to 2 cm, leading to a spatial discretization made of 45000 regions. As angular discretization we used both S_N and MP_N, with the linear and cubic order of the local angular base. We run each calculation using a coarse and a fine angular refinement. In detail, we run:

- S_N calculation using a Chebyshev-Legendre product-type quadrature made of 1800 direction, for the coarse refinement, and 5000 direction, for the fine one;
- MP_N Linear, with 512 and 1800 solid angles;
- MP_N Cubic, with 125 and 512 solid angles.

We selected the number of solid angles such that the total number of angular degrees of freedom involved in the transmission equation, i.e. the number of solid angles times the size of the local angular base, is approximatively equal to the number of directions employed by the S_N coarse and the fine quadratures.

The results of the tests on case $3i$ (without scattering) and case $3ii$ (with scattering) are represented in Figures 4.11 and 4.12, respectively. These show the magnitude of the flux in the points of interest, as well as the ratio between the flux and the reference values. For simplicity, the values of the flux in the points are approximated by average flux in the regions containing the points.

In Figure 4.11 we observe how results of MP_N and S_N are qualitatively similar in terms of error. On average, MP_N with the coarse discretization presents results that deviate most from the reference with respect to S_N coarse. A similar behaviour can be found also in Figure 4.12

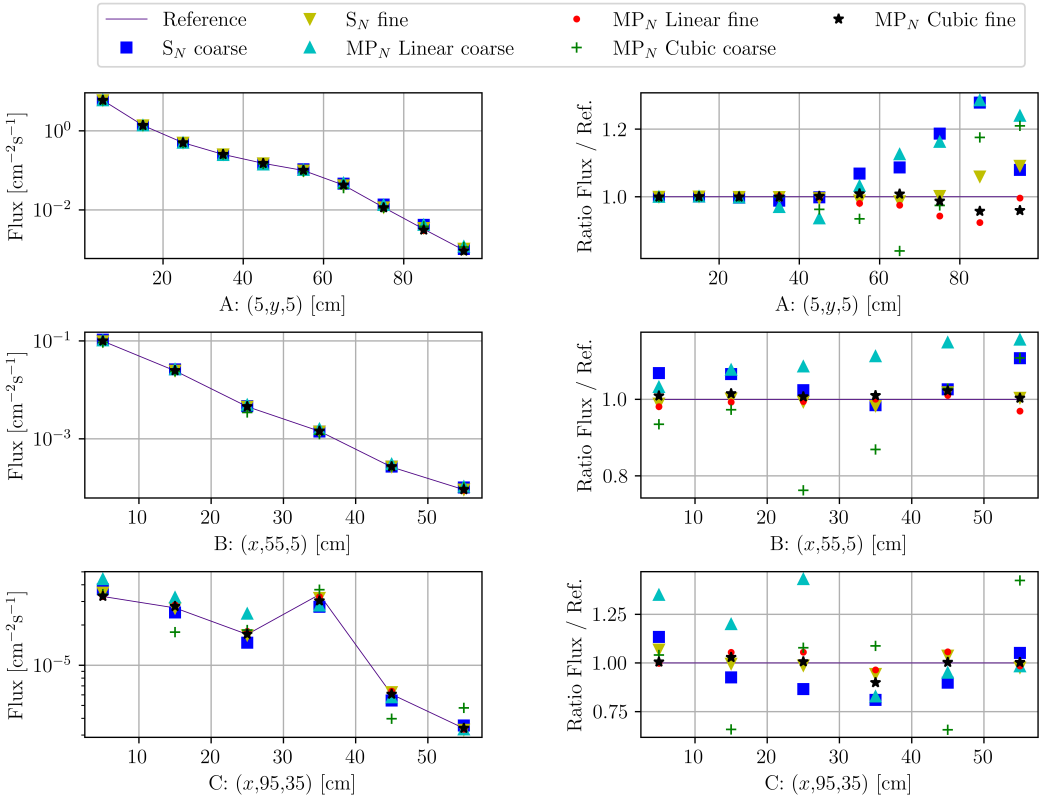


Fig. 4.11 Kobayashi benchmark problem 3i, verification of the MP$_N$ method. The average flux and its ratio with respect to the reference are displayed for sets of points A, B and C (see Section 2.7.2).

for case 3ii. The ratio flux/reference, in this second case, moves within a smaller range of values with respect to case 3i. This is linked to the fact that problems with scattering are less affected by ray-effect then the ones with a purely absorbing media. In this sense, the same accuracy in results may be achieved using a coarser angular discretization while solving diffusive problems.

For all angular discretization methods, results obtained with the fine refinement present errors which are smaller than the one of calculations with the coarse angular discretization. Thus, we can confidently say that MP$_N$ is able to converge the estimation error as the discretization gets refined. The next goal of our analysis is to quantify the accuracy of MP$_N$ method with respect to S$_N$. For this purpose, we have considered a simplified problem, first with a homogeneous medium, and secondly with a heterogeneous one. We will refer to this as the Cube Test problem.

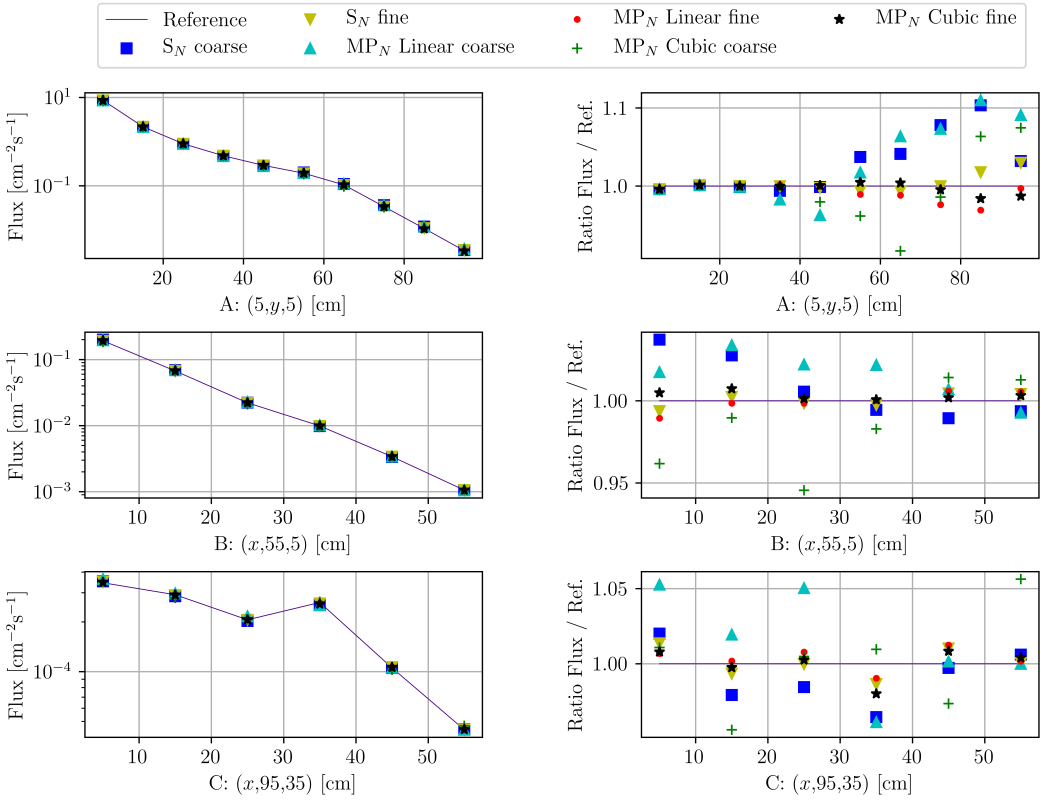


Fig. 4.12 Kobayashi benchmark problem 3*ii*, verification of the MP_N method. The average flux and its ratio with respect to the reference are displayed for sets of points A, B and C (see Section 2.7.2).

4.4.3 Assessment of the accuracy of MP_N method by the homogeneous Cube Test problem

The Homogeneous Cube Test, depicted in Figure 4.13, is made of a cube with side length equal to 50 cm. The cube contains a second smaller cube (coloured in green), with a 30 cm side, centred with it. The domain volume is subdivided into $5 \times 5 \times 5$ equal cubic zones (see the dotted gray lines). The medium is purely absorbing and the total cross section Σ is set everywhere equal to 0.1 cm^{-1}. The source (in pink) is defined in the front-bottom-left corner zone, and it has an intensity of 1 cm^{-3}sr^{-1}. Vacuum boundary conditions are everywhere applied. The test consists in monitoring the root mean square (RMS) of the relative error of the flux, averaged over the different zones. We solved the problem using both S_N and MP_N with the linear-characteristics spatial approximation. Discrete-ordinates calculations are run using a product-type Chebyshev-Legendre quadrature rule, while MP_N is run for the different orders of the local angular base. Calculations are performed using different refinements of the spatial and the angular discretizations. For each of them, we compute the flux in each spatial

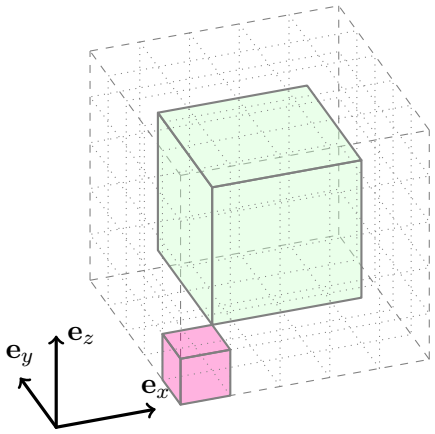


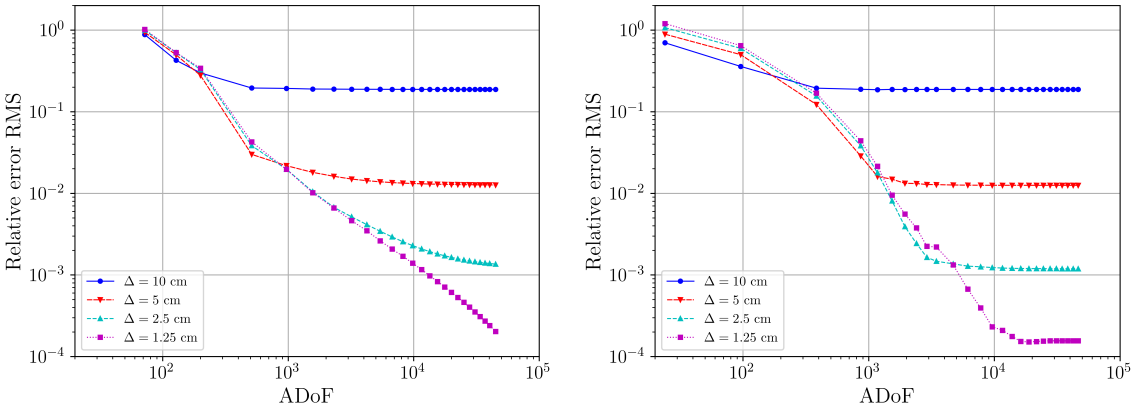
Fig. 4.13 Illustration of Cube Test problem.

region, and then, we average its value over the spatial zones. We calculated the relative error using as reference the result of an extremely fine discrete-ordinates calculation. RMS is computed using

$$RMS = \sqrt{\frac{\sum_{i=1}^{N_{zones}} \left(\frac{\phi^i - \phi^i_{ref}}{\phi^i_{ref}} \right)^2}{N_{zones}}}, \tag{4.38}$$

where ϕ^i is the flux averaged on zone i, evaluated with the considered method, and ϕ^i_{ref} is the reference flux value, averaged on zone i. The evaluation of the RMS of zone-averaged fluxes rather then on regions allows to compare results obtained with different spatial discretizations. The reference calculation is run using linear-characteristics with a step size $\Delta = 0.5$ cm, and a Chebyshev-Legendre quadrature made of 45000 directions. In the calculations under analysis, the angular discretization is progressively refined by increasing the order of the Chebyshev-Legendre angular quadrature for S$_N$, and by refining the solid angle discretization for MP$_N$. The spatial refinement is instead performed by successively splitting each region into 8 equal cubic sub-regions. Namely, we run for step sizes Δ equal to 10 cm, 5 cm, 2.5 cm and 1.25 cm.

Figure 4.14 shows the trend of the RMS with respect to the number of angular degrees of freedom (ADoF), by S$_N$ in (a) and by the Linear MP$_N$ in (b), for different spatial refinement levels. The figure shows how, in order to decrease the RMS of the flux relative error, both spatial and angular discretizations have to be refined. For a given level of spatial refinement, the RMS converges with the angular discretization to an asymptotic error, which is characteristic of the spatial discretization used. As the spatial mesh is refined, the RMS requires a finer angular quadrature in order to converge on its asymptotic value. Furthermore, if the angular discretization is too coarse ($\sim 10^2$ ADoF, in this problem), no gain in terms of



(a) S_N method (b) Linear MP_N method

Fig. 4.14 Homogeneous pure absorber Cube Test, convergence trend of the error with S_N and Linear MP_N. Monitoring the RMS of the relative error of the flux over the spatial zone discretization, with respect to the number of angular degrees of freedom (ADoF). Test run using different spatial discretizations (with Δ as the region sides length) and different angular refinement levels.

error reduction is obtained by refining the spatial variable only. This behaviour is another manifestation of the ray-effect.

By comparing the results of S_N and Linear MP_N on the same graph, as in Figure 4.15, we see that MP_N presents a slow error convergence for low ADoF, while it converges faster than S_N for high ADoF values. As a result, MP_N appears to be convenient over S_N only with fine angular discretizations. Furthermore, the finer the spatial discretization, the more effective MP_N is compared to S_N.

Figure 4.16 compares the results obtained with S_N and MP_N, using different orders of the local angular basis. All results are calculated for a fixed uniform spatial mesh made of cubic regions having sides $\Delta = 2.5$ cm long. The reference is the result of a calculation well converged in angle, but sharing the same spatial discretization of the calculations under analysis. By doing so, the error monitored in Figure 4.16 is representative of the angular discretization only. In other words, the error is representative of how fast the angular discretization method converges to the asymptotic error of the $\Delta = 2.5$ cm spatial discretization. The reference result is generated using a Cubic MP_N simulation run with 2500 solid angles per octant, and a 100 directions quadrature rule in each of solid angle. This particular choice will be motivated later in this section.

Figure 4.16 shows how high-orders MP_N method (orders of the local angular basis larger or equal to the linear one), beyond an entry region characterized by high error values, converges faster than S_N with the number of angular degrees of freedom. Constant MP_N, instead, converges to the asymptotic error with the same rate of the S_N method. The only

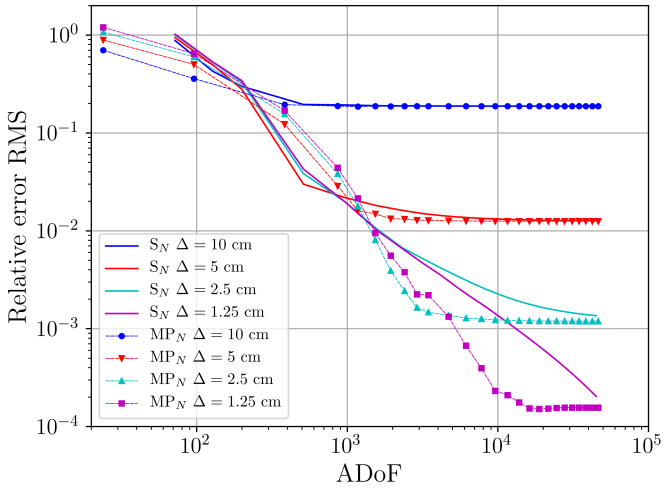


Fig. 4.15 Homogeneous pure absorber Cube Test, comparison of S_N and Linear MP_N error convergence trend for different spatial and angular discretizations.

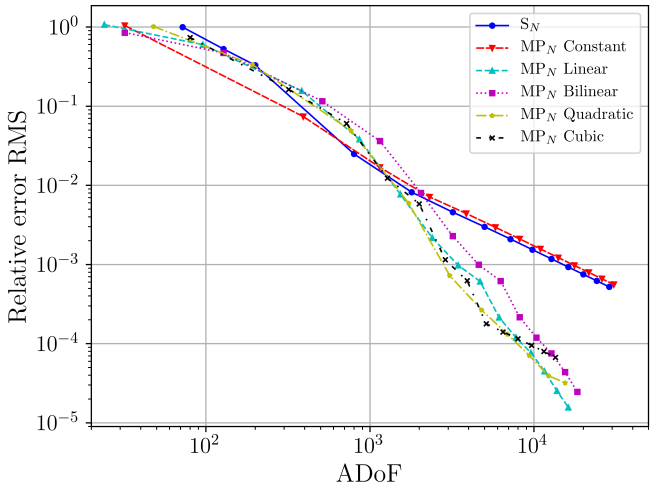


Fig. 4.16 Homogeneous pure absorber Cube Test, convergence trend of the error with S_N and Linear MP_N, for the same spatial discretization. Monitoring the RMS of the relative error of the flux over the spatial zone discretization, with respect to the number of angular degrees of freedom (ADoF). Comparison, for different angular refinement levels, of S_N and MP_N using different local angular bases. The spatial discretization is fixed with step size $\Delta = 2.5$ cm.

appreciable difference between the two methods is for low numbers of ADoF where Constant MP_N behaves more similarly to higher-order MP_N calculations.

