

Rattlesnake Theory Manual

Yaqi Wang¹, Sebastian Schunert¹,
Vincent Labouré²

¹ Nuclear Engineering Methods Development
Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3840

²Reactor Physics Design and Analysis
Idaho National Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3840

April, 2018



INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

Rattlesnake Theory Manual

Yaqi Wang, Sebastian Schunert, Vincent Labouré

April, 2018

**Idaho National Laboratory
Nuclear Systems Design and Analysis
Idaho Falls, Idaho 83415**

<http://www.inl.gov>

**Prepared for the
U.S. Department of Energy
Office of Nuclear Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517**

Rattlesnake Theory Manual

Yaqi Wang, Sebastian Schunert, Vincent Labouré

April, 2018

Contents

1	Introduction	7
2	Multigroup Radiation Transport Equations	10
2.1	Common transport equations	10
2.2	Common diffusion equation	17
2.3	Neutron	21
2.3.1	Physics of neutron transport	21
2.3.2	Neutron transport equation	23
2.4	Thermal radiation	27
2.4.1	Physics of thermal radiation transport	27
2.4.2	Thermal radiation transport equation (TRTE)	30
3	Weak Forms of the Multigroup Radiation Transport Equations	32
3.1	The unstabilized form	32
3.2	Boundary conditions	34
3.3	Spatial discretization and stabilization	36
3.3.1	FEM notations	36
3.3.2	SAAF	38
3.3.3	LS	41
3.3.4	DFEM	42
3.4	Angular discretization and final weak forms	42
3.4.1	Angular discretization	42
3.4.2	SAAF-CFEM-SN	44
3.4.2.1	Steady-state CTE	44
3.4.2.2	Boundary conditions	45
3.4.2.3	Transient kernel	46
3.4.2.4	Neutron	46
3.4.2.5	Thermal radiation	47
3.4.2.6	R-spherical coordinate	47
3.4.2.7	RZ-axial symmetric coordinate	47
3.4.2.8	Mathematical adjoint	47
3.4.2.9	Summary of MOOSE objects	49
3.4.3	SAAF-CFEM-PN	52
3.4.3.1	Steady-state CTE	52

3.4.3.2	Boundary conditions	53
3.4.3.3	Transient kernel	53
3.4.3.4	Neutron	54
3.4.3.5	Thermal radiation	54
3.4.3.6	R-spherical coordinate	54
3.4.3.7	RZ-cylindrical coordinate	54
3.4.3.8	Mathematical adjoint	55
3.4.3.9	Summary of MOOSE objects	55
3.4.4	LS-CFEM-SN	58
3.4.4.1	Steady-state CTE	58
3.4.4.2	Boundary conditions	58
3.4.4.3	Transient kernel	59
3.4.4.4	Neutron	59
3.4.4.5	Thermal radiation	59
3.4.4.6	R-spherical coordinate	59
3.4.4.7	RZ-axial symmetric coordinate	59
3.4.4.8	Mathematical adjoint	60
3.4.4.9	Summary of MOOSE objects	60
3.4.5	LS-CFEM-PN	61
3.4.5.1	Steady-state CTE	61
3.4.5.2	Boundary conditions	62
3.4.5.3	Transient kernel	62
3.4.5.4	Neutron	62
3.4.5.5	Thermal radiation	62
3.4.5.6	R-spherical coordinate	62
3.4.5.7	RZ-cylindrical coordinate	63
3.4.5.8	Mathematical adjoint	63
3.4.5.9	Summary of MOOSE objects	63
3.4.6	DFEM-SN	64
3.4.6.1	The transport sweep	64
3.4.6.2	Steady-state CTE	65
3.4.6.3	Interior face terms (DGKernel)	65
3.4.6.4	Transient	66
3.4.6.5	Neutron	66
3.4.6.6	Boundary Conditions	66

3.4.6.7	Thermal radiation	66
3.4.6.8	R-spherical coordinate	66
3.4.6.9	RZ-cylindrical coordinate	66
3.4.6.10	Mathematical adjoint	66
3.4.6.11	Summary of MOOSE objects	66
3.4.7	DFEM-PN	68
3.4.7.1	Steady-state CTE	68
3.4.7.2	Transient kernel	69
3.4.7.3	Interior facets terms (DGKernel)	69
3.4.7.4	Reflecting boundary	70
3.4.7.5	Incoming Dirichlet boundary	70
3.4.7.6	Filtering	71
3.4.7.7	Thermal radiation	72
3.4.7.8	Neutron	72
3.4.7.9	R-spherical coordinate	72
3.4.7.10	RZ-cylindrical coordinate	72
3.4.7.11	Mass matrix lumping	72
3.4.7.12	DFEM- P_1	75
3.5	Diffusion	78
3.5.1	CFEM-Diffusion	78
3.5.1.1	Steady-state	79
3.5.1.2	Boundary conditions	79
3.5.1.3	Transient kernel	79
3.5.1.4	Neutron	80
3.5.1.5	Thermal radiation	80
3.5.1.6	R-spherical coordinate	80
3.5.1.7	RZ-axial symmetric coordinate	80
3.5.1.8	Mathematical adjoint	80
3.5.1.9	Summary of MOOSE objects	80
3.5.2	DFEM-Diffusion	82
3.5.2.1	Steady-state	82
3.5.2.2	Boundary conditions	83
3.5.2.3	Transient kernel	83
3.5.2.4	Neutron	83
3.5.2.5	Thermal radiation	83

3.5.2.6	R-spherical coordinate	83
3.5.2.7	RZ-axial symmetric coordinate	83
3.5.2.8	Mathematical adjoint	83
3.5.2.9	Summary of MOOSE objects	83
3.6	Nonlinear diffusion acceleration	84
3.6.1	Picard iteration	84
3.6.2	SAAF-CFEM-SN	86
3.6.3	LS-CFEM-SN	88
3.6.4	DFEM-SN	89
3.7	Multi-scheme interface conditions	90
3.7.1	Lagrange multiplier method	91
3.7.1.1	SN-PN	91
3.7.1.2	PN-PN	92
3.7.1.3	SN-SN	92
3.7.2	Upwinding	93
3.7.2.1	SN-PN	93
3.7.2.2	SN-SN	94
3.7.2.3	PN-PN	94
3.8	Reduced order model for NTE	95
3.8.1	Dumping the PKE parameters during a spatial kinetics calculation	97
3.8.2	Multi-region PKE parameters	97
3.8.3	Improved quasi-static	99
4	Rattlesnake Design	103
4.1	Relation to libMesh/MOOSE	103
4.2	Transport system actions	104
4.3	Extendable neutronic materials	107
4.3.1	Flux-Corrected On-The-Fly Group Condensation	112
4.3.2	Cusping Treatment	113
4.4	Transport functions	116
4.5	Method of manufactured solution	116
5	Solvers	118
5.1	PJFNK	119
5.2	Transient	120
5.3	Eigenvalue	120
5.4	Richardson iteration	122

Appendices	123
A Vector Notations	124
B Spherical Harmonics	131
B.1 Basics	131
B.2 Matrices	133
B.3 Boundary types	135
C Streaming in Curvilinear Coordinates	137
D Phase Function	140
D.1 Definition of phase function	140
D.2 Common phase functions	141
D.2.1 Isotropic	141
D.2.2 Rayleigh	141
D.2.3 Henyey-Greenstein	141
D.2.4 Heaviside	141
D.2.5 Dirac	141
D.3 Scattering treatments for SN	142
D.3.1 PN treatment	142
D.3.2 SN treatment	142
D.3.3 S_N - P_N equivalency	144
E Angular Quadratures	145
E.1 General discussions	145
E.1.1 Polar derivative	147
E.1.2 Azimuthal derivative with P_N expansion	149
E.2 Level-Symmetric	149
E.3 Gauss Chebyshev	152
E.4 Bickley-3 Optimized	153
F Multigroup Approximation	156
G Black Body Emission	159
H The Reflection and Refraction on a Medium Interface	161
I Rattlesnake Publications	164

1 Introduction

Rattlesnake is the MOOSE [1, 2] (Multiphysics Object-Oriented Simulation Environment) based application for simulating the physics of *radiation transport*. Radiation transport studies the motion of particles, usually neutrons, photons or electrons, etc. and their interactions with background materials [3, 4, 5, 6, 7, 8]. The fundamental quantity of interest in radiation transport is the statistically averaged radiation distribution or *flux* in space (dimensionality 3), time (1), direction of motion (2) and energy (or frequency) (1), forming a seven-dimensional phase space. Various *reaction rates* are secondary in the sense that they are computed from the primary flux and the background material properties; however, reaction rates are typically more interesting for the analyst or when coupling to other physics, e.g. heat conduction or thermo-hydraulics. Radiation transport calculations receive feedback within a multiphysics environment through temperature and density changes, which predominantly change how radiation particles interact with the background nuclei. Accurate *prediction* of the flux is essential to the design and operation of nuclear facilities for the purpose of *safety and economics*. For example in nuclear fission reactors, neutron-induced fissions and the accumulated radiative isotopes are the main heat source. Predicting whether the heat can be removed from the reactor core under all circumstances without violation of safety limits is critical to the safety of nuclear reactors.

The physics of radiation transport can be described with the linear Boltzmann equation (LBE) or the radiation transport equation (RTE). Interaction among radiation particles can be neglected because the radiation density is typically several orders of magnitude smaller than the density of background nuclei. A radiation particle is thus much more likely to collide with a background nucleus than to collide with another particle. In neutron transport the average neutron life time measured from its birth to its extinction is much smaller than the neutron half life, and hence neutron decay can be ignored. Gravity is also typically neglected in most simulations on the engineering scale because the change of direction of motion due to gravity is negligible during neutron lifetimes. Bulk motion of background nuclei is neglected because we assume its speed is much smaller than the typical speed of individual radiation particles. When the above-mentioned conditions are not met, the RTE needs to be enhanced accordingly. This theory manual will not cover those off-normal conditions. We refer readers to [9, 10, 11] for more information.

There are two fundamentally different ways of modeling radiation transport: statistical or Monte Carlo methods and deterministic methods [3, 12]. The Monte Carlo methods tracks individual particles' histories by sampling the distance to the next collision, the nuclide it interacts with, the type of collision, and the out-going direction in case scattering occurred; the sampling is based on proper probability density functions that can be derived from the geometry and material properties. If a sufficient number of particles is simulated, the statistical errors of the reaction rates or quantities of interest can reach a level that is smaller than the errors introduced by other sources, such as the uncertainty of the nuclear data, present in the simulation. In deterministic methods the RTE is discretized into an algebraic system of equations, which can be solved numerically. The seven-dimensional phase space needs to be discretized which poses a challenge due to the much larger number of unknowns than in other disciplines of physics with smaller dimensionality, such as thermal conduction. Monte Carlo methods and deterministic methods can be combined in certain ways resulting into the so-called hybrid methods [13]. Rattlesnake is a deterministic transport solver and this manual is exclusively about deterministic methods.

Fundamental nuclear data describing the likelihood per unit path length of radiation interacting with matter is required for solving the RTE; this data is referred to as nuclear *cross sections*. Cross sections usually depend on the incident energy in a complicated manner due to nuclear resonances and on the temperature of the background material through changes in density, Doppler broadening and for small incident energies the thermal motion of the targets and molecules ($S(\alpha, \beta)$). Raw nuclear data has been measured and evaluated using our understanding of nuclear physics leading to the evaluated nuclear data library (ENDL) [14, 15, 16, 17, 18, 19]. Because the particle energy spans several orders of magnitude – for example, the neutron energy ranges from 0.001eV to 20MeV in thermal nuclear reactor applications – the computing resources required for performing multi-dimensional transport calculations with continuous or point-wise energy resolution is prohibitive for any real application. Typically, the *multigroup approximation* is applied, where continuous-energy cross sections are collapsed into a selected set of energy ranges, also referred to as *energy groups*, using the energy spectrum as a weight function within these ranges. The details of the group condensation are given in Appendix F. The multigroup equations are exact in the sense that if the correct spectrum everywhere in phase space was known and used for the preparation of the cross sections, they would reproduce the exact group fluxes. However, in reality this cannot be achieved because the flux is unknown by the time the cross sections are prepared. Instead, representative energy spectra are pre-generated targeting certain applications, e.g. light water reactors (LWR) simulation, with some representative geometries. On top of this, group condensation is typically entangled with the spatial homogenization, where a region in space

gets assigned the same cross sections everywhere despite the fact that the energy spectrum varies from location to location, in order to reduce the number of sets of multigroup cross sections and the required angular and spatial resolutions for the transport calculation. Cross section resonance effects and homogenization equivalency need to be carefully treated during this procedure. The application handling the cross section processing is referred to as the lattice physics application because the geometries typically have lattice structures. Rattlesnake is a transport solver assuming the multigroup cross sections are given. Thus, this manual will not talk about lattice physics and assumes the multigroup cross sections appearing in the RTE are known. Rattlesnake is not ready to be used for lattice physics calculations at this moment because of the lack of optimization for calculations with several hundred of energy groups.

Rattlesnake is a transport solver based on the *finite element method (FEM)* [20, 21] discretization of the spatial independent variables. FEM is a numerical technique for finding approximate solutions to partial differential equations (PDE), like the transport equation, by deriving their weak or variational form. The domain is divided into elements forming an unstructured grid and the solution is approximated by linear combinations of basis functions which have local support on these elements. The number of basis functions is equal to the number of unknowns of the FEM problem. The variational form of the PDE can then be converted into a system of algebraic equations that can be solved numerically. The advantages of FEM include an accurate representation of complex geometries, the ability to capture local effects and more. A uniform multiphysics FEM framework can facilitate the data transfer for coupling multiple physics. Both continuous FEM (CFEM) and discontinuous FEM (DFEM) are implemented in Rattlesnake. MOC (method of characteristics) [22], CP (collision probability) method [23], FD (finite difference) [24], etc. are not covered by this manual.

For the angular discretization, Rattlesnake has both the discrete ordinates (SN) and the spherical harmonics expansion (PN) methods, including the diffusion approximation [3]. Fluxes along a finite set of particle directions of motion are evaluated with SN and the angular integral appearing in the RTE is approximated using angular quadratures with weights associated to these directions. The RTE is discretized into a coupled hyperbolic system of equations for each multigroup equation. SN is typically more suitable for heterogeneous calculations, where higher angular resolution is desired, than for problems with significant spatial homogenization. On the other hand, PN expands the flux into spherical harmonics functions in angle introducing a coupled system of equations. Because of the streaming coupling introduced by PN, calculations with high-order PN are costly, and hence PN is typically applied to problems with significant spatial homogenization, where transport effects are limited and low-order PN is sufficient. From a mathematical perspective, SN is a collocation method and PN is a spectral Galerkin method for the angular discretization. The weak forms for various discretization schemes are given in this manual. Time discretization is handled separately from other independent variables by the method of lines. Rattlesnake allows users to apply schemes with different level of resolutions more suitable for subdomains and solve them as a whole without iterations over the interface quantities, which is referred as *multi-scheme* in this manual.

Rattlesnake is built on the MOOSE framework. Rattlesnake manages meshes with libMesh [25] and solves the system equation with PETSc [26] through MOOSE. Support of unstructured meshes, parallelization with domain decomposition, various time integration schemes, most of nonlinear solvers features, multiphysics coupling are all inherited from the framework. Rattlesnake is responsible for setting up the discretization of RTE with interfaces provided by MOOSE. Because Rattlesnake is built on a general framework, it can perform calculations in 1D, 2D and 3D unstructured meshes via MPI (message passing interface) [27] and multi-threading. The weak forms can be directly translated into code under the MOOSE framework, which results in codes easy to maintain and extend. Multiple developers can co-work seamlessly on Rattlesnake. This development pattern allows splitting the work into the physics-related items residing in Rattlesnake, and physics-irrelevant items residing in the framework. The benefit is two-fold: Rattlesnake can push the development on the framework side, which can benefit all the other MOOSE-based applications and any improvements in MOOSE can improve Rattlesnake. For example, the common software development procedure is managed by MOOSE, e.g. the version control, regression tests, on-line documentation. In addition, coupling Rattlesnake with other MOOSE-based applications dealing with other physics is straight-forward because these applications are based on the same framework. As an example, the reactor physics application MAMMOTH [2, 28] leverages components spread over Rattlesnake, MOOSE and its modules, Bison [29, 30, 31], and Relap-7 [32]. In summary, being built on MOOSE makes Rattlesnake powerful and unique. There indeed are constraints imposed by the framework, but this is not insurmountable and outweighed by all the advantages provided by this development pattern.

In Section 2, we first state the RTE for various particles. Currently, we only consider two particles: neutron and thermal radiation but Rattlesnake can be extended easily to other types. Particles mostly share the same time derivative,

streaming, collision, scattering, external source terms and some common types of boundary conditions. These common, physics-agnostic terms are presented first. Basics of the physics of each particle transport are given. We introduce the vector notation for the transport equations for notation simplicity. All vector quantities and operators in this manual are summarized in [Appendix A](#). In [Section 3](#), we derive the weak forms for RTE. Both weak forms with CFEM and DFEM are presented. As a special case of PN, the weak forms for diffusion equation with CFEM and DFEM are given. The links between the weak form and the MOOSE kernels are presented, which bridges this manual and the documentation automatically generated from the codes. Multi-scheme transport, in particular the interface conditions between subdomains where various discretization schemes are applied is also presented in that section. [Section 4](#) presents various code design aspects, including how Rattlesnake is related with MOOSE, how Rattlesnake sets up the radiation transport system, including the weak forms, the material properties, etc. [Section 5](#) presents the solvers for steady-state, eigenvalue and transient transport equations. Most of the solvers are implemented in MOOSE, except Richardson source iteration. Additional considerations such as spherical harmonics, the streaming operator in curvilinear coordinates, scattering phase functions, angular quadrature for SN, multigroup approximation, black body emission, the reflection and refraction on a medium interface for thermal radiation transport, etc. are found in from [Appendix B](#) to [Appendix H](#). This manual represents the current implementation of Rattlesnake and we will keep it being updated along with the code development.

2 Multigroup Radiation Transport Equations

In the following sections, we first give a transport equation for a general type of radiation to describe the features shared by all the transport equations. We refer to it as the common transport equation (CTE). Then we briefly give the background of neutron transport and add the fission related terms to complete the neutron transport equation (NTE). For radiative heat transfer or thermal radiation transport, we add the black body emission. The radiative heat transfer equation (RHTE) is typically coupled with the thermal conduction equation, which is given for completeness. Equations for more types of particles can be included in the future. Various types of boundary conditions and interface conditions are given for these equations. Operator forms of this equation are given to make the presentations shorter. The multigroup diffusion equation is an approximation of the transport equation and is wide used due to its simplicity. It will be given at the end of each section.

2.1 Common transport equations

The *transient*, multigroup transport equation defined on the three-dimensional (3D) open convex spatial domain \mathcal{D} and on the two-dimensional (2D) unit sphere \mathcal{S} over a one-dimensional (1D) time period $[t_0, t_1]$ is

$$\frac{\partial}{\partial t} \left(\frac{\Psi_g}{v_g} \right) + \vec{\Omega} \cdot \vec{\nabla} \Psi_g + \Sigma_{t,g} \Psi_g(\vec{r}, \vec{\Omega}) = Q_g^{\text{ext}} + \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow g} f_{g' \rightarrow g}(\vec{\Omega}' \cdot \vec{\Omega}) \Psi_{g'}(\vec{r}, \vec{\Omega}') d\Omega' \quad (1a)$$

with the boundary condition

$$\Psi_g(\vec{x}, \vec{\Omega}) = \Psi_g^{\text{inc}}(\vec{x}, \vec{\Omega}) + \alpha_g^s \Psi_g(\vec{x}, \vec{\Omega}_r) + \alpha_g^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} |\vec{\Omega}' \cdot \vec{n}_b| \Psi_g(\vec{x}, \vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} |\vec{\Omega}' \cdot \vec{n}_b| d\Omega'}, \text{ on } \vec{x} \in \partial\mathcal{D} \text{ and } \vec{\Omega} \cdot \vec{n}_b < 0, \quad (1b)$$

where

$$\vec{\Omega}_r = \vec{\Omega} - 2(\vec{\Omega} \cdot \vec{n}_b) \vec{n}_b, \quad (1c)$$

and the initial condition

$$\Psi_g(t = t_0) = \Psi_g^0, \quad (1d)$$

for $g = 1, \dots, G$. All symbols appearing in Eq. (1) are listed below:

- \vec{x} independent spatial variable [cm],
- $\vec{\Omega}$ independent angular variable for radiation flying direction,
- t independent time variable [s],
- g index of the energy group or frequency band from 1 to G ,
- G number of groups or bands,
- $\mathcal{D} \subset \mathbb{R}^3$, open 3D convex spatial domain,
- \mathcal{S} angular domain of the 2D unit sphere,
- t_0 initial time [s],
- t_1 end time [s],
- $\partial\mathcal{D}$ boundary of the spatial domain \mathcal{D} ,
- \vec{n}_b outward unit normal vector on the boundary,
- $\vec{\Omega}_r(\vec{n}_b, \vec{\Omega})$ specular reflecting direction corresponding to $\vec{\Omega}$ on the boundary,
- $\vec{\Omega}'$ secondary direction used by scattering and diffusive reflection
- $\Psi_g(\vec{x}, \vec{\Omega}, t)$ radiation angular flux [$\text{cm}^{-2} \text{s}^{-1} \text{st}^{-1}$] (st stands for steradian),
- $Q_g^{\text{ext}}(\vec{x}, \vec{\Omega}, t)$ external source [$\text{cm}^{-3} \text{s}^{-1} \text{st}^{-1}$],

$$\begin{aligned}
\Psi_g^0(\vec{x}, \vec{\Omega}) & \text{ initial angular flux [cm}^{-2} \text{ s}^{-1} \text{ st}^{-1}], \\
\Psi_g^{\text{inc}}(\vec{x}, \vec{\Omega}, t) & \text{ surface source on } \partial\mathcal{D} \text{ and } \vec{\Omega} \cdot \vec{n}_b < 0 \text{ [cm}^{-2} \text{ s}^{-1} \text{ st}^{-1}], \\
v_g(\vec{x}, t) & \text{ group averaged radiation speed [cm s}^{-1}], \\
\Sigma_{t,g}(\vec{x}, t) & \text{ macroscopic total cross section [cm}^{-1}], \\
\Sigma_{s,g' \rightarrow g}(\vec{x}, t) & \text{ macroscopic scattering cross section from group } g' \text{ to group } g \text{ [cm}^{-1}], \\
f_{g' \rightarrow g}(\vec{x}, t, \vec{\Omega}' \cdot \vec{\Omega}) & \equiv \frac{1}{2\pi} f_{g' \rightarrow g}(\vec{x}, t, \mu_0), \text{ scattering phase function [st}^{-1}], \left(\int_{-1}^1 f_{g' \rightarrow g}(\vec{x}, t, \mu_0) d\mu_0 = 1 \right), \\
\alpha_g^s(\vec{x}, \vec{\Omega}, t) & \text{ specular reflectivity on the boundary,} \\
\alpha_g^d(\vec{x}, \vec{\Omega}, t) & \text{ diffusive reflectivity on the boundary.}
\end{aligned}$$

The first term of Eq. (1a) on the left hand side is the *time-derivative term*. The second term is the *streaming term* and the third is the *collision term*. The second term of Eq. (1a) on the right hand side is the *scattering term*, which couples the angular fluxes of all directions and of all groups together. The G-by-G matrix formed by the scattering cross sections is called the scattering matrix. We define the total macroscopic scattering cross section as

$$\Sigma_{s,g} \equiv \sum_{g'=1}^G \Sigma_{s,g \rightarrow g'}. \quad (1e)$$

It is noted that, physically, all cross sections and v_g are positive and $\Sigma_{s,g} \leq \Sigma_{t,g}$. A material with $\Sigma_{s,g} = 0, g = 1, \dots, G$ is called a pure absorber. A material with $\Sigma_{s,g} = \Sigma_{t,g}, g = 1, \dots, G$ is called a pure scatterer, a material with $\Sigma_{t,g} = 0, g = 1, \dots, G$ is void. The second term of Eq. (1b) on the right hand side is the *specular reflection* and the third term is the *diffusive reflection*. We can perform Legendre polynomial expansion of the phase function and apply the spherical harmonics addition theorem (see Appendix B) to treat the scattering,

$$\sum_{g'=1}^G \int_S \Sigma_{s,g' \rightarrow g} f_{g' \rightarrow g}(\vec{\Omega}' \cdot \vec{\Omega}) \Psi_{g'}(\vec{\Omega}') d\Omega' \approx \sum_{g'=1}^G \sum_{\ell=0}^L \frac{2\ell+1}{4\pi} \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g',\ell,m} Y_{\ell,m}(\vec{\Omega}), \quad (1f)$$

where

$$\begin{aligned}
Y_{\ell,m}(\vec{\Omega}) & \text{ spherical harmonics, refer to Appendix B for their definitions,} \\
L & \text{ truncated scattering order,} \\
\Phi_{g,\ell,m}(\vec{x}, t) & \equiv \int_S \Psi_g(\vec{\Omega}) Y_{\ell,m}(\vec{\Omega}) d\Omega, \text{ angular flux moment of degree } \ell \text{ and order } m \text{ [cm}^{-2} \text{ s}^{-1}], \\
\Sigma_{s,\ell}^{g' \rightarrow g} & \equiv \Sigma_{s,g' \rightarrow g} \int_{-1}^1 f_{g' \rightarrow g}(\vec{x}, t, \mu_0) P_\ell(\mu_0) d\mu_0, \ell\text{-th order of scattering cross section [cm}^{-1}].
\end{aligned}$$

It is noted that $\Phi_{g,0,0}(\vec{x}, t)$ is often of particular interest because most of radiation reactions are isotropic and their rates are the product of $\Phi_{g,0,0}(\vec{x}, t)$ and the corresponding reaction cross sections. Hence, for the sake of brevity we use $\Phi_g(\vec{x}, t)$ as a shorthand notation for $\Phi_{g,0,0}(\vec{x}, t)$ and refer to it as the *scalar flux*. The vector formed by the first angular moments $\{\Phi_{g,1,0}, \Phi_{g,1,1}, \Phi_{g,1,-1}\}$ is also typically a quantity of interest. We give a specific notation \vec{J}_g for this vector and call it the *current*. The tensor

$$\vec{\vec{P}}_g \equiv \frac{1}{v_g} \int_S \vec{\Omega} \vec{\Omega} \Psi_g d\Omega, \quad (1g)$$

formed by the second order angular moments are often referred to as the *radiation pressure*. This name originates from the momentum balance equation (1st moment balance equation) in radiation-hydrodynamics. The ratio of the radiation pressure times the radiation speed to the scalar flux is referred to as the *Eddington tensor*

$$\vec{\vec{E}}_g \equiv \frac{v_g \vec{\vec{P}}_g}{\Phi_g}. \quad (1h)$$

When the angular flux is linearly anisotropic, the Eddington tensor is $\frac{1}{3}$, independent of the solution. How far away the Eddington tensor is from this reference constant tensor can be used as a measure of the significance of local transport effects [33].

If we define the in-group scattering source moment as

$$Q_{g,\ell,m}(\vec{x}, t) \equiv \sum_{g'=1}^G \Sigma_{s,\ell}^{g' \rightarrow g} \Phi_{g',\ell,m}, \quad (1i)$$

the scattering term can be written as

$$\sum_{g'=1}^G \sum_{\ell=0}^L \frac{2\ell+1}{4\pi} \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g',\ell,m} Y_{\ell,m}(\vec{\Omega}) = \sum_{\ell=0}^L \frac{2\ell+1}{4\pi} \sum_{m=-\ell}^{\ell} Q_{g,\ell,m} Y_{\ell,m}. \quad (1j)$$

The external source can also be expanded in spherical harmonics as follows:

$$Q_g^{\text{ext}}(\vec{x}, \vec{\Omega}, t) = \sum_{\ell=0}^{L_s} \frac{2\ell+1}{4\pi} \sum_{m=-\ell}^{\ell} Q_{g,\ell,m}^{\text{ext}} Y_{\ell,m}, \quad (1k)$$

where L_s is the truncated spherical harmonics order for expanding the external source.

Beside the multigroup approximation made in CTE for the physics of radiation transport, several additional assumptions have been made:

1. external forces like gravity, electromagnetism on the radiation are neglected: change of the radiation streaming direction due to these forces is typically negligible;
2. bulk material motion is ignored, thereby assuming the medium is stationary or its velocity is significantly smaller than the radiation velocity;
3. the media are assumed to be isotropic, meaning that the velocity and the total, scattering cross sections have no angular dependency and the phase function depends only on the angle suspended by the incoming and out-going directions;
4. the angular fluxes across medium interfaces are continuous along their streaming direction. This could be invalid in some circumstances such as visible light crossing the interface of two media with different refractive index;
5. radiation-radiation interaction is neglected, thereby assuming the radiation density is so low that the chance for a radiation to interact with another is much smaller than interacting with background media.