We estimated the convergence order of the angular discretization by performing a logarithmic regression of the points of Fig. 4.16, neglecting the ones corresponding to the initial transient region (only values for a number of ADoF larger that 800 have been considered). This resulted to be 1.04 for the S_N, and 2.6 for the Linear MP_N. Thus, after a certain threshold, Linear MP_N reduces the error with an order of convergence 2.5 times larger than the one of S_N, with respect to the number of ADoF. We may further observe that the convergence rate with respect to the number of angular degrees of freedom is approximatively the same for higher orders of MP_N.

In summary, MP_N presents errors comparable to those of S_N for low ADoF. The faster convergence of MP_N (high-orders only) may be achieved only above a certain ADoF threshold. Beyond this, MP_N ensures errors smaller than S_N ones. Overall, MP_N appears to be effective for problems which require a very fine discretization.

The error estimation in Figure 4.16 is based on reference values generated using Cubic MP_N, which is also the method in analysis. We opted for this solution only after having found plateau effects, typical of not well converged references, while using S_N as reference. By using S_N, without recurring to Domain Decomposition, we have not been able to achieve precisions comparable to those achieved by MP_N.

Since higher-order MP_N calculations behave similarly, in the following we will often use Linear MP_N as representative of all MP_N approximation orders (excluded the constant one). Furthermore, between the different high-order MP_N, the one whose angular refinement is more easily controllable is the Linear one (smaller variation of ADoF versus the refinement of the solid angles discretization).

4.4.4 Assessment of the effect of heterogeneities on MP_N accuracy

In general, high-order methods, such as MP_N with order larger than the constant one, are less effective than step-approximations (e.g. S_N) in representing medium discontinuities. This is because high-order methods try to represent with linear function the angular flux which, even if continuous, may present very steep variations. In this sense, step-approximations may be better suited in representing flux irregular behaviours. We may expect therefore a lower accuracy of the MP_N method while solving heterogeneous problems with respect to homogeneous ones.

In order to assess these behaviours, the previous analysis performed on a homogeneous Cube Test is now repeated for a heterogeneous model. This is obtained as a variation of the Cube Test by changing the cross sections of outer and inner cubes. We consider two cases presenting different anisotropy degrees. The total cross sections for the outer and the inner cube are respectively, $\Sigma = 0.1$ cm^{-1}, and $\Sigma = 0.3$ cm^{-1} for the first case, and $3 \cdot 10^{-4}$ cm^{-1}

and $\Sigma = 0.3$ cm^{-1} for the second. The inner cube is defined as a block of absorbing material, which, in the first case, presents a total cross section 3 times larger than the outer cube, and 1000 time larger for the second case.

Similarly to the homogeneous test, we now compare the error convergence trend of S$_N$ and Linear MP$_N$ with respect to the number of ADoF. In Figure 4.17 we show, for the two cases, the RMS of the relative error of the flux, evaluated as in Eq. (4.38). The reference is provided by a very fine S$_N$ calculation, using 45000 directions, linear-characteristics and $\Delta = 0.5$ cm. We observe that S$_N$ leads to similar results than MP$_N$ for most spatial and angular discretizations. The threshold behaviour of the convergence speed of MP$_N$ is less pronounced than in the homogeneous case, but anyway present.

In Figure 4.18, different lines distinguish the results obtained using different angular discretization methods, but all sharing the same spatial discretization by linear-characteristics and $\Delta = 2.5$ cm. The reference is obtained using MP$_N$ Cubic with 2500 solid angles per quadrant, and the same spatial discretization, $\Delta = 2.5$ cm. The behaviour is similar to the one already discussed for the homogeneous Cube Test in Figure 4.16. However, the region in which MP$_N$ shows errors smaller that those of S$_N$ appears to be shifted to the right, so to larger ADoF values. Also, in Case 2, which is the most heterogeneous, MP$_N$ shows unreasonably high error values for low ADoF. MP$_N$ is therefore unusable for very heterogeneous problems with coarse angular discretizations. The range of applicability of MP$_N$ narrows as the medium gets more and more heterogeneous. However, above a certain ADoF threshold, the method retains its superior convergence to the asymptotic error, for the given spatial discretization.

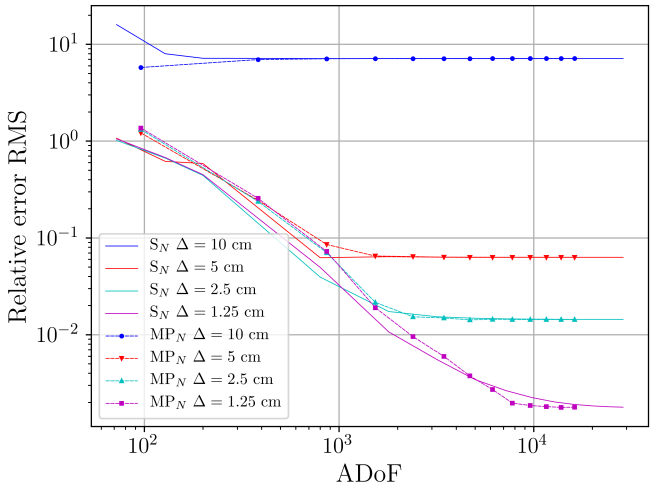
In Case 1 of the heterogeneous Cube Test, the angular convergence rate, estimated for ADoF values larger than 1000, is 1.02 for S$_N$ and 2.79 for Linear MP$_N$. The values are coherent with the convergence rate calculated for the homogeneous test.

As last remark, we observe that while Linear, Quadratic and Cubic MP$_N$ show almost the same convergence trend, Bilinear MP$_N$ seems to converge a little slower.

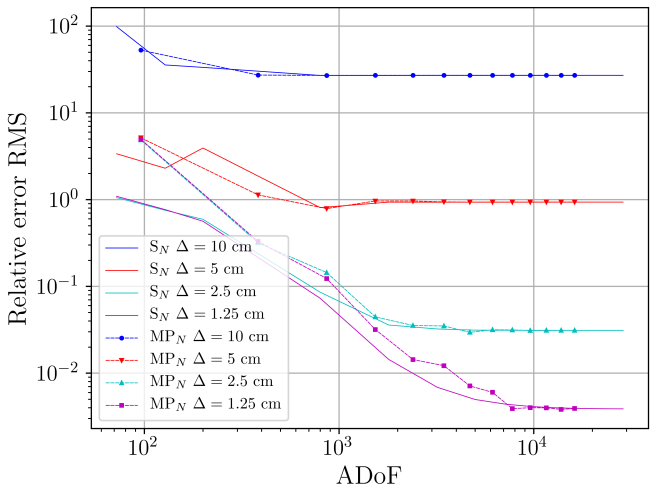
4.4.5 Assessment of the MP$_N$ effectiveness in the presence of scattering

Up to now, we documented the accuracy of MP$_N$ in purely absorbing problems. Our next step is to assess its effectiveness in the presence of scattering. To this extent, we considered the homogeneous Cube Test model and we modified the medium properties by adding isotropic scattering. In particular we ran the model for scattering cross section values equal to 0.0, 0.02, 0.05, and 0.09 cm^{-1}, corresponding, respectively, to values of the scattering ratio $c = \Sigma_s/\Sigma$ equal to 0, 0.2, 0.5, and 0.9.

We ran, for each of these configurations, a spatial-angular convergence study similar to the one presented in Figure 4.14. Namely, we measured the RMS of the error on the zone-averaged fluxes, evaluated for different spatial and angular discretization levels. We

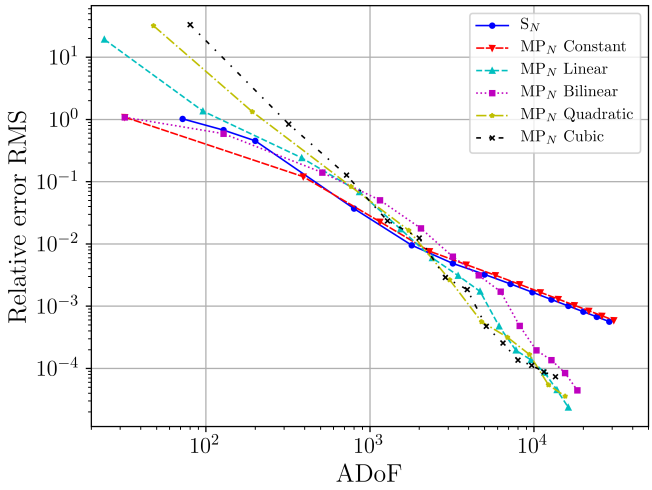


(a) Heterogeneous Case 1

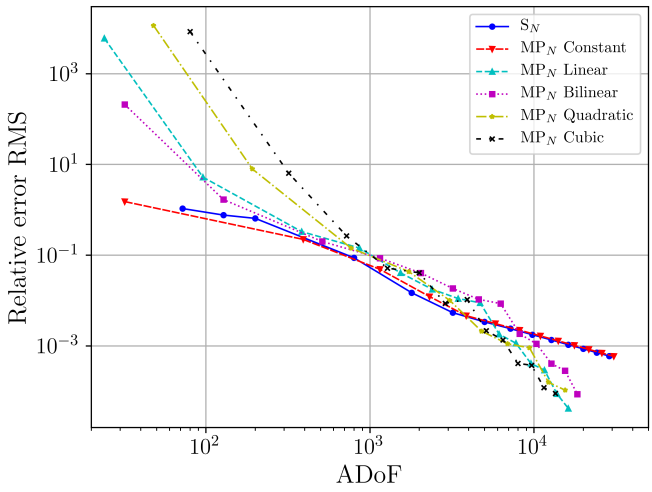


(b) Heterogeneous Case 2

Fig. 4.17 Heterogeneous pure absorber Cube Test, comparison of S_N and Linear MP_N error convergence trend for different spatial and angular discretizations.



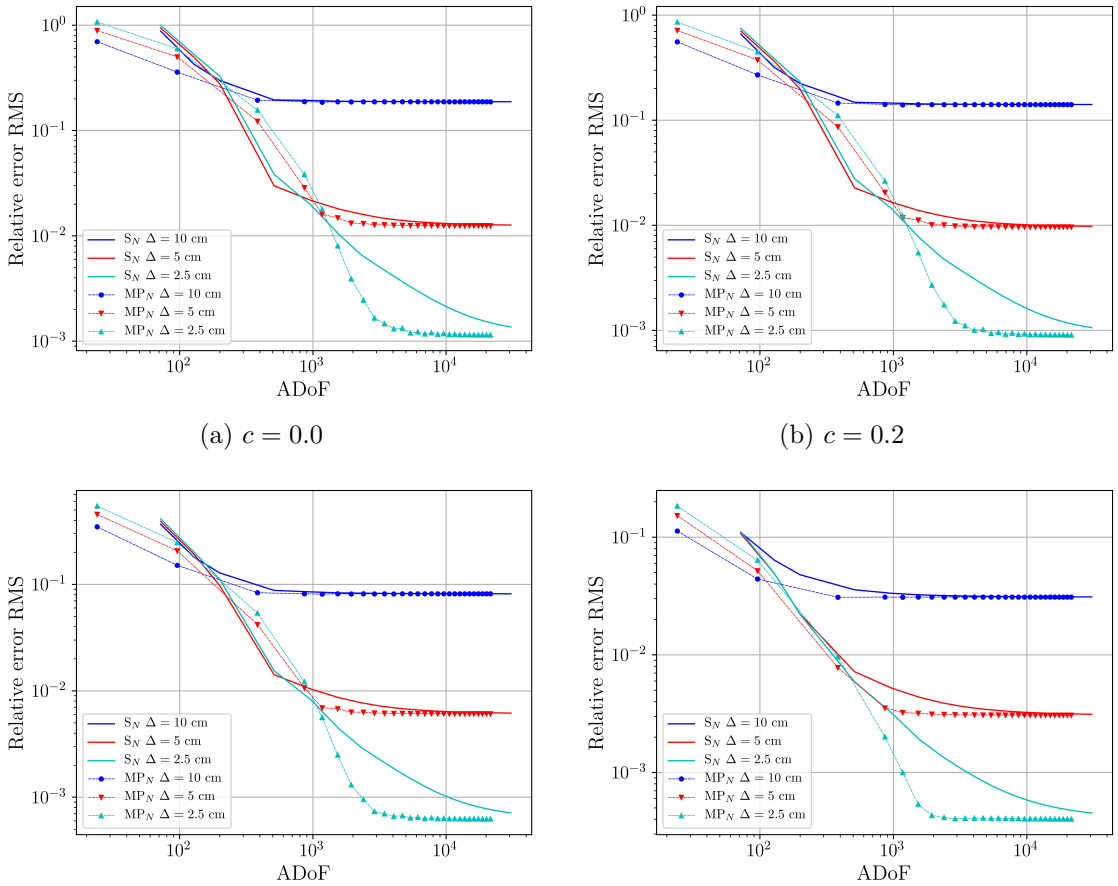
(a) Heterogeneous Case 1



(b) Heterogeneous Case 2

Fig. 4.18 Heterogeneous pure absorber Cube Test, convergence trend of the error with S_N and Linear MP_N, for the same spatial discretization. Monitoring the RMS of the relative error of the flux over the spatial zone discretization, with respect to the number of angular degrees of freedom (ADoF). Comparison, for different angular refinement levels, of S_N and MP_N using different local angular bases. The spatial discretization is fixed with step size $\Delta = 2.5$ cm.

ran for values of the spacing of the spatial mesh $\Delta = 10,\ 5,\ 2.5$ cm, using both S_N and MP_N Linear. For each of the four configurations, the reference is obtained using S_N with a 45000 directions Chebyshev-Legendre product quadrature, and a mesh spacing $\Delta = 1$ cm. Figure 4.19 shows the results of the test displaying S_N and MP_N results on the same graphs.



(a) $c = 0.0$ (b) $c = 0.2$

(c) $c = 0.5$ (d) $c = 0.9$

Fig. 4.19 Homogeneous diffusive Cube Test, assessment of the effectiveness of MP_N in the presence of scattering. Convergence trend of the relative error RMS of the zone-averaged flux, with respect to the number of angular degrees of freedom (ADoF). Calculation run for four different configurations with different scattering cross sections.

At first sight we may appreciate how, in all cases, MP_N converges faster than S_N to the asymptotic error of the given spatial discretization. The accuracy of MP_N appears to be unaltered by the presence of scattering. By comparing the four cases, we note that, as c increases, both S_N and MP_N converges the asymptotic error with a smaller number of ADoF. Further, as c increases, also the intersection point between the S_N and the MP_N curves moves towards lower ADoF values. By looking as this point as the threshold above

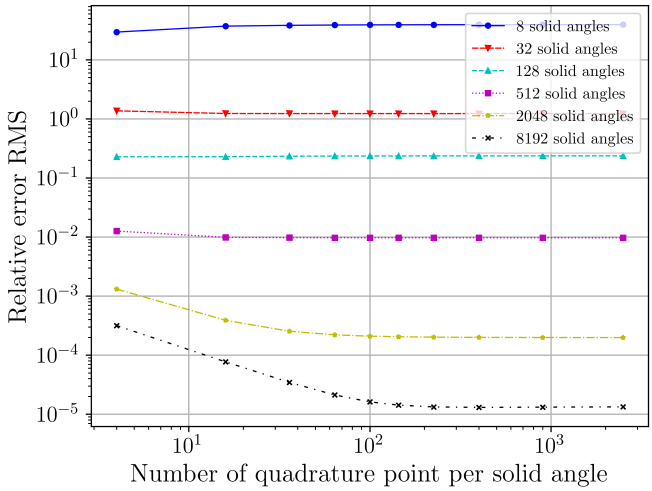


Fig. 4.20 Heterogeneous pure absorber Cube Test, sensitivity analysis of Linear MP$_N$ with respect to the order of the angular quadrature formula for the coefficient integration. Uniform spatial discretization with $\Delta = 5$ cm. The RMS relative error of the zone-averaged flux is computed for different number of solid angles and for different orders of the angular quadrature formula. Errors evaluated using a unique reference calculation obtained with 100×100 solid angles per octant, and a 100×100 points quadrature rule.

which MP$_N$ performs better than S$_N$, we deduce that the MP$_N$ application range (in terms of ADoF) increases with scattering.

The matter being beyond the scope of our research, we have not investigated further the application of MP$_N$ in highly diffusive media. However, this MP$_N$ application is not to be ruled out. To this end, further studies are required.

4.4.6 Assessment of the effect of the accuracy of the coefficient integration in MP$_N$

In Section 4.2.4 we have discussed how MP$_N$ coefficients do not have an analytic formulation, but they are evaluated numerically using Eq. (4.31). The quadrature rule introduces an error which can possibly deteriorate the quality of the estimation. The error does not directly impact the global particle balance, since we verified its conservation, but affects rather the spatial-angular distribution of the particle flux. With the present analysis, we want to quantify the effect of the numerical quadrature on the error of the MP$_N$ results.

The analysis is performed considering the heterogeneous Cube Test with a purely absorbing medium and a regular 5 cm spatial discretization. We solved the problem using the Linear MP$_N$ method and the linear characteristic spatial approximation. We repeated the calculation for different angular discretizations and for different orders of the angular quadrature rule (within each solid angle).

Figure 4.20 shows the relative error RMS of the zone-averaged flux, evaluated using as reference the result of a MP_N calculation with the same spatial discretization, 100×100 solid angles per octant, and 100×100 quadrature points per solid angle. This is a very accurate reference both in terms of angular discretization and in terms of the accuracy of the coefficients integration. The error monitored in the figure is not representative of the actual relative error, since the same spatial discretization is used in both calculations and reference. Instead, it measures the distance of the calculation from the asymptotic error of spatial discretization. The error is monitored with respect to the number of quadrature points, i.e. the quantity N_i in Section 4.2.4.

In Figure 4.20, curves related to small numbers of solid angles appears as flat. On the other hand, as the angular discretization gets refined, the importance of the quality of the coefficient integration comes into play. This means that, for coarse angular discretizations, the effect of the quadrature order is irrelevant since the error is dominated by the MP_N approximation. On the other hand, for large numbers of solid angles, the error of the MP_N approximation diminishes, and it may be worth to ensure a sufficient accuracy of the coefficient integration. However, the error introduced by the calculation of the coefficients is orders of magnitude smaller than the one introduced by the spatial and angular discretizations. A numerical quadrature made of 10×10 directions per solid angle should be enough for most of the calculations. Exceptions might be problems with high order of scattering anisotropy, which, requiring the integration of high-order harmonics, might need finer quadratures. In all MP_N calculations presented in this document, unless differently specified, we have evaluated coefficients using the 10×10 quadrature.

4.4.7 Assessment of the MP_N computational cost

The number of angular degrees of freedom is a very useful parameter allowing to compare different numerical methods, using different mathematical representations of the same function. However, in our case the number of ADoF is not fully representative of the actual computational cost of the methods. Hence, ADoF yields an incomplete information of the actual advantages and disadvantages of MP_N.

As last step of our study on MP_N, we propose an analysis of the computational cost. We considered once again the homogeneous Cube Test with no scattering (anisotropy order $K = 0$). Since the problem is purely absorbing, the analysis focusses on the computational cost of a single inner iteration. We run S_N and MP_N monitoring the calculation times for different angular refinements. All calculations have been run using the linear-characteristics method, a step size equal to 2.5 cm, and 10×10 quadrature points per solid angle for the coefficient integration.

Figure 4.21 shows the computational cost of the different angular discretization methods with respect to the number of angular degrees of freedom. At first, we see that all methods

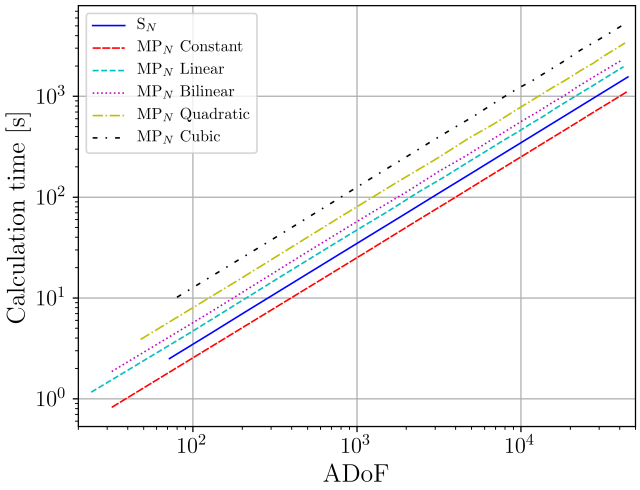


Fig. 4.21 Homogeneous pure absorber Cube Test, representation of the computational cost with respect to the number of angular degrees of freedom (ADoF). Comparison of S$_N$ and MP$_N$ using different local angular basis. The spatial discretization is fixed with $\Delta = 2.5$ cm.

have a linear behaviour with respect to the number of angular degrees of freedom and the cost scaling is of the same order for S$_N$ and MP$_N$. We can also see that the calculation cost of MP$_N$ increases as the dimension of the local angular base increases, which is coherent with the increase of the size of the MP$_N$ matrices with local angular basis size N_ℓ.

We remark that most of the time of an MP$_N$ calculation is spent solving the sweep, while the time spent evaluating the coefficients is minimal. By solving the problem using Linear MP$_N$, linear-characteristics, $\Delta = 2.5$ cm, 10×10 solid angles per octant, the 10×10 quadrature, zero anisotropy, with our actual FORTRAN implementation in an 8-byte machine precision, the time spent evaluating the coefficients is less than the 0.5% of the overall calculation time. Likewise, for other parameters settings, in any of our tests with anisotropy order $K = 0$, this has never been the most relevant component of the computational cost.

In Section 4.2.4 we have discussed how MP$_N$ matrices sizes retain also a dependence with the dimension of the spherical harmonic basis. Consequently, the computational cost is expected to increase with the problem anisotropy. This is the case also for S$_N$ but, in S$_N$ the number of harmonics influences only the size of the moment-to-discrete and discrete-to-moment matrices. The increased size of MP$_N$ matrices impacts not only the computational cost of the sweep, but also the cost of their on-the-fly evaluation.

Figure 4.22 displays the variation of computational cost, normalized to the cost of the calculation at zero-anisotropy, with respect to the number of spherical harmonics used to expand the source term (and so also with respect to the anisotropy order of the scattering operator). The figure shows how the increase of the calculation time with the basis size is much stronger for MP$_N$ than for S$_N$. While for S$_N$ a P$_3$ scattering ($K = 3$) calculation is

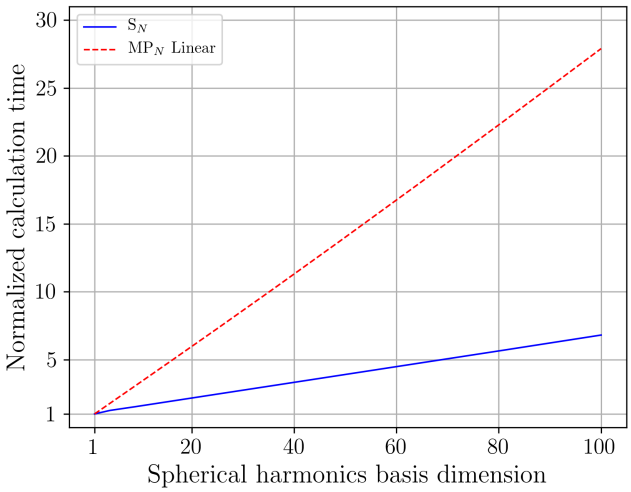


Fig. 4.22 Homogeneous pure absorber Cube Test, representation of the computational cost, normalized to the cost of the zero-anisotropy calculation, with respect to the size of the spherical harmonics basis. Comparison between S_N and MP_N Linear. The spatial discretization is fixed with $\Delta = 2.5$ cm.

about 3 times more expensive than the same calculation at anisotropy zero, for MP_N this calculation is about 10 times more expensive. At anisotropy zero, MP_N spend 99.7% of the time performing the sweep and only the remaining 0.3% evaluating coefficients, while at $K = 9$ the time spent in the sweep reduced to 98.3% of the total. At present, the scaling of the computational cost with the anisotropy order is probably the most limiting factor of the method in practical applications.

4.5 Partial Conclusion

The MP_N method has been conceived with the idea of introducing an angular discretization capable of better representing streaming effects, yet maintaining the same formalism than the S_N short-characteristics discretization method. This is achieved by redefining fluxes at the interfaces of the Cartesian regions using a piecewise continuous angular expansion, and then performing a Galerkin projection of the transmission equation on that basis. The resulting equations couple simultaneously different angular moments of the aforementioned angular expansion within the same solid angle. Since no coupling is present between solid angles, MP_N retains the block diagonal structure of the streaming-plus-removal operator. Moreover, MP_N is conservative, and allows for a straightforward inversion of the system of equations by means of a standard sweep algorithm.

We verified MP_N in both diffusive and purely absorbing problems. Our tests have shown how MP_N, for relatively coarse angular discretizations and with the present choice of the

angular basis, is affected by ray-effect to the same extent as S_N. MP_N exhibits a threshold behaviour: for low angular refinements, it shows a slow error convergence with the angular refinement, and errors generally larger than those yield by S_N. Beyond a certain angular refinement, instead, MP_N has a much faster error convergence than S_N. MP_N allows to achieve results with a very high accuracy, and with a smaller number of degrees of freedom than S_N. MP_N is less effective in highly heterogeneous problems, requiring a very fine angular discretization to succeed in providing better results than S_N (for an equal number of angular degrees of freedom). However, also in these cases the MP_N discretization shows a superior convergence of the error. Its effectiveness has been proven also for moderately diffusive problems. Our tests on a homogeneous problem showed how the gain in accuracy provided by MP_N with respect to S_N increases with scattering. On average, however, the computational cost of MP_N is higher than that of S_N. Moreover, the gap between the two methods in terms of computational cost increases with the anisotropy order of the calculation.

The MP_N formulation allows in principle to discretize the unit sphere using different grids and different angular bases. A further exploration is required in order to assess if there is any angular representation which is better suited for MP_N. In addition, by using a discontinuous finite element-like representation of the interface fluxes, MP_N provides an ideal framework for the application of adaptive angular discretizations. The development of adaptive methods, as well as the application of MP_N to spatial discretizations with in-region heterogeneities, is planned for future researches.

Another research topic consists in the mitigation of the dependence of the computational cost of MP_N with the anisotropy order of the problem. To do so, one could think of replacing the projection onto the spherical harmonics basis in the MP_N matrices definition. By maintaining the same MP_N formulation, this is possible only by introducing a different angular representation of the source term. In order to limit the angular coupling, the basis function expanding the source should have a limited angular support, or possibly not generating a coupling between all source angular moments within each solid angles. For example, this can be obtained by expanding the source using the same local angular basis used in the expansion of the interface fluxes. The major difficulty of such a strategy lies in the definition of a conservative transformation between this representation, and the spherical harmonics basis employed by the scattering kernel. Alternatively, one may also think of employing a different scattering kernels, coupling fluxes and sources between solid angles, [18].

Although MP_N does not require the storage of its coefficients, as these are computed on-the-fly, the memory occupation is still a limiting factor for the method. Since shielding calculations typically require very fine spatial and angular discretization, even the only storage of the flux may require terabytes of memory to be stored. Without the use of Domain Decomposition methods, this type of calculations would not be feasible. At present, IDT is already capable of running in domain decomposition, and MP_N can be easily integrated in

the standard DDM computational scheme. In Chapter 5, we will verify and test the MP_N method in domain decomposition, when applied to realistic shielding calculations.

Chapter 5

Analysis and verification of the IDT solver in radiation shielding calculations

Throughout the previous chapters we have developed a number of numerical methods aimed to improve the accuracy of deterministic radiation-shielding calculations. Primarily, we dealt with localised sources for ray-effect mitigation through a specialized treatment of the uncollided flux and the calculation of the first collision source. The FCS module has been improved so as to perform both serial and domain decomposed calculations. Then, we conceived the Multi-P_N method: an alternative angular discretization for transport calculations, aimed to provide a better representation of streaming effects.

In this chapter, we will first see how FCS and MP_N have been integrated into the DDM calculation scheme of IDT and we will discuss their main features. By focusing on the calculation of the collided component of the flux, we will assess the scalability of the S_N and the MP_N methods in domain decomposition calculations. Then, we will assess the convergence properties of the Block-Jacobi iterations used in DDM, and the actual computational cost of the calculation. Finally, we will verify the DDM calculation scheme on the Prob500 benchmark problem presented earlier in Chapter 4.

5.1 Presentation of the domain decomposition calculation scheme for radiation shielding applications

A transport calculation using the FCS method, as seen in Section 1.3.4, is comprised of two steps: a first calculation of the uncollided flux, and a second calculation for the collided component. While performing calculations on distributed-memory parallel architectures,

both steps have to run in domain decomposition mode. Let us now briefly summarize the calculation scheme, putting the different ingredients together.

At first, the external source Q_{ext} is decomposed into a set of point sources, as in Eq. (1.77). As described in Chapter 3, the uncollided flux is integrated for each point source, and for each subdomain $\mathcal{D}_i$, solving Eqs. (3.2) and (3.3). Then, the uncollided flux ψ_i^u is retrieved for each subdomain $\mathcal{D}_i$ summing the contributions of the different point sources, as in Eq. (3.4). By independently applying in each subdomain the scattering operator to the uncollided flux,

$$Q_{FCS,i} = \mathcal{H}_i \psi_i^u, \quad i = 1, \ldots, N_i, \tag{5.1}$$

the first collision source $Q_{FCS,i}$ is retrieved, for each subdomain $\mathcal{D}_i$. The next step consists in a standard transport calculation for the evaluation of the collided component of the flux. This is performed by rewriting the original problem in Eq. (1.75), i.e.

$$\mathcal{L}\psi^c = \mathcal{H}\psi^c + \mathcal{H}\psi^u, \tag{5.2}$$

in the form

$$\begin{cases} (\mathcal{L} - \mathcal{H})_i \ \psi_i^{c,(e+1)}(\mathbf{x}) = Q_i^{(e+1)} & \text{for } \mathbf{x} \in \mathcal{X}_i, \\ \psi_i^{c,(e+1)}(\mathbf{x}) = \psi^{c,-,(e+1)}(\mathbf{x}) & \text{for } \mathbf{x} \in \partial\mathcal{X}_i^-, \end{cases} \tag{5.3}$$

with the external source

$$Q_i^{(e+1)} = Q_{FCS,i} + \mathcal{F}\psi_i^{c,(e)}. \tag{5.4}$$

Incoming interface fluxes on the domain boundaries $\partial\mathcal{D}$ are retrieved by applying boundary conditions, while fluxes on boundaries between subdomain are exchanged and updated by the Gauss-Jacobi iterations. The calculation is solved using the PMBJ scheme described in Section 1.4.3. Once the calculation of the collided flux $\psi_i^{c,(e+1)}$ is converged on the asymptotic solution ψ_i^c, the flux is retrieved by simply summing the two flux contributions,

$$\psi_i = \psi_i^u + \psi_i^c, \quad i = 1, \ldots, N_i. \tag{5.5}$$

Remark that this operation does not need inter-communications between subdomains and can be performed independently.

Transport calculations with FCS require the storage of two multigroup fluxes, namely the collided and the uncollided components. The memory required for the flux storage is therefore double than a standard transport calculation, or even triple, if the multigroup first collision source is also stored in memory. Therefore, the increased memory demand of FCS makes DDM even more important.

The calculation of the collided flux, described in Eq. (5.3), can be performed either with S_N or MP_N. From the point of view of the DDM algorithm, the only difference between the two lies in the representation of the fluxes at the region interfaces. In S_N, at each iteration,

subdomains exchange, for each direction, the spatial moments of the boundary flux. In MP_N, instead, subdomains exchange the spatial-angular moments of the boundary flux for each solid angle. As for Discrete Ordinates, the continuity of the flux at the subdomains interfaces is also preserved when using MP_N. No additional approximation is done when exchanging interface fluxes (due to the conformal spatial mesh and to the uniform angular discretization across the domain). Hence, the transmission between subdomains is exact.