Spatial symmetries allow the problem to be represented in an angular space of reduced size. In 2D, we only need to solve the equation over the half sphere $\vec{\Omega} \in \mathcal{S}, \vec{\Omega} \cdot \vec{e}_z > 0$ due to the symmetry with respect to the 2D x-y spatial plane. Thus we can align the polar direction of the angular coordinate with \vec{e}_x , so that the angular flux moments with $m < 0$ are equal to zero. Consequently, the index m in Eq. (1j) only ranges from 0 to ℓ instead of from $-\ell$ to ℓ in 3D. Then, $\Phi_{g,1,0}$ is the current in the x-direction and $\Phi_{g,1,1}$ in the y-direction. We also scale the external source and the scattering source by a factor two, so that the angular flux is doubled and its moments (and reaction rates) remain unchanged whenever the angular integration is done on the half sphere. This factor two is often reflected in the spherical harmonics normalization factor $\frac{2\ell+1}{4\pi}$ in Eq. (1j). If the external source is expanded with spherical harmonics, that same scaling is automatically applied to it. It also need be applied on the boundary source Ψ_g^{inc} . In 1D, we redefine the angular domain as $\mu \in [-1, 1]$ and align the polar direction of the angular coordinate with \vec{e}_x , so that the solution only depends on μ and the angular flux moments with $m \neq 0$ are then equal to zero. The index m is thus dropped in 1D. The external source and the scattering source are then scaled by a factor 2π , so that the angular flux will be scaled by 2π and the angular flux moments remain unchanged. The same factor needs to be applied to the surface sources as in 2D. The streaming term $\vec{\Omega} \cdot \Psi_g$ can be simplified in both 2D and 1D by dropping those zero spatial components due to the symmetry. The special treatment of the phase function in 2D and 1D will be detailed in Appendix D.

Writing the transport equation in operator form makes the notation and all subsequent derivations much cleaner. We therefore define

$$\Psi \equiv \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \vdots \\ \Psi_G \end{bmatrix}, \quad \frac{\Psi}{\mathbf{v}} \equiv \begin{bmatrix} \frac{\Psi_1}{v_1} \\ \frac{\Psi_2}{v_2} \\ \vdots \\ \frac{\Psi_G}{v_G} \end{bmatrix}, \quad \mathbf{Q}^{\text{ext}} \equiv \begin{bmatrix} Q_1^{\text{ext}} \\ Q_2^{\text{ext}} \\ \vdots \\ Q_G^{\text{ext}} \end{bmatrix},$$

over $\mathcal{D} \times \mathcal{S} \times [t_0, t_1]$, henceforth referred to as the *phase space*. We also define the operators mapping a function in the phase space to another function in the same phase space

$$\mathbb{L}\Psi \equiv \begin{bmatrix} \vec{\nabla} \cdot \vec{\Omega} \Psi_1 + \Sigma_{t,1} \Psi_1 \\ \vec{\nabla} \cdot \vec{\Omega} \Psi_2 + \Sigma_{t,2} \Psi_2 \\ \vdots \\ \vec{\nabla} \cdot \vec{\Omega} \Psi_G + \Sigma_{t,G} \Psi_G \end{bmatrix}, \mathbb{S}\Psi \equiv \begin{bmatrix} \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow 1} f_{g' \rightarrow 1} \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow 2} f_{g' \rightarrow 2} \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \vdots \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow G} f_{g' \rightarrow G} \Psi_{g'}(\vec{\Omega}') d\Omega' \end{bmatrix}, \quad (2)$$

where \mathbb{L} is the streaming-collision operator; \mathbb{S} is the scattering operator. Note that we intentionally wrote $\vec{\nabla} \cdot \vec{\Omega} \Psi_g$ which is true in both Cartesian and curvilinear coordinate system; in Cartesian coordinates systems the derivative of $\vec{\Omega}$ with respect to space is zero and hence we obtain $\vec{\nabla} \cdot \vec{\Omega} \Psi_g = \vec{\Omega} \cdot \vec{\nabla} \Psi_g$. We further define the operator for the homogeneous part of the boundary condition mapping from $\partial\mathcal{D} \times \mathcal{S}_{\vec{n}_b}^+ \times [t_0, t_1]$ to $\partial\mathcal{D} \times \mathcal{S}_{\vec{n}_b}^- \times [t_0, t_1]$

$$\mathbb{B}\Psi \equiv \begin{bmatrix} \alpha_1^s \Psi_1(\vec{\Omega}_r) + \alpha_1^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} |\vec{\Omega}' \cdot \vec{n}_b| \Psi_1(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} |\vec{\Omega}' \cdot \vec{n}_b| d\Omega'} \\ \alpha_2^s \Psi_2(\vec{\Omega}_r) + \alpha_2^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} |\vec{\Omega}' \cdot \vec{n}_b| \Psi_2(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} |\vec{\Omega}' \cdot \vec{n}_b| d\Omega'} \\ \vdots \\ \alpha_G^s \Psi_G(\vec{\Omega}_r) + \alpha_G^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} |\vec{\Omega}' \cdot \vec{n}_b| \Psi_G(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} |\vec{\Omega}' \cdot \vec{n}_b| d\Omega'} \end{bmatrix}$$

where $\mathcal{S}_{\vec{v}}^\pm = \{\vec{\Omega} \in \mathcal{S}, \vec{\Omega} \cdot \vec{v} \gtrless 0\}$ stands for the half angular space defined with respect to a unit vector \vec{v} . We can think of $\frac{1}{\mathbf{v}}$ as an operator, although here we adopt the notation of the *multiplication of two same sized vectors*. It is noted that all of these operators \mathbb{L} , \mathbb{S} and \mathbb{B} have no time derivative terms, so they are also valid operators on $\mathcal{D} \times \mathcal{S}$ or $\partial\mathcal{D} \times \mathcal{S}_{\vec{n}_b}^+$. We also define the surface source on $\partial\mathcal{D} \times \mathcal{S}_{\vec{n}_b}^- \times [t_0, t_1]$

$$\Psi^{\text{inc}} \equiv \begin{bmatrix} \Psi_1^{\text{inc}} \\ \Psi_2^{\text{inc}} \\ \vdots \\ \Psi_G^{\text{inc}} \end{bmatrix},$$

and the initial solution on $\mathcal{D} \times \mathcal{S}$

$$\Psi^0 \equiv \begin{bmatrix} \Psi_1^0 \\ \Psi_2^0 \\ \vdots \\ \Psi_G^0 \end{bmatrix}.$$

Finally Eq. (1) becomes

$$\frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) + \mathbb{L}\Psi = \mathbf{Q}^{\text{ext}} + \mathbb{S}\Psi, \quad (3a)$$

with the boundary condition on $\partial\mathcal{D} \times \mathcal{S}_{\vec{n}_b}^- \times [t_0, t_1]$

$$\Psi = \Psi^{\text{inc}} + \mathbb{B}\Psi, \quad (3b)$$

and the initial condition

$$\Psi|_{t=t_0} = \Psi^0. \quad (3c)$$

'-' superscript is used to denote the incoming angular fluxes on the boundary.

With this operator notation, the *steady-state* CTE can be written as

$$\mathbb{L}\Psi = \mathbf{Q}^{\text{ext}} + \mathbb{S}\Psi, \quad (4a)$$

with the boundary condition on $\partial\mathcal{D} \times \mathcal{S}_{\vec{n}_b}^-$

$$\Psi = \Psi^{\text{inc}} + \mathbb{B}\Psi. \quad (4b)$$

It is noted that the time dependence is dropped whenever possible and the operators \mathbb{L} , \mathbb{S} and \mathbb{B} are re-interpreted as acting on time-independent functions.

The *adjoint* equation can be derived through the variational form of CTE. We first define the following inner products

$$(\mathbf{a}, \mathbf{b}) \equiv \sum_{g=1}^G \int_{t_0}^{t_1} dt \int_S d\Omega \int_D dx a_g(\vec{x}, \vec{\Omega}, t) b_g(\vec{x}, \vec{\Omega}, t), \quad (5a)$$

$$(\mathbf{a}, \mathbf{b})|_{t=t'} \equiv \sum_{g=1}^G \int_S d\Omega \int_D dx a_g(\vec{x}, \vec{\Omega}, t=t') b_g(\vec{x}, \vec{\Omega}, t=t'), \quad (5b)$$

$$\langle \mathbf{a}, \mathbf{b} \rangle^\pm \equiv \sum_{g=1}^G \int_{t_0}^{t_1} dt \int_{\partial D} dx \int_{S_{\vec{n}_b}^\pm} d\Omega \left| \vec{\Omega} \cdot \vec{n}_b \right| a_g(\vec{x}, \vec{\Omega}, t) b_g(\vec{x}, \vec{\Omega}, t), \quad (5c)$$

where \mathbf{a} and \mathbf{b} stand for two arbitrary functions defined on the phase space. Then, we define the functional \mathbb{G} :

$$\begin{aligned} \mathbb{G}(\Psi^*, \Psi) \equiv & \left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) + \left(\frac{\Psi^*}{\mathbf{v}}, \Psi \right) \Big|_{t=t_0} - \left(\frac{\Psi^*}{\mathbf{v}}, \Psi_0 \right) \Big|_{t=t_0} - \left(\Psi_1^*, \frac{\Psi}{\mathbf{v}} \right) \Big|_{t=t_1} \\ & + (\Psi^*, \mathbb{L}\Psi) + \langle \Psi^*, \Psi \rangle^- - \langle \Psi^*, \mathbb{B}\Psi \rangle^- - \langle \Psi^*, \Psi_b^{\text{inc}} \rangle^- - \langle \Psi_b^{\text{out}}, \Psi \rangle^+ \\ & - (\Psi^*, \mathbb{S}\Psi) - (\Psi^*, \mathbf{Q}^{\text{ext}}) - (\mathbf{Q}^{\text{ext}}, \Psi), \end{aligned} \quad (6)$$

where

$$\Psi^* = \begin{bmatrix} \Psi_1^* \\ \Psi_2^* \\ \vdots \\ \Psi_G^* \end{bmatrix}; \mathbf{Q}^{\text{ext}} = \begin{bmatrix} Q_1^{\text{ext}} \\ Q_2^{\text{ext}} \\ \vdots \\ Q_G^{\text{ext}} \end{bmatrix},$$

are the adjoint and the adjoint external source defined on the phase space and

$$\Psi_1^* = \begin{bmatrix} \Psi_1^{*1} \\ \Psi_2^{*1} \\ \vdots \\ \Psi_G^{*1} \end{bmatrix}; \Psi^{\text{out}} = \begin{bmatrix} \Psi_1^{\text{out}} \\ \Psi_2^{\text{out}} \\ \vdots \\ \Psi_G^{\text{out}} \end{bmatrix},$$

are the final adjoint conditions at $t = t_1$ and the boundary out-flow angular fluxes. We can see that the solution of Eq. (1) is the stationary point of this functional because it is the corresponding Euler-Lagrange equation. We will now derive the adjoint equations as stationary point of the functional \mathbb{G} with respect to variations in Ψ . Using integration by parts it is easy to show that

$$\left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) + \left(\frac{\Psi^*}{\mathbf{v}}, \Psi \right) \Big|_{t=t_0} = - \left(\frac{1}{\mathbf{v}} \frac{\partial}{\partial t} \Psi^*, \Psi \right) + \left(\Psi^*, \frac{\Psi}{\mathbf{v}} \right) \Big|_{t=t_1}, \quad (7)$$

$$(\Psi^*, \mathbb{L}\Psi) + \langle \Psi^*, \Psi \rangle^- = (\mathbb{L}^* \Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+, \quad (8)$$

where we defined

$$\mathbb{L}^* \Psi \equiv \begin{bmatrix} \left(-\vec{\Omega} \cdot \vec{\nabla} + \Sigma_{t,1} \right) \Psi_1 \\ \left(-\vec{\Omega} \cdot \vec{\nabla} + \Sigma_{t,2} \right) \Psi_2 \\ \vdots \\ \left(-\vec{\Omega} \cdot \vec{\nabla} + \Sigma_{t,G} \right) \Psi_G \end{bmatrix}.$$

Similarly, using integration by parts on the scattering term reveals that

$$(\Psi^*, \mathbb{S}\Psi) = (\mathbb{S}^* \Psi^*, \Psi), \quad (9)$$

where

$$\mathbb{S}^* \Psi \equiv \begin{bmatrix} \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,1 \rightarrow g'} f_{1 \rightarrow g'} \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,2 \rightarrow g'} f_{2 \rightarrow g'} \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \vdots \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,G \rightarrow g'} f_{G \rightarrow g'} \Psi_{g'}(\vec{\Omega}') d\Omega' \end{bmatrix},$$

and finally the reflecting boundary terms satisfies

$$\langle \Psi^*, \mathbb{B} \Psi \rangle^- = \langle \mathbb{B}^* \Psi^*, \Psi \rangle^+ \quad (10)$$

where

$$\mathbb{B}^* \Psi \equiv \begin{bmatrix} \alpha_1^s \Psi_1(\vec{\Omega}_r) + \alpha_1^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} |\vec{\Omega}' \cdot \vec{n}_b| \Psi_1(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} |\vec{\Omega}' \cdot \vec{n}_b| d\Omega'} \\ \alpha_2^s \Psi_2(\vec{\Omega}_r) + \alpha_2^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} |\vec{\Omega}' \cdot \vec{n}_b| \Psi_2(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} |\vec{\Omega}' \cdot \vec{n}_b| d\Omega'} \\ \vdots \\ \alpha_G^s \Psi_G(\vec{\Omega}_r) + \alpha_G^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} |\vec{\Omega}' \cdot \vec{n}_b| \Psi_G(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} |\vec{\Omega}' \cdot \vec{n}_b| d\Omega'} \end{bmatrix}.$$

The adjoint equation can now be obtained as the Euler-Lagrange equation of the transformed functional \mathbb{G} with the variation of Ψ :

$$-\frac{1}{\mathbf{v}} \frac{\partial}{\partial t} \Psi^* + \mathbb{L}^* \Psi^* = \mathbf{Q}^{\text{ext}} + \mathbb{S}^* \Psi^*, \quad (11a)$$

along with the boundary condition on $\partial \mathcal{D} \times \mathcal{S}_{\vec{n}_b}^+ \times [t_0, t_1]$

$$\Psi^* = \Psi^{\text{out}} + \mathbb{B}^* \Psi^*, \quad (11b)$$

and the final condition

$$\Psi^*|_{t=t_1} = \Psi^{*1}. \quad (11c)$$

Solution of the adjoint equation is also the stationary point of the variational of Eq. (6). It is typical to redefine the adjoint angular flux $\Psi^*(\vec{\Omega}) \leftarrow \Psi^*(-\vec{\Omega})$. Then the adjoint equation can be solved by simply solving the forward equation with the scattering matrix being transposed, a factor (-1) being applied to the odd spherical harmonics and the direction of time being inverted, i.e. the problem is solved starting at final times and ending at the initial time. The solution obtained with the discretization applied to the adjoint equation is called the *physical adjoint*.

If we integrate Eq. (3) over the 2D unit sphere \mathcal{S} , we obtain the balance equation

$$\frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}} \right) + \vec{\nabla} \cdot \vec{\mathbf{J}} + \mathbb{L}_2 \Phi = \mathbf{Q}_0^{\text{ext}} + \mathbb{S}_0 \Phi, \quad (12)$$

where

$$\Phi \equiv \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_G \end{bmatrix}, \quad \frac{\Phi}{\mathbf{v}} \equiv \begin{bmatrix} \frac{\Phi_1}{v_1} \\ \frac{\Phi_2}{v_2} \\ \vdots \\ \frac{\Phi_G}{v_G} \end{bmatrix}, \quad \vec{\mathbf{J}} \equiv \begin{bmatrix} \vec{J}_1 \\ \vec{J}_2 \\ \vdots \\ \vec{J}_G \end{bmatrix}, \quad \mathbf{Q}_0^{\text{ext}} \equiv \begin{bmatrix} \int_{\mathcal{S}} Q_1^{\text{ext}} d\Omega \\ \int_{\mathcal{S}} Q_2^{\text{ext}} d\Omega \\ \vdots \\ \int_{\mathcal{S}} Q_G^{\text{ext}} d\Omega \end{bmatrix},$$

and

$$\mathbb{L}_2 \Phi \equiv \begin{bmatrix} \Sigma_{t,1} \Phi_1 \\ \Sigma_{t,2} \Phi_2 \\ \vdots \\ \Sigma_{t,G} \Phi_G \end{bmatrix}, \quad \mathbb{S}_0 \Phi \equiv \begin{bmatrix} \sum_{g'=1}^G \Sigma_{s,g' \rightarrow 1} \Phi_{g'} \\ \sum_{g'=1}^G \Sigma_{s,g' \rightarrow 2} \Phi_{g'} \\ \vdots \\ \sum_{g'=1}^G \Sigma_{s,g' \rightarrow G} \Phi_{g'} \end{bmatrix}.$$

It is noted that we use the subscript 0 to indicate the isotropic part of the external source and the zeroth moment of the scattering operator. It is common to merge the ingroup scattering into the total collision by defining the macroscopic removal cross sections

$$\Sigma_{r,g} \equiv \Sigma_{t,g} - \Sigma_{s,g \rightarrow g}, \quad g = 1, \dots, G, \quad (13)$$

and the two operators

$$\mathbb{L}_3 \Phi \equiv \begin{bmatrix} \Sigma_{r,1} \Phi_1 \\ \Sigma_{r,2} \Phi_2 \\ \vdots \\ \Sigma_{r,G} \Phi_G \end{bmatrix}, \mathbb{S}_d \Phi \equiv \begin{bmatrix} \sum_{g' \neq 1} \Sigma_{s,g' \rightarrow 1} \Phi_{g'} \\ \sum_{g' \neq 2} \Sigma_{s,g' \rightarrow 2} \Phi_{g'} \\ \vdots \\ \sum_{g' \neq G} \Sigma_{s,g' \rightarrow G} \Phi_{g'} \end{bmatrix},$$

the balance equation becomes

$$\frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}} \right) + \vec{\nabla} \cdot \vec{\mathbf{J}} + \mathbb{L}_3 \Phi = \mathbf{Q}_0^{\text{ext}} + \mathbb{S}_d \Phi. \quad (14)$$

If we multiply Eq. (3b) with $|\vec{\Omega} \cdot \vec{n}_b|$ and integrate over $\mathcal{S}_{\vec{n}_b}^-$, we obtain the boundary condition for the balance equation,

$$\mathbf{J}^i = \mathbf{J}^{\text{inc}} + \alpha \mathbf{J}^o, \quad (15)$$

where

$$\mathbf{J}^i \equiv \begin{bmatrix} \int_{|\vec{\Omega} \cdot \vec{n}_b| < 0} |\vec{\Omega} \cdot \vec{n}_b| \Psi_1 d\Omega \\ \int_{|\vec{\Omega} \cdot \vec{n}_b| < 0} |\vec{\Omega} \cdot \vec{n}_b| \Psi_2 d\Omega \\ \vdots \\ \int_{|\vec{\Omega} \cdot \vec{n}_b| < 0} |\vec{\Omega} \cdot \vec{n}_b| \Psi_G d\Omega \end{bmatrix}, \mathbf{J}^o \equiv \begin{bmatrix} \int_{|\vec{\Omega} \cdot \vec{n}_b| > 0} |\vec{\Omega} \cdot \vec{n}_b| \Psi_1 d\Omega \\ \int_{|\vec{\Omega} \cdot \vec{n}_b| > 0} |\vec{\Omega} \cdot \vec{n}_b| \Psi_2 d\Omega \\ \vdots \\ \int_{|\vec{\Omega} \cdot \vec{n}_b| > 0} |\vec{\Omega} \cdot \vec{n}_b| \Psi_G d\Omega \end{bmatrix}, \mathbf{J}^{\text{inc}} \equiv \begin{bmatrix} J_1^{\text{inc}} \\ J_2^{\text{inc}} \\ \vdots \\ J_G^{\text{inc}} \end{bmatrix},$$

and

$$\alpha \equiv \begin{bmatrix} \alpha_1^s + \alpha_1^d \\ \alpha_2^s + \alpha_2^d \\ \vdots \\ \alpha_G^s + \alpha_G^d \end{bmatrix}.$$

The incoming boundary partial current is defined as:

$$J_g^{\text{inc}} = \int_{|\vec{\Omega} \cdot \vec{n}_b| < 0} |\vec{\Omega} \cdot \vec{n}_b| \Psi_g^{\text{inc}} d\Omega \quad (16)$$

This is also known as the albedo boundary condition. We use superscript 'o' and 'i' to denote the out-going and incoming partial currents with respect to the solution domain. When $\alpha = \mathbf{I}$, the condition is

$$-\vec{\mathbf{J}} \cdot \vec{n}_b = \mathbf{J}^{\text{inc}}, \quad (17)$$

where $\vec{\mathbf{J}} \cdot \vec{n}_b$ is the net out-going current and equal to $\mathbf{J}^o - \mathbf{J}^i$.

If we multiply Eq. (3) with $\vec{\Omega}$ and integrate over the 2D unit sphere \mathcal{S} , we obtain the set of momentum equations or the first moment equations

$$\frac{\partial}{\partial t} \left(\frac{\vec{\mathbf{J}}}{\mathbf{v}} \right) + \vec{\nabla} \cdot (\vec{\mathbf{E}} \Phi) + \mathbb{L}_2 \vec{\mathbf{J}} = \vec{\mathbf{Q}}_1^{\text{ext}} + \mathbb{S}_1 \vec{\mathbf{J}}, \quad (18)$$

where

$$\vec{\mathbf{E}} \equiv \begin{bmatrix} \vec{E}_1 \\ \vec{E}_2 \\ \vdots \\ \vec{E}_G \end{bmatrix}, \mathbb{S}_1 \vec{\mathbf{J}} \equiv \begin{bmatrix} \sum_{g'=1}^G \Sigma_{s,g' \rightarrow 1,1} \vec{J}_{g'} \\ \sum_{g'=1}^G \Sigma_{s,g' \rightarrow 2,1} \vec{J}_{g'} \\ \vdots \\ \sum_{g'=1}^G \Sigma_{s,g' \rightarrow G,1} \vec{J}_{g'} \end{bmatrix}.$$

If we assume that the angular flux varies only linearly with angle we know that $\vec{\mathbf{E}} = \frac{\vec{\mathbf{I}}}{3}$. If we additionally neglect the time derivative term of the currents, the first angular moments of the external source and the off-group linearly anisotropic scattering terms, we obtain

$$\vec{\mathbf{J}} = -\mathbb{L}_{\text{tr}}^{-1} \vec{\nabla} \Phi, \quad (19)$$

where we defined

$$\Sigma_{\text{tr},g} \equiv \Sigma_{\text{t},g} - \Sigma_{\text{s},g \rightarrow g,1}, \quad g = 1, \dots, G, \quad (20)$$

and the operator

$$\mathbb{L}_{\text{tr}} \vec{J} \equiv \begin{bmatrix} \Sigma_{\text{tr},1} \vec{J}_1 \\ \Sigma_{\text{tr},2} \vec{J}_2 \\ \vdots \\ \Sigma_{\text{tr},G} \vec{J}_G \end{bmatrix}.$$

Eq. (19) is a relationship between the current and the gradient of the scalar fluxes and it provides a way of obtaining the diffusion coefficient in Fick's law.

2.2 Common diffusion equation

The common diffusion equation (CDE) can be derived from the CTE with a P_0 projection or asymptotic analysis. If Fick's law

$$\vec{J}_g = D_g \vec{\nabla} \Phi_g, \quad g = 1, \dots, G \quad (21)$$

is valid for evaluating the current, we can get the CDE from the balance equation. The transient CDE on the 3D open convex spatial domain \mathcal{D} within the 1D time period $[t_0, t_1]$ is

$$\frac{\partial}{\partial t} \left(\frac{\Phi_g}{v_g} \right) - \vec{\nabla} \cdot D_g \vec{\nabla} \Phi_g + \Sigma_{\text{r},g} \Phi_g = Q_{g,0}^{\text{ext}} + \sum_{g' \neq g} \Sigma_{\text{s},g' \rightarrow g} \Phi_{g'}, \quad (22a)$$

with the Dirichlet boundary condition

$$\Phi_g = \Phi_{\text{b},g}, \quad \vec{x} \in \partial \mathcal{D}_1, \quad (22b)$$

the Neumann boundary condition

$$D_g \vec{\nabla} \Phi_g \cdot \vec{n}_{\text{b}} = J_{\text{b},g}^{\text{net}}, \quad \vec{x} \in \partial \mathcal{D}_2, \quad (22c)$$

and the Robin boundary condition

$$\frac{\Phi_g}{4} + \frac{e_g}{2} D_g \vec{\nabla} \Phi_g \cdot \vec{n}_{\text{b}} = J_{\text{b},g}^{\text{inc}}, \quad \vec{x} \in \partial \mathcal{D}_3, \quad (22d)$$

where $\partial \mathcal{D} = \partial \mathcal{D}_1 \cup \partial \mathcal{D}_2 \cup \partial \mathcal{D}_3$, and the initial condition

$$\Phi_g(t = t_0) = \Phi_g^0, \quad (22e)$$

for $g = 1, \dots, G$. All the symbols in Eq. (22) that were not already defined in Eq. (1) are listed below:

$\partial \mathcal{D}_1$	Dirichlet boundary,
$\partial \mathcal{D}_2$	Neumann boundary,
$\partial \mathcal{D}_3$	Robin boundary,
$\Phi_g(\vec{x}, t)$	scalar flux [$\text{cm}^{-2} \text{s}^{-1}$],
$D_g(\vec{x}, t)$	diffusion coefficient [cm],
$Q_{g,0}^{\text{ext}}(\vec{x}, t)$	$\equiv \int_S Q_g^{\text{ext}} d\Omega$, isotropic external source [$\text{cm}^{-3} \text{s}^{-1}$],
$\Phi_g^0(\vec{x})$	initial scalar flux [$\text{cm}^{-2} \text{s}^{-1}$],
$\Phi_{\text{b},g}(\vec{x}, t)$	scalar flux on Dirichlet boundary [$\text{cm}^{-2} \text{s}^{-1}$],
$J_{\text{b},g}^{\text{net}}(\vec{x}, t)$	net incoming current on Neumann boundary [$\text{cm}^{-2} \text{s}^{-1}$],
$J_{\text{b},g}^{\text{inc}}(\vec{x}, t)$	partial incoming partial current on Robin boundary [$\text{cm}^{-2} \text{s}^{-1}$],
$\Sigma_{\text{r},g}(\vec{x}, t)$	$\equiv \Sigma_{\text{t},g} - \Sigma_{\text{s},g \rightarrow g}$, macroscopic removal cross section [cm^{-1}],

$e_g(\vec{x}, t)$ a factor used to have the diffusion solution better match the transport solution.