The DDM employed by IDT has been primarily conceived to solve reactor core problems. In the presence of diffusive materials, the flux is mostly isotropic, and the error estimator based on partial currents (see Eq. (1.89)) is representative of the actual convergence of the calculation, [58]. However, in radiation shielding calculations, the flux can be very anisotropic, and its angular representation may involve higher order moments. The error on partial currents is representative of the convergence of diffusive angular moments only, and not of the more anisotropic ones. This is why we opted to replace this with an error estimator based on an L^∞ norm of the error of the interfaces fluxes, i.e.

$$\epsilon_\psi^{(e+1)} = \underset{\substack{s\in\partial\mathcal{D}_i \\ i=1,N_i \\ g=1,G}}{max} \left\{ \underset{\substack{j\in\Gamma_i^s \\ d=1,D^-}}{max} \left| 1 - \frac{\psi_{i,g,j,d}^{-,(e+1)}}{\psi_{i,g,j,d}^{-,(e)}} \right| \right\}. \tag{5.6}$$

Solving with S_N, $\psi_{i,g,j,d}^{-,(e+1)}$ stays for the constant spatial component of the flux, defined on the region interface j, lying on surface s of subdomain i, in group g, and along direction d. D^- denotes the set of incoming directions of the angular quadrature, i.e. $\mathbf{\Omega}_d \cdot \mathbf{n} < 0$, with $\mathbf{n}$ the outgoing normal direction of interface Γ_i^s. Using MP_N method instead, the error is evaluated on the first spatial-angular component (constant in space and in angle) of the interface fluxes, for each incoming solid angle d (here using subscript d in place of n, which we have used in the derivation of MP_N in Section 4).

5.2 Quantification of the parallel performances of the DDM algorithm

The cost of uncollided flux calculation has been extensively discussed in Chapter 3. Due to the dependence between fluxes entering and exiting different subdomains, the calculation of subdomains must be performed sequentially. The parallel efficiency of this calculation is difficult to further optimize (while maintaining a purely spatial domain decomposition). However, for discretizations of the FCS streaming-plus-removal operator not excessively refined, it makes sense to believe that most of the computational time will be spent solving the second part of the calculation, namely the collided flux.

We now want to evaluate the parallel performances of the solver, excluding the contribution of the uncollided flux calculation. To this extent, we will first perform a strong scaling test.

Then, we will assess its effectiveness by monitoring the calculation time and the number of external iterations, for different numbers of cores and subdomains.

5.2.1 Assessment of the parallel performances of the PMBJ algorithm with S_N and MP_N

The strong scaling test is useful to quantify to what extent the solver is capable of exploiting an increased amount of hardware resources for speeding up the calculation. In practice, the test consists in solving the same problem with an increasing number of cores, and monitoring at each step the elapsed time of the calculation.

For this analysis we considered the Prob500 benchmark problem, presented in Section 4.1.1. Since the analysis is focussed exclusively on the properties of the transport calculation of the collided flux, FCS has not been employed. The problem has been solved using a spatial mesh made of about 52000 regions and the linear characteristics. We employed first an S_N angular discretization with 200 directions, and then Linear MP_N with 72 solid angles. Calculations are run limiting the scattering anisotropy order K to 3. The test is done by progressively increasing the number of subdomains, and proportionally the number of cores. Specifically, the number of cores is set exactly equal to the number of subdomains, so that they can all be solved in parallel. In order to be conservative, all subdomains are solved on different MPI processes, which are generally associated to larger computational costs because of communications. The calculation is done for a fixed number of iterations. The number of external iterations as well as the number of inner iterations, in each of the 42 energy groups, are set equal to 10. Since the problem has no up-scattering, thermal iterations are not performed.

Figure 5.1 shows the speedup factor of the calculation with respect to the number of subdomains. The blue and red lines distinguish respectively the speedup of S_N and MP_N, while the values on the sides of the curves are parallel efficiencies, expressed in percent (defined in Section 1.4.1).

With S_N, the PMBJ domain decomposition shows values of the efficiency above 50% up to a thousand cores, which is coherent with the results of previous publications, [1, 58]. However, the performances obtained with MP_N result to be degraded as compared to those of S_N. This strong under-performance can be mainly attributed to the on-the-fly calculation of MP_N coefficients. In fact, as we will discuss in the following, their calculation may cause load imbalances between the different subdomains.

On the MP_N parallel performances and the calculation of coefficients in domain decomposition

We recall that in the current implementation, the MP_N coefficients are recomputed at each inner iteration, and their calculation is more computationally expensive than the one of the

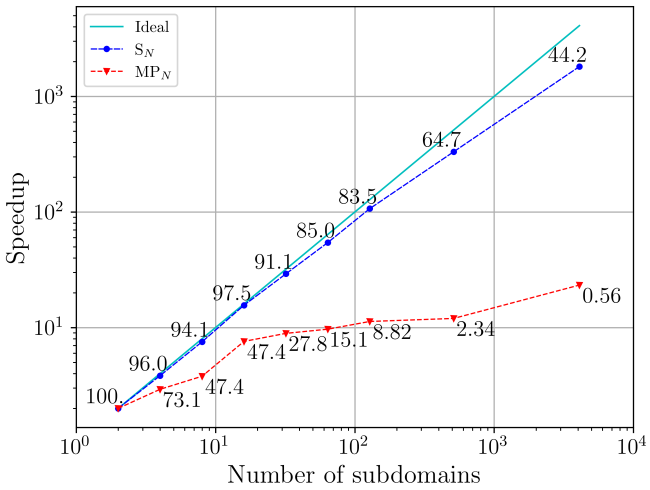


Fig. 5.1 Prob500, strong scaling test, representation of speedup factor with S_N and MP_N for different number of subdomains/cores. The values indicate the parallel efficiency.

short-characteristics coefficients. A single set of coefficients is evaluated for all "optically" equal regions, i.e. having equal sizes and medium. The cost of the coefficient evaluation, therefore, does not scale linearly with the number of spatial regions per subdomain, but it is proportional to the number of "region types" contained in the subdomain. Numerically, the time of the serial calculation can be approximated as

$$t(1) = N_m T_c + N_r T_s$$

where N_r and N_m are respectively the number of regions and the number of region types. T_c is the time spent evaluating coefficients per region type, and T_s is the average time spent solving for the flux in one single region.

By using DDM, the domain is partitioned into smaller subdomains while keeping the number of regions per subdomain approximatively constant. Therefore, the time spent performing the spatial-angular sweep decreases proportionally as the number of subdomains increases. However, one single subdomain may contain several region types, and, in the worst case scenario, the number of region types contained in a subdomain does not change as the number of subdomains varies. In such a case, N_m remains constant while the number of regions reduces to N_r/N_i, with N_i being the number of subdomains. The time spent solving the problem subdivided into N_i subdomains is approximatively

$$t(N_i) = N_m T_c + \frac{N_r}{N_i} T_s.$$

This corresponds to the case where the calculation of the coefficients is not parallelized, so, the parallel performances are limited by the Amdahl's law. The maximum speedup achievable is therefore

$$S_A(N_i) = 1 + \frac{N_r T_s}{N_m T_c}.$$

The parallel efficiency, being S_A/N_i, follows a hyperbolic trend.

From the results of our analysis on the computational cost of MP_N at varying anisotropy order, presented in Section 4.4.7, we estimated that, for a calculation with $K = 3$, Linear MP_N, and a 10×10 product-quadrature for the coefficients integration, the ratio T_c/T_s is about 42. Taking this estimate for good, and using $N_r = 52000$ and $N_m = 26$, which are the number of regions and region types for the Prob500 test, the maximum speed up is about 48. The resulting parallel efficiency at 4000 cores is equal to 1.2%. Thus, we see how the coefficients storage may be largely responsible for the reduced parallel efficiency of MP_N, which is compatible with the measured efficiency shown in Figure 5.1. In order to improve the parallel scalability of MP_N calculations, it is therefore essential to store these coefficients.

However, this option could be quite expensive in terms of memory occupation. For example, in a calculation using linear-characteristics, linear MP_N, 200 solid angles and anisotropy order $K = 3$, the storage of a single set of coefficients would require about 6.7 MB of memory, per energy group and per region type. Namely, for a problem similar to Prob500, using 42 groups and 26 region types, the memory occupation would rise up to 7.3GB. Alternatively, the coefficients can be computed and cached for each group at a time, allowing them to be reused during internal iterations.

5.2.2 Assessment of the effect of Block-Jacobi iterations on the convergence properties of the transport operator

As previously mentioned, the domain-decomposition method resorts to the iterative Block-Jacobi method to solve the transport problem. The convergence properties of the Block-Jacobi depend on the choice of the decomposition. In particular, in the presence of optically thin media, where subdomains are strongly spatially-coupled, the convergence may be extremely slow, [95, 96]. In order to cope with this problem, as well as to improve the convergence speed of internal and thermal iterations, a common practice is to use preconditioning techniques. Radiation shielding calculations could certainly benefit from accelerations. Still, this topic is left to future investigations.

In order to measure the effective speedup of the DDM, we performed a second test on Prob500, this time by fixing the convergence criterion to $\tau = 10^{-6}$. For each subdomain, we also required inner iterations to be fully converged. We underline that these tests do not use any preconditioner of the Block-Jacobi iterations.

Figure 5.2 shows how the number of Block-Jacobi (or externals) iterations required to reach the convergence increases with the number of subdomains. This behaviour can be

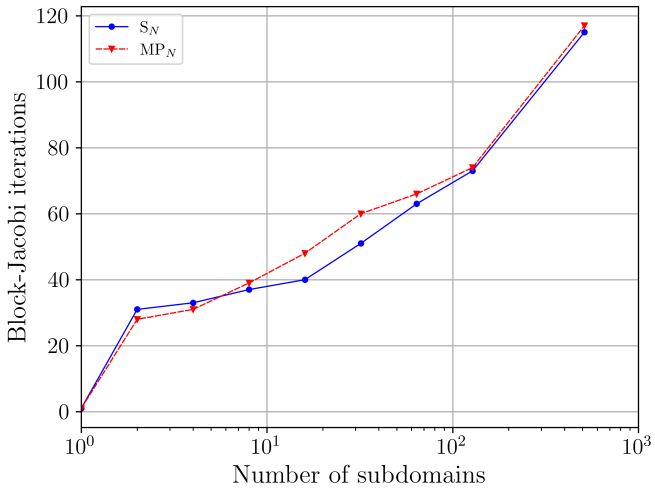


Fig. 5.2 Prob500, assessment of the number of Block-Jabobi iterations required for the convergence of the DDM calculation with different subdomain numbers.

observed both using the S_N and the MP_N angular discretizations. The two methods display similar increase in terms of the number of iterations.

Figure 5.3 instead shows how the computational time varies as the number of subdomains and cores is increased (the number of cores is set equal to the number of subdomains). At present, without any coefficient storage, the use of DDM does not provide any relevant benefit in terms of computing time while using MP_N. By using S_N, instead, an increase of the number of cores from 4 to 4000 has reduced the calculation time of about a factor 50.

5.3 Assessment of the accuracy of the IDT solver with First Collision Source and Domain Decomposition

We now want to assess the actual accuracy of the solver with respect to a Monte Carlo reference. To do so, we considered once again the Prob500 test, presented in Section 4.1.1.

The problem is solved using a fine spatial discretization made of about $3.8 \cdot 10^5$ regions, and linear characteristics. As for the angle, we employed:

- a coarse S_N discretization, using a product-type Chebyshev-Legendre angular quadrature made of 800 directions;
- a fine S_N discretization, using the same type of quadrature, but with 3528 directions;
- a coarse MP_N discretization, with the linear order of the local angular basis, and 6×6 solid angles per octant (ADoF= 864);

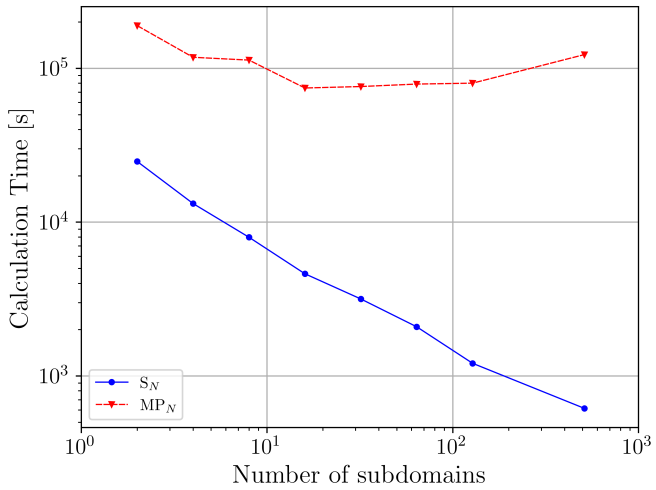


Fig. 5.3 Prob500, assessment of the computational cost of the DDM calculation with different subdomain numbers.

- a fine Linear MP_N discretization, with 12×12 solid angles per octant (ADoF= 3456).

The problem is solved using the macroscopic cross section library presented in Section 4.1.1, using the Vitamin-B6 energy discretization made of 42 groups. We set the scattering anisotropy order $K = 0$, due to the incapability of solving a Monte Carlo reference calculation with higher scattering anisotropies and multigroup cross sections. The FCS calculation is run using the MOC method with a trajectory spacing equal to 0.01 cm and an EXS grid spacing equal to 0.002 cm. The source is approximated by a single point source. Since the point source lies also on the symmetry axis of the problem, the reflection boundary condition applied on the y- plane is taken into account exactly. The calculation is solved subdividing the domain into $20 \times 10 \times 10$ subdomains and running on 2000 cores, one for each subdomain, in a cluster using AMD EPYC Milan 7763 CPUs. The convergence criterion is set imposing $\tau = 10^{-6}$ for both inner and external iterations.

The reference result has been evaluated using the Tripoli-4 Monte Carlo code, [4], with multigroup cross sections and isotropic scattering. Finding no relevant benefits through the use of variance reduction methods, the calculation has been carried out with the only use of population control methods, such as splitting and implicit capture. The reference is obtained simulating about $2 \cdot 10^{12}$ particle histories. In both the deterministic and the Monte Carlo calculations the source has been approximated as a point emitting uniformly in the energy range 1.33-0.8 MeV, corresponding to groups 22 and 23.

The objective of the calculations is the estimation of the average flux in the five detectors coloured in green in Figure 4.1. Detectors are numbered from right to left, i.e. detector 1 is the nearest to the source, and detector 5 is the farthest.

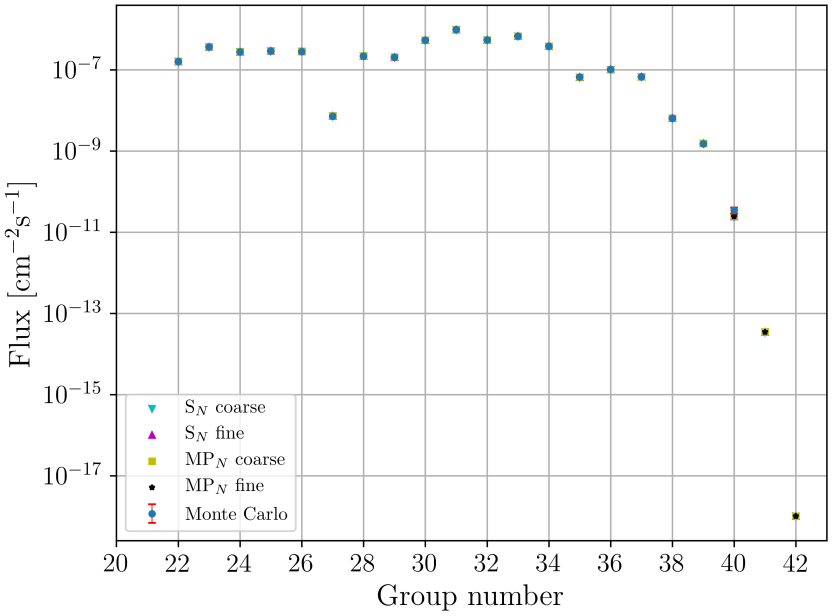


Fig. 5.4 Prob500, verification of the IDT calculation with FCS and DDM, flux in detector 1.

Figure 5.4 shows the flux measured in detectors 1 using IDT and Tripoli-4. The flux in other detectors is not displayed because showing results similar to detector 1. Figures 5.5, 5.6 and 5.7 show, for the five detectors, the values of the relative error of the flux estimation with respect to the Monte Carlo reference. We remark that errors can be quantified only for groups where the reference is associated to a sufficiently small standard deviation. Where the value of the error is smaller than the standard deviation, we can not draw any reliable conclusions. The error estimation is impossible in groups 41 and 42 because the Monte Carlo calculation has given no estimate for these two groups.

Results are summarized in Tables 5.1 and 5.2. Table 5.1 shows the value of the average flux, integrated over all energy groups, in the five detectors, together with their relative error. Table 5.2, instead, shows the L^1, the RMS, and the L^∞ norm of the relative error of the average flux, over the five detectors.