The first term of Eq. (22a) on the left hand side is the *time-derivative* term. The second term is the *diffusion term* and the third is the *removal term*. The second term on the right hand side is the *scattering term*, which couples the scalar fluxes of all groups together. We can define the in-group scattering source as

$$Q_g(\vec{x}, t) \equiv \sum_{g' \neq g} \Sigma_{s,g' \rightarrow g} \Phi_{g'}, \quad (23)$$

which is different from the transport in-scattering source $Q_{g,0,0}$ in that $\Sigma_{s,g \rightarrow g} \Phi_g$ is not included. Because the CTE does not have any angular dependency, the 3D CDE notation also applies to 2D and 1D. The steady-state equation can be obtained by simply dropping the time-derivative term, the initial condition and the time dependencies of all material properties and variables. With a derivation similar to that described above for the CTE, we obtain the following CDE adjoint equation,

$$-\frac{1}{v_g} \frac{\partial}{\partial t} \Phi_g^* - \vec{\nabla} \cdot D_g \vec{\nabla} \Phi_g^* + \Sigma_{r,g} \Phi_g^* = Q_{g,0}^{*ext} + \sum_{g' \neq g} \Sigma_{s,g \rightarrow g'} \Phi_{g'}^*, \quad (24a)$$

with the Dirichlet boundary condition

$$\Phi_g^* = \Phi_{b,g}^*, \vec{x} \in \partial \mathcal{D}_1, \quad (24b)$$

the Neumann boundary condition

$$D_g \vec{\nabla} \Phi_g^* \cdot \vec{n}_b = J_{b,g}^{net*}, \vec{x} \in \partial \mathcal{D}_2, \quad (24c)$$

the Robin boundary condition

$$\frac{\Phi_g^*}{4} + \frac{e_g}{2} D_g \vec{\nabla} \Phi_g^* \cdot \vec{n}_b = J_{b,g}^{*inc}, \vec{x} \in \partial \mathcal{D}_3, \quad (24d)$$

and the final condition

$$\Phi_g^*(t = t_1) = \Phi_g^{*1}, \quad (24e)$$

for $g = 1, \dots, G$. The Robin boundary condition is the analog of the surface source boundary condition in the CTE. The homogeneous Neumann boundary condition can be derived from the reflecting boundary condition. If we assume the angular flux can be expanded as

$$\Psi_g = \frac{1}{4\pi} \Phi_g - \frac{3}{4\pi} D_g \vec{\nabla} \Phi_g \cdot \vec{\Omega}, \quad (25)$$

we can substitute it into

$$\begin{aligned} \int_{\vec{\Omega} \cdot \vec{n}_b < 0} \Psi_g \left| \vec{\Omega} \cdot \vec{n}_b \right| d\Omega &= \int_{\vec{\Omega} \cdot \vec{n}_b < 0} \Psi_g^{inc} \left| \vec{\Omega} \cdot \vec{n}_b \right| d\Omega + \\ &\alpha_g^s \int_{\vec{\Omega} \cdot \vec{n}_b < 0} \Psi_g(\vec{\Omega}_r) \left| \vec{\Omega} \cdot \vec{n}_b \right| d\Omega + \\ &\alpha_g^d \int_{\vec{\Omega} \cdot \vec{n}_b < 0} \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} \left| \vec{\Omega}' \cdot \vec{n}_b \right| \Psi_g(\vec{x}, \vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} \left| \vec{\Omega}' \cdot \vec{n}_b \right| d\Omega'} \left| \vec{\Omega} \cdot \vec{n}_b \right| d\Omega \end{aligned}$$

to obtain

$$\frac{1}{4} \Phi_g + \frac{1}{2} D_g \vec{\nabla} \Phi_g \cdot \vec{n}_b = J_g^{inc} + \left(\alpha_g^s + \alpha_g^d \right) \left(\frac{1}{4} \Phi_g - \frac{1}{2} D_g \vec{\nabla} \Phi_g \cdot \vec{n}_b \right).$$

If the reflectivity $\alpha_g^s + \alpha_g^d$ is zero, we retrieve the Robin boundary condition with the factor e_g being equal to 1. If the reflectivity $\alpha_g^s + \alpha_g^d$ is 1, we retrieve the Neumann boundary condition. Asymptotic analysis for the diffusion limit of the transport equation with isotropic scattering gives the following boundary condition when the reflectivity is zero:

$$\frac{1}{4} \Phi_g + \frac{3z_0}{2} \frac{1}{2} D_g \vec{\nabla} \Phi_g \cdot \vec{n}_b = \int_{\vec{\Omega} \cdot \vec{n}_b < 0} \frac{1}{2} W(\left| \vec{\Omega} \cdot \vec{n}_b \right|) \Psi_g^{inc} d\Omega, \quad (26)$$

where

$$\begin{aligned}
 z_0 &\equiv \int_0^1 W(\mu) \mu d\mu \approx 0.710446, \\
 W(\mu) &\equiv \frac{\sqrt{3}}{2} \mu H_{c=1}(\mu), \\
 H_{c=1}(\mu) &\text{ Chandrasekhar's H-function for a } c = 1 \text{ isotropic scattering medium [34],} \\
 D_g &\equiv \frac{1}{3\Sigma_{t,g}}, \text{ asymptotic diffusion coefficient with isotropic scattering.}
 \end{aligned}$$

If Ψ_g^{inc} is isotropic, because $\int_{\vec{\Omega} \cdot \vec{n}_b < 0} \frac{1}{2} W(|\vec{\Omega} \cdot \vec{n}_b|) d\Omega = \pi$, we see that $\int_{\vec{\Omega} \cdot \vec{n}_b < 0} \frac{1}{2} W(|\vec{\Omega} \cdot \vec{n}_b|) \Psi_g^{\text{inc}} d\Omega = J_g^{\text{inc}}$. Eq. (26) gives a better equivalent incoming partial current for the diffusion equation when the surface source is anisotropic. It also explains why we keep the e_g factor, which is often understood as the extrapolation factor. Extension of asymptotic analysis for anisotropic scattering with boundary reflection is possible but not pursued in this manual. The Dirichlet boundary condition is valid only for the diffusion equation and does not have a correspondence in the transport equation.

If we define the operator

$$\mathbb{D}\vec{J} \equiv \begin{bmatrix} D_1 \vec{J}_1 \\ D_2 \vec{J}_2 \\ \vdots \\ D_G \vec{J}_G \end{bmatrix},$$

the CDE in operator form is the balance equation with $\vec{\nabla} \cdot \vec{J}$ being replaced with $-\vec{\nabla} \cdot \mathbb{D}\vec{\nabla}\Phi$:

$$\frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}} \right) - \vec{\nabla} \cdot \mathbb{D}\vec{\nabla}\Phi + \mathbb{L}_3\Phi = \mathbf{Q}_0^{\text{ext}} + \mathbb{S}_d\Phi, \quad (27a)$$

with the boundary condition

$$\Phi = \Phi_b, \quad \vec{x} \in \partial\mathcal{D}_1, \quad (27b)$$

$$-\mathbb{D}\vec{\nabla}\Phi \cdot \vec{n}_b = -\mathbf{J}^{\text{net}}, \quad \vec{x} \in \partial\mathcal{D}_2, \quad (27c)$$

$$-\mathbb{D}\vec{\nabla}\Phi \cdot \vec{n}_b = \frac{1}{2}\mathbf{e}\Phi - 2\mathbf{e}\mathbf{J}^{\text{inc}}, \quad \vec{x} \in \partial\mathcal{D}_3, \quad (27d)$$

where

$$\mathbf{e} \equiv \begin{bmatrix} \frac{1}{e_1} \\ \frac{1}{e_2} \\ \vdots \\ \frac{1}{e_G} \end{bmatrix},$$

and the initial condition

$$\Phi(t = t_0) = \Phi^0. \quad (27e)$$

Similarly we can write down the operator form of the adjoint CDE:

$$-\frac{1}{\mathbf{v}} \frac{\partial}{\partial t} \Phi^* - \vec{\nabla} \cdot \mathbb{D}\vec{\nabla}\Phi^* + \mathbb{L}_3\Phi^* = \mathbf{Q}_0^{\text{ext}*} + \mathbb{S}_d^*\Phi^*, \quad (28a)$$

where

$$\Phi^* \equiv \begin{bmatrix} \Phi_1^* \\ \Phi_2^* \\ \vdots \\ \Phi_G^* \end{bmatrix}, \quad \mathbb{S}_d^*\Phi^* \equiv \begin{bmatrix} \sum_{g' \neq 1} \Sigma_{s,1 \rightarrow g'} \Phi_{g'}^* \\ \sum_{g' \neq 2} \Sigma_{s,2 \rightarrow g'} \Phi_{g'}^* \\ \vdots \\ \sum_{g' \neq G} \Sigma_{s,G \rightarrow g'} \Phi_{g'}^* \end{bmatrix},$$

with the boundary condition

$$\Phi^* = \Phi_b^*, \quad \vec{x} \in \partial\mathcal{D}_1, \quad (28b)$$

$$-\mathbb{D} \vec{\nabla} \Phi^* \cdot \vec{n}_b = -\mathbf{J}^{\text{net}*}, \quad \vec{x} \in \partial \mathcal{D}_2, \quad (28c)$$

$$-\mathbb{D} \vec{\nabla} \Phi^* \cdot \vec{n}_b = \frac{1}{2} \mathbf{e} \Phi^* - 2\mathbf{e} \mathbf{J}^{\text{inc}*}, \quad \vec{x} \in \partial \mathcal{D}_3, \quad (28d)$$

and the final condition

$$\Phi^*(t = t_1) = \Phi^{*1}. \quad (28e)$$

The diffusion coefficient is typically a scalar value based on the P_0 projection or asymptotic analysis. But in general it can be a tensor. For instance, we can have a tensor diffusion coefficient evaluated from the transport angular fluxes so that the CDE can reproduce the scalar fluxes of the CTE:

$$D_{u,v,g} = \begin{cases} \frac{\int_{4\pi} \partial_u \Psi_g d\Omega}{\int_{4\pi} \Omega_u \Psi_g d\Omega}, & u=v \text{ and } \int_{4\pi} \Omega_u \Psi_g d\Omega \neq 0; \\ \min \left(\frac{1}{\Sigma_{t,g} - \Sigma_{s,g \rightarrow g}}, D_0 \right), & u=v \text{ and } \int_{4\pi} \Omega_u \Psi_g d\Omega = 0; \\ 0, & u \neq v; \end{cases} \quad (29)$$

where D_0 is a threshold to ensure the value is not infinity in void. Theoretically speaking, there exists a tensor diffusion coefficient which can let the diffusion scalar flux match the transport scalar flux.

The standard derivation of the diffusion approximation takes the 0th and 1st angular moment of the transport equation and then proceeds to derive Fick's law by assuming that the flux varies at most linearly in angle (this assumption is incorporated in Eq. (25)) and that the time derivative in the 1st angular moment equation can be neglected. Applying these approximations leads to the following 1st moment equation:

$$\frac{1}{3} \vec{\nabla} \Phi_g + \Sigma_{t,g} \vec{J}_g = \sum_{g'=1}^G \Sigma_{s,1}^{g' \rightarrow g} \vec{J}_{g'}. \quad (30)$$

Without additional approximation, it is impossible to define a scalar diffusion coefficient fitting the definition of Eq. (21) from Eq. (30). Therefore, another approximation is introduced that we refer to as the transport approximation:

$$\sum_{g'=1}^G \Sigma_{s,1}^{g' \rightarrow g} \vec{J}_{g'} = \sum_{g'=1}^G \Sigma_{s,1}^{g \rightarrow g'} \vec{J}_g \quad (31)$$

Using Eq. (31) results in the well-known definition of the diffusion coefficient:

$$D_g = \frac{1}{3 \left(\Sigma_{t,g} - \sum_{g'=1}^G \Sigma_{s,1}^{g \rightarrow g'} \right)} \quad (32)$$

Within the Rattlesnake code, this definition of the diffusion coefficient is referred to as the local diffusion coefficient as it can be directly computed from the local cross sections. The local diffusion coefficient is incompatible with voids.

The non-local diffusion theory was developed by Morel and Larsen in [35] and [36], respectively, with a recent extension to include anisotropic scattering in [37]. This section will briefly review the relevant points, but the reader is encouraged to consult the references for a more detailed description. The difference in the derivation of non-local diffusion theory and standard diffusion theory is that we assume the following:

- The scalar flux is of order one: $\Phi_g = \mathcal{O}(1)$,
- Weak spatial derivatives: $\vec{\nabla} \Psi_g = \mathcal{O}(\epsilon)$,
- Small currents: $\vec{J}_g = \mathcal{O}(\epsilon)$,

where ϵ is a small parameter typically used in asymptotic diffusion analysis. Instead of deriving Fick's law from the 1st angular moment equation, it is derived from formally inverting the streaming and collision operator onto the right hand side of the transport equation featuring linearly anisotropic scattering, and then deriving an equation for

the current by multiplication with $\vec{\Omega}$ and integration over the unit sphere. Application of the stated assumptions leads to a tensor diffusion coefficient defined by:

$$\vec{D}_g = \left(\vec{I} - \vec{M}_{g,1} \right)^{-1} \vec{M}_{g,0}, \quad (33a)$$

where

$$\vec{M}_{g,\ell} = \frac{1}{4\pi} \int_S \vec{\Omega} \vec{\Omega} F_{g,\ell}(\vec{x}, \vec{\Omega}) d\Omega, \text{ for } \ell = 0, 1. \quad (33b)$$

The auxiliary fluxes $F_{g,\ell}(\vec{x}, \vec{\Omega})$ are the solutions of the following auxiliary problems:

$$\begin{aligned} \vec{\Omega} \cdot \vec{\nabla} F_{g,0} + \Sigma_{t,g}(\vec{x}) F_{g,0}(\vec{x}, \vec{\Omega}) &= 1, \quad \vec{x} \in \mathcal{D} \\ \vec{\Omega} \cdot \vec{\nabla} F_{g,1} + \Sigma_{t,g}(\vec{x}) F_{g,1}(\vec{x}, \vec{\Omega}) &= 3\Sigma_{s,1}^g, \quad \vec{x} \in \mathcal{D} \\ F_{g,0} &= F_{g,1} = 0, \quad \vec{x} \in \partial\mathcal{D}_v \text{ and } \vec{\Omega} \cdot \vec{n}_b < 0 \\ F_{g,0}(\vec{x}, \vec{\Omega}) &= F_{g,0}(\vec{x}, \vec{\Omega}_r), \quad F_{g,1}(\vec{x}, \vec{\Omega}) = F_{g,1}(\vec{x}, \vec{\Omega}_r), \quad \vec{x} \in \partial\mathcal{D}_r \text{ and } \vec{\Omega} \cdot \vec{n}_b < 0, \end{aligned} \quad (33c)$$

where the final equations are identical to the ones found in [37] with the exception of the $F_{g,1}$ equation that has been extended to multigroup by using the transport approximation Eq. (31). The question of the correct boundary conditions has not been answered conclusively. First, it should be mentioned that the vacuum and reflective pieces of the boundary $\partial\mathcal{D}_v$ and $\partial\mathcal{D}_r$, respectively, need to coincide with the respective fractions in the original transport problem. Trahan and Larsen, [36], suggested imposing reflective boundary conditions everywhere because it would simplify the diffusion tensor to the standard $1/3\Sigma_t$ in case of an infinite homogeneous medium. Later Larsen showed that vacuum boundary conditions should be applied everywhere [37]. Finally, without solid mathematical backing, natural boundary conditions where $\partial\mathcal{D}_v$ and $\partial\mathcal{D}_r$ coincide with their respective counterparts of the original transport problem could be used. The main importance of non-local diffusion theory is that it is well defined in voids. Moreover it does not assume at most linear variation of the angular flux with direction, but allows deviations from it inside a small cone as long as these deviations do not affect the overall scalar flux too much [37]. The best diffusion coefficient for compensating transport effects in realistic simulations without resorting the transport solution is still an open research area. Evaluating diffusion coefficient with Monte Carlo method is also an active research area.

2.3 Neutron

2.3.1 Physics of neutron transport

Neutron transport is one of the most important physics occurring in nuclear reactors. Neutrons are born from fission reactions or from spontaneous fission events of some heavy actinides; due to the nature of the reactions creating neutrons energies above 20MeV are rare. Since the temperature of nuclear reactor components are close to or above the ambient temperature (20.46°C, where nuclear data are measured and evaluated), corresponding to 0.025301eV, free neutrons with energies below 0.001eV are also rare. Neutrons in fast reactors have a higher chance to be absorbed or leak out of the reactor due to insufficient moderating before reaching the thermal range than in thermal reactors. The energy range of importance for neutron transport is typically between 0.001eV and 20MeV. This range may vary slightly depending on the multigroup libraries generated by different organizations or on the reactor type. Traditionally, we order groups from fast to thermal with descending energy. The upper bound of the 1-st energy group for neutron transport is thus equal to the highest energy of the library, e.g. 20MeV, and the lower bound of the G -th energy group is the lowest energy of the library, e.g. 0.001eV for most thermal reactor applications. The number of energy groups G can range from 2 to a few hundred depending on the application. Rattlesnake is currently not designed for energy groups notably larger than 50.

In the neutron transport equation (NTE), the physical meaning of the flux $\Psi_g(\vec{x}, \vec{\Omega}, t) dx d\Omega = v_g N_g(\vec{x}, \vec{\Omega}, t) dx d\Omega$ is the expected number of neutrons in dx about \vec{x} , with directions of motion within $d\Omega$ about the direction $\vec{\Omega}$ at time t in the g -th energy group (N_g) times the averaged neutron speed in the group (v_g). Neutron transport is a special case of radiation transport with the following properties:

1. neutrons of all energies can cause fission of heavy nuclei; fission reactions can promptly emit more than one neutron per event.

2. fission fragments usually have a surplus of neutrons and some of them (delayed neutron precursors (DNP)) emit delayed neutrons when decaying. DNP are lumped based on and their decay constants into several groups from 1 to about 8 for different applications.
3. neutrons usually lose energy in scattering events (slowing-down).
4. neutrons can gain energy from scattering with nuclei in thermal motion. Such "upscattering" events are significant for incident neutron energies up to about $E_c = 10kT$. This cut-off energy E_c is 0.43eV with the temperature T equal to 500K (k being the Boltzmann constant).
5. high-energy neutrons can interact with nuclei in non-fission reactions and cause more than one neutron with lower energies to be emitted; these reactions are referred to as (n, xn) reactions.
6. non-neutron particles could be emitted after neutron absorption, including photons, protons, etc.
7. free neutrons decay with a half life of roughly ten minutes. However, because the averaged neutron life time is much smaller than the neutron mean generation time ($\sim 10^{-6} - 10^{-4}$ sec), this phenomenon can be neglected in most reactor applications.

The first two items of this list describe the extra terms in the NTE. An equation for delayed neutron precursors (DNP) need be taken into account for transient simulations. The third and fourth items lead to the following structure of the scattering matrix: Typically we set another energy boundary above the thermal cut-off E_c around

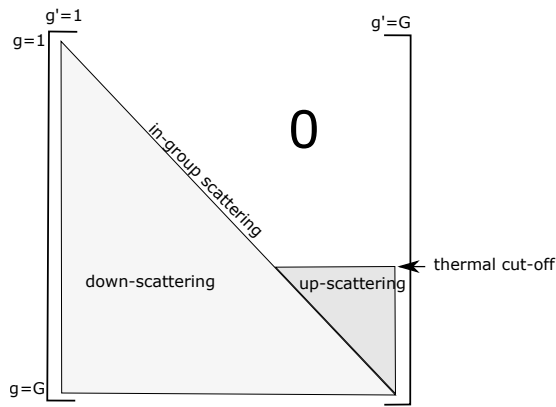


Figure 1 Pattern of neutron scattering matrix.

18eV to ensure that no neutrons are non-physically up-scattered into the resonance energy regions. The physics of neutron slowing down and thermalization in relation with the generation of multigroup cross sections are not considered in this manual. The 5-th item is typically treated with changing the microscopic scattering cross section as

$$\sigma_s = \sum_{x=1}^X x \sigma_{n,xn}, \quad (34)$$

where x is the number of emitted neutrons and X is the maximum number of neutrons that can be emitted. Typically X is set to 3. This change on the scattering cross sections could result into situations where macroscopic scattering cross sections in the high-energy range are larger than the macroscopic total cross sections, i.e. $\Sigma_{s,g} > \Sigma_{t,g}$. Because of this, we cannot evaluate the absorption reaction rates by subtracting the scattering rates from the total rates. Absorption cross sections need to be provided if the corresponding rates are desired. It is noted that this treatment seldom creates numerical issues because the neutron energy quickly drops below the threshold for this multiple-neutron-emission reaction. All the secondary particles other than neutrons are not considered in the NTE and lumped into the absorption event. Cross sections for these reactions are needed for coupled radiation transport simulations such as the neutron-photon coupled simulations, where neutron absorption may induce photon production and vice versa. The assumption of isotropic media is considered sufficiently accurate for neutron transport in nuclear reactor analysis. Neutron angular fluxes are continuous across the medium interface along the streaming direction. This interface condition is implicitly assumed.

Neutron transport is coupled with other physics in reality. Background media temperature can change the neutron cross sections with the Doppler effect. Resonances are broadened and cross sections in thermal range are increased when temperature is increased due to the effect. Neutron cross sections are also proportional to the atomic densities of composing nuclides. Thus thermal expansion of solid or density change in fluid affects the possibility of neutron interacting with materials. Lower cross sections also typically mean more leakage of neutron out of the system. Rattlesnake can set up the transport system on a displaced mesh to bring in the thermal expansion effect of solid. It can obtain the density and temperature variables from other physics to interpolate the pre-generated multigroup cross sections. Neutron fission, scattering, absorption reactions can promptly deposit thermal energies locally or create compound nuclides, which release radiations through decay. We define prompt energy deposition by neutron with a special *energy-deposition cross section* Σ_e in unit of J/cm. It is noted that the kinetic energy of fission fragments in a fission event is considered part of this energy on engineering scale because the flying distance of fragments is in micrometers. Because of this, Σ_e is sometimes referred as $\kappa\Sigma_f$, where κ is energy deposition per fission. Neutron reactions can produce promptly photon as well, which is used as source of the photon transport equation. If photon transport is neglected, we typically assume all energy carried by photon is deposited immediately and locally. We can similarly assume energy carried by electron is deposited immediately and locally. Interaction of neutrons with lattices gives rise to defects/imperfections such as vacancies, self-interstitials, ionization and electron excitation etc. The microscopic defects produced in materials due to irradiation, including neutron, are referred to as radiation damage and are typically measured with also cross sections. Radiation damage is important to the nuclear fuel performance modeling at meso-scale.

2.3.2 Neutron transport equation

In summary, the NTE is obtained by adding the following to the right hand-side

$$\frac{1}{4\pi} (1 - \beta) \chi_{p,g} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} + \frac{1}{4\pi} \sum_{i=1}^I \chi_{d,i,g} \lambda_i C_i, \quad (35)$$

where

- I number of groups of DNP,
- $C_i(\vec{x}, t)$ concentration of delayed neutron precursor [cm^{-3}],
- $\beta_i(\vec{x}, t)$ delayed neutron fraction,
- $\beta(\vec{x}, t) \equiv \sum_{i=1}^I \beta_i$, total delayed neutron fraction,
- λ_i DNP decay constant [s^{-1}],
- $\chi_{p,g}(\vec{x}, t)$ prompt fission spectrum,
- $\chi_{d,i,g}(\vec{x}, t)$ delayed fission spectrum,
- $\Sigma_{f,g}(\vec{x}, t)$ macroscopic fission cross section [cm^{-1}],
- ν averaged neutron emitted per fission,

and the DNP equations read

$$\frac{\partial}{\partial t} C_i = \beta_i \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} - \lambda_i C_i, \quad i = 1, \dots, I. \quad (36)$$

It is noted that we assume that DNPs are deposited locally and never diffuse or are convected away after their generation from fission. Prompt and delayed fission neutrons are both emitted isotropically. We also assume the prompt fission spectrum is independent of the energy of the neutron causing the fission, i.e. it has no dependency on g' . This assumption allows us to define the fission source as

$$F \equiv \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'}. \quad (37)$$

If we define the averaged fission spectrum

$$\chi_g \equiv (1 - \beta) \chi_{p,g} + \sum_{i=1}^I \beta_i \chi_{d,i,g}, \quad (38)$$

we observe that as soon as the DNPs reach the equilibrium state, i.e.

$$C_i = \frac{\beta_i}{\lambda_i} F, \quad i = 1, \dots, I, \quad (39)$$

Eq. (35) can be simplified to $\frac{1}{4\pi} \chi_g F$.

If we define the following vector notation for the DNPs

$$\mathbf{C} \equiv \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_I \end{bmatrix},$$

and the fission operators as

$$\begin{aligned} \mathbb{F}_p \Psi &\equiv \frac{1}{4\pi} \mathbb{F}_{p,0} \Phi \equiv \begin{bmatrix} \frac{1}{4\pi} (1 - \beta) \chi_{p,1} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \frac{1}{4\pi} (1 - \beta) \chi_{p,2} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \frac{1}{4\pi} (1 - \beta) \chi_{p,G} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}, \\ \mathbb{F}_d \Psi &\equiv \frac{1}{4\pi} \mathbb{F}_{d,0} \Phi \equiv \begin{bmatrix} \frac{1}{4\pi} \sum_{i=1}^I \beta_i \chi_{d,i,1} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \frac{1}{4\pi} \sum_{i=1}^I \beta_i \chi_{d,i,2} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \frac{1}{4\pi} \sum_{i=1}^I \beta_i \chi_{d,i,G} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}, \\ \mathbb{F} \Psi &\equiv \frac{1}{4\pi} \mathbb{F}_0 \Phi \equiv \begin{bmatrix} \frac{1}{4\pi} \chi_1 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \frac{1}{4\pi} \chi_2 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \frac{1}{4\pi} \chi_G \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}, \\ \mathbb{C} \mathbf{C} &\equiv \frac{1}{4\pi} \mathbb{C}_0 \mathbf{C} \equiv \begin{bmatrix} \frac{1}{4\pi} \sum_{i=1}^I \chi_{d,i,1} \lambda_i C_i \\ \frac{1}{4\pi} \sum_{i=1}^I \chi_{d,i,2} \lambda_i C_i \\ \vdots \\ \frac{1}{4\pi} \sum_{i=1}^I \chi_{d,i,G} \lambda_i C_i \end{bmatrix}, \\ \mathbb{P} \Phi &\equiv \begin{bmatrix} \beta_1 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \beta_2 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \beta_I \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}, \\ \lambda &\equiv \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_I \end{bmatrix}, \end{aligned}$$

the NTE reads

$$\frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) + \mathbb{L} \Psi = \mathbf{Q}^{\text{ext}} + \mathbb{S} \Psi + \mathbb{F}_p \Psi + \mathbb{C} \mathbf{C}, \quad (40a)$$

$$\frac{\partial}{\partial t} \mathbf{C} = \mathbb{P} \Phi - \lambda \mathbf{C}, \quad (40b)$$

with the same boundary and initial conditions for Ψ as for the CTE, along with the initial condition for the DNPs

$$\mathbf{C}|_{t=t_0} = \mathbf{C}_0, \quad (40c)$$

where \mathbf{C}_0 are the initial DNP concentrations. It is noted that $\mathbb{F} \Psi = \mathbb{F}_d \Psi + \mathbb{F}_p \Psi$. We also define few DNP related operators:

$$\begin{aligned} \mathbb{F}_{d,i} \Psi &\equiv \frac{1}{4\pi} \mathbb{F}_{d,i,0} \Phi \equiv \begin{bmatrix} \frac{1}{4\pi} \beta_i \chi_{d,i,1} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \frac{1}{4\pi} \beta_i \chi_{d,i,2} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \frac{1}{4\pi} \beta_i \chi_{d,i,G} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}, \\ \mathbb{C}_i C_i &\equiv \frac{1}{4\pi} C_{i,0} C_i \equiv \frac{1}{4\pi} \begin{bmatrix} \chi_{d,i,1} \lambda_i C_i \\ \chi_{d,i,2} \lambda_i C_i \\ \vdots \\ \chi_{d,i,G} \lambda_i C_i \end{bmatrix}, \\ \mathbb{P}_i \Phi &\equiv \beta_i F. \end{aligned}$$

It can be seen that

$$\mathbb{F}_d \Psi = \sum_{i=1}^I \mathbb{F}_{d,i} \Psi, \quad (41)$$

$$\mathbb{C} \mathbf{C} = \sum_{i=1}^I \mathbb{C}_i C_i, \quad (42)$$

$$\mathbb{P} \Phi = \begin{bmatrix} \mathbb{P}_1 \Phi \\ \mathbb{P}_1 \Phi \\ \vdots \\ \mathbb{P}_I \Phi \end{bmatrix}. \quad (43)$$

The steady-state NTE is

$$\mathbb{L} \Psi = \mathbf{Q}^{\text{ext}} + \mathbb{S} \Psi + \mathbb{F} \Psi, \quad (44)$$

with the appropriate boundary conditions. It is often important to obtain the neutron multiplication factor by solving the following generalized eigenvalue problem

$$\mathbb{L} \Psi = \mathbb{S} \Psi + \frac{1}{k} \mathbb{F} \Psi, \quad (45)$$

with the homogeneous boundary condition, i.e. the surface source $\Psi_b^{\text{inc}} = 0$. The multiplication factor corresponds to the largest eigenvalue k satisfying Eq. (45). It is also often called k -effective. This largest eigenvalue and its corresponding eigenvector are often referred to as the fundamental mode. Higher eigenmodes could be of interest for certain applications like flux reconstruction due to small perturbations, etc. Eq. (45) is known as the k -eigenvalue problem, as opposed to the α -eigenvalue problem, which is

$$\mathbb{L} \Psi = \mathbb{S} \Psi + \mathbb{F} \Psi - \alpha \frac{\Psi}{v}. \quad (46)$$

In addition, eigenvalue problems are steady-state problems. The k -eigenvalue problem is usually used to provide the initial condition for fast transient problems (time scale of seconds to minutes) and a factor of $\frac{1}{k}$ is applied to the fission cross section to make sure the initial condition is self-sustained without changing the problem configuration during the transient. A sequence of k -eigenvalue problems is often solved in lieu of transients for slow transients (time scale of days to weeks) for computing the change of the isotope inventory. This specific, slow transient is referred to as depletion in nuclear reactor analysis and the solution method is referred to as quasi-static method.