The IDT results manifest a good agreement with the reference calculation. The errors of the fine S_N and MP_N calculations are similar, with errors smaller than 1% for most of the energy groups. Differently from what we observed in Chapter 4, the results shown by MP_N with the coarse refinement are significantly more accurate than those provided by S_N, at approximatively equal number of ADoF. In fact, in Chapter 4, we have seen how MP_N outperforms S_N for very fine angular discretizations, and not for coarse ones. This, while highlighting the quality of the method, suggests that further analyses on MP_N are still required.

Detector	Calculation	Flux $[\cdot 10^{-6}$ cm^{-2}s$^{-1}]$	Relative Error [%]
Detector 1	Tripoli-4	5.153	–
	S_N coarse	5.086	1.30
	S_N fine	5.146	0.13
	MP_N coarse	5.186	0.64
	MP_N fine	5.142	0.22
Detector 2	Tripoli-4	4.774	–
	S_N coarse	4.806	0.67
	S_N fine	4.797	0.47
	MP_N coarse	4.797	0.47
	MP_N fine	4.789	0.30
Detector 3	Tripoli-4	4.392	–
	S_N coarse	4.547	3.51
	S_N fine	4.393	0.01
	MP_N coarse	4.343	1.13
	MP_N fine	4.407	0.33
Detector 4	Tripoli-4	4.017	–
	S_N coarse	4.169	3.79
	S_N fine	4.005	0.31
	MP_N coarse	3.985	0.80
	MP_N fine	4.052	0.86
Detector 5	Tripoli-4	3.642	–
	S_N coarse	3.718	2.09
	S_N fine	3.676	0.94
	MP_N coarse	3.683	1.12
	MP_N fine	3.675	0.91

Table 5.1 Prob500, energy integrated flux and relative error

We remark that, since the multigroup approximation is used in both IDT calculation and the reference, the error is representative of the only spatial and angular discretization we used. In this respect, in order to further decrease the error we should also refine the spatial discretization, and not only the angular one.

5.4 Partial conclusion

In this chapter we tested the IDT solver for applications in radiation shielding calculations. To do so, we considered Prob500, a model representative of the class of problems we want to address.

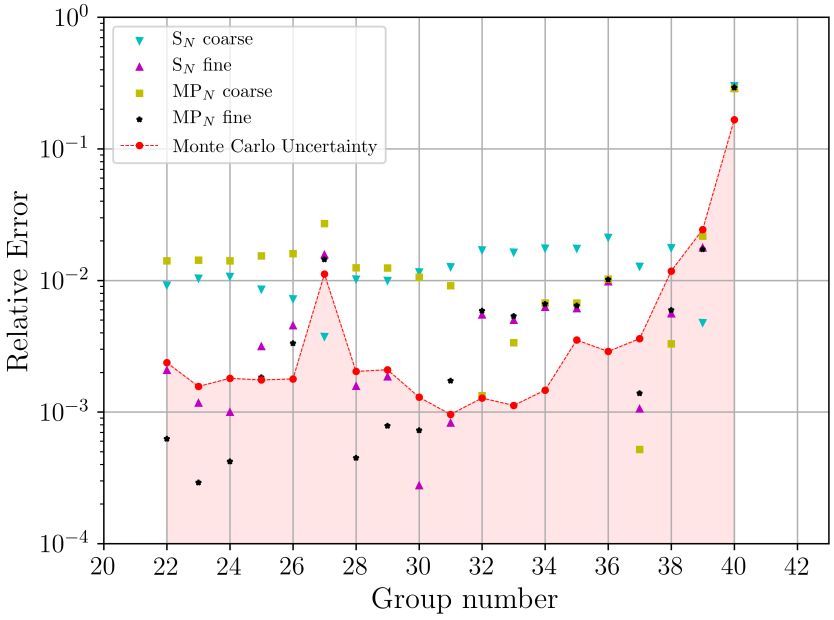
Calculation	L^1-norm [%]	RMS [%]	L^∞-norm [%]
S_N coarse	2.27	2.57	3.79
S_N fine	0.37	0.49	0.94
MP_N coarse	0.83	0.87	1.13
MP_N fine	0.52	0.60	0.91

Table 5.2 Prob500, relative error norms of the energy integrated flux, evaluated on the 5 detectors

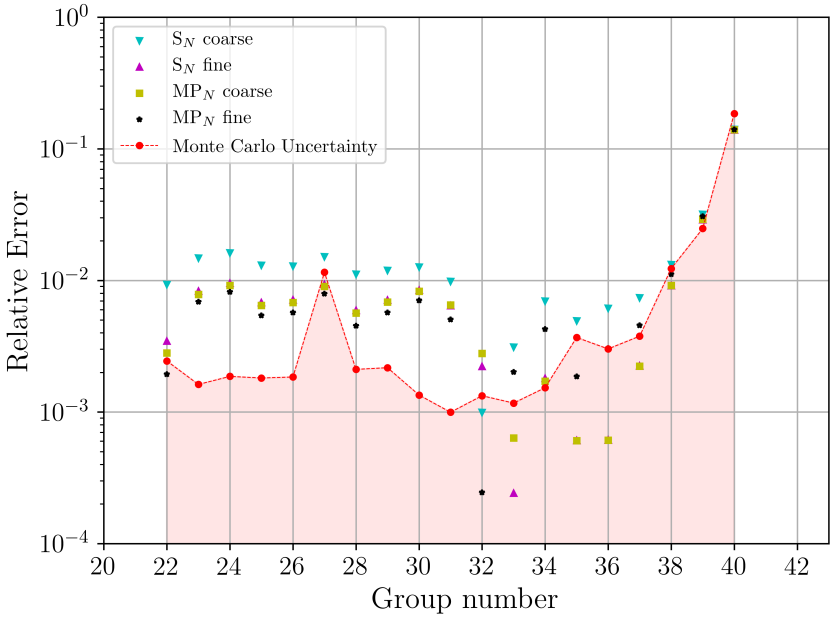
With an initial strong scaling test, we have highlighted that the present implementation of MP_N does not scale well for a high number of cores. These poor performances are probably due to the strategy used for the coefficients calculation.

Then, we put in evidence how the convergence properties of the PMBJ deteriorate as the number of subdomains is increased. These, as expected, are found for both S_N and MP_N angular discretizations. Large numbers of Block-Jabobi iterations are generally associated with increased computational costs. In order to speed up transport calculations, the use of acceleration methods is required. An effective acceleration must be able to speed up the calculation by converging all the modes that are slower to converge with transport. If in diffusive problems the slowest mode is represented by the constant moment of the flux, in problems dominated by particle propagation we expect that the acceleration method will have to deal also with the slow convergence of higher-order moments. Transport synthetic accelerations, such as the BPA acceleration, may be better suited for this type of applications than Diffusion synthetic accelerations. Angular multi-grid accelerations may also be a viable alternative. We expect that MP_N may lend itself well to this type of accelerations. This topic is left to further investigations.

Finally, we verified the DDM calculation on a representative problem. We showed how the solver is effectively capable of solving radiation shielding calculations. The joint use of FCS, with both S_N and MP_N, resulted in accurate estimations with error smaller than 1% for most of the energy groups. MP_N also showed more accurate results than S_N when using a relatively coarse refinement of the angular discretization.

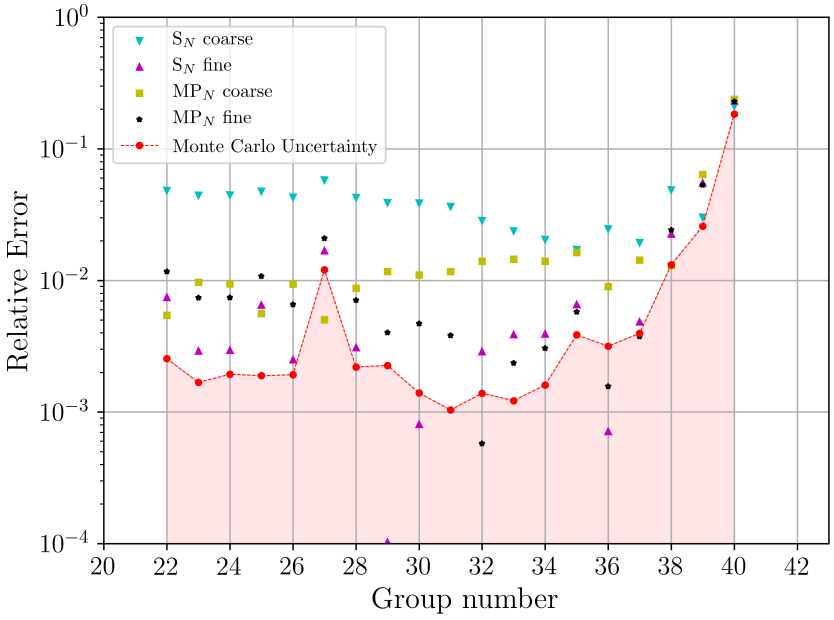


(a) Detector 1

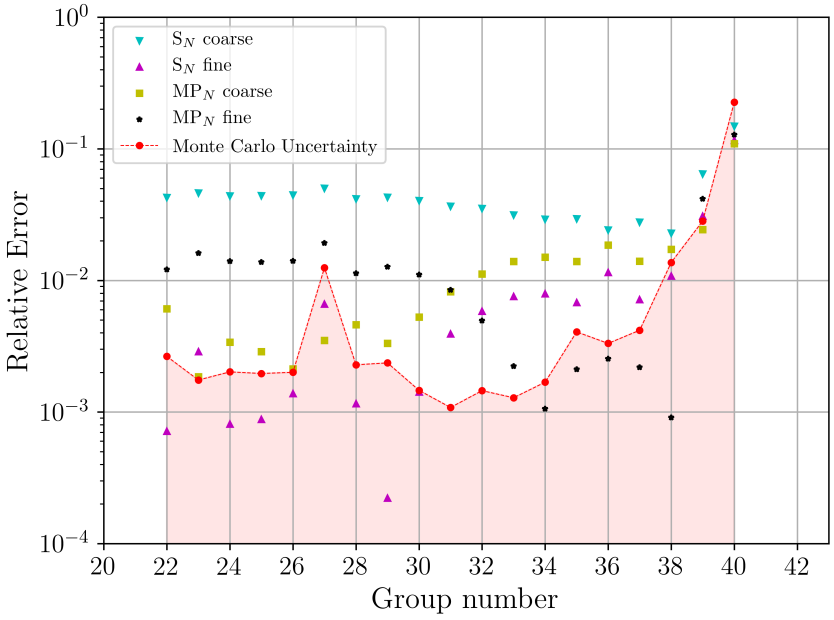


(b) Detector 2

Fig. 5.5 Prob500, verification of the IDT calculation with FCS and DDM, relative error of the flux in detectors 1 and 2.



(a) Detector 3



(b) Detector 4

Fig. 5.6 Prob500, verification of the IDT calculation with FCS and DDM, relative error of the flux in detectors 3 and 4.

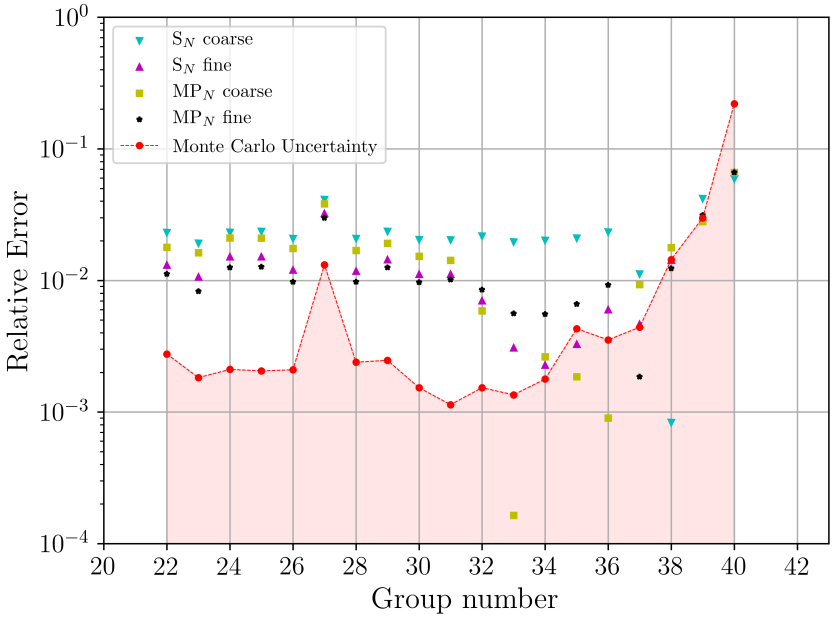


Fig. 5.7 Prob500, verification of the IDT calculation with FCS and DDM, relative error of the flux in detector 5.

Conclusion

In this manuscript, new numerical methods have been developed for the solution of deterministic shielding calculations.

In order to adequately represent the flux emitted by localised sources, four different integration strategies have been developed within the First Collision Source (FCS) module of the IDT solver. There are the MOC, the TS-MOC, the MC-MOC, and the QP methods. Techniques such as MOC and TS-MOC, relying on the use of Method of Characteristics and angular quadratures, have provided results with the highest accuracy. However, while the MOC method showed accurate results for problems of different sizes and varying degrees of heterogeneity, TS-MOC proved to be convenient only for very large size problems. The QP method, which preforms a region-by-region integration of the uncollided flux by means of a spatial quadrature, has provided satisfactory results in terms of calculation time only for very small problems. The QP method, however, is able to progressively refine the order of the quadrature until a stopping criterion, based on an error estimator, is reached. The less accurate technique resulted to be the MC-MOC method, which relies on the Method of Characteristics, but uses a random sampling of the propagation direction.

A Domain Decomposition Method for FCS has been also developed. This relies on an interpolative approximation of the equivalent cross sections (EXS) to reconstruct the flux at the interfaces of subdomains not containing the source. The method uses two different boundary meshes: one as support for MOC integration, and another defining the points in which the equivalent cross section is computed. The refinement of the first mesh determines the accuracy on the uncollided flux integration in each subdomain. The refinement of the second, instead, controls the accuracy of the EXS approximation. Tests have shown that the error introduced by the EXS approximation is typically larger than the one of the MOC integration, thus justifying our choice of controlling the two sources of error separately. The parallel efficiency of the FCS method is highly dependent on the refinement of the two boundary meshes. In the typical use case, i.e. with EXS boundary mesh finer than the MOC mesh, the efficiency is lower than in the case with a coarser EXS mesh. However, in the presence of several point sources, it has been qualitatively demonstrated that the parallel efficiency of the algorithm is higher.

In order to mitigate ray-effect generated by scattering sources, the MP_N angular discretization method has been designed. The MP_N method is based on the integral formulation of the transport equation, discretized using the short characteristics method, and on a piecewise polynomial angular expansion of the interface fluxes. For sufficiently large numbers of angular degrees of freedom, the method has shown an error convergence order about 2.5 larger than the one measured with the S_N method. The results have been confirmed in both homogeneous and heterogeneous problems. Furthermore, it has been seen that the accuracy of MP_N benefits from the presence of diffusive materials. As a drawback, for an equal number of angular degrees of freedom, MP_N has a higher computational cost than the S_N method.

In conclusion, the IDT solver has been tested using domain decomposition, with FCS, and employing both S_N and MP_N. The results have shown parallel performances larger than 50% up to 1000 cores using S_N. On the other hand, much lower parallel efficiencies have been observed with MP_N. It has been shown how this result is due in great part to the on-the-fly calculation of the MP_N coefficients. The solver has been verified on a realistic test case, comparing the results with a reference provided by the Tripoli-4 Monte Carlo solver. In this test, for relatively coarse angular discretizations, MP_N has provided more accurate results than S_N.

Perspectives

In the course of the present research, for reasons of time, a large number of ideas and possible developments have been left for further investigations. Among these, the most relevant are here summarized.

The strategy that has been employed for the treatment of spatially distributed sources consists in modelling the source as a set of uniformly-distributed point sources. This method is practical, but it can be poorly effective in modelling optically thick source volumes, [54]. In fact, in this case, the only particles that succeeds in exiting the source region are the ones that are emitted near the region boundaries. Therefore, it may be better to model this type of sources differently. A further investigation aimed at quantifying the impact of this source of error is therefore necessary.

Together with the FCS calculation, solvers such as ATTILA employ the Last-Collided Flux (LCF) method, i.e. they uses a method to reconstruct the flux in certain points of interest of the domain, e.g. in detectors, [10]. LCF method uses a fine transport operator similarly to the FCS, allowing to evaluate the flux with great accuracy, using the emission density calculated with a coarse angular discretization. These types of methods have not been addressed in this thesis, but their development can benefit from the the research on FCS methods.

At the time being, the major limitation of the FCS method is the impossibility of accounting for boundary conditions different from vacuum conditions. Further developments are therefore necessary.