If we define the following adjoint operators,

$$\begin{aligned}
 \mathbb{F}_p^* \Psi^* &\equiv \frac{1}{4\pi} \mathbb{F}_{p,0}^* \Phi^* \equiv \begin{bmatrix} \frac{1}{4\pi} (1-\beta) \nu \Sigma_{f,1} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ \frac{1}{4\pi} (1-\beta) \nu \Sigma_{f,2} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ \vdots \\ \frac{1}{4\pi} (1-\beta) \nu \Sigma_{f,G} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \end{bmatrix}, \\
 \mathbb{F}_d^* \Psi^* &\equiv \frac{1}{4\pi} \mathbb{F}_{d,0}^* \Phi^* \equiv \begin{bmatrix} \frac{1}{4\pi} \nu \Sigma_{f,1} \sum_{g'=1}^G \sum_{i=1}^I \beta_i \chi_{d,i,g'} \Phi_{g'}^* \\ \frac{1}{4\pi} \nu \Sigma_{f,2} \sum_{g'=1}^G \sum_{i=1}^I \beta_i \chi_{d,i,g'} \Phi_{g'}^* \\ \vdots \\ \frac{1}{4\pi} \nu \Sigma_{f,G} \sum_{g'=1}^G \sum_{i=1}^I \beta_i \chi_{d,i,g'} \Phi_{g'}^* \end{bmatrix}, \\
 \mathbb{F}^* \Psi^* &\equiv \frac{1}{4\pi} \mathbb{F}_0^* \Phi^* \equiv \begin{bmatrix} \frac{1}{4\pi} \nu \Sigma_{f,1} \sum_{g'=1}^G \chi_{g'} \Phi_{g'}^* \\ \frac{1}{4\pi} \nu \Sigma_{f,2} \sum_{g'=1}^G \chi_{g'} \Phi_{g'}^* \\ \vdots \\ \frac{1}{4\pi} \nu \Sigma_{f,G} \sum_{g'=1}^G \chi_{g'} \Phi_{g'}^* \end{bmatrix}, \\
 \mathbb{P}^* \mathbf{C}^* &\equiv \frac{1}{4\pi} \mathbb{P}_0^* \mathbf{C}^* \equiv \begin{bmatrix} \frac{1}{4\pi} \nu \Sigma_{f,1} \sum_{i=1}^I \beta_i C_i^* \\ \frac{1}{4\pi} \nu \Sigma_{f,2} \sum_{i=1}^I \beta_i C_i^* \\ \vdots \\ \frac{1}{4\pi} \nu \Sigma_{f,G} \sum_{i=1}^I \beta_i C_i^* \end{bmatrix}, \\
 \mathbb{C}_0^* \Phi^* &\equiv \begin{bmatrix} \lambda_1 \sum_{g'=1}^G \chi_{d,1,g'} \Phi_{g'}^* \\ \lambda_2 \sum_{g'=1}^G \chi_{d,2,g'} \Phi_{g'}^* \\ \vdots \\ \lambda_I \sum_{g'=1}^G \chi_{d,I,g'} \Phi_{g'}^* \end{bmatrix},
 \end{aligned}$$

the adjoint NTE is

$$-\frac{1}{v} \frac{\partial}{\partial t} \Psi^* + \mathbb{L}^* \Psi^* = \mathbf{Q}^{*ext} + \mathbb{S}^* \Psi^* + \mathbb{F}_p^* \Psi^* + \mathbb{P}^* \mathbf{C}^*, \quad (47a)$$

$$-\frac{\partial}{\partial t} \mathbf{C}^* = \mathbb{C}_0^* \Phi^* - \lambda \mathbf{C}^*, \quad (47b)$$

with the same boundary condition and the final condition as for the adjoint CTE and the final condition for the adjoint DNPs

$$\mathbf{C}^*|_{t=t_1} = \mathbf{C}_1^*, \quad (47c)$$

where \mathbf{C}_1^* are the final adjoint DNP concentrations. From the above equations, we can define the adjoint fission source as

$$F_p^* \equiv \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^*, \quad (48a)$$

$$F^* \equiv \sum_{g'=1}^G \chi_{g'} \Phi_{g'}^*, \quad (48b)$$

$$F_{d,i}^* \equiv \sum_{g'=1}^G \chi_{d,i,g'} \Phi_{g'}^*. \quad (48c)$$

We can easily see that the equilibrium adjoint DNP C_i^* is equal to the delayed adjoint fission source $F_{d,i}^*$. The steady-state adjoint NTE is

$$\mathbb{L}^* \Psi^* = \mathbf{Q}^{*ext} + \mathbb{S}^* \Psi^* + \mathbb{F}^* \Psi^*. \quad (49a)$$

and the adjoint k -eigenvalue NTE is

$$\mathbb{L}^* \Psi^* = \mathbb{S}^* \Psi^* + \frac{1}{k} \mathbb{F}^* \Psi^*. \quad (50)$$

We can prove that the adjoint k -effective is the same as the primal one. The adjoint α -eigenvalue NTE is

$$\mathbb{L}^* \Psi^* = \mathbb{S}^* \Psi^* + \mathbb{F}^* \Psi^* - \alpha \frac{\Psi^*}{v}. \quad (51)$$

Similarly, the neutron diffusion equation (NDE) is

$$\frac{\partial}{\partial t} \left(\frac{\Phi}{v} \right) - \vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi + \mathbb{L}_3 \Phi = \mathbb{Q}_0^{\text{ext}} + \mathbb{S}_d \Phi + \mathbb{F}_{p,0} \Phi + \mathbb{C}_0 \mathbf{C}, \quad (52a)$$

$$\frac{\partial}{\partial t} \mathbf{C} = \mathbb{P} \Phi - \lambda \mathbf{C}, \quad (52b)$$

with the same boundary condition and the initial condition as those for the CDE and the same NTE initial condition for the DNPs. The steady-state NDE is

$$-\vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi + \mathbb{L}_3 \Phi = \mathbb{Q}_0^{\text{ext}} + \mathbb{S}_d \Phi + \mathbb{F}_0 \Phi \quad (53)$$

with the corresponding boundary conditions. The k -eigenvalue problem is

$$-\vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi + \mathbb{L}_3 \Phi = \mathbb{S}_d \Phi + \frac{1}{k} \mathbb{F}_0 \Phi \quad (54)$$

with homogeneous boundary conditions. The α -eigenvalue problem is

$$-\vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi + \mathbb{L}_3 \Phi = \mathbb{S}_d \Phi + \mathbb{F}_0 \Phi - \alpha \frac{\Phi}{v}, \quad (55)$$

with the same homogeneous boundary condition. The adjoint NDE is

$$-\frac{1}{v} \frac{\partial}{\partial t} \Phi^* - \vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi^* + \mathbb{L}_3 \Phi^* = \mathbb{Q}_0^{\text{ext}*} + \mathbb{S}_d^* \Phi^* + \mathbb{F}_{p,0}^* \Phi^* + \mathbb{P}_0^* \mathbf{C}^*, \quad (56a)$$

$$-\frac{\partial}{\partial t} \mathbf{C}^* = \mathbb{C}^* \Phi^* - \lambda \mathbf{C}^*, \quad (56b)$$

with the same boundary condition and the final condition as those for the adjoint CDE and the same NTE final condition as for the DNPs. The steady-state adjoint NDE is

$$-\vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi^* + \mathbb{L}_3 \Phi^* = \mathbb{Q}_0^{\text{ext}*} + \mathbb{S}_d^* \Phi^* + \mathbb{F}_0^* \Phi^* \quad (57)$$

with the corresponding boundary conditions. The k -eigenvalue problem is

$$-\vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi^* + \mathbb{L}_3 \Phi^* = \mathbb{S}_d^* \Phi^* + \frac{1}{k} \mathbb{F}_0^* \Phi^* \quad (58)$$

with homogeneous boundary conditions. The α -eigenvalue problem is

$$-\vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi^* + \mathbb{L}_3 \Phi^* = \mathbb{S}_d^* \Phi^* + \mathbb{F}_0^* \Phi^* - \alpha \frac{\Phi^*}{v}, \quad (59)$$

with the same homogeneous boundary conditions.

2.4 Thermal radiation

2.4.1 Physics of thermal radiation transport

Thermal radiation is created by the energy released by the oscillations or transitions of the many electrons contained in matter. All forms of matter emit thermal radiation. Energy of thermal radiation particles – or photons – is

measured with the standard wave properties of *frequency*, ν , and/or *wavelength*, λ . These two properties are related by

$$\lambda = \frac{c}{\nu}, \quad (60)$$

where c is the speed of light in a medium. It is noted that the notations λ for wavelength and ν for frequency are widely accepted in literature and their difference with decay constant and averaged neutron emission per fission is obvious in context. So we do not introduce a new notation to distinguish these quantities here. After introducing the multigroup approximation, these quantities will not appear in the equations.

For thermal radiation, only the portion of the electromagnetic wave spectrum is usually of interest, which extends from approximately 0.1 to $1000 \mu\text{m}$; this range encompasses a portion of the ultraviolet and all of the visible and infrared (IR) light. It is noted that these limits are to some degree arbitrary. The range considered thermal radiation is roughly related to the frequencies a *black body* (refer to Appendix C) is likely to emit at temperatures of interest. For example, barely more than 1 percent of the total radiation is emitted above $1000 \mu\text{m}$ by a black body with a temperature below 9K . Similarly, a black body only releases a few percent of its energy in the range $\lambda < 0.1 \mu\text{m}$ if its temperature lies around $15,000\text{K}$.

Thermal radiation transport is only one type of a more general phenomenon called *photon radiation transport*, which include the study of:

- high-energy *photons* (x-rays and γ -rays) with energies of starting with roughly 100 eV . The main interaction mechanisms are Compton Scattering, photoelectric absorption, and pair production (pair production has an energy threshold of 1022 keV). No black-body radiation is present because typically all bodies have temperatures far too cold to contribute to the considered energy range. A body would have to be at $T = 116,750 \text{ K}$ to emit more than 1% of its energy above 100 eV . The dominant energy redistribution mechanism is Compton scattering.
- *thermal photons* this is the intermediate energy range, where energies are comparable to electron shell energy levels. Typically, the energies are high enough to get some Compton scattering acting as competing energy redistribution in addition to black body re-emission. In addition to Compton scattering, Rayleigh scattering, photo-electric absorption and resonance scattering (resonance fluorescence) are important. Thermal photons originate from very hot black body emitters $T > 5000 \text{ K}$. The energy redistribution mechanisms are absorption/emission and Compton scattering.
- thermal radiation is the radiation emitted by black body emitters up to moderate temperatures ($< 5000 \text{ K}$). At these temperatures the energy is sufficiently low [at 1000 K 99.7 % of photons have energies $< 1 \text{ eV}$, at 5000 K 99.7 % of photons have energies $< 5 \text{ eV}$] so that Compton scattering is negligible¹ and even electron shell excitation levels are sufficiently larger in energy. In this case, the redistribution of energies in energy occurs only through black body emission. Note, though that in general the spectrum at some point can be a superposition of several Planckians which make the local spectrum non-Planckian. The photon energies are comparable to molecular torsion and vibrational modes that humans would sense as heat. Energy redistribution occurs through absorption/emission. Most of the photons have energies below the energy threshold of photoelectric absorption which is of the order of electronic shell energies preventing predominant absorption of certain wave-lengths.

The manner in which photons interact with the background medium differs significantly for these three categories and hence they are usually considered separately as different physics. Rigorously, radiation exerts a force on the background material while it propagates through and accelerates the material in the direction of the radiation flow. In thermal radiation transport though, this effect is typically too small to be considered. Therefore, we couple the radiation equation with a stationary medium, i.e. a motionless material. The fully coupled problem with radiation transport and material motion is referred to as *radiation-hydrodynamics* [9, 10, 11] and will not be discussed here.

Typical applications well suited for simulation using radiative heat transfer include the following:

¹It should be noted that coherent Thomson scattering is the low energy level of Compton scattering and hence it is technically incorrect to talk about Compton scattering not occurring. More correctly, one would state that in the low energy limit Compton scattering becomes coherent and does not act as energy re-distribution mechanism [11].

- Radiative heat transfer from flames;
- Surface-to-surface radiant heating or cooling;
- Coupled radiation, convection, and/or conduction heat transfer;
- Radiation through windows in HVAC (Heating, Ventilation and Air Conditioning) applications, and cabin heat transfer analysis in automotive applications;
- Radiation in glass processing, glass fiber drawing, and ceramic processing.

Physics of radiative heat transfer should be considered when the radiant heat flux is large compared to the heat transfer rate due to convection or conduction. Typically this will occur at high temperatures where the fourth-order dependence of the radiative heat flux on temperature implies that radiative effects are dominant.

The angular flux used for describing the distribution of thermal radiation in the phase space has a particular name, the *specific intensity*, and is denoted by $I(\vec{x}, \vec{\Omega}, \nu, t)$ in the literature. Here, we will continue using Ψ for consistency. The physical meaning of Ψ can be understood by considering a small element of area in phase space $\vec{\Omega} \cdot \vec{n} d\nu d\Omega ds$; then $\Psi(\vec{x}, \vec{\Omega}, \nu, t) \vec{\Omega} \cdot \vec{n} d\nu d\Omega ds$ is the rate at which radiation energy with frequencies of $(\nu, \nu + d\nu)$, direction of motions in $d\Omega$ about $\vec{\Omega}$ is transported across an element of area ds in a time interval dt , where \vec{n} is the normal to the surface. Throughout our study, we use the centimeter-gram-second (cgs) units by default, though more unit systems are available in Rattlesnake. While the fundamental quantity for neutron transport is the neutron density, the energy density is more valued in radiation transport because it is conserved, unlike the number of photons. In the thermal radiation transport community, $\frac{\Phi}{c}$ is called *monochromatic energy density*, the current \vec{j} is called *flux*, the total macroscopic cross section $\Sigma_{t,g}$ is referred to as the *opacity*. Although absorption can be – in all generality – be angularly dependent, it is usually isotropic in a stationary medium, i.e. independent of the angle of the incident radiation. This approximation is known as *diffuse absorption*. In thermal radiation transport, isotropic scattering is often assumed due to the dominated by coherent scattering denoting scattering events that do not change the frequency of the photon. Scattering theory for low energy photons is separated into Rayleigh scattering and Mie scattering. Rayleigh scattering describes cases, where the particle the photon scatters with is much smaller than the wavelength of the photon, while Mie scattering describes cases where the particle can be of the same size or larger compared with the photon wavelength. Rayleigh scattering is almost isotropic in the laboratory frame, while Mie scattering can be significantly anisotropic. In radiative heat transfer, Rayleigh scattering from molecules and atoms is usually more important while Mie scattering applies to cases where scattering off of dust or water droplets is considered. Within the energy range of thermal radiative transfer, scattering can often be neglected because Rayleigh scattering cross sections has a $1/\lambda^4$ dependence and hence occurs predominantly at small wavelengths and is practically insignificant for $\lambda > 1\mu\text{m}$. It is noted that all frequencies are coupled indirectly through black body emission but not by down-scattering as in the NTE.

In radiative heat transfer, no distinction is made between absorption and emissivity (ϵ) when *LTE (local thermodynamic equilibrium)* is assumed. This is stated in *Kirchhoff's law*: for a body of any arbitrary material, emitting and absorbing thermal electromagnetic radiation at every wavelength in *thermodynamic equilibrium*, the ratio of its *emissive power* to its *dimensionless coefficient of absorption* is equal to a universal function depending only on the radiative wavelength and temperature, the perfect black body emissive power.

In the general case, the emission characteristic of materials depend on the population of energy states in the material. By the process of induced emission, the population of these states depends on the radiation field Ψ itself. LTE assumes that the population is dominated by atomic collisions and hence is independent of the radiation field. This allows the emission characteristics to be described by a few thermodynamic variables such as temperature and density which in LTE govern the . Under LTE, absorptivity and emissivity are closely related [11]. Essentially, LTE establishes that $\epsilon_g = \Sigma_{a,g}$. and in a lot of cases, this assumption is valid. With the multigroup approximation however, because the weighting of total interaction and emission terms are different, the emissivity is no longer equal to the absorption. Hence, we keep the emissivity for the black body emission term.

In most solids and liquids, radiation emitted from interior molecules is strongly absorbed by adjoining molecules and are within a distance of approximately $1\mu\text{m}$ from the exposed surface. For this reason, the solid or liquid body is removed from the solution domain, and instead, we impose proper *boundary conditions* on their surfaces and view their emissions as surface phenomena. We call this *opaque treatment*. The surface emission has energy and directional dependencies. The spectral distribution varies with the nature and temperature of the emitting surface. The surface

emissivity $\epsilon_s(\lambda, \vec{\Omega}, T)$ is defined as the ratio of the radiation emitted by the surface to the radiation emitted by a black body at the same temperature. Typically isotropic emission is a good approximation for most surfaces and the normal emissivity $\epsilon_n(\lambda, T) = \epsilon_s(\lambda, \vec{\Omega} \cdot \vec{n} = 0, T)$ is a good measure of the emissivity. The manner in which ϵ_n varies with λ depends on whether the solid is a conductor or nonconductor, as well as the nature of the surface coating. The spectral distribution of $\epsilon_n(\lambda, T)$ is approximately independent of temperature. More comprehensive emissivity compilations can be found in [38, 39, 40, 41]. Measurements of surface emissivity are often limited to determining its normal to the surface, for a few chosen wavelengths or only the hemispherical total emissivity. Its strong dependence on the surface condition is an additional difficulty. Impurities also play a role, alongside roughness. In view of the experimental difficulties a theory for radiation properties is desirable. Unfortunately the electromagnetic theory is only valid under a series of limiting suppositions, so that the emissivities calculated often differ from reality. Typically, it depends on the temperature, the polar angle, the wavelength and some other properties like the electrical resistivity. The dependence of the absorptivity on the surface temperature is generally small. If the surface is opaque, absorptivity and reflectivity sum to one. The assumption of diffuse reflection (white boundary condition) is reasonable for most engineering applications.

In technical applications, gas radiation in the infrared – i.e. at wavelengths above $1\mu m$ – is of interest. The main emitters in that portion of the spectrum are CO_2 and H_2O , although other gases such as CO , SO_2 , NH_3 , CH_4 and other hydrocarbons also emit therein. In contrast, N_2 and O_2 , the main constituents of air allow radiation in the infrared region to pass through with virtually no attenuation; they do not absorb, and therefore – according to Kirchhoff's law – do not emit either. Gases only absorb and emit radiation in narrow wavelength regions, called bands. Gases cannot be idealized as gray radiators [a gray radiator is defined as a body emitting less energy than a black body but featuring a black body spectrum] without loss of accuracy because the available energy state are not sufficiently dense to justify a black body emission spectrum. We will typically assume that radiation is absorbed but not scattered. This applies to the infrared wavelengths because Rayleigh scattering by molecules only takes place at very small wavelengths and is practically insignificant for $\lambda > 1\mu m$.

Radiation speed could change either abruptly at interfaces or smoothly within a medium. The former requires a general interface condition Eq. (608a) in Appendix H. When the radiation speeds in the two interfacing media are the same, the condition reduces to the normal continuity condition as in neutron transport.

The latter introduces one more term to account for the effect from the varying refractive index.

$$\left(\frac{1}{n} \vec{\nabla} n \cdot \vec{\nabla}_S - \frac{2}{n} \vec{\Omega} \cdot \vec{\nabla} n \right) \Psi, \quad (61)$$

where $\vec{\nabla}_S$ is the spherical gradient operator on the 2D unit sphere (see Appendix B). This term is typically treated with the spherical harmonics expansion.

2.4.2 Thermal radiation transport equation (TRTE)

The TRTE requires an extra term, the black body emission:

$$\frac{1}{4\pi} S_{BB,g}(T) \equiv \frac{1}{4\pi} \epsilon_g (F(n\lambda_g T) - F(n\lambda_{g+1} T)) c_0 a n^2 T^4, \quad (62)$$

where

- $\epsilon_g(\vec{x}, t)$ emissivity [cm^{-1}],
- $n(\vec{x}) \equiv \frac{c}{c_0}$ refractive index of medium,
- $T(\vec{x}, t)$ medium temperature [K],
- F fraction of black body emission, function of $n\lambda T$,
- c_0 speed of light in vacuum $2.99792458 \times 10^{10} [cm\ s^{-1}]$,
- λ_g upper boundary of the wavelength of the frequency band [cm],
- λ_{g+1} lower boundary of the wavelength of the frequency band [cm],
- $a \equiv 8\pi^5 k^4 / (15h^3 c_0^3)$ radiation constant $7.5646 \times 10^{-15} [erg\ cm^{-3}\ K^{-4}]$,

$$\begin{aligned}
k & \text{ Boltzmann constant } 1.38064852(79) \times 10^{-16} [\text{erg K}^{-1}], \\
\hbar & \text{ Planck constant } 6.6260755(40) \times 10^{-27} [\text{erg s}].
\end{aligned}$$

The black body emission operator is nonlinear with respect to the medium's temperature. More details about the black body emission can be found in Appendix G. The TRTE is often solved along with the material temperature equation

$$\rho C_V \frac{\partial T}{\partial t} = \vec{\nabla} \cdot (\mathbf{D}_e \vec{\nabla} T) - \sum_{g=1}^G (S_{\text{BB},g} - \Sigma_{a,g} \Phi_g) + Q \quad (63a)$$

where

$$\begin{aligned}
\rho(\vec{x}, t) & \text{ density } [\text{g cm}^{-3}], \\
C_V(\vec{x}, t) & \text{ specific heat } [\text{erg g}^{-1} \text{K}^{-1}], \\
\mathbf{D}_e(\vec{x}, t) & \text{ thermal conductivity } [\text{erg s}^{-1} \text{cm}^{-1} \text{K}^{-1}], \\
Q(\vec{x}, t) & \text{ volumetric heat source } [\text{erg s}^{-1} \text{cm}^{-3}], \\
\Phi_g(\vec{x}, t) & \text{ scalar intensity } [\text{erg cm}^{-2} \text{s}^{-1}], \\
\Sigma_{a,g}(\vec{x}, t) & \text{ absorptivity } [\text{cm}^{-1}],
\end{aligned}$$

with the Dirichlet boundary condition

$$T(\vec{x}, t) = T_b(\vec{x}, t), \quad \vec{x} \in \partial \mathcal{D}_1, \quad (63b)$$

the heat flux boundary condition

$$\vec{\nabla} T \cdot \vec{n}_b = h_b, \quad \vec{x} \in \partial \mathcal{D}_2. \quad (63c)$$

and the initial condition

$$T(\vec{x}, t = t_0) = T_0(\vec{x}), \quad \vec{x} \in \mathcal{D}. \quad (63d)$$

If we neglect thermal conduction, no boundary condition is needed for the temperature equation. The second term on the right hand side means that energy transferred from the radiation field must be added to the material. When the material and radiation fields are instantaneously at equilibrium, i.e., $\sum_{g=1}^G (S_{\text{BB},g} - \Sigma_{a,g} \Phi_g) = 0$, the material temperature equation is decoupled from the radiation transport equation. Thermal conduction due to other mechanism is negligibly small. The challenge of thermal radiation transport mainly lies in the equation's nonlinearities. The primal variable of medium temperature is often redefined to make the problem easier to solve. Rattlesnake provides three options of defining the temperature primal variable, T , T^4 or $c_0 a T^4$.

We define

$$\mathbf{S}_{\text{BB}} \equiv \begin{bmatrix} S_{\text{BB},1} \\ S_{\text{BB},2} \\ \vdots \\ S_{\text{BB},G} \end{bmatrix},$$

TRTE becomes

$$\frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) + \mathbb{L} \Psi = \mathbf{Q}^{\text{ext}} + \mathbb{S} \Psi + \frac{1}{4\pi} \mathbf{S}_{\text{BB}}, \quad (64)$$

with the same boundary and initial conditions for Ψ as for the CTE. The thermal radiation diffusion equation (TRDE) is

$$\frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}} \right) - \vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi + \mathbb{L}_3 \Phi = \mathbf{Q}_0^{\text{ext}} + \mathbb{S}_0 \Phi + \mathbf{S}_{\text{BB}}, \quad (65)$$

with the same boundary condition and the initial condition as those for the CDE.

3 Weak Forms of the Multigroup Radiation Transport Equations

This section presents the weak forms for the multigroup radiation transport equations with the spatial and angular discretizations available in Rattlesnake. The weak form without considering the numerical stabilization of the streaming term is presented first. Unstabilized weak form for the adjoint equation will also be given. Various vacuum and reflecting boundary conditions are considered. It is noted that those conditions apply to all later numerical weak forms. Then the form of different formulations including SAAF (self-adjoint angular flux) [42, 43, 44, 45, 46] and LS (least-square) [47, 48, 49] suitable for spatial discretization with CFEM are derived. The similarity and difference between SAAF and LS are briefly discussed. We then present the upwinding scheme with DFEM for the original transport equation (first-order form of the streaming). We present these weak forms for the CTE only. The complete weak forms with angular discretization are finally given along with terms specific for each particle. The extra kernels required by curvilinear coordinates are given. The kernels appearing in the complete weak forms are linked with the code. Weak forms with both CFEM and DFEM for the multigroup diffusion equation are given independently. The conditions for equivalence between the diffusion and PN projections are discussed. In all these weak forms, the time derivative is kept un-discretized. We leave its discretization to the method of lines and will briefly discuss it Section 5. Then the interface conditions for the multi-scheme transport are given. Reduced order method (ROM) derived from the weak forms for the neutron transport equation is discussed in general. Its applications are presented including, dumping the PKE (point kinetics equation) parameters along with a transient solve, evaluation of the multi-region PKE parameters and IQS (improved quasi-static). Solving techniques associated with various weak forms will be deferred to Section 5.

Because we use the method of lines to treat the time discretization, we drop the time integration and dependency in the definition of the inner product. If the functions have time dependency, the inner product will be evaluated at a particular time. We define our inner product as

$$(\mathbf{a}, \mathbf{b})_{\mathcal{D} \otimes \mathcal{S}} \equiv \sum_{g=1}^G \int_{\mathcal{S}} d\Omega \int_{\mathcal{D}} dx a_g(\vec{x}, \vec{\Omega}) b_g(\vec{x}, \vec{\Omega}), \quad (66a)$$

where \mathbf{a} and \mathbf{b} are generic multigroup functions defined on $\mathcal{D} \otimes \mathcal{S}$. We drop the subscript $\mathcal{D} \otimes \mathcal{S}$ for notation simplicity. If the integration domain is different from this default domain, it needs to be included as the subscript, such as $(\mathbf{a}, \mathbf{b})_{\mathcal{D} \otimes \mathcal{S}_n^+}$. We also find the definitions of the following boundary integrals useful:

$$\langle \mathbf{a}, \mathbf{b} \rangle^{\pm} \equiv \sum_{g=1}^G \oint_{\partial \mathcal{D}} dx \int_{\mathcal{S}_{\vec{n}_b}^{\pm}} d\Omega \left| \vec{\Omega} \cdot \vec{n}_b \right| a_g(\vec{x}, \vec{\Omega}) b_g(\vec{x}, \vec{\Omega}), \quad (66b)$$

$$\langle \mathbf{a}, \mathbf{b} \rangle \equiv \sum_{g=1}^G \oint_{\partial \mathcal{D}} dx \int_{\mathcal{S}} d\Omega \left| \vec{\Omega} \cdot \vec{n}_b \right| a_g(\vec{x}, \vec{\Omega}) b_g(\vec{x}, \vec{\Omega}), \quad (66c)$$

where \mathbf{a} and \mathbf{b} are generic multigroup functions defined on $\partial \mathcal{D} \otimes \mathcal{S}$. If the integration is on a general surface Γ other than $\partial \mathcal{D}$, it needs to be included in the subscript as $\langle \mathbf{a}, \mathbf{b} \rangle_{\Gamma}$. We also define the inner product of two generic functions over subdomain \mathcal{D} as

$$(\mathbf{a}, \mathbf{b})_{\mathcal{D}} \equiv \sum_{g=1}^G \int_{\mathcal{D}} dx a_g(\vec{x}) b_g(\vec{x}). \quad (66d)$$

We use the subscript \mathcal{D} to explicitly indicate this inner product only involves a spatial integration. This inner product is evaluated at a particular direction and a particular time if the two functions have such dependencies. We also define the inner product of two generic functions over a surface such as $\partial \mathcal{D}$ as

$$(\mathbf{a}, \mathbf{b})_{\partial \mathcal{D}} \equiv \sum_{g=1}^G \oint_{\partial \mathcal{D}} dx a_g(\vec{x}) b_g(\vec{x}). \quad (66e)$$

3.1 The unstabilized form

We multiply the test function Ψ^* with Eq. (3) and integrate over the phase space

$$\left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) + (\Psi^*, \mathbb{L} \Psi) = (\Psi^*, \mathbf{Q}^{\text{ext}}) + (\Psi^*, \mathbb{S} \Psi),$$

and apply the property with integration by parts,

$$(\mathbf{a}, \mathbb{L}\mathbf{b}) + \langle \mathbf{a}, \mathbf{b} \rangle^- = (\mathbb{L}^*\mathbf{a}, \mathbf{b}) + \langle \mathbf{a}, \mathbf{b} \rangle^+, \quad (67)$$

on the streaming term,

$$\left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) + (\mathbb{L}^*\Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+ - \boxed{\langle \Psi^*, \Psi \rangle^-} = (\Psi^*, \mathbf{Q}^{\text{ext}}) + (\Psi^*, \mathbb{S}\Psi),$$

then we substitute the boundary condition in the boxed term and obtain

$$\left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) + (\mathbb{L}^*\Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+ - \langle \Psi^*, \Psi^{\text{inc}} + \mathbb{B}\Psi \rangle^- = (\Psi^*, \mathbf{Q}^{\text{ext}}) + (\Psi^*, \mathbb{S}\Psi).$$

We denote the test function as Ψ^* , the same notation of adjoint solution. It will be obvious why we use this notation after deriving the adjoint weak form. The weak form is:

Find a solution Ψ in the function space V , such that

$$\mathbb{b}(\Psi^*, \Psi) = \mathbb{l}(\Psi^*), \quad \forall \Psi^* \in V, \quad (68a)$$

where

$$\mathbb{b}(\Psi^*, \Psi) \equiv \left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) + (\mathbb{L}^*\Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+ - \langle \Psi^*, \mathbb{B}\Psi \rangle^- - (\Psi^*, \mathbb{S}\Psi), \quad (68b)$$

$$\mathbb{l}(\Psi^*) \equiv \langle \Psi^*, \Psi^{\text{inc}} \rangle^- + (\Psi^*, \mathbf{Q}^{\text{ext}}). \quad (68c)$$

The function space V include all possible analytical transport solution functions. Its precise mathematical definition is not the interest of this manual. The function space V where a solution is sought and the function space for the test functions are identical (i.e. we use Galerkin FEM). Because the test function is arbitrary in the space, the solution satisfying Eq. (68b) is uniquely determined. Each individual term of the bilinear and linear form is called a *kernel* with the MOOSE terminology.

We can do the same procedure for the adjoint equation and end up with:

Find a solution $\Psi^* \in V$, such that

$$\mathbb{b}^*(\Psi^*, \Psi) = \mathbb{l}^*(\Psi), \quad \forall \Psi \in V, \quad (69a)$$

where

$$\mathbb{b}^*(\Psi^*, \Psi) \equiv - \left(\frac{1}{\mathbf{v}} \frac{\partial}{\partial t} \Psi^*, \Psi \right) + (\Psi^*, \mathbb{L}\Psi) + \langle \Psi^*, \Psi \rangle^- - \langle \mathbb{B}^*\Psi^*, \Psi \rangle^+ - (\mathbb{S}^*\Psi^*, \Psi), \quad (69b)$$

$$\mathbb{l}^*(\Psi) \equiv \langle \Psi^{\text{out}}, \Psi \rangle^+ + (\mathbf{Q}^{\text{ext}}, \Psi). \quad (69c)$$

For steady state problems, the transient kernels $\left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right)$ and $\left(\frac{1}{\mathbf{v}} \frac{\partial}{\partial t} \Psi^*, \Psi \right)$ can be dropped. It can be seen that with zero initial and final conditions or steady-state, $\mathbb{b}(\Psi^*, \Psi) = \mathbb{b}^*(\Psi^*, \Psi)$. The same bilinear form for the primal equation can be applied to the adjoint equation.

In curvilinear coordinates, the transport equation has an extra term due to the streaming term $\mathbb{E}\Psi$,

$$\mathbb{E}\Psi \equiv \begin{bmatrix} E\Psi_1 \\ E\Psi_2 \\ \vdots \\ E\Psi_G \end{bmatrix},$$

where the definition of operator E can be found in Appendix C and the \mathbb{L} operator is given by Eq. (2.1) that is valid in both Cartesian and curvilinear coordinate systems. It should be noted that we use the It is noted that the the adjoint operator \mathbb{L}^* is still accurate. It will be useful to define another extra streaming operator $\mathbb{E}_2\Psi$ here:

$$\mathbb{E}_2\Psi \equiv \begin{bmatrix} E_2\Psi_1 \\ E_2\Psi_2 \\ \vdots \\ E_2\Psi_G \end{bmatrix},$$

where the definition of operator E_2 can also be found in Appendix C. Rattlesnake is based on MOOSE, which supports 1D R-spherical and 2D R-Z cylindrical coordinates besides 1D, 2D and 3D Cartesian coordinates. The definition of this extra term in R-spherical is

$$\mathbb{E}\Psi \equiv \begin{bmatrix} \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi_1] \\ \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi_2] \\ \vdots \\ \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi_G] \end{bmatrix}.$$

It is in R-Z coordinates

$$\mathbb{E}\Psi \equiv \begin{bmatrix} -\frac{1}{r} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi_1) \\ -\frac{1}{r} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi_2) \\ \vdots \\ -\frac{1}{r} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi_G) \end{bmatrix}.$$

When using either of these curvilinear coordinates, an extra term need be added to Eq. (68b):

$$(\Psi^*, \mathbb{E}\Psi). \quad (70)$$

3.2 Boundary conditions

In the previous section, the boundary conditions are applied on the boxed term $\langle \Psi^*, \Psi \rangle^-$. However, there are other ways to apply them [50]. To derive them, we first split the boundary terms into four parts:

$$\begin{aligned} \langle \Psi^*, \Psi \rangle^+ - \langle \Psi^*, \Psi \rangle^- &= \int_S (\vec{\Omega} \cdot \vec{n}_b \Psi^*, \Psi)_{\partial \mathcal{D}} d\Omega \\ &= \int_S (\vec{\Omega} \cdot \vec{n}_b \Psi_e^*, \Psi_o)_{\partial \mathcal{D}} d\Omega + \int_S (\vec{\Omega} \cdot \vec{n}_b \Psi_o^*, \Psi_e)_{\partial \mathcal{D}} d\Omega \\ &= \langle \Psi_e^*, \Psi_o \rangle^+ - \langle \Psi_e^*, \Psi_o \rangle^- + \langle \Psi_o^*, \Psi_e \rangle^+ - \langle \Psi_o^*, \Psi_e \rangle^-, \end{aligned} \quad (71)$$

with the even parity Ψ_e^*, Ψ_e and odd parity Ψ_o^*, Ψ_o fluxes being defined as

$$\Psi_e^* \equiv \frac{\Psi^*(\vec{\Omega}) + \Psi^*(-\vec{\Omega})}{2}; \Psi_e \equiv \frac{\Psi(\vec{\Omega}) + \Psi(-\vec{\Omega})}{2}; \quad (72a)$$

$$\Psi_o^* \equiv \frac{\Psi^*(\vec{\Omega}) - \Psi^*(-\vec{\Omega})}{2}; \Psi_o \equiv \frac{\Psi(\vec{\Omega}) - \Psi(-\vec{\Omega})}{2}. \quad (72b)$$

With these definitions, we have

$$\Psi_e^*(-\vec{\Omega}) = \Psi_e^*(\vec{\Omega}); \quad \Psi_o^*(-\vec{\Omega}) = -\Psi_o^*(\vec{\Omega}), \quad (72c)$$

$$\Psi_e(-\vec{\Omega}) = \Psi_e(\vec{\Omega}); \quad \Psi_o(-\vec{\Omega}) = -\Psi_o(\vec{\Omega}), \quad (72d)$$

$$\Psi^* = \Psi_e^* + \Psi_o^*; \quad \Psi = \Psi_e + \Psi_o. \quad (72e)$$

We will split the boundary into three *non-overlapping* parts $\partial \mathcal{D} = \partial \mathcal{D}_s \cup \partial \mathcal{D}_r \cup \partial \mathcal{D}_w$ and assume that only one type of boundary condition is applied on each part.

For the external non-homogeneous part Ψ_b^{inc} , we convert the boundary condition into parity form

$$\Psi_e + \Psi_o = \Psi_b^{\text{inc}}, \quad \vec{\Omega} \cdot \vec{n}_b < 0 \text{ and } \vec{x} \in \partial \mathcal{D}_s \quad (73a)$$

$$\Psi_e - \Psi_o = \Psi_b^{\text{inc}}(-\vec{\Omega}), \quad \vec{\Omega} \cdot \vec{n}_b > 0 \text{ and } \vec{x} \in \partial \mathcal{D}_s. \quad (73b)$$

If we substitute Eq. (73a) into the second and the fourth term of Eq. (71), we have

$$\langle \Psi_e^*, \Psi_o \rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi_e^*, \Psi_b^{\text{inc}} - \Psi_e \rangle_{\partial \mathcal{D}_s}^- + \langle \Psi_o^*, \Psi_e \rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi_o^*, \Psi_b^{\text{inc}} - \Psi_o \rangle_{\partial \mathcal{D}_s}^-$$

$$\begin{aligned}
&= \langle \Psi_e^*, \Psi_o \rangle_{\partial \mathcal{D}_s}^+ + \langle \Psi_e^*, \Psi_e \rangle_{\partial \mathcal{D}_s}^- + \langle \Psi_o^*, \Psi_e \rangle_{\partial \mathcal{D}_s}^+ + \langle \Psi_o^*, \Psi_o \rangle_{\partial \mathcal{D}_s}^- - \langle \Psi^*, \Psi^{\text{inc}} \rangle_{\partial \mathcal{D}_s}^- \\
&= \langle \Psi_e^*, \Psi_o \rangle_{\partial \mathcal{D}_s}^+ + \langle \Psi_e^*, \Psi_e \rangle_{\partial \mathcal{D}_s}^+ + \langle \Psi_o^*, \Psi_e \rangle_{\partial \mathcal{D}_s}^+ + \langle \Psi_o^*, \Psi_o \rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi^*, \Psi^{\text{inc}} \rangle_{\partial \mathcal{D}_s}^- \\
&= \langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi^*, \Psi^{\text{inc}} \rangle_{\partial \mathcal{D}_s}^-,
\end{aligned} \tag{74}$$

which is the same as the boundary terms in Section 3.1. Note that we applied the following property in the second to last step of the above derivation

$$\int_{\partial \mathcal{D}} dx \int_{\vec{\Omega} \cdot \vec{n}_b > 0} d\Omega \left| \vec{\Omega} \cdot \vec{n}_b \right| b_e = \int_{\partial \mathcal{D}} dx \int_{\mathcal{H}} d\Omega \left| \vec{\Omega} \cdot \vec{n}_b \right| b_e, \tag{75a}$$

$$\int_{\partial \mathcal{D}} dx \int_{\vec{\Omega} \cdot \vec{n}_b > 0} d\Omega \left| \vec{\Omega} \cdot \vec{n}_b \right| b_o = \int_{\partial \mathcal{D}} dx \int_{\mathcal{H}} d\Omega \vec{\Omega} \cdot \vec{n}_b b_o, \tag{75b}$$

where $\mathcal{H}(\vec{n}_0) = \left\{ \vec{\Omega} \in \mathcal{S}, \vec{\Omega} \cdot \vec{n}_0 > 0 \right\}$ is the half angular space with respect to an arbitrarily oriented normal \vec{n}_0 .

If we substitute Eq. (73a) into the second term and Eq. (73b) into the first term of Eq. (71), we have

$$\begin{aligned}
&\left\langle \Psi_e^*, \Psi_e - \Psi^{\text{inc}}(-\vec{\Omega}) \right\rangle_{\partial \mathcal{D}_s}^+ - \left\langle \Psi_e^*, \Psi^{\text{inc}} - \Psi_e \right\rangle_{\partial \mathcal{D}_s}^- + \langle \Psi_o^*, \Psi_e \rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi_o^*, \Psi_e \rangle_{\partial \mathcal{D}_s}^- \\
&= 2 \langle \Psi_e^*, \Psi_e \rangle_{\partial \mathcal{D}_s}^+ - 2 \left\langle \Psi_e^*, \Psi^{\text{inc}} \right\rangle_{\partial \mathcal{D}_s}^- + 2 \langle \Psi_o^*, \Psi_e \rangle_{\partial \mathcal{D}_s}^+ \\
&= \langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_s}^+ + \left\langle \Psi^*, \Psi(-\vec{\Omega}) - \Psi_b^{\text{inc}}(-\vec{\Omega}) \right\rangle_{\partial \mathcal{D}_s}^+ - \left\langle \Psi^*, \Psi^{\text{inc}} \right\rangle_{\partial \mathcal{D}_s}^-,
\end{aligned} \tag{76}$$

It can be seen that the coupling from odd parity to even parity due to the surface flux is removed. If there is no anisotropic scattering, the even parity components can be solved independently.

If we substitute Eq. (73a) into the last term and Eq. (73b) into the third term of Eq. (71), we have

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_s}^+ - \left\langle \Psi^*, \Psi(-\vec{\Omega}) - \Psi^{\text{inc}}(-\vec{\Omega}) \right\rangle_{\partial \mathcal{D}_s}^+ - \left\langle \Psi^*, \Psi^{\text{inc}} \right\rangle_{\partial \mathcal{D}_s}^-. \tag{77}$$

Another type often referred to as parity-equivalent (PE) is also seen in literature

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_s}^+ + \left\langle \Psi^*, \Psi - \Psi^{\text{inc}} \right\rangle_{\partial \mathcal{D}_s}^- - \left\langle \Psi^*, \Psi^{\text{inc}} \right\rangle_{\partial \mathcal{D}_s}^-.$$

Derivation of the reflecting boundary condition starts from splitting

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \left\langle \Psi^*, \Psi(\vec{\Omega}_r) \right\rangle_{\partial \mathcal{D}_r}^-$$

into

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \left\langle \Psi_e^*, \Psi_e(\vec{\Omega}_r) \right\rangle_{\partial \mathcal{D}_r}^- - \left\langle \Psi_e^*, \Psi_o(\vec{\Omega}_r) \right\rangle_{\partial \mathcal{D}_r}^- - \left\langle \Psi_o^*, \Psi_e(\vec{\Omega}_r) \right\rangle_{\partial \mathcal{D}_r}^- - \left\langle \Psi_o^*, \Psi_o(\vec{\Omega}_r) \right\rangle_{\partial \mathcal{D}_r}^-. \tag{78}$$

If we substitute the condition

$$\Psi_e = \Psi_e(\vec{\Omega}_r), \vec{x} \in \partial \mathcal{D}_r, \tag{79a}$$

$$\Psi_o = \Psi_o(\vec{\Omega}_r), \vec{x} \in \partial \mathcal{D}_r. \tag{79b}$$

into the second and forth terms of Eq. (78), we obtain

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \left\langle \Psi^*, \frac{1}{2}(\Psi + \Psi(\vec{\Omega}_r)) \right\rangle_{\partial \mathcal{D}_r}^- - \left\langle \Psi^*, \frac{1}{2}(\Psi - \Psi(\vec{\Omega}_r)) \right\rangle_{\partial \mathcal{D}_r}^+. \tag{80}$$

The above condition is asymmetric because of the terms involving $\Psi(\vec{\Omega}_r)$. We can obtain a symmetric reflecting boundary condition type by simply change the sign of the last term in Eq. (80):

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \left\langle \Psi^*, \frac{1}{2}(\Psi + \Psi(\vec{\Omega}_r)) \right\rangle_{\partial \mathcal{D}_r}^- + \left\langle \Psi^*, \frac{1}{2}(\Psi - \Psi(\vec{\Omega}_r)) \right\rangle_{\partial \mathcal{D}_r}^+. \quad (81)$$

It is noted that this change does not affect the consistency of the condition because analytically $\Psi = \Psi(\vec{\Omega}_r)$ on $\partial \mathcal{D}_r$. Notation can be simplified if we introduce numerical flux $\bar{\Psi}$ on the boundary and define them properly.

$$\langle \Psi^*, \bar{\Psi} \rangle^+ - \langle \Psi^*, \bar{\Psi} \rangle^- \quad (82)$$

We summarize these four different types for vacuum into Table 1, and three types for reflecting into Table 2.

Table 1 Vacuum boundary condition types.

Name	$\bar{\Psi}$ on $\mathcal{S}_{\vec{n}_b}^+$	$\bar{\Psi}$ on $\mathcal{S}_{\vec{n}_b}^-$	Odd-to-even decoupled
SAAF	Ψ	Ψ^{inc}	No
EVEN	$\Psi + \Psi(-\vec{\Omega}) - \Psi^{\text{inc}}(-\vec{\Omega})$	Ψ^{inc}	Yes
ODD	$\Psi - \Psi(-\vec{\Omega}) + \Psi^{\text{inc}}(-\vec{\Omega})$	Ψ^{inc}	No
PE	Ψ	$\Psi - \Psi^{\text{inc}} - \Psi^{\text{inc}}$	Yes

Table 2 Reflecting boundary condition types.

Name	$\bar{\Psi}$ on $\mathcal{S}_{\vec{n}_b}^+$	$\bar{\Psi}$ on $\mathcal{S}_{\vec{n}_b}^-$	Symmetric
SAAF	Ψ	$\Psi(\vec{\Omega}_r)$	No
NSYM	$\Psi - \frac{1}{2}(\Psi - \Psi(\vec{\Omega}_r))$	$\frac{1}{2}(\Psi + \Psi(\vec{\Omega}_r))$	No
SYM	$\Psi + \frac{1}{2}(\Psi - \Psi(\vec{\Omega}_r))$	$\frac{1}{2}(\Psi + \Psi(\vec{\Omega}_r))$	Yes

No extra types for white boundary is implemented.

3.3 Spatial discretization and stabilization

Because the transport equation is hyperbolic in nature, stabilization must be applied to the spatial discretization. After presenting some fundamental FEM notations, we will discuss three stabilization techniques implemented in Rattlesnake: SAAF, LS and upwinding with DFEM. We use CTE for all of the stabilization techniques because generalization for various particle types is trivial. We give kernels including the kernels for various particles in the fully discretized (both space and angle are discretized) weak forms in Section 3.4 later. Balance equations with the isotropic test function are given. These equations are closely related to transport corrected diffusion equations for nonlinear diffusion accelerations (NDA).

3.3.1 FEM notations

With FEM we perform the spatial discretization by *first* triangulating or meshing the domain \mathcal{D} with non-overlapping elements $\mathcal{D} = \cup_{e \in \mathbb{T}_h} \mathcal{D}_e$, where \mathbb{T}_h is used to denote the triangulation or the mesh and \mathcal{D}_e is the partial domain covered by an element indexed with e . Elements are typically not arbitrarily shaped; rather, they are described with nodes positioned in \mathcal{D} . The number of nodes of an element depends on the element type: for example, 8 for a hexahedron, 4 for a tetrahedron, etc. in 3D or 4 for a quadrilateral, 3 for a triangle, etc. in 2D or 2 for a so-called edge in 1D. These nodes are implicitly ordered to define an element. Elements can be unstructured. They can also be high order isoparametric. For example, a 2D triangle is typically defined with three vertices (linear), but we can use six nodes to define a triangle with curved sides: three on the vertices and three on the sides to have a quadratic element.

Elements have sides. When the mesh is conforming, i.e. no hanging nodes in the mesh, every side can at most be shared by two elements except those on the boundary. We define $\Gamma_{\text{int}} \equiv \cup_{e \in \mathbb{T}_h} (\partial \mathcal{D}_e \setminus \partial \mathcal{D})$, the set of all the interior sides. On every interior side, a unit normal direction \vec{n} can be defined. Weak forms should be independent of the orientation of this interior normal. Elements on the boundary may not exactly preserve the domain boundary $\partial \mathcal{D}$ especially when $\partial \mathcal{D}$ is curved, i.e. $\Gamma \equiv \cup_{e \in \mathbb{T}_h} (\partial \mathcal{D}_e \cap \partial \mathcal{D})$ may not be equal to $\partial \mathcal{D}$. If so, we do have $\mathcal{D} \neq \cup_{e \in \mathbb{T}_h} \mathcal{D}_e$. But this inequality becomes exact as the number of elements on the domain goes to infinity. By convention, the boundary normal is always chosen to point outside of the domain.

We then define the shape functions on local elements. These shape functions are the polynomial functions on the reference elements. The number of shape functions are determined by the order of the local function space. For example, the number of shape functions is $p + 1$ for 1D edges, $p \times (p + 1)/2$ for 2D triangles, $(p + 1)^2$ for 2D quadrilaterals. With discontinuous FEM, these shape functions directly form the basis functions of the discretized scalar function space V_h . With continuous FEM, shape functions on adjacent elements must be constrained to enforce continuity of the solution, consequently basis functions with local support spanning several elements are typically associated with mesh nodes. Continuous and discontinuous linear shape functions for 1D and 2D are illustrated in Fig. 2a and Fig. 2b respectively. Basis functions associated with nodes are continuous and functions associated with elements are discontinuous.

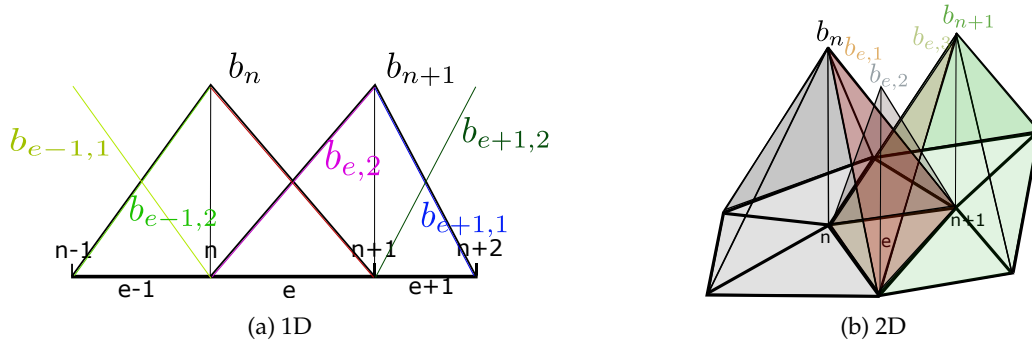


Figure 2 FEM linear basis functions.

Finally, with the definition of basis functions, any spatially varying function can be approximated with

$$f(\vec{x}) = \sum_{i=1}^N f_i b_i(\vec{x}), \quad (83)$$

where N is the dimension of the function space and b_i is the basis function with local support. N is also often referred to as the number of degrees of freedom (DoF). We can also define an index set E_e containing the indices of the basis functions who have support on element e (i.e. that are non-zero therein). The number of indices in E_e depends on the local polynomial order of the element e . Similarly we can define an index set E_s containing the indices of the basis functions that have support on a side of index s . With this spatial discretization, inner products can be explicitly spelled out as

$$(\mathbf{a}, \mathbf{b}) \equiv \sum_{g=1}^G \int_S d\Omega \sum_{e \in \mathbb{T}_h} \int_{\mathcal{D}_e} dx \sum_{i \in E_e} a_{g,i}(\vec{\Omega}) b_i(\vec{x}) \sum_{j \in E_e} b_{g,j}(\vec{\Omega}) b_j(\vec{x}), \quad (84a)$$

$$\langle \mathbf{a}, \mathbf{b} \rangle^\pm \equiv \sum_{g=1}^G \sum_{s \in \partial \mathcal{D}} \int_s dx \int_{S_{\vec{n}_b}^\pm} d\Omega \left| \vec{\Omega} \cdot \vec{n}_b \right| \sum_{i \in E_s} a_{g,i}(\vec{\Omega}) b_i(\vec{x}) \sum_{j \in E_s} b_{g,j}(\vec{\Omega}) b_j(\vec{x}), \quad (84b)$$

$$\langle \mathbf{a}, \mathbf{b} \rangle \equiv \sum_{g=1}^G \sum_{s \in \partial \mathcal{D}} \int_s dx \int_S d\Omega \left| \vec{\Omega} \cdot \vec{n}_b \right| \sum_{i \in E_s} a_{g,i}(\vec{\Omega}) b_i(\vec{x}) \sum_{j \in E_s} b_{g,j}(\vec{\Omega}) b_j(\vec{x}), \quad (84c)$$

$$(\mathbf{a}, \mathbf{b})_{\mathcal{D}} \equiv \sum_{g=1}^G \int_{\mathcal{D}} dx \sum_{i \in E_e} a_{g,i} b_i(\vec{x}) \sum_{j \in E_e} b_{g,j} b_j(\vec{x}), \quad (84d)$$

$$(\mathbf{a}, \mathbf{b})_{\partial \mathcal{D}} \equiv \sum_{g=1}^G \sum_{s \in \partial \mathcal{D}} \int_s dx \sum_{i \in E_s} a_{g,i} b_i(\vec{x}) \sum_{j \in E_s} b_{g,j} b_j(\vec{x}). \quad (84e)$$

However, we still keep the same notation without spatial expansion for simplicity.

3.3.2 SAAF

In the SAAF method, we split the streaming and collision operator \mathbb{L}

$$\mathbb{L}\Psi \equiv \mathbb{L}_1\Psi + \mathbb{L}_2\Psi, \quad (85)$$

where

$$\mathbb{L}_1\Psi \equiv \begin{bmatrix} \vec{\Omega} \cdot \vec{\nabla} \Psi_1 \\ \vec{\Omega} \cdot \vec{\nabla} \Psi_2 \\ \vdots \\ \vec{\Omega} \cdot \vec{\nabla} \Psi_G \end{bmatrix},$$

and define

$$\mathbb{L}_2^{-1}\Psi \equiv \begin{bmatrix} \Psi_1 \\ \frac{\Psi_1}{\Sigma_{t,1}} \\ \Psi_2 \\ \frac{\Psi_2}{\Sigma_{t,2}} \\ \vdots \\ \Psi_G \\ \frac{\Psi_G}{\Sigma_{t,G}} \end{bmatrix},$$

We can easily see that $\mathbb{L}_2^{-1}\mathbb{L}_2 = \mathbb{I}$, $\mathbb{L}_2^* = \mathbb{L}_2$, $\mathbb{L}_1^* = -\mathbb{L}_1$ and $\mathbb{L}_1\mathbb{L}_2 = \mathbb{L}_2\mathbb{L}_1$.

With these definitions we can rearrange the CTE Eq. (3) as

$$\Psi = \mathbb{L}_2^{-1} \left(\mathbf{Q}^{\text{ext}} + \mathbb{S}\Psi - \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) - \mathbb{L}_1\Psi \right), \quad (86)$$

which we call the *angular flux equation* (AFE). Then we substitute this AFE into the streaming kernel of the unstabilized bilinear form Eq. (68b) copied here,

$$\mathbb{b}(\Psi^*, \Psi) \equiv \left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) + \boxed{(\mathbb{L}_1^* \Psi^*, \Psi)} + (\mathbb{L}_2^* \Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+ - \langle \Psi^*, \mathbb{B}\Psi \rangle^- - \langle \Psi^*, \mathbb{S}\Psi \rangle,$$

and obtain the following after some manipulations

$$\begin{aligned} \mathbb{b}(\Psi^*, \Psi) &= \boxed{(\mathbb{L}_1 \Psi^*, \mathbb{L}_2^{-1} \mathbb{L}_1 \Psi)} + (\mathbb{L}_2^* \Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+ \\ &\quad + \left(\mathbb{L}_2^{-1} \mathbb{L}_1 \Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) - \langle \Psi^*, \mathbb{B}\Psi \rangle^- - (\mathbb{L}_2^{-1} \mathbb{L}_1 \Psi^*, \mathbb{S}\Psi), \\ &\quad - (\mathbb{L}_2^{-1} \mathbb{L}_1 \Psi^*, \mathbf{Q}^{\text{ext}}). \end{aligned}$$

We noticed that the boxed kernels are symmetric positive definite (SPD). We move the extra source term into the linear form and finally end up with the SAAF weak form, given by:

$$\begin{aligned} \mathbb{b}_{\text{SAAF}}(\Psi^*, \Psi) &\equiv (\mathbb{L}_1 \Psi^*, \mathbb{L}_2^{-1} \mathbb{L}_1 \Psi) + (\mathbb{L}_2^* \Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+ \\ &\quad - \left(\mathbb{L}_2^{-1} \mathbb{L}_1 \Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) - (\mathbb{L}_2^{-1} \mathbb{L}_1 \Psi^*, \mathbb{S}\Psi) - \langle \Psi^*, \mathbb{B}\Psi \rangle^-, \end{aligned} \quad (87)$$

$$\mathbb{l}_{\text{SAAF}}(\Psi^*) \equiv \langle \Psi^*, \Psi^{\text{inc}} \rangle^- + (\mathbb{L}_2^{-1} \mathbb{L}_1 \Psi^*, \mathbf{Q}^{\text{ext}}). \quad (88)$$

SAAF is equivalent with SUPG (Streamline upwind/Petrov-Galerkin) [51] with the inverse of group-wise total cross sections as the stabilization parameter for all directions of corresponding groups. SAAF must be modified to work with void or near void where the total cross sections are zero or close to zero. We define the following operator

$$\boldsymbol{\tau} \equiv \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_G \end{bmatrix},$$

where

$$\tau_g = \begin{cases} \frac{1}{c\Sigma_{t,g}} & ch\Sigma_{t,g} \geq \varsigma \\ \frac{h}{\varsigma}, & ch\Sigma_{t,g} < \varsigma \end{cases}, \quad (89)$$

where h is the maximum vertex separation for points within an element; ς is usually chosen to be a constant equal to 0.5. c is a constant to set the maximum stabilization from $1/\Sigma_{t,g}$ to $1/(c\Sigma_{t,g})$. Its default value is 1. We then redefine our AFE as

$$\Psi = (\mathbb{I} - \tau\mathbb{L}_2) \Psi + \tau \left(\mathbf{Q}^{\text{ext}} + \mathbb{S}\Psi - \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) - \mathbb{L}_1 \Psi \right), \quad (90)$$

If there is no void and ς is set to zero, $\tau\mathbb{L}_2 = \mathbb{I}$ and the modified SAAF is identical to the original SAAF regardless of the mesh used. Substituting Eq. (90) into the streaming term of Eq. (68b), we obtain the weak form with this void treatment (VT),

$$\begin{aligned} \mathbb{b}_{\text{SAAF-VT}}(\Psi^*, \Psi) &\equiv (\mathbb{L}_1 \Psi^*, (\tau\mathbb{L}_1 - \mathbb{I} + \tau\mathbb{L}_2) \Psi) + (\mathbb{L}_2 \Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+ \\ &\quad + \left((\mathbb{I} + \tau\mathbb{L}_1) \Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) - ((\mathbb{I} + \tau\mathbb{L}_1) \Psi^*, \mathbb{S}\Psi) - \langle \Psi^*, \mathbb{B}\Psi \rangle^-, \end{aligned} \quad (91)$$

$$\mathbb{l}_{\text{SAAF-VT}}(\Psi^*) \equiv \langle \Psi^*, \Psi^{\text{inc}} \rangle^- + ((\mathbb{I} + \tau\mathbb{L}_1) \Psi^*, \mathbf{Q}^{\text{ext}}). \quad (92)$$

It is noted that group-wise global balance can be shown by substituting the constant 1 for a particular group and 0 for the rest as the test function (using the fact that the constant test function belongs to V_h). However, local or element-wise particle balance is not guaranteed and subjects to discretization errors. If τ is zero, we retrieve the unstabilized weak form.