In this thesis, the MP_N method has been investigated and analysed but its development remains at a preliminary stage. Further researches are therefore required. Among these, it is necessary to investigate the use of different angular bases for the interface flux expansion and the implementation of an effective strategy for storing in memory the MP_N coefficients. It is also worth investigating the possibility of using an angular representation of the source, different from the currently used spherical harmonics expansion, so as to avoid the couplings with all angular moments of the source within the MP_N matrices. Finally, it is advisable the development of acceleration methods for MP_N.

The IDT spatial discretization by short characteristics offers the possibility of representing medium heterogeneities within each Cartesian region. By doing so, in principle, it is possible

to represent any type of geometry. This option may be very useful in shielding calculations, however, this is not yet available for the MP_N method. Further development are therefore required.

In this research, only the PMBJ domain decomposition method has been considered. An exploration of other methods, together with the development of acceleration methods, may be of great interest.

Summary

Transport calculations for radiation shielding applications are nowadays of considerable practical interest. These include calculations of activation of structural materials, estimations of radiation doses, heating of components, and sizing of shieldings. In order to provide accurate estimations within a reduced calculation time, deterministic codes require appropriate discretization methods to represent the actual physics of the problem. In a special way, streaming effects are particularly challenging for common angular discretization methods such as S_N. The solutions yield by S_N may be affected by numerical artifacts, known with the name of ray-effect. The smaller the source, and the coarser the angular discretization, the more severe will be the ray-effect. Moreover, given the characteristic size of radiation shielding problems, the use of Domain Decomposition Methods is essential. The objective of this thesis is to investigate and develop a number of numerical methods, within the deterministic IDT solver, in order to improve the accuracy and the performances of the solver when applied to shielding calculations.

At first, the research dealt with the First Collision Source (FCS), which is a set of methods useful for calculating the uncollided flux emitted from localized sources. Four different discretization techniques for FCS have been developed. These come with the aim to identify the best method for integrating the uncollided flux. The four methods have been compared in a series of test problems, highlighting their advantages and disadvantages. The MOC and TS-MOC methods appeared to be the ones leading to the lowest errors within the smallest computational times.

Then, the FCS method has been integrated within the domain decomposition environment of the IDT solver. The Equivalent Cross Section approximation has been introduced in order to interpolate fluxes at the interfaces between subdomains. A parallel FCS method relying on an efficient scheduling algorithm has been developed. Finally, the quality of FCS has been verified and analysed by solving calculations on parallel distributed-memory architectures. Results showed how the method allows for a precise control of the error of the FCS approximations.

In order to adequately represent particle streaming effects, it is useful to employ a suitable angular discretization. After a detailed presentation of the problem, an alternative angular discretization method, called MP_N has been developed. It is based on a discontinuous

polynomial expansion of the flux at the interfaces of Cartesian regions. A series of tests showed how the error of S_N and MP_N are comparable when the number of degrees of freedom is relatively small. However, the MP_N method has better convergence properties once the asymptotic regime of convergence is reached.

Later, the ability of IDT to solve real-size radiation shielding calculations with FCS and domain decomposition has been verified. It has been seen how the parallel performances of MP_N, at present, are still limited due to the cost of the evaluation of the coefficients. However, the parallel capabilities of the DDM scheme based on PMBJ iterations have been verified. Finally, the results provided by IDT, with both S_N and MP_N, have been verified using the Tripoli-4 Monte Carlo code. In conclusion, a number of possible improvements, as well as several topics for future research, have been identified.

Sommaire

Les calculs de transport pour les applications de radioprotection présentent aujourd'hui un intérêt pratique considérable. Il s'agit notamment des calculs d'activation des matériaux de structure, des estimations des doses de rayonnement, de l'échauffement des composants et du dimensionnement des protections. Afin de fournir des estimations précises dans un temps de calcul réaliste, les codes déterministes nécessitent des méthodes de discrétisation appropriées pour représenter la physique réelle du problème. En particulier, les effets de streaming sont particulièrement difficiles pour les méthodes de discrétisation angulaire courantes telles que la méthode S_N. Les solutions produites en S_N peuvent être affectées par des artifacts numériques, connus sous le nom d'effet de raie. Plus la source est petite, et plus la discrétisation angulaire est grossière, plus l'effet de raie sera important. De plus, étant donné la taille caractéristique des problèmes de radioprotection, l'utilisation des méthodes de décomposition de domaine est essentielle. L'objectif de cette thèse est d'étudier un certain nombre de méthodes numériques, dans le cadre du solveur déterministe IDT, afin d'effectuer des calculs de radioprotection avec une grande précision et un coût de calcul relativement faible.

Dans un premier temps, la recherche s'est concentrée sur les méthodes pour le calcul de la source de première collision (FCS), qui visent à obtenir le flux sans choc provenant de sources localisées. Quatre techniques de discrétisation différentes ont été développées pour la méthode FCS. Elles ont été comparées sur une série de problèmes tests, et leurs avantages et inconvénients ont été mis en évidence. Les méthodes MOC et TS-MOC se sont avérées être celles qui présentaient les erreurs les plus faibles dans les temps de calcul les plus courts.

Ensuite, la méthode FCS a été intégrée dans l'environnement de décomposition de domaine du solveur IDT. Une approximation appelée section efficace équivalente a été introduite pour interpoler les flux aux interfaces entre les sous-domaines. Un algorithme d'ordonnancement a ensuite été développé pour gérer les communications et les synchronisations. Enfin, la qualité des calculs de FCS a été vérifiée et analysée sur des architectures parallèles à mémoire distribuée. La méthode mise en place permet un contrôle précis de l'erreur des approximations. De plus, l'erreur augmente lentement avec le nombre de sous-domaines.

Afin de représenter correctement les effets de propagation, il est utile de recourir à une discrétisation angulaire appropriée. Suite à une analyse détaillée du problème, une méthode alternative de discrétisation angulaire, appelée MP_N, a été développée. La méthode est basée

sur un développement polynomial discontinu du flux aux interfaces des régions cartésiennes. La méthode MP_N a été vérifiée sur une série de tests avec des comparaisons avec la méthode S_N. Elle s'est avéré peu efficace pour atténuer l'effet de raie pour des petits nombres de degrés de liberté angulaire. Cependant, elle permet une convergence beaucoup plus rapide que la méthode S_N pour des discrétisations angulaires plus fines.

Ensuite, le schéma de calcul global avec FCS et décomposition de domaine a été analysé et la capacité de IDT à résoudre des calculs de radioprotection de taille réelle a été vérifiée. Les performances parallèles du MP_N sont, à l'heure actuelle, encore limitées en raison du coût de l'évaluation des coefficients. Cependant, les performances en parallèle du schéma DDM basé sur les itérations PMBJ ont pu être confirmées. Enfin, une validation a pu être réalisée en comparant les résultats IDT à ceux fournis par le code Monte Carlo Tripoli-4. En conclusion, un certain nombre d'améliorations possibles ont été identifiées qui constituent autant de sujets potentiels pour des travaux de recherche futurs.

References

[1] E. Masiello, R. Lenain et W. Ford, "3D heterogeneous Cartesian cells for transport-based core simulations," *Annals of Nuclear Energy*, t. 142, p. 107 364, 2020.

[2] I. Zmijarevic, "Multidimensional Discrete Ordinates Nodal and Characteristics Methods for APOLLO2 Code," in *Proc. Int. Cong. on Math. and Computations*, 1999.

[3] E. Masiello, "Improvements of the boundary projection acceleration technique applied to the discrete ordinates transport solver in XYZ geometries," in *Proc. Int. Cong. on Math. and Computations*, 2013.

[4] E. Brun et al., "TRIPOLI-4®, CEA, EDF and AREVA reference Monte Carlo code," *Annals of Nuclear Energy*, t. 82, p. 151-160, 2015.

[5] A. Haghighat et J. C. Wagner, "Monte Carlo variance reduction with deterministic importance functions," *Progress in Nuclear Energy*, t. 42, n° 1, p. 25-53, 2003.

[6] M. Nowak et al., "Accelerating Monte Carlo Shielding Calculations in TRIPOLI-4 with a Deterministic Adjoint Flux," *Nuclear Science and Engineering*, t. 193, p. 1-16, 2019.

[7] Falabino, Matteo et al., "Comparison of variance reduction methods in shielding problems with source multiplication," *Eur. Phys. J. Plus*, t. 136, n° 5, p. 598, 2021.

[8] T. A. Wareing et al., "Discontinuous Finite Element SN Methods on Three-Dimensional Unstructured Grids," *Nuclear Science and Engineering*, t. 138, n° 3, p. 256-268, 2001.

[9] T. M. Evans et al., "Denovo: A New Three-Dimensional Parallel Discrete Ordinates Code in SCALE," *Nuclear Technology*, t. 171, n° 2, p. 171-200, 2010.

[10] T. Wareing et al., "Capabilities of Attila for Radiation Protection and Shielding," *Transactions of American Nuclear Society*, t. 95, n° 1, p. 659-660, 2006.

[11] K. R. Koch, R. S. Baker et R. E. Alcouffe, "Solution of the first-order form of the 3-D discrete ordinates equations on a massively parallel machine," in *Proceedings of American Nuclear Society annual meeting*, 1991.

[12] G. I. Bell et S. Glasstone, *Nuclear Reactor Theory*. Van Nostrand Reinhold, 1970.

[13] R. Macfarlane et al., "The NJOY Nuclear Data Processing System, Version 2016," Los Alamos National Lab. (LANL), Los Alamos, NM (United States), Tech.Report, 2017.

[14] S. Mengelle, "Notice d'identification de la bibliothèque CEA V6 d'APOLLO2, DEN/DANS/DM2S/SERMA/LLPR/NT/14-5717/B," CEA, Tech.Report, 2015.

[15] D. Brown et al., "ENDF/B-VIII.0: The 8th Major Release of the Nuclear Reaction Data Library with CIELO-project Cross Sections, New Standards and Thermal Scattering Data," *Nuclear Data Sheets*, t. 148, p. 1-142, 2018.

[16] M. Coste-Delclaux, "Modélisation du phénomène d'autoprotection dans le code de transport multigroupe APOLLO2, Rapport CEA, ISSN 0429-3460," CEA Saclay, Direction des systèmes d'information, Tech.Report, 2006.

[17] A. Hébert, "A Review of Legacy and Advanced Self-Shielding Models for Lattice Calculations," *Nuclear Science and Engineering*, t. 155, p. 310-320, 2007.

[18] A. Calloo, "Développement d'une nouvelle modélisation de la loi de choc dans les codes de transport neutronique multigroupes," Theses, Université de Grenoble, 2012.

[19] R. Sanchez, "On SN-PN Equivalence," *Transport Theory and Statistical Physics*, t. 41, n° 5-6, p. 418-447, 2012.

[20] R. Sanchez et J. Ragusa, "On the Construction of Galerkin Angular Quadratures," *Nuclear Science and Engineering*, t. 169, n° 2, p. 133-154, 2011.

[21] J. E. Morel, "On the Validity of the Extended Transport Cross-Section Correction for Low-Energy Electron Transport," *Nuclear Science and Engineering*, t. 71, n° 1, p. 64-71, 1979.

[22] C. R. Drumm et al., "An Analysis of the Extended-Transport Correction with Application to Electron Beam Transport," *Nuclear Science and Engineering*, t. 155, n° 3, p. 355-366, 2007.

[23] R. Sanchez et N. J. McCormick, "Discrete Ordinates Solutions for Highly Forward Peaked Scattering," *Nuclear Science and Engineering*, t. 147, n° 3, p. 249-274, 2004.

[24] R. Sanchez et N. J. McCormick, "A Review of Neutron Transport Approximations," *Nuclear Science and Engineering*, t. 80, n° 4, p. 481-535, 1982.

[25] E. W. Larsen et R. E. Alcouffe, "Linear characteristic method for spatially discretizing the discrete ordinates equations in (X,Y)-geometry," in *Proceeding of ANS/ENS Joint Topical Meeting, Mathematical Methods in Nuclear Engineering*, 1981.

[26] I. Zmijarevic, "Résolution de l'equation de transport par des méthodes nodales et des caractéristiques dans les domaines à deux et trois dimensions," Theses, Université de Provence Aix-Marseille I, 1998.

[27] K. A. Mathews, "On the Propagation of Rays in Discrete Ordinates," *Nuclear Science and Engineering*, t. 132, n° 2, p. 155-180, 1999.

[28] E. Masiello, "Résolution de l'équation du transport des neutrons par les méthodes des éléments finis et des caractéristiques structurées appliquées à des maillages hétérogènes, CEA-R-6106," CEA, Tech.Report, 2006.

[29] F. Filiciotto et al., "New Modelling Capabilities in IDT," in *Proceedings of International Conference on Physics of Reactors 2022 (PHYSOR 2022)*, 2022.

[30] K. D. Lathrop, "Ray Effects in Discrete Ordinates Equations," *Nuclear Science and Engineering*, t. 32, n° 3, p. 357-369, 1968.

[31] K. D. Lathrop, "Remedies for Ray Effects," *Nuclear Science and Engineering*, t. 45, n° 3, p. 255-268, 1971.

[32] E. E. Lewis et W. F. Miller Jr., *Computational methods of neutron transport*. Wiley-Interscience, 1984.

[33] R. Baker, "A stochastic first collision source for TWODANT," in *8th International Conference on Radiation Shielding*, 1994.

[34] J. E. Morel et al., "Analysis of Ray-Effect Mitigation Techniques," *Nuclear Science and Engineering*, t. 144, n° 1, p. 1-22, 2003.

[35] E. W. Larsen et A. B. Wollaber, "A Quantitative Theory of Angular Truncation Errors in Three-Dimensional SN Calculations," *Nuclear Science and Engineering*, t. 160, n° 3, p. 267-283, 2008.

[36] Y. Azmy et E. Sartori, *Nuclear Computational Science, A Century in Review*. Springer Dordrecht, 2010.

[37] J. J. Jarrell, "An adaptive angular discretization method for neutral-particle transport in three-dimensional geometries," Theses, Texas A&M University, 2010.

[38] J. J. Jarrell et M. L. Adams, "Discrete-ordinates quadrature sets based on linear discontinuous finite elements," in *M&C 2011: International conference on mathematics and computational methods applied to nuclear science and engineering*, 2011.

[39] C. D. Ahrens, "Highly Efficient, Exact Quadratures for Three-Dimensional Discrete Ordinates Transport Calculations," *Nuclear Science and Engineering*, t. 170, n° 1, p. 98-101, 2012.

[40] K. Manalo, C. D. Ahrens et G. Sjoden, "Advanced quadratures for three-dimensional discrete ordinate transport simulations: A comparative study," *Annals of Nuclear Energy*, t. 81, p. 196 -206, 2015.

[41] J. Jung et al., "Discrete Ordinate Neutron Transport Equation Equivalent to PL Approximation," *Nuclear Science and Engineering*, t. 49, n° 1, p. 1-9, 1972.

[42] W. H. Reed, "Spherical Harmonic Solutions of the Neutron Transport Equation from Discrete Ordinate Codes," *Nuclear Science and Engineering*, t. 49, n° 1, p. 10-19, 1972.

[43] W. F. Miller Jr. et W. H. Reed, "Ray-Effect Mitigation Methods for Two-Dimensional Neutron Transport Theory," *Nuclear Science and Engineering*, t. 62, n° 3, p. 391-411, 1977.

[44] P. N. Brown, B. Chang et U. R. Hanebutte, "Spherical harmonic solutions of the Boltzmann transport equation via discrete ordinates," *Progress in Nuclear Energy*, t. 39, n° 2, 2001.

[45] J. Kópházi et D. Lathouwers, "A space–angle DGFEM approach for the Boltzmann radiation transport equation with local angular refinement," *Journal of Computational Physics*, t. 297, p. 637-668, 2015.

[46] Y. Favennec et al., "Ad hoc angular discretization of the radiative transfer equation," *Journal of Quantitative Spectroscopy and Radiative Transfer*, t. 225, p. 301-318, 2019.

[47] M. A. Goffin et al., "Goal-based angular adaptivity applied to a wavelet-based discretisation of the neutral particle transport equation," *Journal of Computational Physics*, t. 281, p. 1032-1062, 2015.

[48] L. L. Briggs, W. F. Miller Jr. et E. E. Lewis, "Ray-Effect Mitigation in Discrete Ordinate-Like Angular Finite Element Approximations in Neutron Transport," *Nuclear Science and Engineering*, t. 57, n° 3, p. 205-217, 1975.

[49] M. Natelson, "Variational Derivation of Discrete Ordinate-Like Approximations," *Nuclear Science and Engineering*, t. 43, n° 2, p. 131-144, 1971.