The balance equation with isotropic test functions Φ^* is

$$\begin{aligned} &\left(\Phi^*, \frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}} \right) \right)_D - (\vec{\nabla} \Phi^*, \vec{\mathbf{J}})_D + (\mathbb{L}_2 \Phi^*, \Phi)_D - (\Phi^*, \mathbb{S}_0 \Phi)_D + (\Phi^*, (\mathbf{I} - \alpha) \mathbf{J}^o)_{\partial D} + \\ &\left(\tau \vec{\nabla} \Phi^*, \boxed{\frac{\partial}{\partial t} \left(\frac{\vec{\mathbf{J}}}{\mathbf{v}} \right) + \vec{\nabla} \cdot (\vec{\mathbf{E}} \Phi) + \mathbb{L}_2 \vec{\mathbf{J}} - \mathbb{S}_1 \vec{\mathbf{J}} - \vec{\mathbf{Q}}_1^{\text{ext}}} \right)_D = \\ &(\Phi^*, \mathbf{J}^{\text{inc}})_{\partial D} + (\Phi^*, \mathbf{Q}_0^{\text{ext}})_D, \end{aligned} \quad (93)$$

where the boxed equation is simply the residual of the momentum equation of Eq. (18) and is analytically equal to zero. We can further obtain the P_0 projection of the unmodified weak form with $\tau = \mathbb{L}_2^{-1}$ by assuming $\Psi = \frac{1}{4\pi} \Phi$:

$$\begin{aligned} &\left(\Phi^*, \frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}} \right) \right)_D + \left(\vec{\nabla} \Phi^*, \frac{1}{3} \mathbb{L}_2^{-1} \vec{\nabla} \Phi \right)_D + (\mathbb{L}_2 \Phi^*, \Phi)_D - (\Phi^*, \mathbb{S}_0 \Phi)_D + \left(\Phi^*, \frac{1}{4} (\mathbf{I} - \alpha) \Phi \right)_{\partial D} = \\ &(\Phi^*, \mathbf{J}^{\text{inc}})_{\partial D} + (\Phi^*, \mathbf{Q}_0^{\text{ext}})_D, \end{aligned} \quad (94)$$

which is the weak form for a multigroup diffusion equation. It can be seen that this equation is slightly different from the bilinear form for CDE, Eq. (255), with CFEM in Section 3.5

The streaming operator in the AFE needs to be updated as $\mathbb{L}_1 \Psi \rightarrow \mathbb{L}_1 \Psi + \mathbb{E}\Psi + \mathbb{E}_2 \Psi$ in curvilinear coordinates. The additional term to add to Eq. (91) is thus

$$((\mathbb{I} + \tau\mathbb{L}_1) \Psi^*, \mathbb{E}\Psi) + (\tau\mathbb{L}_1 \Psi^*, \mathbb{E}_2 \Psi). \quad (95)$$

We present the derivation of the mathematical adjoint equations for the stabilized SAAF equations in an abstract form. The presented development is a generalization of the original derivation published in [52] for SAAF- S_N . The mathematical adjoint is different than the physical adjoint. The physical adjoint is obtained from the continuous PDEs and then the discretization is applied, while for the mathematical adjoint the equations are discretized first and then the adjoint is obtained from the discretized equations. Obtaining the adjoint from the discretized equation is achieved by transposing the discretized set of equations. For the first order transport equation, the physical and mathematical adjoint equations are identical, while for the SAAF equations, there is a distinct difference. The general advantage of the mathematical adjoint is that its property to commute under a discrete representation of

the inner product is exactly satisfied, while it is only approximately satisfied for the physical adjoint. Adjoints are extremely useful for perturbation theory, optimization, and sensitivity analysis.

First, the rules for the transformation leading to the mathematical adjoint are explained:

- $\vec{\Omega} \rightarrow -\vec{\Omega}$, hence $\mathbb{L}_1 \rightarrow -\mathbb{L}_1$, $\mathbb{S} \rightarrow \bar{\mathbb{S}}$, where

$$\bar{\mathbb{S}}\Psi \equiv \begin{bmatrix} \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow 1} f_{g' \rightarrow 1} (-\vec{\Omega} \cdot \vec{\Omega}') \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow 2} f_{g' \rightarrow 2} (-\vec{\Omega} \cdot \vec{\Omega}') \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \vdots \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow G} f_{g' \rightarrow G} (-\vec{\Omega} \cdot \vec{\Omega}') \Psi_{g'}(\vec{\Omega}') d\Omega' \end{bmatrix},$$

- The solution is reinterpreted as test function and the test function is reinterpreted as solution.

It should again be stressed that we now solve for the adjoint flux Ψ^* . The transformation is applied to the stabilized bilinear SAAF form Eq. (91) on a term by term basis:

1. The streaming terms:

$$(\mathbb{L}_1 \Psi^*, (\tau \mathbb{L}_1 - \mathbb{I} + \tau \mathbb{L}_2) \Psi) + \langle \Psi^*, \Psi \rangle^+ \rightarrow (-\mathbb{L}_1 \Psi^*, (-\tau \mathbb{L}_1 - \mathbb{I} + \tau \mathbb{L}_2) \Psi) + \langle \Psi^*, \Psi \rangle^- \quad (96)$$

Note the change of the half-sphere over which boundary terms are integrated. This change originates from inverting the angular directions. Simple manipulations of the term in Eq. (96) lead to:

$$(\mathbb{L}_1 \Psi^*, \tau(\mathbb{L}_1 - \mathbb{L}_2) \Psi) + (\mathbb{L}_1 \Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^-. \quad (97)$$

Application of integration by parts on the second term in Eq. (97) reveals that:

$$(\mathbb{L}_1 \Psi^*, \Psi) = -(\Psi^*, \mathbb{L}_1 \Psi) + \langle \Psi^*, \Psi \rangle^+ - \langle \Psi^*, \Psi \rangle^-. \quad (98)$$

Substituting Eq. (98) into Eq. (97) gives:

$$(\mathbb{L}_1 \Psi^*, \tau(\mathbb{L}_1 - \mathbb{L}_2) \Psi) - (\Psi^*, \mathbb{L}_1 \Psi) + \langle \Psi^*, \Psi \rangle^+. \quad (99)$$

2. The total interaction term remains unchanged:

$$(\mathbb{L}_2 \Psi^*, \Psi) \rightarrow (\mathbb{L}_2 \Psi^*, \Psi) = (\Psi^*, \mathbb{L}_2 \Psi). \quad (100)$$

3. We do not discuss the time derivative term here because the mathematical adjoint equations are currently only implemented for steady state and eigenvalue calculations.
4. The mathematical adjoint equations can only use specular reflective boundary conditions, so for the operator \mathbb{B} we set $\alpha_g^s = 1$ and $\alpha_g^d = 0$. Applying the transformation gives:

$$\langle \Psi^*, \mathbb{B} \Psi \rangle^- \rightarrow \langle \Psi^*, \mathbb{B} \Psi \rangle^+ = \langle \mathbb{B}^* \Psi^*, \Psi \rangle^-. \quad (101)$$

5. The scattering term is transformed to obtain:

$$((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{S} \Psi) \rightarrow ((\mathbb{I} - \tau \mathbb{L}_1) \Psi^*, \bar{\mathbb{S}} \Psi) = (\bar{\mathbb{S}}^* (\mathbb{I} - \tau \mathbb{L}_1) \Psi^*, \Psi). \quad (102)$$

The linear form is simply given by:

$$\mathbb{I}^*(\Psi) = (\mathbf{Q}^*, \Psi) + \langle \Psi^{*,\text{inc}}, \Psi \rangle^-. \quad (103)$$

The source and inflow adjoint fluxes often constitute degrees of freedom to achieve a suitable adjoint flux solution for a specific purpose. For example, the source could be the cross section in a region where the sensitivity of a reaction rate with respect to this cross section is desired. The final mathematical adjoint SAAF equations are given by:

$$\mathbb{b}_{\text{SAAF-VT}}^*(\Psi, \Psi^*) = \mathbb{I}^*(\Psi) \quad (104)$$

with

$$\begin{aligned} \mathbb{b}_{\text{SAAF-VT}}^*(\Psi, \Psi^*) = & (\tau(\mathbb{L}_1 - \mathbb{L}_2) \Psi, \mathbb{L}_1 \Psi^*) - (\mathbb{L}_1 \Psi, \Psi^*) + \langle \Psi, \Psi^* \rangle^+ \\ & + (\mathbb{L}_2 \Psi, \Psi^*) - (\Psi, \bar{\mathbb{S}}^* (\mathbb{I} - \tau \mathbb{L}_1) \Psi^*) - \langle \Psi, \mathbb{B}^* \Psi^* \rangle^-. \end{aligned} \quad (105)$$

3.3.3 LS

If we use $\mathbb{L}\Psi^*$ as the test function, we obtain the LS formulation

$$\left(\mathbb{L}\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}}\right)\right) + (\mathbb{L}\Psi^*, \mathbb{L}\Psi) = (\mathbb{L}\Psi^*, \mathbf{S}^{\text{ext}}) + (\mathbb{L}\Psi^*, \mathbf{S}\Psi).$$

This formulation does not have the boundary condition terms. We can strongly enforce the boundary conditions or we can add terms to weakly impose them

$$\left(\mathbb{L}\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}}\right)\right) + (\mathbb{L}\Psi^*, \mathbb{L}\Psi) + \langle \mathbb{C}\Psi^*, \Psi - \Psi_b^{\text{inc}} - \mathbb{B}\Psi \rangle^- = (\mathbb{L}\Psi^*, \mathbf{Q}^{\text{ext}}) + (\mathbb{L}\Psi^*, \mathbf{S}\Psi),$$

where the options for $\mathbb{C} = \text{diag}\{c_g, g = 1, \dots, G\}$ are given in Table 3. This results into the following bilinear and linear form

$$\begin{aligned} \mathbb{b}_{\text{LS}}(\Psi^*, \Psi) &\equiv (\mathbb{L}\Psi^*, \mathbb{L}\Psi) + \langle \mathbb{C}\Psi^*, \Psi \rangle^- \\ &\quad + \left(\mathbb{L}\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}}\right)\right) - (\mathbb{L}\Psi^*, \mathbf{S}\Psi) - \langle \mathbb{C}\Psi^*, \mathbb{B}\Psi \rangle^-, \end{aligned} \quad (106)$$

$$\mathbb{l}_{\text{LS}}(\Psi^*) \equiv \langle \mathbb{C}\Psi^*, \Psi_b^{\text{inc}} \rangle^- + (\mathbb{L}\Psi^*, \mathbf{Q}^{\text{ext}}). \quad (107)$$

We can prove that when \mathbb{L}_2 is spatially independent and $\mathbb{C} = \mathbb{L}_2$, the LS weak form is equivalent to the original SAAF weak form. In general, the LS weak form does not have local and global balance. This would destroy the super convergence of k -effective relying on this feature.

Table 3 Boundary condition type for LS schemes, c_0 being a constant positive value.

Type	Expression of c_g
1	c_0
2	$c_0 \Sigma_{t,g}$
3	$c_0(3 \vec{\Omega} \cdot \vec{n} + 2)$
4	$c_0(1 + \Sigma_{t,g})(3 \vec{\Omega} \cdot \vec{n} + 2)$
5	$c_0(4\frac{1}{h} + \Sigma_{t,g})(3 \vec{\Omega} \cdot \vec{n} + 2)$
6	$h_0 = \min\left(4, \max\left(c_0, \frac{V_{\text{elem}}}{A_{\text{elem,BC}}}\right)\right)$
	$c_0(4\frac{1}{h_0} + \Sigma_{t,g})(3 \vec{\Omega} \cdot \vec{n} + 2)$
7	$c_0 \tau_g^{-1}$

The weighted least-squares (WLS) transport equation addresses some issues of the unweighted LS equation on material interfaces and in voids. Due to the second order nature of the LS equation, downstream information can influence the upstream solution. An optically thick material further downstream in the current direction reduces the flux in the unweighted LS. Therefore the LS equation does not have causality. The introduction of a weight function diminishes this problem significantly. We denote a weight function as

$$\mathbf{W} \equiv \begin{bmatrix} W_1(\vec{x}, t) \\ W_2(\vec{x}, t) \\ \vdots \\ W_G(\vec{x}, t) \end{bmatrix}.$$

And WLS weak form becomes

$$\begin{aligned} \mathbb{b}_{\text{WLS}}(\Psi^*, \Psi) &\equiv (\mathbb{L}\Psi^*, \mathbf{W}\mathbb{L}\Psi) + \langle \mathbb{C}\Psi^*, \mathbf{W}\Psi \rangle^- \\ &\quad + \left(\mathbb{L}\Psi^*, \mathbf{W} \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}}\right)\right) - (\mathbb{L}\Psi^*, \mathbf{W}\mathbf{S}\Psi) - \langle \mathbb{C}\Psi^*, \mathbf{W}\mathbb{B}\Psi \rangle^-, \end{aligned} \quad (108)$$

$$\mathbb{l}_{\text{WLS}}(\Psi^*) \equiv \langle \mathbb{C}\Psi^*, \mathbf{W}\Psi_b^{\text{inc}} \rangle^- + (\mathbb{L}\Psi^*, \mathbf{W}\mathbf{Q}^{\text{ext}}). \quad (109)$$

If \mathbf{W} is equal to \mathbf{I} , we retrieve the original LS form. Ref. [49] proposes the following weight function:

$$W_g(\vec{x}, t) = \min\left(\frac{1}{\Sigma_{t,g}}, w_{\max}\right), \quad g = 1, \dots, G, \quad (110)$$

where w_{\max} is a preset maximum value to make the weight function well-defined over void or near-void. The default value of w_{\max} is 1000cm.

3.3.4 DFEM

We first define

$$a_{\pm}(\vec{x}, \vec{\Omega}) \equiv \lim_{s \rightarrow 0^{\pm}} a(\vec{x} + s\vec{\Omega}, \vec{\Omega}), \quad (111a)$$

$$\llbracket a(\vec{x}, \vec{\Omega}) \rrbracket \equiv a_+(\vec{x}, \vec{\Omega}) - a_-(\vec{x}, \vec{\Omega}), \quad (111b)$$

where a is a generic function having both spatial and angular dependency. The jump $\llbracket a(\vec{x}, \vec{\Omega}) \rrbracket$ of a on internal sides Γ_{int} are not necessarily zero with DFEM. We use the subscripts \pm to differentiate this definition based on the streaming direction $\vec{\Omega}$ from the other one appearing in [DFEM-Diffusion](#) and [upwinding interface condition](#) based on the face normal \vec{n} .

With DFEM, the stabilization is usually achieved with upwinding, i.e. the numerical angular flux on all interior sides are using the upwinding flux with respect to the streaming direction $\vec{\Omega}$. This results into the following form

$$\mathbb{b}(\Psi^*, \Psi) \equiv \left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) + (\mathbb{L}^* \Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+ - \langle \llbracket \Psi^* \rrbracket, \Psi_- \rangle_{\Gamma_{\text{int}}} - \langle \Psi^*, \mathbb{B} \Psi \rangle^- - \langle \Psi^*, \mathbb{S} \Psi \rangle, \quad (112)$$

$$\mathbb{l}(\Psi^*) \equiv \left\langle \Psi^*, \Psi_{\text{b}}^{\text{inc}} \right\rangle^- + \langle \Psi^*, \mathbf{Q}^{\text{ext}} \rangle. \quad (113)$$

A desirable feature of the DFEM weak forms is that the mathematical adjoint weak form is identical with that of the physical adjoint. In addition, both local and global balances are preserved.

3.4 Angular discretization and final weak forms

3.4.1 Angular discretization

The S_N method can be thought of as a collocation method for the angular discretization. Given an angular quadrature set $\left\{ \vec{\Omega}_d, w_d, d = 1, \dots, N_d \right\}$ consisting of N_d directions $\vec{\Omega}_d$ and weights w_d , transport equations are solved along these directions and all angular integrations in the kernels are numerically evaluated with the angular quadrature. The available angular quadratures in Rattlesnake are given in [Appendix E](#). Angular derivative terms like those in curvilinear coordinates need special requirements on the angular quadrature with S_N methods. The reflecting directions on reflecting boundaries of all directions in the angular quadrature need also be therein. Otherwise angular interpolation schemes need to be provided by the angular quadrature for evaluating the angular fluxes in the reflected directions. The primal variables to be solved for with S_N method are $\Psi_{g,n}, n = 1, \dots, M; g = 1, \dots, G$.

The P_N method is a spectral Galerkin FEM method in angle with the basis and test functions being the spherical harmonics defined on 2D unit sphere. More specifically, the angular fluxes are expanded with spherical harmonics truncated to an order L , often referred to as the P_N order. Boundary conditions require the proper rotation of spherical harmonics on boundaries where the unit outward normal is not aligned with the axis defining the polar angle. The primal variables to be solved for with the P_N method are $\Phi_{g,\ell,m}, m = \text{lowm}(\ell), \dots, \text{upm}(\ell); \ell = 0, \dots, L; g = 1, \dots, G$, where two functions for upper and lower bounds depending on ℓ are

$$\text{lowm}(\ell) = \begin{cases} 0, & 1\text{D} \\ 0, & 2\text{D} \\ -\ell, & 3\text{D} \end{cases}, \quad (114)$$

$$\text{upm}(\ell) = \begin{cases} 0, & 1D \\ \ell, & 2D \\ \ell, & 3D \end{cases}. \quad (115)$$

Thus the number of primal variables is

$$\mathcal{N}(L) = \begin{cases} L + 1, & 1D \\ \frac{1}{2}(L + 1)(L + 2), & 2D \\ (L + 1)^2, & 3D \end{cases} \quad (116)$$

times G . We often use a single index to loop through spherical harmonics for simpler notation:

$$\sum_{p=1}^{\mathcal{N}(L)} (\cdot) = \sum_{\ell=0}^L \sum_{m=\text{lowm}(\ell)}^{\text{upm}(\ell)} (\cdot). \quad (117)$$

The conversion from (ℓ, m) to p and is

$$p(\ell, m) = \begin{cases} \ell + 1 & , \quad 1D \\ \ell(\ell + 1)/2 + m + 1 & , \quad 2D \\ \ell^2 + \ell + m + 1 & , \quad 3D \end{cases}. \quad (118)$$

It is noted that $\mathcal{N}(L) = p(L, L)$.

For even-parity P_N , the primal variables are $\Phi_{g,\ell,m}, m = \text{lowm}(\ell), \dots, \text{upm}(\ell); \ell \leq L$ and even; $g = 1, \dots, G$. Accordingly, the number of primal variables is

$$\mathcal{N}_e(L) = \begin{cases} \frac{1}{2}(L + 1), & 1D \\ \frac{1}{4}(L + 1)^2, & 2D \\ \frac{1}{2}L(L + 1), & 3D \end{cases} \quad (119)$$

times G , where L is an odd number.

The diffusion approximation is a special case of the P_N method and discussed in Section 3.5. It can be show that SAAF-P0 without anisotropic scattering is the diffusion equation with the diffusion coefficient being defined as $D_g = \frac{1}{3\Sigma_{t,g}}$. The connection between diffusion and transport can be clearly revealed in the NDA section.

The following six subsections – each dedicated to a particular discretization scheme – are arranged with the same subsections. The first subsection describes the kernels of the steady-state CTE. The second subsection is for the boundary conditions. The third shows the time derivative kernel for transient calculations. The fourth contains the kernels for neutron transport. The fifth describes kernels for thermal radiation transport. Extra kernels for R-spherical and RZ (axial-symmetric) coordinates are in the next two subsections. Kernels for mathematical adjoint are at the end. The last subsection summarizes all kernels, auxiliary kernels, BCs, auxiliary materials, etc. into tables. Not all capabilities are supported by all schemes at the moment. The capability matrix is showed in Table 4.

Table 4 The capability of schemes.

Scheme	Neutron	Thermal radiation	Transient	R-spherical	RZ	Mathematical adjoint	Multi-scheme
CFEM-Diffusion	Y	Y	Y	Y	Y	Y	Y
DFEM-Diffusion	Y	N	Y	Y	Y	N	N
SAAF-CFEM-SN	Y	Y	Y	Y	Y	Y	Y
SAAF-CFEM-PN	Y	N	Y	N	Y	N	Y
LS-CFEM-SN	Y	N	Y	N	N	N	N
LS-CFEM-PN	Y	N	N	N	N	N	N
DFEM-SN	Y	N	Y	N	Y	Y	N
DFEM-PN	N	Y	Y	N	N	N	N

3.4.2 SAAF-CFEM-SN

We will expand all the kernels in order to make a direct correspondence to the code.

3.4.2.1 Steady-state CTE

Streaming kernel:

$$\begin{aligned} (\mathbb{L}_1 \Psi^*, (\tau \mathbb{L}_1 - \mathbb{I} + \tau \mathbb{L}_2) \Psi) &= \sum_{g=1}^G \int_S d\Omega \left(\vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g - (1 - \tau_g \Sigma_{t,g}) \Psi_g \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} \left(\vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} - (1 - \tau_g \Sigma_{t,g}) \Psi_{g,d} \right)_{\mathcal{D}} \end{aligned} \quad (120)$$

Collision kernel:

$$(\Psi^*, \mathbb{L}_2 \Psi) = \sum_{g=1}^G \int_S d\Omega \left(\Psi_g^*, \Sigma_{t,g} \Psi_g \right)_{\mathcal{D}} = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^*, \Sigma_{t,g} \Psi_{g,d} \right)_{\mathcal{D}} \quad (121)$$

Rattlesnake provides two treatments of the scattering kernels. One is the P_N treatment:

$$\begin{aligned} ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{S} \Psi) &= \sum_{g=1}^G \int_S d\Omega \left(\Psi_g^* + \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \sum_{g'=1}^G \sum_{\ell=0}^L \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g',\ell,m} R_{\ell,m} \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_n \sum_{g'=1}^G \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \sum_{\ell=0}^L \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g',\ell,m} R_{\ell,m}(\vec{\Omega}_d) \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_n \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} S_{g,\ell,m} R_{\ell,m}(\vec{\Omega}_d) \right)_{\mathcal{D}} \end{aligned} \quad (122)$$

with

$$\Phi_{g',\ell,m} = \sum_{d=1}^{N_d} w_d Y_{\ell,m}(\vec{\Omega}_d) \Psi_{g',d}, \quad (123)$$

and

$$S_{g,\ell,m} = \sum_{g'=1}^G \Sigma_{s,\ell}^{g' \rightarrow g} \Phi_{g',\ell,m}. \quad (124)$$

The other is the direct S_N treatment:

$$\begin{aligned} ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{S} \Psi) &= \sum_{g=1}^G \int_S d\Omega \left(\Psi_g^* + \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \sum_{g'=1}^G \int_S \Sigma_{s,g' \rightarrow g} f_{g' \rightarrow g}(\vec{\Omega}' \cdot \vec{\Omega}) \Psi_{g'}(\vec{\Omega}') d\Omega' \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \sum_{g'=1}^G \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \Sigma_{s,g' \rightarrow g} \sum_{d'=1}^{N_d} f_{g' \rightarrow g,d' \rightarrow d} \Psi_{g',d'} \right)_{\mathcal{D}}, \end{aligned} \quad (125)$$

where

$$f_{g' \rightarrow g,d' \rightarrow d} = \begin{cases} w_{d'} f_{g' \rightarrow g}(\vec{\Omega}_{d'} \cdot \vec{\Omega}_d), & \text{sampling method} \\ \int_{P_{d'}} f_{g' \rightarrow g}(\vec{\Omega}' \cdot \vec{\Omega}_d) d\Omega', & \text{patch method} \end{cases}. \quad (126)$$

The patch method requires the angular quadrature to build the patches on the 2D unit sphere for all the directions. The external source kernel:

$$((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbf{Q}^{\text{ext}}) = \sum_{g=1}^G \int_S d\Omega \left(\Psi_g^* + \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, Q_g^{\text{ext}} \right)_{\mathcal{D}}$$

$$= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, Q_g^{\text{ext}}(\vec{\Omega}_d) \right)_{\mathcal{D}}. \quad (127)$$

3.4.2.2 Boundary conditions

Vacuum boundary conditions:

$$\langle \Psi^*, \bar{\Psi} \rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi^*, \bar{\Psi} \rangle_{\partial \mathcal{D}_s}^- = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^*, \vec{\Omega}_d \cdot \vec{n}_b \bar{\Psi}_{g,d,\text{vacuum}} \right)_{\partial \mathcal{D}_s}, \quad (128)$$

where numerical angular flux $\bar{\Psi}_{g,n}$ is

$$\bar{\Psi}_{g,d,\text{vacuum}} = \begin{cases} \begin{cases} \Psi_{g,d}, & \text{SAAF or PE} \\ \Psi_{g,d} + \Psi_{g,d,\text{reverse}} - \Psi_{g,d,\text{reverse}}^{\text{inc}}, & \text{EVEN} \\ \Psi_{g,d} - \Psi_{g,d,\text{reverse}} + \Psi_{g,d,\text{reverse}}^{\text{inc}}, & \text{ODD} \end{cases}, & \vec{\Omega}_d \cdot \vec{n}_b > 0, \vec{x} \in \partial \mathcal{D}_s \\ \begin{cases} \Psi_{g,d}^{\text{inc}}, & \text{SAAF, EVEN or ODD} \\ 2\Psi_{g,d}^{\text{inc}} - \Psi_{g,d}, & \text{PE} \end{cases}, & \vec{\Omega}_d \cdot \vec{n}_b < 0, \vec{x} \in \partial \mathcal{D}_s \end{cases}, \quad (129)$$

where d_{reverse} stands for the direction index such that $\vec{\Omega}_{d_{\text{reverse}}} = -\vec{\Omega}_d$. Angular quadratures typically have both the direction and its reverse in the set.

Reflecting boundary conditions:

$$\langle \Psi^*, \bar{\Psi} \rangle_{\partial \mathcal{D}_r}^+ - \langle \Psi^*, \bar{\Psi} \rangle_{\partial \mathcal{D}_r}^- = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^*, \vec{\Omega}_d \cdot \vec{n}_b \bar{\Psi}_{g,d,\text{reflecting}} \right)_{\partial \mathcal{D}_r}, \quad (130)$$

$$\bar{\Psi}_{g,d,\text{reflecting}} = \begin{cases} \begin{cases} \Psi_{g,d}, & \text{SAAF} \\ \Psi_{g,d} - \frac{\Psi_{g,d} - \Psi_{g,d,\text{reflecting}}}{2}, & \text{NSYM} \\ \Psi_{g,d} + \frac{\Psi_{g,d} - \Psi_{g,d,\text{reflecting}}}{2}, & \text{SYM} \end{cases}, & \vec{\Omega}_d \cdot \vec{n}_b > 0, \vec{x} \in \partial \mathcal{D}_r \\ \begin{cases} \Psi_{g,d,\text{reflecting}}', & \text{SAAF} \\ -\frac{\Psi_{g,d} + \Psi_{g,d,\text{reflecting}}}{2}, & \text{NSYM or SYM} \end{cases}, & \vec{\Omega}_d \cdot \vec{n}_b < 0, \vec{x} \in \partial \mathcal{D}_r \end{cases}, \quad (131)$$

where $d_{\text{reflecting}}$ stands for the direction index such that $\vec{\Omega}_{d_{\text{reflecting}}} = \vec{\Omega}_d - 2(\vec{\Omega}_d \cdot \vec{n}_b)\vec{n}_b$. Special care is needed to make sure angular quadratures contains the reflecting directions for all boundary normals if the angular interpolation scheme is not available.

White boundary conditions currently only have one option:

$$\begin{aligned} & \langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_w}^+ - \langle \Psi^*, \mathbb{B}_w \Psi \rangle_{\partial \mathcal{D}_w}^- \\ &= \sum_{g=1}^G \left(\sum_{\vec{\Omega}_d \cdot \vec{n}_b > 0} w_d \left(\Psi_{g,d}^*, \vec{\Omega}_d \cdot \vec{n}_b \Psi_{g,d} \right)_{\partial \mathcal{D}_w} + \sum_{\vec{\Omega}_d \cdot \vec{n}_b < 0} w_d \left(\Psi_{g,d}^*, \vec{\Omega}_d \cdot \vec{n}_b \frac{J_{\text{out}}}{W_{\text{half}}(\vec{n}_b)} \right)_{\partial \mathcal{D}_w} \right), \end{aligned} \quad (132)$$

where

$$J_{\text{out}} = \sum_{\vec{\Omega}_d \cdot \vec{n}_b > 0} w_d \vec{\Omega}_d \cdot \vec{n}_b \Psi_{g,d}, \quad (133)$$

$$W_{\text{half}}(\vec{n}_b) = \sum_{\vec{\Omega}_d \cdot \vec{n}_b > 0} w_d \vec{\Omega}_d \cdot \vec{n}_b. \quad (134)$$

It is noted that the half weighting $W_{\text{half}}(\vec{n}_b)$ depends on \vec{n}_b and is not exactly equal to $\frac{\pi}{4}$, the analytical integration. $W_{\text{half}}(\vec{n}_b)$ are provided by the angular quadratures.

3.4.2.3 Transient kernel

Time derivative kernel:

$$\begin{aligned} \left((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{v} \right) \right) &= \sum_{g=1}^G \int_{\mathcal{S}} d\Omega \left(\Psi_g^* + \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{\partial}{\partial t} \left(\frac{\Psi_g}{v_g} \right) \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{\partial}{\partial t} \left(\frac{\Psi_{g,d}}{v_g} \right) \right)_{\mathcal{D}}. \end{aligned} \quad (135)$$

If we assume $v_g, g = 1, \dots, G$ is time independent, the kernel can be simplified into

$$\sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{v_g} \frac{\partial \Psi_{g,d}}{\partial t} \right)_{\mathcal{D}}. \quad (136)$$

3.4.2.4 Neutron

Prompt fission kernel:

$$\begin{aligned} ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{F} \Psi) &= \sum_{g=1}^G \int_{\mathcal{S}} d\Omega \left(\Psi_g^* + \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{1}{4\pi} \frac{\bar{\chi}_g}{k} \sum_{g'=1}^G \nu \Sigma_{f,g'} \int_{\mathcal{S}} \Psi_{g'}(\vec{\Omega}') d\Omega' \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{4\pi} \frac{\bar{\chi}_g}{k} F \right)_{\mathcal{D}}, \end{aligned} \quad (137)$$

where F is the isotropic fission source:

$$F = \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'}. \quad (138)$$

It is noted that for transients, $\bar{\chi}_g = \chi_{p,g}(1 - \beta)$ and for steady-state problems $\bar{\chi}_g = \chi_g$.

Delayed neutron kernel:

$$\begin{aligned} ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{C} \mathbb{C}) &= \sum_{g=1}^G \int_{\mathcal{S}} d\Omega \left(\Psi_g^* + \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{1}{4\pi} \frac{1}{k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{4\pi} \frac{1}{k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i \right)_{\mathcal{D}}. \end{aligned} \quad (139)$$

The DNP equation does not have the spatial derivative term, so DNP concentrations are typically evaluated directly as:

$$C_i(\vec{x}, t) = \begin{cases} \gamma_i C_i(\vec{x}, t_{\text{old}}) + (1 - \gamma_i) \frac{\beta_i}{\lambda_i} F(\vec{x}, t), & \text{constant} \\ \gamma_i C_i(\vec{x}, t_{\text{old}}) + \left(\frac{1 - \gamma_i}{\lambda_i \Delta t} - \gamma_i \right) \frac{\beta_i}{\lambda_i} F(\vec{x}, t_{\text{old}}) + \left(1 - \frac{1 - \gamma_i}{\lambda_i \Delta t} \right) \frac{\beta_i}{\lambda_i} F(\vec{x}, t), & \text{linear} \end{cases}, \quad (140a)$$

where

$$\gamma_i(\lambda_i \Delta t) = \begin{cases} \frac{1}{1 + \lambda_i \Delta t}, & \text{backward Euler} \\ \frac{2 - \lambda_i \Delta t}{2 + \lambda_i \Delta t}, & \text{Crank-Nicholson} \\ e^{-\lambda_i \Delta t}, & \text{exponential} \end{cases}. \quad (140b)$$

We assume that both λ_i and β_i do not change from t_{old} to t and define $\Delta t \equiv t - t_{\text{old}}$. It is not surprising that the backward Euler scheme will give the same equation for constant and linear approximation of the fission source from t_{old} to t . It is also noted that this equation is independent of the weak forms and the time integration schemes in the transient solver. It is noted that Eq. (138) and Eq. (140a) are identical for other discretization schemes with neutron transport.

3.4.2.5 Thermal radiation

Planckian emission kernel:

$$\begin{aligned} \left((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \frac{1}{4\pi} \mathbf{S}_{\text{BB}}(T) \right) &= \sum_{g=1}^G \int_S d\Omega \left(\Psi_g^* + \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{1}{4\pi} S_{\text{BB},g} \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{4\pi} S_{\text{BB},g} \right)_{\mathcal{D}}. \end{aligned} \quad (141)$$

$S_{\text{BB},g}$ is defined by Eq. (62).

3.4.2.6 R-spherical coordinate

The extra kernel for R-spherical coordinate is:

$$\begin{aligned} &((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{E} \Psi) + (\tau \mathbb{L}_1 \Psi^*, \mathbb{E}_2 \Psi) \\ &= \sum_{g=1}^G \int_S d\Omega \left\{ \left(\Psi_g^* + \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi_g] \right)_{\mathcal{D}} + \left(\tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{2}{r} \mu \Psi_g \right) \right\}. \end{aligned} \quad (142)$$

Special care is needed with S_N methods for evaluating $\frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi_g]$, which are explained in Section E.1.1. With S_N , the kernel becomes

$$\sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left\{ \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{r} \sum_{d'=1}^{N_d} \beta_{d,d'} \Psi_{g,d'} \right)_{\mathcal{D}} + \left(\tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{2}{r} \mu_d \Psi_{g,d} \right) \right\}, \quad (143)$$

where the expression for β can be found in Section E.1.1. We keep the notation $\vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*$ although in R-spherical coordinate, it is simply equal to $\mu_d \frac{\partial \Psi_{g,d}}{\partial r}$. The Jacobian of this kernel must be evaluated for good linear convergence. It is noted that the evaluation of μ -derivative may introduce *dummy directions*. Dummy here means that, although the directions are in the angular quadrature, their weights for the angular integration are zero. Introduction of dummy directions results into more primal variables and special care on their mathematical adjoint needs to be taken. The scaling of dummy directions are set to a fixed non-zero value $w_{\text{dummy}} = \sum_{d=1}^{N_d} w_d / N_d$ while the scaling for other normal directions is w_d .

3.4.2.7 RZ-axial symmetric coordinate

The extra kernel for RZ-axial symmetric coordinate is:

$$\begin{aligned} &((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{E} \Psi) + (\tau \mathbb{L}_1 \Psi^*, \mathbb{E}_2 \Psi) = (\tau \mathbb{L}_1 \Psi^*, (\mathbb{E} + \mathbb{E}_2) \Psi) + (\Psi^*, \mathbb{E} \Psi) \\ &= - \sum_{g=1}^G \int_S d\Omega \left\{ \left(\Psi_g^* + \tau_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{\Omega_\theta}{r} \frac{\partial \Psi_g}{\partial \omega} \right)_{\mathcal{D}} + \left(\Psi_g^*, \frac{\Omega_r}{r} \Psi_g \right) \right\}, \\ &= - \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left\{ \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{r} \sum_{d'=1}^N \alpha_{d,d'} \Psi_{g,d'} \right)_{\mathcal{D}} + \left(\Psi_{g,d}^*, \frac{\Omega_{r,d}}{r} \Psi_{g,d} \right) \right\}, \end{aligned} \quad (144)$$

where the expression for α can be found in the appendix (see Eq. (555)).

3.4.2.8 Mathematical adjoint

It is noted again that for the mathematical adjoint kernels, $\Psi_{g,n}^*$ is the solution and $\Psi_{g,n}$ is a member of the test space. Streaming term for the SAAF mathematical adjoint equations implemented in SAAFAdjointStreaming:

$$(\mathbb{L}_1 \Psi^*, (\tau (\mathbb{L}_1 - \mathbb{L}_2) \Psi)) - (\Psi^*, \mathbb{L}_1 \Psi) = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left[\left(\tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} - \Sigma_{t,g} \Psi_{g,d} \right)_{\mathcal{D}} \right]$$

$$- \left(\Psi_{g,d}^*, \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} \right)_{\mathcal{D}} \Big] \quad (145)$$

The collision term is the same as for the forward SAAF- S_N equations.

The mathematical adjoint scattering term for the SAAF equations is:

$$\begin{aligned} ((\mathbb{I} - \tau \mathbb{L}_1) \Psi^*, \mathbb{S} \Psi) &= (\mathbb{S}^* (\mathbb{I} - \tau \mathbb{L}_1) \Psi^*, \Psi) \\ &= \sum_{g=1}^G \sum_{g'=1}^G \sum_{\ell=0}^L \sum_{k=-\ell}^{\ell} \frac{2\ell+1}{4\pi} \left((-1)^{\ell} \Sigma_{s,\ell}^{g \rightarrow g'} \left[\Phi_{g',\ell,k}^* + \tau_{g'} \vec{\nabla} \cdot \vec{X}_{g',\ell,k}^* \right], \sum_{d=1}^{N_d} w_d \Psi_{g,d} Y_{\ell,k}(-\vec{\Omega}_d) \right)_{\mathcal{D}}, \end{aligned} \quad (146)$$

where

$$\vec{X}_{g,\ell,k}^* = \sum_{d=1}^{N_d} w_d \vec{\Omega}_d Y_{\ell,k}(\vec{\Omega}_d) \Psi_{g,d}^*. \quad (147)$$

We define the adjoint fission source as:

$$F^* = F_0^* - F_1^*, \quad (148a)$$

$$F_0^* = \sum_{g=1}^G \chi_{p,g} \Phi_g^*, \quad (148b)$$

$$F_1^* = \sum_{g=1}^G \tau_g \chi_{p,g} \vec{\nabla} \cdot \vec{J}_g^*. \quad (148c)$$

The fission term for the SAAF mathematical adjoint equations is:

$$((\mathbb{I} - \tau \mathbb{L}_1) \Psi^*, \mathbb{F} \Psi) = (\mathbb{F}^* (\mathbb{I} - \tau \mathbb{L}_1) \Psi^*, \Psi) = \sum_{g=1}^G \frac{1}{4\pi} \left(F^*, \nu \Sigma_{f,g} \sum_{d=1}^{N_d} w_d \Psi_{g,d} \right)_{\mathcal{D}}. \quad (149)$$

The distributed source term in the SAAF equations is given by:

$$\sum_{g=1}^G \sum_{d=1}^{N_d} w_d \sum_{\ell=0}^{L_s} \sum_{k=-\ell}^{\ell} \left(Q_{g,\ell,k}^* Y_{\ell,k}(\vec{\Omega}_d), \Psi_{g,d} \right)_{\mathcal{D}}, \quad (150)$$

where L_s is the spherical harmonics expansion order of the source and $Q_{g,\ell,k}^*$ is the angular, adjoint source moment.

For derivation of mathematical adjoint kernels in curvilinear geometry, it is important to note that the integration by parts performed in Eq. (98) introduces an *extra term* $(\Psi^*, \mathbb{E}_2 \Psi)$, which can be merged into Eq. (95) as

$$((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, (\mathbb{E} + \mathbb{E}_2) \Psi). \quad (151)$$

It can then be expanded as

$$\sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{r} \sum_{d'=1}^{N_d} (\beta_{d,d'} + 2\delta_{d,d'} \mu_d) \Psi_{g,d'} \right)_{\mathcal{D}}. \quad (152)$$

We apply the math adjoint transform:

$$\begin{aligned} & \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^* - \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{r} \sum_{d'=1}^{N_d} (\beta_{d_{\text{reverse}},d'_{\text{reverse}}} - 2\delta_{d,d'} \mu_d) \Psi_{g,d'} \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d'=1}^{N_d} w_{d'} \left(\sum_{d=1}^{N_d} \frac{w_d}{w_{d'}} (\beta_{d_{\text{reverse}},d'_{\text{reverse}}} - 2\delta_{d,d'} \mu_d) (\Psi_{g,d}^* - \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*), \frac{1}{r} \Psi_{g,d'} \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d'=1}^{N_d} w_d \left(\sum_{d'=1}^{N_d} \frac{w_{d'}}{w_d} (\beta_{d'_{\text{reverse}},d_{\text{reverse}}} - 2\delta_{d,d'} \mu_d) (\Psi_{g,d'}^* - \tau_g \vec{\Omega}_{d'} \cdot \vec{\nabla} \Psi_{g,d'}^*), \frac{1}{r} \Psi_{g,d} \right)_{\mathcal{D}}. \end{aligned} \quad (153)$$

When there are dummy directions, which are denoted by a set Dummy (the rest is denoted as Non-dummy), because angular fluxes in these directions do not contribute to the angular flux moments, extra terms are needed for the special treatment of scattering and fission. The scattering term is

$$- \sum_{g=1}^G \sum_{g'=1}^G \sum_{d \in \text{Non-dummy}} w_d \left(\sum_{\ell=0}^L \sum_{k=-\ell}^{\ell} \Sigma_{s,\ell}^{g \rightarrow g'} R_{\ell,k}(-\vec{\Omega}_d) \sum_{d' \in \text{Dummy}} w_{\text{dummy}} Y_{\ell,k}(-\vec{\Omega}_{d'}) (-\tau_{g'} \vec{\Omega}_{d'} \cdot \vec{\nabla} \Psi_{g',d'}^* + \Psi_{g',d'}^*), \Psi_{g,d} \right)_{\mathcal{D}}.$$

If we define

$$\Phi_{g',\ell,k}^{\text{Dummy}} = \sum_{m \in \text{Dummy}} w_{\text{dummy}} Y_{\ell,k}(-\vec{\Omega}_m) (-\tau_{g'} \vec{\Omega}_m \cdot \vec{\nabla} \Psi_{g',m}^* + \Psi_{g',m}^*), \quad (154)$$

Then the term can be simplified into

$$- \sum_{g=1}^G \sum_{g'=1}^G \sum_{d \in \text{Non-dummy}} w_d \left(\sum_{\ell=0}^L \sum_{k=-\ell}^{\ell} \Sigma_{s,\ell}^{g \rightarrow g'} R_{\ell,k}(-\vec{\Omega}_d) \Phi_{g',\ell,k}^{\text{Dummy}}, \Psi_{g,d} \right)_{\mathcal{D}}. \quad (155)$$

This dummy part represents a coupling of dummy angular fluxes to the normal fluxes in the math-adjoint equation. Similarly, we can have the dummy part of the fission bilinear form

$$- \sum_{g=1}^G \sum_{d \in \text{Non-dummy}} w_d \left(\frac{1}{4\pi} \sum_{g'=1}^G \chi_{g'} \Phi_{g',0,0}^{\text{Dummy}}, \frac{1}{k} \nu \Sigma_{f,g} \Psi_{g,d} \right)_{\mathcal{D}}. \quad (156)$$

3.4.2.9 Summary of MOOSE objects

The MOOSE objects used by SAAF-CFEM-SN are listed in Table 5, Table 6, Table 7, Table 8 and Table 9.

Table 5 SAAF-CFEM-SN kernels.

Kernel	Type	Jacobian
$\left(\vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} - (1 - \tau_g \Sigma_{t,g}) \Psi_{g,d}\right)_D$	SAAFStreaming	Y
$\left(\Psi_{g,d}^*, \Sigma_{t,g} \Psi_{g,d}\right)_D$	Reaction	Y
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \sum_{\ell=0}^L \sum_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g,\ell,m} R_{\ell,m}(\vec{\Omega}_d)\right)_D$	SAAFScattering	Y
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} S_{g,\ell,m} R_{\ell,m}(\vec{\Omega}_d)\right)_D$	SAAFTotalScattering	N
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \Sigma_{s,g' \rightarrow g} \sum_{d'=1}^{N_d} f_{g' \rightarrow g,d' \rightarrow d} \Psi_{g',d'}\right)_D$	SAAFPFScattering	N
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, Q_g^{\text{ext}}(\vec{\Omega}_d)\right)_D$	SAAFSource	N/A
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{4\pi} \frac{\bar{\chi}_g}{k} F\right)_D$	SAAFFission	N
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{4\pi} \frac{1}{k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i\right)_D$	SAAFDelayedNeutron	N
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{4\pi} S_{\text{BB},g}\right)_D$	SAAFFPlanck	N
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{v_g} \frac{\partial}{\partial t} \Psi_{g,d}\right)_D$	SAAFTimeDerivative	Y
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{1}{r} \sum_{d'=1}^{N_d} \beta_{d,d'} \Psi_{g,d'}\right)_D + \left(\tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{2}{r} \mu_d \Psi_{g,d}\right)_D$	SAAFStreamingRSpherical	Y
$\left(\Psi_{g,d}^* + \tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, -\frac{1}{r} \sum_{d'=1}^{N_d} \alpha_{d,d'} \Psi_{g,d'}\right)_D - \left(\Psi_{g,d}^*, \frac{\Omega_{r,d}}{r} \Psi_{g,d}\right)_D$	SAAFStreamingRZCylindrical	Y
$\left(\tau_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} - (1 - \Sigma_{t,g} \tau_g) \Psi_{g,d}, \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*\right)_D$	SAAFAdjunctStreaming	Y
$\left(\Psi_{g,d} R_{\ell,k}(-\vec{\Omega}_d), (-1)^\ell \sum_{s,\ell}^{g \rightarrow g'} \left[\Phi_{g',\ell,k}^* + \tau_{g'} \vec{\nabla} \cdot \vec{X}_{g',\ell,m}^*\right]\right)_D$	SAAFAdjunctScattering	D ^a
$\left(\frac{1}{4\pi} \frac{\nu \Sigma_{f,g}}{k} F^*, \Psi_{g,d}\right)_D$	SAAFAdjunctFission	N
$\left(\Psi_{g,d}, Q_{g,\ell,k}^* R_{\ell,k}(\vec{\Omega}_d)\right)_D$	SAAFAdjunctSource	N/A
$\left(\sum_{d'=0}^{N_d} \frac{w_{d'}}{w_d} (\beta_{d',\text{reverse},d,\text{reverse}} - 2\delta_{d,d'} \mu_d) (\Psi_{g,d'}^* - \tau_g \vec{\Omega}_{d'} \cdot \vec{\nabla} \Psi_{g,d'}^*), \frac{1}{r} \Psi_{g,d}\right)_D$	SAAFStreamingRSphericalAdjunct	Y
$\left(\sum_{\ell=0}^L \sum_{k=-\ell}^{\ell} \sum_{s,\ell}^{g \rightarrow g'} R_{\ell,k}(-\vec{\Omega}_d) \Phi_{g',\ell,k}^{\text{Dummy}}, \Psi_{g,d}\right)_D$	SAAFScatteringRSphericalAdjunct	N
$\left(\frac{1}{4\pi} \sum_{g'=1}^G \chi_{g'} \Phi_{g',0,0}^{\text{Dummy}}, \frac{1}{k} \nu \Sigma_{f,g} \Psi_{g,d}\right)_D$	SAAFFissionRSphericalAdjunct	N

^a D means that only the diagonal part (within-group) is assembled.

Table 6 SAAF-CFEM-SN kernels (continued).

Type	Condition	Number of kernels
SAAFStreaming	$\Sigma_{t,g}^a$	$N_d \times G$
Reaction	$\Sigma_{t,g}$	$N_d \times G$
SAAFScattering	$\Sigma_{s,\ell}^{g' \rightarrow g}, \ell = 0, \dots, L$	$N_d \times \text{NNZ}^b$
SAAFTotalScattering	$S_{g,\ell,m}$	$N_d \times G$
SAAFPFScattering	$f_{g' \rightarrow g}(\mu_0)$	$N_d \times \text{NNZ}$
SAAFSource	Q_g^{ext}	$N_s^c \times N_d \times G$
SAAFFission	$\bar{\chi}_g$	$N_d \times G$
SAAFDelayedNeutron	$\chi_{d,g,i}, i = 1, \dots, I$	$N_d \times G$
SAAFPPlanck	$\Sigma_{t,g}$	$N_d \times G$
SAAFTimeDerivative	v_g	$N_d \times G$
SAAFStreamingRSpherical	R-spherical, $\Sigma_{t,g}$	$N_d \times G$
SAAFStreamingRZCylindrical	RZ-cylindrical, $\Sigma_{t,g}$	$N_d \times G$
SAAFAdjointStreaming	$\Sigma_{t,g}$	$N_d \times G$
SAAFAdjointScattering	$\Sigma_{s,\ell}^{g' \rightarrow g}, \ell = 0, \dots, L$	$N_d \times \text{NNZ}$
SAAFAdjointFission	$\bar{\chi}_g$	$N_d \times G$
SAAFAdjointSource	$Q_{g,l,k}^*$	$N_d \times G \times \mathcal{N}(L_s)$
SAAFStreamingRSphericalAdjoint	R-spherical, $\Sigma_{t,g}$	$N_d \times G$
SAAFScatteringRSphericalAdjoint	R-spherical, $\Sigma_{t,g}, N_{\text{dummy}} \neq 0$	$N_{\text{non-dummy}} \times G$
SAAFFissionRSphericalAdjoint	R-spherical, $\Sigma_{t,g}, N_{\text{dummy}} \neq 0$	$N_{\text{non-dummy}} \times G$

^a Means where $\Sigma_{t,g}$ is available.

^b Number of non-zeros in the scattering matrix, i.e. all (g', g) s whose

$$\Sigma_s^{g' \rightarrow g} = \left\{ \Sigma_{s,\ell}^{g' \rightarrow g}, \forall \ell = 0, \dots, L \right\} \neq 0.$$

^c Number of external source functions.

Table 7 SAAF-CFEM-SN boundary conditions.

Type	Equation	Number of BCs	Jacobian
SNVacuum	$\left(\Psi_{g,n}^*, \vec{\Omega} \cdot \vec{n}_b \bar{\Psi}_{g,n} \right)_{\partial \mathcal{D}_s}$	$N_s^a \times N_d \times G$	Y
SNReflecting	$\left(\Psi_{g,n}^*, \vec{\Omega} \cdot \vec{n}_b \bar{\Psi}_{g,n} \right)_{\partial \mathcal{D}_r}$	$N_d \times G$	Y
SNWhite	$\sum_{\vec{\Omega} \cdot \vec{n}_b > 0} w_n \left(\Psi_{g,n}^*, \vec{\Omega} \cdot \vec{n}_b \bar{\Psi}_{g,n} \right)_{\partial \mathcal{D}_w}$	$N_d \times G$	Y

^a N_s is the number of surface source transport functions. If there are no source functions, it is equal to 1.

Table 8 SAAF-CFEM-SN auxiliary kernels.

Name	Equation	Type	Number of kernels	Condition
Angular flux moments ($\Phi_{g,\ell,m}$)	Eq. (123)	FluxMoment	$\mathcal{N}(L_{\text{input}})^a \times G$	\mathcal{D}
Fission source (F)	Eq. (138)	MaterialFissionSource	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$
DNP ($C_i, i = 1, \dots, I$)	Eq. (140a)	DelayedNeutronPrecursor	I	$\nu \Sigma_{f,g}, g = 1, \dots, G$
Adjoint fission source F_1^*	Eq. (148a)	AdjointFissionSource	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$

^a L_{input} is a user input. It thus can be used to truncate the scattering order when $L_{\text{input}} < L$ or ask Rattlesnake to evaluate more flux moments when $L_{\text{input}} > L$.

Table 9 SAAF-CFEM-SN auxiliary materials.

Name	Equation	Type	Number of materials	Condition
τ_g	Eq. (89)	SAAFTau	G	$\Sigma_{t,g}$
$S_{g,\ell,m}$	Eq. (124)	SsrcMaterial	$\mathcal{N}(L) \times G$	$\Sigma_{s,\ell}^{g' \rightarrow g}, g' = 1, \dots, G; g = 1, \dots, G$
F	Eq. (138)	FsrcMaterial	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$
C_i	Eq. (140a)	DNPMaterial	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$
J_{out}	Eq. (133)	BoundaryCurrent	G	$\partial \mathcal{D}_w$
F_1^*	Eq. (148a)	AdjointFsrcMaterial	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$

3.4.3 SAAF-CFEM-PN

The angular discretization agnostic weak form is given in Section 3.3.2. Spherical harmonics definitions and notation can be found in Appendix B. In particular, matrix definitions are detailed in Section B.2. We further define:

$$\Sigma_{s,g' \rightarrow g} = \text{diag} \left\{ \Sigma_{s,\ell}^{g' \rightarrow g}, m = -\ell, \dots, \ell; \ell = 0, \dots, N \right\}, \quad (157)$$

$$\nu \Sigma_{f,g} = \text{diag} \left\{ \nu \Sigma_{f,g} \delta_{\ell,0}, m = -\ell, \dots, \ell; \ell = 0, \dots, N \right\}. \quad (158)$$

We also define the vector of moments for the external source :

$$\mathbf{Q}_g \equiv \int_S Q_g^{\text{ext}} \mathbf{R} d\Omega \quad (159)$$

Furthermore, a globally conservative void treatment for the SAAF-PN method was developed based on [53]. The weak formulation is very similar to that of the SAAF with the void treatment, initially developed for S_N methods in [46]. The difference lies in the definition of τ . We refer to this method as SAAF-CLS and it is constructed as a hybrid method using the SAAF scheme in the non-void regions and the Conservative Least-Squares (CLS) scheme in the void or near-void regions. The latter is a modified LS method which preserves global conservation by adding a non-symmetric term to the weak form [53]. For the SAAF-CLS-PN method, we define the following material property, instead of Eq. (89) (we add a hat to the notation to emphasize that they are different):

$$\hat{\tau}_g^{-1} \equiv \begin{cases} c_g & , \quad \vec{x} \in \mathcal{D}_0 \\ \Sigma_{t,g} & , \quad \vec{x} \in \mathcal{D}_1 \end{cases} \quad (160)$$

where c_g is a constant, \mathcal{D}_0 is the void or near-void region and $\mathcal{D}_1 = \mathcal{D} \setminus \mathcal{D}_0$.