[50] C. R. E. de Oliveira, "An arbitrary geometry finite element method for multigroup neutron transport with anisotropic scattering," *Progress in Nuclear Energy*, t. 18, n° 1, p. 227-236, 1986.

[51] G. Coppa, G. Lapenta et P. Ravetto, "Angular finite element techniques in neutron transport," *Annals of Nuclear Energy*, t. 17, n° 7, p. 363-378, 1990.

[52] R. Sanchez, "On the singular structure of the uncollided and first-collided components of the Green's function," *Annals of Nuclear Energy*, t. 27, n° 13, p. 1167 -1186, 2000.

[53] I. Zmijarevic et D. Sciannandrone, "First Collision Source in the IDT Discrete Ordinates Transport Code," in *M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*, 2017.

[54] M. Hanuš et al., "Uncollided flux techniques for arbitrary finite element meshes," *Journal of Computational Physics*, t. 398, p. 108 848, 2019.

[55] W. J. Walters et A. Haghighat, "The adaptive collision source method for discrete ordinates radiation transport," *Annals of Nuclear Energy*, t. 105, p. 45-58, 2017.

[56] X. Wang et al., "Multicollision source method for discrete ordinates neutron transport calculation," *Annals of Nuclear Energy*, t. 156, p. 108 216, 2021.

[57] W. Xinyu, Z. Bin et C. Yixue, "Goal-oriented multi-collision source algorithm for discrete ordinates transport calculation," *Nuclear Engineering and Technology*, t. 54, n° 7, p. 2625-2634, 2022.

[58] R. Lenain, "Amélioration des méthodes de calcul de cœurs de réacteurs nucléaires dans APOLLO3 : décomposition de domaine en théorie du transport pour des géométries 2D et 3D avec une accélération non linéaire par la diffusion," Theses, Université Paris-Sud, 2015.

[59] G. Sjoden et A. Haghighat, "PENTRAN code system: Users guide to version 9. 4X.5 series," HSW Technologies, Tech.Report, 2008.

[60] R. S. Baker et K. R. Koch, "An Sn Algorithm for the Massively Parallel CM-200 Computer," *Nuclear Science and Engineering*, t. 128, n° 3, p. 312-320, 1998.

[61] M. P. Adams et al., "Provably optimal parallel transport sweeps on semi-structured grids," *Journal of Computational Physics*, t. 407, p. 109 234, 2020.

[62] N. Odry et al., "Coarse Mesh Rebalance Acceleration Applied to an Iterative Domain Decomposition Method on Unstructured Mesh," *Nuclear Science and Engineering*, t. 187, n° 3, 2017.

[63] B. Kochunas, T. J. Dovvnar et Z. Liu, "Parallel 3-D method of characteristics in MPACT," in *M and C 2013: 2013 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2013.

[64] E. Masiello, B. Martin et J.-M. Do, "Domain decomposition and CMFD acceleration applied to discrete-ordinate methods for the solution of the neutron transport equation in XYZ geometries," in *M&C 2011: International conference on mathematics and computational methods applied to nuclear science and engineering*, 2011.

[65] R. Lenain et al., "A Parallel Full Core Transport Calculation Based On Domain Decomposition Method," in *SNA + MC 2013 - Joint International Conference on Supercomputing in Nuclear Applications + Monte Carlo*, 2014.

[66] R. Lenain et al., "Domain decomposition method for 2d and 3d transport calculations using hybrid mpi/openmp parallelism," in *ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*, 2015.

[67] V. Dolean, P. Jolivet et F. Nataf, *An Introduction to Domain Decomposition Methods, Algorithms, Theory, and Parallel Implementation*. Society for Industrial et Applied Mathematics, 2015.

[68] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Society for Industrial et Applied Mathematics, 2003.

[69] D. Sciannandrone, "État de l'art des codes SN en trois dimensions pour des calculs de radioprotection, DEN/DANS/DM2S/SERMA/LLPR/NT/16-6100/A," CEA, Tech.Report, 2017.

[70] M. Falabino et al., "Computation of the uncollided-flux moments with advanced MOC and QP methods," *Journal of Computational Physics*, t. 460, p. 111 156, 2022.

[71] R. E. Alcouffe, R. D. O'Dell et F. W. B. Jr., "A First-Collision Source Method That Satisfies Discrete Sn Transport Balance," *Nuclear Science and Engineering*, t. 105, n° 2, p. 198-203, 1990.

[72] T. A. Wareing, J. E. Morel et D. K. Parsons, "A first collision source method for ATTILA, an unstructured tetrahedral mesh discrete ordinates code," Los Alamos National Lab., NM (United States), Tech.Report LA-UR-98-1373, 1998.

[73] L. Harbour et J. C. Ragusa, "Uncollided-Flux Treatment for Discrete-Ordinate Radiation Transport Solutions in the Rattlesnake Code System," in *PHYTRA4*, 2018.

[74] K. Kosako et C. Konno, "FNSUNCL3: First collision source code for TORT," *Journal of Nuclear Science and Technology*, t. 37, p. 475-478, 2000.

[75] C. Yang et al., "Development of 3-D parallel first-collision source method for discrete ordinate code JSNT-S," *Annals of Nuclear Energy*, t. 135, 2020.

[76] R. J. Mathar, "Solid Angle of a Rectangular Plate," *Max-Planck institute of astronomy*, p. 1-9, 2015.

[77] P. J. Davis et P. Rabinowitz, *Methods of Numerical Integration (Second Edition)*. Academic Press, 1984.

[78] J. Spanier et E. M. Gelbard, *Monte Carlo Principles and Neutron Transport Problems*. Dover Publications Inc., 2008.

[79] W. Gander et W. Gautschi, "Adaptive quadrature - Revisited," *BIT Numerical Mathematics*, t. 40, p. 84-101, 1 2000.

[80] P. Gonnet, "A Review of Error Estimation in Adaptive Quadrature," *ACM Comput. Surv.*, t. 44, n° 4, 2012.

[81] R. Piessens et al., *Quadpack, A Subroutine Package for Automatic Integration*. Springer-Verlag Berlin Heidelberg, 1983.

[82] P. van Dooren et L. de Ridder, "An adaptive algorithm for numerical integration over an n-dimensional cube," *Journal of Computational and Applied Mathematics*, t. 2, n° 3, p. 207 -217, 1976.

[83] A. Genz et A. Malik, "Remarks on algorithm 006: An adaptive algorithm for numerical integration over an N-dimensional rectangular region," *Journal of Computational and Applied Mathematics*, t. 6, n° 4, p. 295 -302, 1980.

[84] J. Berntsen, T. O. Espelid et A. Genz, "An Adaptive Algorithm for the Approximate Calculation of Multiple Integrals," *ACM Trans. Math. Softw.*, t. 17, n° 4, 437–451, 1991.

[85] D. P. Laurie, "Practical error estimation in numerical integration," *Journal of Computational and Applied Mathematics*, t. 12-13, p. 425 -431, 1985.

[86] S. Ehrich, "Stopping functionals for Gaussian quadrature formulas," *Journal of Computational and Applied Mathematics*, t. 127, n° 1, p. 153 -171, 2001.

[87] K. Kobayashi, N. Sugimura et Y. Nagaya, "3D radiation transport benchmark problems and results for simple geometries with void region," *Progress in Nuclear Energy*, t. 39, n° 2, 2001.

[88] D. Shepard, "A Two-Dimensional Interpolation Function for Irregularly-Spaced Data," in *Proceedings of the 1968 23rd ACM National Conference*, Association for Computing Machinery, 1968, 517–524.

[89] NEA, “"Nuclear Energy Agency - Databank Computer Program Services", https://www.oecd-nea.org/dbprog/ [Accessed 26-Apr-2016],” NEA, Online Resource, 2016.

[90] S. Ghazaie, M. Abbasi et A. Zolfaghari, “The Multi-PN approximation to neutron transport equation,” *Progress in Nuclear Energy*, t. 110, 2019.

[91] M. Falabino et al., “The Multi-PN angular discretization method of the neutral-particle transport equation for radiation shielding calculations,” *Annals of Nuclear Energy*, t. 177, p. 109 301, 2022.

[92] M. Falabino et al., “The MP_N Method: an New Angular Discretization Method Based on Piecewise Polynomial Interface Fluxes,” in *Proceedings of International Conference on Physics of Reactors 2022 (PHYSOR 2022)*, 2022.

[93] J. Stepanek, “The Multiple PN Surface Flux Integral Transport Method in X-Y Geometry Using General PN Expansions of the Flux in Angle and Space,” Theses, Federal Institute of Technology in Zurich, Switzerland, 1983.

[94] T. Mori et M. Nakagawa, “MVP/GMVP: general purpose Monte Carlo codes for neutron and photon transport calculations based on continuous energy and multigroup methods,” JAERI, Tech.Report 1348, 1994.

[95] W. Ford, “The Advancement of Stable, Efficient and Parallel Acceleration Methods for the Neutron Transport Equation,” Theses, Université Paris Saclay (COmUE), 2019.

[96] W. Ford et al., “A spatially variant rebalancing method for discrete-ordinates transport equation,” *Annals of Nuclear Energy*, t. 133, p. 589-602, 2019.

Appendix A

Derivation of the MOC-discretized mass matrix formulation

Along with the integration of the spatial-angular moments of the uncollided flux, the First Collision Source method numerically computes also the mass matrix of the spatial angular basis $\mathcal{A}_H \times \mathcal{P}_M$. In particular in MOC and TS-MOC methods, the numerical integration by characteristic lines also applies to the integral in Eq. (2.12), leading to

$$M_{(m,h)(m',h')} = \delta_{h,h'} a_m a_{m'} \sum_{i \cap \mathcal{T}} \int_{\Delta\mathbf{\Omega}_i} d\mathbf{\Omega} \int_{L_i^-}^{L_i^+} dr\ r^2 P_m(r\mathbf{\Omega}) P_{m'}(r\mathbf{\Omega}). \tag{A.1}$$

By further rewriting the spatial basis functions as in Eq. (2.19), we obtain

$$\begin{aligned} M_{(m,h)(m',h')} &= \delta_{h,h'} a_m a_{m'} \\ &\times \begin{cases} \sum_{i \cap \mathcal{T}} \Delta\mathbf{\Omega}_i \zeta_{0,i}, & \text{for } m = 1 \text{ and } m' = 1 \\ \sum_{i \cap \mathcal{T}} \left(a_{0,m} \Delta\mathbf{\Omega}_i \zeta_{0,i} + a_{1,m} \upsilon_{m,i} \zeta_{1,i} \right), & \text{for } m = 2,3,4 \text{ and } m' = 1 \\ \sum_{i \cap \mathcal{T}} \Big(a_{0,m} a_{0,m'} \Delta\mathbf{\Omega}_i \zeta_{0,i} + a_{0,m} a_{1,m'} \upsilon_{m',i} \zeta_{1,i} + \\ \qquad a_{1,m} a_{0,m'} \upsilon_{m,i} \zeta_{1,i} + a_{1,m} a_{1,m'} \upsilon_{m,m',i} \zeta_{2,i} \Big), & \text{for } m = 2,3,4 \text{ and } m' = 2,3,4 \end{cases} \end{aligned} \tag{A.2}$$

where a_m are the Legendre Polynomial normalization coefficients, $a_{0,m}$ and $a_{1,m}$ are the coefficients defined in Eq. (2.20), ζ are the integrals along trajectory i,

$$\zeta_{0,i} = \int_{L_i^-}^{L_i^+} dr\ r^2 = \frac{(L_i^+)^3 - (L_i^-)^3}{3}, \tag{A.3}$$

$$\zeta_{1,i} = \int_{L_i^-}^{L_i^+} dr\ r^3 = \frac{(L_i^+)^4 - (L_i^-)^4}{4}, \tag{A.4}$$

$$\zeta_{2,i} = \int_{L_i^-}^{L_i^+} dr\ r^4 = \frac{(L_i^+)^5 - (L_i^-)^5}{5}, \tag{A.5}$$

and υ are the solid angle integrals of the m and m' polar cosines

$$\upsilon_{m,i} = \int_{\Delta\mathbf{\Omega}_i} d\mathbf{\Omega}\ \mathbf{n}_m \cdot \mathbf{\Omega}, \tag{A.6}$$

$$\upsilon_{m,m',i} = \int_{\Delta\mathbf{\Omega}_i} d\mathbf{\Omega}\ (\mathbf{n}_m \cdot \mathbf{\Omega})\ (\mathbf{n}_{m'} \cdot \mathbf{\Omega})\,. \tag{A.7}$$

While ζ values are computed analytically, the υ solid angle integrals are solved using the numerical quadratures defined in Section 2.3.1, coherently with Γ.

Appendix B

Definition of the tabulated orders for the numerical integration of Γ weights in TS-MOC

Due to the large variability of the amplitude of solid angles in TS-MOC, we have tabulated the orders of the numerical quadrature needed to integrate Γ weights, up to spherical harmonic order H, with accuracy τ, and for any solid angle with amplitude $\Delta\Omega$. We have evaluated these tables thanks to a simplified analysis on the accuracy of numerically-evaluated Γ weights. Instead of considering every possible solid angle having amplitude $\Delta\Omega$, we restricted the analysis to a single solid angle. In particular, we considered solid angle

$$\mathcal{S}_t^2 = [0, \varphi] \times \Big[0, cos\left(\pi/2 - \theta\right)\Big] ,$$

generated by the x-axis and extending over the first octant of the unit sphere. The angles φ and θ are equal to each other and are determined such that the amplitude of the solid angle is exactly $\Delta\Omega$. Since the most anisotropic harmonics are also the most difficult to integrate, given K the maximum order of the harmonics expansion (here we are considering the double-index spherical harmonics notation $A_{k,l}$), the analysis of the error has been performed only on the weight associated to harmonics of order K, for each degree $l = -K, \ldots, K$. We considered uniquely $\Gamma_{0,h,i}$ (see Eq. (2.23)), i.e. the gamma weight associated to the constant spatial order. We evaluated the Γ weights for orders K equal to

$$K = \{0, 1, 3, 5, 7, 9\} ,$$

and solid angle amplitudes

$$\Delta\Omega = \{1., 5 \times 10^{-1}, 10^{-1}, 5 \times 10^{-2}, 10^{-2}, 5 \times 10^{-3}, 10^{-3}, \ldots, 10^{-10}\}$$

steradians, using different refinements of the numerical quadrature described in Section 2.3.1. In particular, we evaluated Γ weights for

$$N = 0, 1, 2, 3, \ldots, 11 \ ,$$

where N is the quadrature order, corresponding to $2^N \times 2^N$ quadrature points. For each set of parameters $(K, \Delta\Omega, N)$, we estimated the integration error as

$$\epsilon_K^{\Delta\Omega,N} = \underset{l=-K,\ldots,K}{max} \left\{ \frac{\Gamma_{K,l}^{\Delta\Omega,N} - \Gamma_{K,l}^{\Delta\Omega,12}}{\Gamma_{K,l}^{\Delta\Omega,12}} \right\},$$

where the reference value is evaluated using a fine quadrature of order $N = 12$.

For each couple $(K, \Delta\Omega)$, for values of the error bound τ equal to

$$\tau = 10^{-1}, 10^{-2}, \ldots, 10^{-7},$$

we tabulated the minimum quadrature order N such that $\epsilon_K^{\Delta\Omega,N} < \tau$.

Table B.1 shows the values of tabulated quadrature orders required to integrate weights Γ up to spherical harmonic order $K = 1,\ 3,\ 5$. This is a reduced version of the tables which are actually used by the solver. As expected, we observe that the minimum quadrature order N increases with K, meaning that higher order spherical harmonics are more difficult to integrate. At the same time, N increases as the error upper bound decreases, and as the solid angle amplitude increases.

Table B.1 Tabulated quadrature orders for the Γ coefficients integration in TS-MOC.

K	$\Delta\Omega$ [steradians]	τ 10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}
1	10^{-0}	2	3	5	6	8	10	11
	10^{-1}	0	1	3	4	6	8	9
	10^{-2}	0	0	1	3	4	6	8
	10^{-3}	0	0	0	1	3	4	6
	10^{-4}	0	0	0	0	1	3	4
	10^{-5}	0	0	0	0	0	1	3
	10^{-6}	0	0	0	0	0	0	1
	10^{-7}	0	0	0	0	0	0	0
	10^{-8}	0	0	0	0	0	0	0
3	10^{-0}	5	6	8	9	11	11	11
	10^{-1}	1	3	4	6	8	9	11
	10^{-2}	0	1	2	4	6	7	9
	10^{-3}	0	0	1	2	4	6	7
	10^{-4}	0	0	0	1	3	4	6
	10^{-5}	0	0	0	0	1	3	4
	10^{-6}	0	0	0	0	0	1	3
	10^{-7}	0	0	0	0	0	0	1
	10^{-8}	0	0	0	0	0	0	0
5	10^{-0}	5	7	8	10	11	11	11
	10^{-1}	1	3	5	6	8	9	11
	10^{-2}	0	1	3	5	6	8	10
	10^{-3}	0	1	2	3	5	7	8
	10^{-4}	0	1	1	3	4	6	8
	10^{-5}	0	0	0	1	2	4	5
	10^{-6}	0	0	0	0	1	2	3
	10^{-7}	0	0	0	0	0	1	2
	10^{-8}	0	0	0	0	0	0	1

Appendix C

Derivation of conservation relations for the MP_N method

The particle balance in $d\mathbf{r}$ around $\mathbf{r}$ and in $d\mathbf{\Omega}$ around $\mathbf{\Omega}$ is given by the differential form of the transport equation,

$$(\mathbf{\Omega} \cdot \boldsymbol{\nabla} + \Sigma)\psi(\mathbf{r}, \mathbf{\Omega}) = q(\mathbf{r}, \mathbf{\Omega}). \tag{C.1}$$

For each spatial region $\mathcal{T}$ and for each solid angle $\mathcal{S}_n^2$, the angular flux is assumed to be expanded in

$$\begin{aligned} \psi(\mathbf{r}, \mathbf{\Omega}) &= \sum_{h=1,H} \alpha_h A_h(\mathbf{\Omega}) \mathbf{P}(\mathbf{r}) \cdot \boldsymbol{\psi}_h^n \\ \text{for } \mathbf{r} &\in \mathcal{T} \text{ and } \mathbf{\Omega} \in \mathcal{S}_n^2 \end{aligned} \tag{C.2}$$

and

$$\begin{aligned} \psi_n^s(\mathbf{r}, \mathbf{\Omega}) &= \sum_{\ell=1,N_\ell} B_\ell^n(\mathbf{\Omega}) \mathbf{P}^s(\mathbf{r}) \cdot \boldsymbol{\phi}_\ell^{s,n} \\ \text{for } \mathbf{r} &\in \Gamma_n^s = \Gamma^s(\mathbf{\Omega}) \text{ and } \mathbf{\Omega} \in \mathcal{S}_n^2, \end{aligned} \tag{C.3}$$

where the angular coefficients $\boldsymbol{\psi}_h^n$ for the flux in solid angle $\mathcal{S}_n^2$ differ from angular moments $\boldsymbol{\phi}_h^n$, defined in Eq. (4.21), for the flux on the unit sphere.

The projection of the flux expansion, Eq. (C.2), onto the subspace $span\{A_h(\mathbf{\Omega})\mathbf{P}(\mathbf{r})\}$,

$$(A_h\mathbf{P},\ \cdot\ )_n = \frac{1}{4\pi V} \int_{\mathcal{S}_n^2} d\mathbf{\Omega}\ A_h(\mathbf{\Omega}) \int_{\mathcal{T}} d\mathbf{r}\ \mathbf{P}(\mathbf{r}) \cdot \tag{C.4}$$

gives the relation between the expansion coefficients $\boldsymbol{\psi}_h^n$ and the flux angular moments $\boldsymbol{\phi}_h^n$,

$$\boldsymbol{\phi}_h^n = (A_h\mathbf{P}, \psi)_n = \sum_{h'=1,H} \alpha_{h'} (A_h\mathbf{P}, \mathbf{P}A_{h'})_n \cdot \boldsymbol{\psi}_{h'}^n. \tag{C.5}$$

By defining the mass matrix minors in Eq. (C.5) as

$$\mathbb{M}^n_{h,h'} = \alpha_{h'}(A_h\mathbf{P}, \mathbf{P}A_{h'})_n \tag{C.6}$$

the relation can be inverted and rewritten as

$$\boldsymbol{\psi}^n_{h'} = \sum_{h=1,H} [\mathbb{M}^n]^{-1}_{h',h} \ \boldsymbol{\phi}^n_h. \tag{C.7}$$

By factorizing $\mathbb{M}^n$ into

$$\mathbb{M}^n = \overline{\overline{\mathbf{1}}}_{M\times M} \otimes \overline{\overline{\mathbf{M}}}^n,$$

where the unit matrix $\overline{\overline{\mathbf{1}}}_{M\times M}$ coincides with the mass matrix for the linear spatial basis $\mathbf{P}$, and

$$\overline{\overline{\mathbf{M}}}^n = (\mathbf{A}, \alpha\mathbf{A})_n = \frac{1}{4\pi}\int_{\mathcal{S}^2_n} d\boldsymbol{\Omega}\ \mathbf{A}(\boldsymbol{\Omega}) \otimes \alpha\mathbf{A}(\boldsymbol{\Omega}),$$

the inverse mass matrix can be expressed by

$$[\mathbb{M}^n]^{-1}_{h',h} = \overline{\overline{\mathbf{1}}}_{4\times 4} \otimes \left[\overline{\overline{\mathbf{M}}}^n\right]^{-1}_{h',h}.$$

For convenience, the MP$_N$ Eqs. (4.28) and (4.16) are rewritten for each angular moment as

$$\boldsymbol{\phi}^n_h = \sum_{h'=1,H} \mathbb{C}_{n,h,h'}\mathbf{q}_{h'} + \sum_{s'\in\Gamma^-_n}\sum_{\ell'=1,N_\ell} \mathbb{I}^{s'}_{n,h,\ell'}\boldsymbol{\phi}^{s',n}_{\ell'}, \tag{C.8}$$

$$\boldsymbol{\phi}^{s,n}_\ell = \sum_{h'=1,H} \mathbb{E}^{*,s}_{n,\ell,h'}\mathbf{q}_{h'} + \sum_{s'\in\Gamma^-_n}\sum_{\ell'=1,N_\ell} \mathbb{T}^{*,s,s'}_{n,\ell,\ell'}\boldsymbol{\phi}^{s',n}_{\ell'}\ , \tag{C.9}$$

with the matrix $\mathbb{E}^{*,s}_{n,\ell,h'}$ and $\mathbb{T}^{*,s,s'}_{n,\ell,\ell'}$ representing the minors of the global matrices in Eq. (4.18).

The projection of the balance equation, Eq. (C.1), on subspace $span\{A_h(\boldsymbol{\Omega})\mathbf{P}(\mathbf{r})\}$, which entails the surface projection

$$\langle A_h\mathbf{P},\ \cdot\ \rangle^s_n = \frac{1}{4\pi V}\int_{\mathcal{S}^2_n} d\boldsymbol{\Omega}\ \mu^s(\boldsymbol{\Omega})A_h(\boldsymbol{\Omega})\int_{\Gamma^s_n} d\mathbf{r}\ \mathbf{P}(\mathbf{r})\ \cdot, \tag{C.10}$$

because of the derivative (Gauss theorem), transforms the equation first into

$$\begin{aligned}\sum_{s\in\Gamma^+_n}\langle A_h\mathbf{P}, \psi^s_n\rangle^s_n - \sum_{s'\in\Gamma^-_n}\left\langle A_h\mathbf{P}, \psi^{s'}_n\right\rangle^{s'}_n - (A_h\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\mathbf{P}, \psi)_n \\ +\ \ \Sigma(A_h\mathbf{P}, \psi)_n = (A_h\mathbf{P}, q)_n,\end{aligned} \tag{C.11}$$

and then, by further substituting the expansion for the flux and for the source, into

$$\begin{aligned}
&\sum_{s\in\Gamma_n^+}\sum_{\ell=1,N_\ell}\langle A_h\mathbf{P},\mathbf{P}^s B_\ell^n\rangle_n^s\,\boldsymbol{\phi}_\ell^{s,n}-\sum_{s'\in\Gamma_n^-}\sum_{\ell'=1,N_\ell}\left\langle A_h\mathbf{P},\mathbf{P}^{s'}B_{\ell'}^n\right\rangle_n^{s'}\boldsymbol{\phi}_{\ell'}^{s',n} \qquad \text{(C.12)}\\
-&\sum_{h'=1,H}\alpha_{h'}(A_h\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\mathbf{P},\mathbf{P}A_{h'})_n\;\boldsymbol{\psi}_{h'}^n+\sum_{h'=1,H}\alpha_{h'}\Sigma(A_h\mathbf{P},\mathbf{P}A_{h'})_n\;\boldsymbol{\psi}_{h'}^n\\
=&\sum_{h'=1,H}\alpha_{h'}(A_h\mathbf{P},\mathbf{P}A_{h'})_n\mathbf{q}_{h'}.
\end{aligned}$$

By using Eqs. (C.5) and (C.7), and defining:

- the boundary mass matrices

$$\mathbb{B}_{h,\ell}^{s,n}=\langle A_h\mathbf{P},\mathbf{P}^s B_\ell\rangle_n^s\ \text{and}\ \mathbb{B}_{h,\ell'}^{s',n}=\left\langle A_h\mathbf{P},\mathbf{P}^{s'}B_{\ell'}\right\rangle_n^{s'}\ ,$$

- and the streaming plus removal matrix

$$\mathbb{S}_{h,h'}^n=\sum_{h''=1,H}\alpha_{h''}\Sigma(A_h\mathbf{P},\mathbf{P}A_{h''})_n\,[\mathbb{M}^n]_{h'',h'}^{-1}-\sum_{h''=1,H}\alpha_{h''}(A_h\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\mathbf{P},\mathbf{P}A_{h''})_n\;[\mathbb{M}^n]_{h'',h'}^{-1}\ ,$$

Eq. (C.12) is rewritten as

$$\begin{aligned}
&\sum_{s\in\Gamma_n^+}\sum_{\ell=1,N_\ell}\mathbb{B}_{h,\ell}^{s,n}\boldsymbol{\phi}_\ell^{s,n}-\sum_{s'\in\Gamma_n^-}\sum_{\ell'=1,N_\ell}\mathbb{B}_{h,\ell'}^{s',n}\boldsymbol{\phi}_{\ell'}^{s',n} \qquad \text{(C.13)}\\
&+\sum_{h'=1,H}\mathbb{S}_{h,h'}^n\boldsymbol{\phi}_{h'}^n=\sum_{h'=1,H}\mathbb{M}_{h,h'}^n\mathbf{q}_{h'}.
\end{aligned}$$

By substituting Eqs. (C.8) and (C.9) for $\boldsymbol{\phi}_{h'}^n$ and $\boldsymbol{\phi}_\ell^{s,n}$, the equation becomes

$$\begin{aligned}
&\sum_{s\in\Gamma_n^+}\sum_{\ell=1,N_\ell}\mathbb{B}_{h,\ell}^{s,n}\left\{\sum_{h'=1,H}\mathbb{E}_{n,\ell,h'}^{*,s}\mathbf{q}_{h'}+\sum_{s'\in\Gamma_n^-}\sum_{\ell'=1,N_\ell}\mathbb{T}_{n,\ell,\ell'}^{*,s,s'}\boldsymbol{\phi}_{\ell'}^{s',n}\right\} \qquad \text{(C.14)}\\
-&\sum_{s'\in\Gamma_n^-}\sum_{\ell'=1,N_\ell}\mathbb{B}_{h,\ell'}^{s',n}\boldsymbol{\phi}_{\ell'}^{s',n}\\
+&\sum_{h'=1,H}\mathbb{S}_{h,h'}^n\left\{\sum_{h''=1,H}\mathbb{C}_{n,h',h''}\mathbf{q}_{h''}+\sum_{s'\in\Gamma_n^-}\sum_{\ell'=1,N_\ell}\mathbb{I}_{n,h',\ell'}^{s'}\boldsymbol{\phi}_{\ell'}^{s',n}\right\}\\
=&\sum_{h'=1,H}\mathbb{M}_{h,h'}^n\mathbf{q}_{h'}.
\end{aligned}$$

Exchanging the index $h'\to h''$ within the first term in the LHS and on the RHS of the equation, and grouping the terms in $\mathbf{q}_{h''}$ and $\boldsymbol{\phi}_{\ell'}^{s',n}$, one obtains two systems of conservation

relations. The first,

$$\mathbb{M}^n_{h,h''} = \sum_{s\in\Gamma_n^+}\sum_{\ell=1,N_\ell} \mathbb{B}^{s,n}_{h,\ell}\mathbb{E}^{*,s}_{n,\ell,h''} + \sum_{h'=1,H} \mathbb{S}^n_{h,h'}\mathbb{C}_{n,h',h''} \tag{C.15}$$

expresses the conservation for any spatial degrees of freedom for any couple of angular moments (h, h'') in $\mathcal{S}_n^2$. In particular, the zero-order term represents the conservation of particles within the volume of the region. The second,

$$\mathbb{B}^{s',n}_{h,\ell'} = \sum_{s\in\Gamma_n^+}\sum_{\ell=1,N_\ell} \mathbb{B}^{s,n}_{h,\ell}\mathbb{T}^{*,s,s'}_{n,\ell,\ell'} + \sum_{h'=1,H} \mathbb{S}^n_{h,h'}\mathbb{I}^{s'}_{n,h',\ell'} \tag{C.16}$$

expresses the conservation for any spatial moment in $\Gamma^{s'}$ and $\mathcal{S}_n^2$ for any couples of angular moments (h, ℓ').

If the conservation Eqs. (C.15) and (C.16) are rewritten for the subspace $span\{\mathbf{A}(\mathbf{\Omega}) \otimes \mathbf{P}(\mathbf{r})\}$, they can be represented in two global system of equations for any region $\mathcal{T}$ and any solid angle $\mathcal{S}_n^2$,

$$\mathbb{M}^n = \sum_{s\in\Gamma_n^+} \mathbb{B}^{s,n}\mathbb{E}^{*,s}_n + \mathbb{S}^n\mathbb{C}_n \tag{C.17}$$

$$\mathbb{B}^{s',n} = \sum_{s\in\Gamma_n^+} \mathbb{B}^{s,n}\mathbb{T}^{*,s,s'}_n + \mathbb{S}^n\mathbb{I}^{s'}_n \tag{C.18}$$

where

$$\mathbb{M}^n = \begin{bmatrix} \mathbb{M}^n_{0,0} & \mathbb{M}^n_{0,1} & \cdots & \mathbb{M}^n_{0,H} \\ \mathbb{M}^n_{1,0} & \mathbb{M}^n_{1,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbb{M}^n_{H,0} & \cdots & \cdots & \mathbb{M}^n_{H,H} \end{bmatrix}, \qquad \mathbb{S}^n = \begin{bmatrix} \mathbb{S}^n_{0,0} & \mathbb{S}^n_{0,1} & \cdots & \mathbb{S}^n_{0,H} \\ \mathbb{S}^n_{1,0} & \mathbb{S}^n_{1,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbb{S}^n_{H,0} & \cdots & \cdots & \mathbb{S}^n_{H,H} \end{bmatrix},$$

$$\mathbb{B}^{s,n} = \begin{bmatrix} \mathbb{B}^{s,n}_{0,0} & \mathbb{B}^{s,n}_{0,1} & \cdots & \mathbb{B}^{s,n}_{0,N_\ell} \\ \mathbb{B}^{s,n}_{1,0} & \mathbb{B}^{s,n}_{1,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbb{B}^{s,n}_{H,0} & \cdots & \cdots & \mathbb{B}^{s,n}_{H,N_\ell} \end{bmatrix}, \qquad \mathbb{B}^{s',n} = \begin{bmatrix} \mathbb{B}^{s',n}_{0,0} & \mathbb{B}^{s',n}_{0,1} & \cdots & \mathbb{B}^{s',n}_{0,N_\ell} \\ \mathbb{B}^{s',n}_{1,0} & \mathbb{B}^{s',n}_{1,1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbb{B}^{s',n}_{H,0} & \cdots & \cdots & \mathbb{B}^{s',n}_{H,N_\ell} \end{bmatrix}.$$

Then, the collision and the incoming global matrices can be easily obtained by,

$$\mathbb{C}_n = [\mathbb{S}^n]^{-1} \left[\mathbb{M}^n - \sum_{s \in \Gamma_n^+} \mathbb{B}^{s,n} \mathbb{E}_n^{*,s} \right], \tag{C.19}$$

$$\mathbb{I}_n^{s'} = [\mathbb{S}^n]^{-1} \left[\mathbb{B}^{s',n} - \sum_{s \in \Gamma_n^+} \mathbb{B}^{s,n} \mathbb{T}_n^{*,s,s'} \right]. \tag{C.20}$$

One can take advantage of Eqs. (C.19) and (C.20) to relate together MP_N coefficients, making their calculation easier. Moreover, in case of coefficients storage, these can help in reducing the number of coefficient to be stored, and so reducing the memory footprint. Although Eqs. (C.19) and (C.20) could be computationally effective, they have not been implemented in the current version of the MP_N yet.