3.4.3.1 Steady-state CTE

Streaming kernel:

$$\begin{aligned}
& (\mathbb{L}_1 \Psi^*, (\tau \mathbb{L}_1 - \mathbb{I} + \tau \mathbb{L}_2) \Psi) \\
&= \sum_{g=1}^G \int_S d\Omega \left(\left(\vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \hat{\tau}_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g \right)_{\mathcal{D}} - \left((1 - \hat{\tau}_g \Sigma_{t,g}) \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \Psi_g \right)_{\mathcal{D}} \right) \\
&= \sum_{g=1}^G \left(\left(\vec{\nabla} \Phi_g^*, \hat{\tau}_g \vec{H} \cdot \vec{\nabla} \Phi_g \right)_{\mathcal{D}} - \left((1 - \hat{\tau}_g \Sigma_{t,g}) \vec{D} \cdot \vec{\nabla} \Phi_g^*, \Phi_g \right)_{\mathcal{D}} \right) \\
&= \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left(\left(\vec{\nabla} \Phi_{g,\ell,m}^*, \hat{\tau}_g \vec{H}_{p,p'} \cdot \vec{\nabla} \Phi_{g,\ell',m'} \right)_{\mathcal{D}} - \left((1 - \hat{\tau}_g \Sigma_{t,g}) \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, \Phi_{g,\ell',m'} \right)_{\mathcal{D}} \right). \quad (161)
\end{aligned}$$

Collision kernel:

$$(\Psi^*, \mathbb{L}_2 \Psi) = \sum_{g=1}^G \left(\Phi_g^*, \Sigma_{t,g} \mathbf{P} \Phi_g \right)_{\mathcal{D}} = \sum_{g=1}^G \sum_{p=1}^{\mathcal{N}} \left(\Phi_{g,\ell,m}^*, \Sigma_{t,g} P_{p,p} \Phi_{g,\ell,m} \right)_{\mathcal{D}}. \quad (162)$$

Scattering kernel:

$$\begin{aligned}
((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{S} \Psi) &= \sum_{g=1}^G \int_S d\Omega \left(\Psi_g^* + \hat{\tau}_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \sum_{g'=1}^G \sum_{\ell=0}^L \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g',\ell,m} R_{\ell,m} \right)_{\mathcal{D}} \\
&= \sum_{g,g'=1}^G \left(\mathbf{P} \Phi_g^* + \hat{\tau}_g \vec{D} \cdot \vec{\nabla} \Phi_g^*, \Sigma_{s,g' \rightarrow g} \Phi_{g'} \right)_{\mathcal{D}} \\
&= \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left(P_{p',p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, \sum_{g'=1}^G \Sigma_{s,\ell'}^{g' \rightarrow g} \Phi_{g',\ell',m'} \right)_{\mathcal{D}}. \tag{163}
\end{aligned}$$

Source kernel:

$$\begin{aligned}
((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbf{Q}^{\text{ext}}) &= \sum_{g=1}^G \left(\mathbf{P} \Phi_g^* + \hat{\tau}_g \vec{D} \cdot \vec{\nabla} \Phi_g^*, \mathbf{Q}_g \right)_{\mathcal{D}} \\
&= \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left(P_{p',p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, Q_{g,\ell',m'} \right)_{\mathcal{D}}. \tag{164}
\end{aligned}$$

3.4.3.2 Boundary conditions

Vacuum boundary conditions:

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi^*, \Psi^{\text{inc}} \rangle_{\partial \mathcal{D}_s}^- = \sum_{g=1}^G \left(\Phi_{g'}^*, \mathbf{L}^+ \Phi_g \right)_{\partial \mathcal{D}_s} - \left(\Phi_{g'}^*, \mathbf{L}^- \Phi_g^{\text{inc}} \right)_{\partial \mathcal{D}_s} \tag{165}$$

$$= \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left(\Phi_{g,\ell,m}^*, L_{p,p'}^+ \Phi_{g,\ell',m'} \right)_{\partial \mathcal{D}_s} - \left(\Phi_{g,\ell,m}^*, L_{p,p'}^- \Phi_{g,\ell',m'}^{\text{inc}} \right)_{\partial \mathcal{D}_s}. \tag{166}$$

Several options are available for \mathbf{L}^+ and \mathbf{L}^- , as described in Section B.3.

Reflecting boundary conditions:

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \langle \Psi^*, \Psi_r \rangle_{\partial \mathcal{D}_r}^- = \sum_{g=1}^G \left(\Phi_{g'}^*, \mathbf{N}^+ \Phi_g \right)_{\partial \mathcal{D}_r} = \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left(\Phi_{g,\ell,m}^*, N_{p,p'}^+ \Phi_{g,\ell',m'} \right)_{\partial \mathcal{D}_r}. \tag{167}$$

Several options are available for \mathbf{N}^+ , as described in Section B.3.

White boundary conditions are currently not supported.

3.4.3.3 Transient kernel

Time-derivative kernel:

$$\begin{aligned}
\left((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) &= \sum_{g=1}^G \left(\mathbf{P} \Phi_g^* + \hat{\tau}_g \vec{D} \cdot \vec{\nabla} \Phi_g^*, \frac{\partial}{\partial t} \left(\frac{\Phi_g}{v_g} \right) \right)_{\mathcal{D}} \\
&= \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left(P_{p',p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, \frac{\partial}{\partial t} \left(\frac{\Phi_{g,\ell',m'}}{v_g} \right) \right)_{\mathcal{D}}. \tag{168}
\end{aligned}$$

If we assume $v_g, g = 1, \dots, G$ is time independent, the kernel can be simplified into

$$\sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left(P_{p',p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, \frac{1}{v_g} \frac{\partial \Phi_{g,\ell',m'}}{\partial t} \right)_{\mathcal{D}}. \tag{169}$$

3.4.3.4 Neutron

Prompt fission kernel:

$$\begin{aligned}
 ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{F} \Psi) &= \sum_{g=1}^G \int_S d\Omega \left(\Psi_g^* + \hat{\tau}_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{1}{4\pi} \frac{\chi_{p,g}}{k} \sum_{g'=1}^G \nu \Sigma_{f,g'} \int_S \Psi_{g'}(\vec{\Omega}') d\Omega' \right)_{\mathcal{D}} \\
 &= \sum_{g=1}^G \left(\mathbf{P} \Phi_g^* + \hat{\tau}_g \vec{D} \cdot \vec{\nabla} \Phi_g^*, \frac{1}{4\pi} \frac{\chi_{p,g}}{k} F \right)_{\mathcal{D}}, \\
 &= \sum_{g=1}^G \sum_{p=1}^{\mathcal{N}} \left(P_{0,p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{0,p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, \frac{1}{4\pi} \frac{\chi_{p,g}}{k} F \right)_{\mathcal{D}}, \tag{170}
 \end{aligned}$$

where the fission source F is defined by Eq. (138).

Delayed neutron kernel:

$$\begin{aligned}
 ((\mathbb{I} + \tau \mathbb{L}_1) \Phi^*, \mathbf{DC}) &= \sum_{g=1}^G \int_S d\Omega \left(\Psi_g^* + \hat{\tau}_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{1}{4\pi} \frac{1}{k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i \right)_{\mathcal{D}} \\
 &= \sum_{g=1}^G \left(\mathbf{P} \Phi_g^* + \hat{\tau}_g \vec{D} \cdot \vec{\nabla} \Phi_g^*, \frac{1}{4\pi} \frac{1}{k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i \right)_{\mathcal{D}}, \\
 &= \sum_{g=1}^G \sum_{p=1}^{\mathcal{N}} \left(P_{0,p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{0,p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, \frac{1}{4\pi} \frac{1}{k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i \right)_{\mathcal{D}}, \tag{171}
 \end{aligned}$$

where the DNP concentrations C_i are evaluated using Eq. (140a).

3.4.3.5 Thermal radiation

This weak form currently does not support thermal radiation.

3.4.3.6 R-spherical coordinate

This weak form currently does not support R-spherical coordinate.

3.4.3.7 RZ-cylindrical coordinate

The implementation of the SAAF-PN method in cylindrical geometries with axial symmetry relies on the ability given by MOOSE to redefine the scalar products introduced by Eq. (66) to include the volumetric term $2\pi r$ as well as to efficiently implement the additional cylindrical streaming terms given by Eq. (95) using a P_N expansion. As a reminder, Eq. (95) is

$$(\mathbb{L}_1 \Phi^*, \tau(\mathbf{E} + \mathbf{E}_2) \Phi) + (\Phi^*, \mathbf{E} \Phi), \tag{172}$$

which, in cylindrical coordinates with axial symmetry, becomes (see Table 52):

$$-\left(\mathbb{I} + \tau \mathbb{L}_1 \Phi^*, \frac{\Omega_\theta}{r} \frac{\partial \Phi}{\partial \omega} \right) - \left(\Phi^*, \frac{\Omega_r}{r} \Phi \right). \tag{173}$$

We now define a few additional matrices and give the expression of the corresponding kernels. In the RZ case, the direction $\vec{\Omega}$ is characterized by the cosine μ of its polar angle and by its azimuthal angle ω , which are then also the

angles used in the definition of the spherical harmonics (see Appendix B). Using the notation from Appendix C, $\vec{\Omega} \equiv (\Omega_z, \Omega_r, \Omega_\theta)^T \equiv (\mu, \sqrt{1-\mu^2} \cos \omega, \sqrt{1-\mu^2} \sin \omega)^T$, we define the operator \mathcal{P} using the following identities:

$$\frac{\partial \mathbf{R}}{\partial \omega} \equiv \mathcal{P} \mathbf{R} \quad \text{with} \quad \frac{\partial R_\ell^m}{\partial \omega} = -m R_\ell^{-m}. \quad (174)$$

Then, the following matrices are being defined:

$$\mathbf{D}_\theta \equiv \int_S \Omega_\theta \mathbf{R} (\mathcal{P} \mathbf{R})^T d\Omega \quad , \quad \mathbf{D}_r \equiv \int_S \Omega_r \mathbf{R} \mathbf{R}^T d\Omega, \quad (175)$$

$$\mathbf{H}_{r,\theta} \equiv \int_S \Omega_r \Omega_\theta \mathbf{R} (\mathcal{P} \mathbf{R})^T d\Omega \quad , \quad \mathbf{H}_{z,\theta} \equiv \int_S \Omega_z \Omega_\theta \mathbf{R} (\mathcal{P} \mathbf{R})^T d\Omega, \quad (176)$$

$$\vec{\mathbf{H}}_\theta \equiv (\mathbf{H}_{r,\theta}, \mathbf{H}_{z,\theta})^T. \quad (177)$$

The additional kernels are thus given by:

$$\begin{aligned} & - \sum_{g=1}^G \left(\left(\frac{\partial \Phi_g^*}{\partial r}, \frac{\hat{\tau}_g}{r} \mathbf{H}_{r,\theta} \Phi_g \right)_{\mathcal{D}} + \left(\frac{\partial \Phi_g^*}{\partial z}, \frac{\hat{\tau}_g}{r} \mathbf{H}_{z,\theta} \Phi_g \right)_{\mathcal{D}} + \left(\Phi_{g'}^* \frac{1}{r} (\mathbf{D}_r + \mathbf{D}_\theta) \Phi_g \right)_{\mathcal{D}} \right) \\ & = - \sum_{g=1}^G \left(\left(\vec{\nabla} \Phi_{g'}^*, \frac{\hat{\tau}_g}{r} \vec{\mathbf{H}}_\theta \Phi_g \right)_{\mathcal{D}} + \left(\Phi_{g'}^* \frac{1}{r} (\mathbf{D}_r + \mathbf{D}_\theta) \Phi_g \right)_{\mathcal{D}} \right) \\ & = - \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left(\left(\vec{\nabla} \Phi_{g,\ell,m}^*, \frac{\hat{\tau}_g}{r} \vec{H}_{\theta,p,p'} \Phi_{g,\ell',m'} \right)_{\mathcal{D}} + \left(\Phi_{g,\ell,m'}^* \frac{1}{r} (D_{r,p,p'} + D_{\theta,p,p'}) \Phi_{g,\ell',m'} \right)_{\mathcal{D}} \right). \quad (178) \end{aligned}$$

3.4.3.8 Mathematical adjoint

This weak form currently does not support math adjoint.

3.4.3.9 Summary of MOOSE objects

The MOOSE objects used by SAAF-CFEM-PN are listed in Table 10, Table 11, Table 12, Table 13 and Table 14.

Table 10 SAAF-CFEM-PN kernels.

Kernel	Type	Jacobian
$\left(\vec{\nabla} \Phi_{g,\ell,m}^* \cdot \hat{\tau}_g \vec{H}_{p,p} \cdot \vec{\nabla} \Phi_{g,\ell,m} \right)_{\mathcal{D}}$	SAAFPNStreaming ^a	Y
$\left(\vec{\nabla} \Phi_{g,\ell,m}^* \cdot \hat{\tau}_g \vec{H}_{p,p'} \cdot \vec{\nabla} \Phi_{g,\ell',m'} \right)_{\mathcal{D}}$	SAAFPNCouplingStreaming ^b	Y
$\left((1 - \hat{\tau}_g \Sigma_{t,g}) \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^* \cdot \Phi_{g,\ell',m'} \right)_{\mathcal{D}}$	SAAFPNStreamingAsymmetric	Y
$\left(\Phi_{g,\ell,m}^* \cdot \Sigma_{t,g} P_{p,p} \Phi_{g,\ell,m} \right)_{\mathcal{D}}$	SAAFPNRemoval	Y
$\left(P_{p',p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^* \cdot \sum_{g'=1}^G \Sigma_{s,\ell'}^{g' \rightarrow g} \Phi_{g',\ell',m'} \right)_{\mathcal{D}}$	SAAFPNScattering	Y
$\left(P_{p',p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^* \cdot Q_{g,\ell',m'} \right)_{\mathcal{D}}$	SAAFPNSource	N/A
$\left(P_{0,p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{0,p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^* \cdot \frac{1}{4\pi} \frac{\chi_{p,g}}{k} F \right)_{\mathcal{D}}$	SAAFPNFission	Y
$\left(P_{0,p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{0,p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^* \cdot \frac{1}{4\pi} \frac{1}{k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i \right)_{\mathcal{D}}$	SAAFPNDelayedNeutronMat	N
$\left(P_{p',p} \Phi_{g,\ell,m}^* + \hat{\tau}_g \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^* \cdot \frac{1}{v_g} \frac{\partial \Phi_{g,\ell',m'}}{\partial t} \right)_{\mathcal{D}}$	SAAFPNTimeDerivative	Y
$\left(\vec{\nabla} \Phi_{g,\ell,m}^* \cdot \frac{\hat{\tau}_g}{r} \vec{H}_{\theta,p,p'} \Phi_{g,\ell',m'} \right)_{\mathcal{D}}$	SAAFPNStreamingRZCylindrical ^a	Y
$\left(\vec{\nabla} \Phi_{g,\ell,m}^* \cdot \frac{\hat{\tau}_g}{r} \vec{H}_{\theta,p,p'} \Phi_{g,\ell',m'} \right)_{\mathcal{D}}$	SAAFPNCouplingStreamingRZCylindrical ^b	Y
$\left(\Phi_{g,\ell,m}^* \cdot \frac{1}{r} (D_{r,p,p'} + D_{\theta,p,p'}) \Phi_{g,\ell',m'} \right)_{\mathcal{D}}$		

^a Only the terms such that $p = p'$ (i.e. $(\ell, m) = (\ell', m')$) are included in this kernel.

^b Only the terms such that $p \neq p'$ (i.e. $(\ell, m) \neq (\ell', m')$) are included in this kernel.

Table 11 SAAF-CFEM-PN kernels (continued).

Type	Condition	Number of kernels
SAAFPNStreaming	$\Sigma_{t,g}$ ^a	$\mathcal{N} \times G$
SAAFPNCouplingStreaming	$\Sigma_{t,g}$	$\mathcal{N}_1^b \times G$
SAAFPNStreamingAsymmetric	void treatment, $\Sigma_{t,g}$	$\mathcal{N}_2^c \times G$
SAAFPNRemoval	$\Sigma_{t,g}$	$\mathcal{N} \times G$
SAAFPNScattering	$S_{g,\ell,m}$	$\mathcal{N} \times G \times (L + 1)$
SAAFPNSource	Q_g^{ext}	$\mathcal{N}^d \times G$
SAAFPNFission	$\tilde{\chi}_g$	$\mathcal{N} \times G$
SAAFPNDelayedNeutronMat	$\chi_{d,g,i}, i = 1, \dots, I$	$\mathcal{N} \times G$
SAAFPNTimeDerivative	v_g	$\mathcal{N} \times G$
SAAFPNStreamingRZCylindrical	RZ-cylindrical, $\Sigma_{t,g}$	$\mathcal{N} \times G$
SAAFPNCouplingStreamingRZCylindrical	RZ-cylindrical, $\Sigma_{t,g}$	$\mathcal{N}_4^e \times G$

^a Means where $\Sigma_{t,g}$ is available.

^b Number of non-zero elements in the set $\{\mathbf{H}_{p,p'}; p, p' \in \{1, \dots, \mathcal{N}\}, p \neq p'\}$.

^c Number of non-zero elements in the set $\{\vec{D}_{p,p'}; p, p' \in \{1, \dots, \mathcal{N}\}, p \neq p'\}$.

^d At the moment, only isotropic sources are available.

^e Number of non-zero elements in the set $\{(p, p') \in \{1, \dots, \mathcal{N}\}^2; \|\vec{H}_{\theta,p,p'}\| \neq 0 \text{ or } (\mathbf{D}_r + \mathbf{D}_{\theta})_{p,p'} \neq 0\}$.

Table 12 SAAF-CFEM-PN boundary conditions.

Equation	Type	Number of kernels	Jacobian
$\left\langle \Phi_{g,\ell,m}, \sum_{p'} \mathbf{L}_{p,p'}^+ \Phi_{g,\ell',m'} - \mathbf{L}_{p,p'}^- \Phi_{g,\ell',m'}^{\text{inc}} \right\rangle_{\partial \mathcal{D}_s}$	SAAFPNVacuumBC	$\mathcal{N} \times G \times N_s^a$	Y
$\left\langle \Phi_{g,\ell,m}, \sum_{p'} \mathbf{N}_{p,p'}^+ \Phi_{g,\ell',m'} \right\rangle_{\partial \mathcal{D}_r}$	SAAFPNReflectingBC	$\mathcal{N} \times G$	Y

^a N_s is the number of source transport functions. If there are no source functions, it is equal to 1.

Table 13 SAAF-CFEM-PN auxiliary kernels.

Name	Equation	Type	Number of kernels	Condition
Fission source (F)	Eq. (138)	MaterialFissionSource	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$
DNP ($C_i, i = 1, \dots, I$)	Eq. (140a)	DelayedNeutronPrecursor	I	$\nu \Sigma_{f,g}, g = 1, \dots, G$

Table 14 SAAF-CFEM-PN auxiliary materials.

Name	Type	Number of kernels	Condition
$\hat{\tau}_g^{-1}$	SAAFCLSTauInv	G	\mathcal{D}_0
$S_{g,\ell,m}$	SsrcMaterial	$\mathcal{N}(L) \times G$	$\Sigma_{s,\ell}^{g' \rightarrow g}, g' = 1, \dots, G; g = 1, \dots, G$
F	FsrcMaterial	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$
C_i	DNPMaterial	I	$\nu \Sigma_{f,g}, g = 1, \dots, G$

Table 15 SAAF-CFEM-PN userobjects. Refer to Section B.3 for the matrix definitions.

Content	Type
\mathbf{L}^+	LMatrixOnSide
\mathbf{L}^-	LMinusMatrixOnSide
\mathbf{N}^+	NMatrixOnSide

3.4.4 LS-CFEM-SN

We will expand all the kernels in order to make a direct correspondence to the code.

3.4.4.1 Steady-state CTE

Streaming kernel:

$$\begin{aligned} (\mathbf{W}\mathbb{L}_1\Psi^*, \mathbb{L}\Psi) &= \sum_{g=1}^G \int_S d\Omega \left(W_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \vec{\Omega} \cdot \vec{\nabla} \Psi_g + \Sigma_{t,g} \Psi_g \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} \left(W_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} + \Sigma_{t,g} \Psi_{g,d} \right)_{\mathcal{D}}. \end{aligned} \quad (179)$$

Collision kernel:

$$\begin{aligned} (\mathbf{W}\mathbb{L}_2\Psi^*, \mathbb{L}\Psi) &= \sum_{g=1}^G \int_S d\Omega \left(W_g \Sigma_{t,g} \Psi_g^*, \vec{\Omega} \cdot \vec{\nabla} \Psi_g + \Sigma_{t,g} \Psi_g \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} \left(W_g \Sigma_{t,g} \Psi_{g,d}^*, \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} + \Sigma_{t,g} \Psi_{g,d} \right)_{\mathcal{D}}. \end{aligned} \quad (180)$$

Scattering kernel:

$$\begin{aligned} (\mathbb{L}\Psi^*, \mathbf{W}\mathbb{S}\Psi) &= \sum_{g=1}^G \int_S d\Omega \left(\Sigma_{t,g} \Psi_g^* + \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, W_g \sum_{g'=1}^G \sum_{\ell=0}^L \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g,\ell,m} R_{\ell,m} \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \sum_{g'=1}^G \left(\Sigma_{t,g} \Psi_{g,d}^* + \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, W_g \sum_{\ell=0}^L \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g',\ell,m} R_{\ell,m}(\vec{\Omega}_d) \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Sigma_{t,g} \Psi_{g,d}^* + \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, W_g \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} S_{g,\ell,m} R_{\ell,m}(\vec{\Omega}_d) \right)_{\mathcal{D}}. \end{aligned} \quad (181)$$

with $\Phi_{g',\ell,m}$ and $S_{g,\ell,m}$ being respectively defined by Eq. (123) and Eq. (124).

External source kernel:

$$\begin{aligned} (\mathbb{L}\Psi^*, \mathbf{W}\mathbf{Q}^{\text{ext}}) &= \sum_{g=1}^G \int_S d\Omega \left(\Sigma_{t,g} \Psi_g^* + \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, W_g Q_g^{\text{ext}} \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Sigma_{t,g} \Psi_{g,d}^* + \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, W_g Q_g^{\text{ext}}(\vec{\Omega}_d) \right)_{\mathcal{D}}. \end{aligned} \quad (182)$$

3.4.4.2 Boundary conditions

This scheme supports two kind of boundary conditions: nodal (i.e. strongly imposed) and integrated (weakly imposed) boundary conditions.

Nodal vacuum boundary conditions: this imposes the following condition at each corresponding boundary node and each direction index n such that $\vec{\Omega}_d \cdot \vec{n}_b < 0$

$$\Psi_{g,d} = \Psi_{g,d}^{\text{inc}}. \quad (183)$$

Nodal reflecting boundary conditions: this imposes the following condition at each corresponding boundary node and each direction index n such that $\vec{\Omega}_d \cdot \vec{n}_b < 0$

$$\Psi_{g,d} = \Psi_{g,d,\text{reflecting}} \quad (184)$$

with $\vec{\Omega}_{d,\text{reflecting}} = \vec{\Omega}_d - 2(\vec{\Omega}_d \cdot \vec{n}_b)\vec{n}_b$.

The other boundary conditions are weakly imposed and in particular involve the variable c whose different possible values are summarized in Table 3.

Integrated vacuum boundary conditions:

$$\left\langle \mathbb{C}\Psi^*, \Psi - \Psi^{\text{inc}} \right\rangle_{\partial\mathcal{D}_s}^- = \sum_{g=1}^G \sum_{\vec{\Omega}_d \cdot \vec{n}_b < 0} w_d \left(\Psi_{g,d}^*, c_g(\vec{\Omega}_d) |\vec{\Omega}_d \cdot \vec{n}_b| (\Psi_{g,d} - \Psi_{g,d}^{\text{inc}}) \right)_{\partial\mathcal{D}_s}, \quad (185)$$

Integrated reflecting boundary conditions:

$$\left\langle \mathbb{C}\Psi^*, \Psi - \Psi(\vec{\Omega}_r) \right\rangle_{\partial\mathcal{D}_r}^- = \sum_{g=1}^G \sum_{\vec{\Omega}_d \cdot \vec{n}_b < 0} w_d \left(\Psi_{g,d}^*, c_g(\vec{\Omega}_d) |\vec{\Omega}_d \cdot \vec{n}_b| (\Psi_{g,d} - \Psi_{g,d,\text{reflecting}}) \right)_{\partial\mathcal{D}_r}, \quad (186)$$

with $\vec{\Omega}_{d,\text{reflecting}} = \vec{\Omega}_d - 2(\vec{\Omega}_d \cdot \vec{n}_b)\vec{n}_b$.

3.4.4.3 Transient kernel

This weak form currently does not support time-dependent problems.

3.4.4.4 Neutron

Prompt fission kernel:

$$\begin{aligned} (\mathbb{L}\Psi^*, \mathbf{WF}\Psi) &= \sum_{g=1}^G \int_{\mathcal{S}} d\Omega \left(\Sigma_{t,g} \Psi_g^* + \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{W_g}{4\pi} \frac{\tilde{\chi}_g}{k} \sum_{g'=1}^G \nu \Sigma_{f,g'} \int_{\mathcal{S}} \Psi_{g'}(\vec{\Omega}') d\Omega' \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Sigma_{t,g} \Psi_{g,d}^* + \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \frac{W_g}{4\pi} \frac{\tilde{\chi}_g}{k} F \right)_{\mathcal{D}}, \end{aligned} \quad (187)$$

where F is defined by Eq. (138). Delayed neutron kernels are not implemented, which is required for transient calculations.

3.4.4.5 Thermal radiation

This weak form currently does not support thermal radiation.

3.4.4.6 R-spherical coordinate

This weak form currently does not support R-spherical coordinate.

3.4.4.7 RZ-axial symmetric coordinate

This weak form currently does not support RZ coordinate.

3.4.4.8 Mathematical adjoint

This weak form currently does not support mathematical adjoint.

3.4.4.9 Summary of MOOSE objects

The MOOSE objects used by LS-CFEM-SN are listed in Table 16, Table 17, Table 18, Table 19 and Table 20.

Table 16 LS-CFEM-SN kernels.

Kernel	Type	Jacobian
$\left(W_g \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^* \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} + \tau_g \Sigma_{t,g} \Psi_{g,d} \right)_{\mathcal{D}}$	LSStreaming	Y
$\left(W_g \Sigma_{t,g} \Psi_{g,d}^* \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} + \Sigma_{t,g} \Psi_{g,d} \right)_{\mathcal{D}}$	LSCollision	Y
$\left(\Sigma_{t,g} \Psi_{g,d}^* + \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^* W_g \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} S_{g,\ell,m} R_{\ell,m}(\vec{\Omega}_d) \right)_{\mathcal{D}}$	LSScattering	Y
$\left(\Sigma_{t,g} \Psi_{g,d}^* + \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^* W_g S_g^{\text{ext}}(\vec{\Omega}_d) \right)_{\mathcal{D}}$	LSSource	N/A
$\left(\Sigma_{t,g} \Psi_{g,d}^* + \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^* \frac{W_g}{4\pi} \frac{\tilde{\chi}_g}{k} F \right)_{\mathcal{D}}$	LSFission	Y

Table 17 LS-CFEM-SN kernels (continued).

Type	Condition	Number of kernels
LSStreaming	$\Sigma_{t,g}^a$	$N \times G$
LSCollision	$\Sigma_{t,g}$	$N \times G$
LSScattering	$\Sigma_{s,\ell}^{g' \rightarrow g}, \ell = 0, \dots, L$	$N \times NNZ^b$
LSSource	Q_g^{ext}	$N \times G$
LSFission	$\tilde{\chi}_g$	$N \times G$

^a Means where $\Sigma_{t,g}$ is available.

^b Number of non-zeros in the scattering matrix, i.e. as many (g', g) such that

$$\Sigma_s^{g' \rightarrow g} = \left\{ \Sigma_{s,\ell}^{g' \rightarrow g}, \forall \ell = 0, \dots, L \right\} \neq 0.$$

Table 18 LS-CFEM-SN boundary conditions.

Type	Equation	Number of BCs	Jacobian
LSIncFluxBC	$\Psi_{g,n} = \Psi_{g,n}^{\text{inc}}$	(nodal)	N/A
LSReflectBC	$\Psi_{g,n} = \Psi_{g,n,\text{reflecting}}$	(nodal)	N/A
LSIncFluxBC_Weak	$\left(\Psi_{g,n}^* c_g(\vec{\Omega}_n) \frac{ \vec{\Omega}_n \cdot \vec{n}_b - \vec{\Omega}_n \cdot \vec{n}_b}{2} (\Psi_{g,n} - \Psi_{g,n}^{\text{inc}}) \right)_{\partial \mathcal{D}_s}$	$N_s^a \times N \times G$	Y
LSReflectBC_Weak	$\left(\Psi_{g,n}^* c_g(\vec{\Omega}_n) \frac{ \vec{\Omega}_n \cdot \vec{n}_b - \vec{\Omega}_n \cdot \vec{n}_b}{2} (\Psi_{g,n} - \Psi_{g,n,\text{reflecting}}) \right)_{\partial \mathcal{D}_r}$	$N \times G$	Y

^a N_s is the number of source transport functions. If there are no source functions, it is equal to 1.

Table 19 LS-CFEM-SN auxiliary kernels.

Name	Equation	Type	Number of kernels	Condition
Angular flux moments ($\Phi_{g,\ell,m}$)	Eq. (123)	FluxMoment	$\mathcal{N}(L_{\text{input}})^3 \times G$	\mathcal{D}
Fission source (F)	Eq. (138)	MaterialFissionSource	1	$\nu\Sigma_{t,g}, g = 1, \dots, G$

^a L_{input} is a user input. It thus can be used to truncate the scattering order when $L_{\text{input}} < L$ or ask Rattlesnake to evaluate more flux moments when $L_{\text{input}} > L$.

Table 20 LS-CFEM-SN auxiliary materials.

Name	Type	Number of materials	Condition
W_g	LSWeight	G	$\Sigma_{t,g}$
$S_{g,\ell,m}$	SsrcMaterial	$\mathcal{N}(L) \times G$	$\Sigma_{s,\ell}^{g' \rightarrow g}, g' = 1, \dots, G; g = 1, \dots, G$
F	FsrcMaterial	1	$\nu\Sigma_{t,g}, g = 1, \dots, G$
J_{out}	BoundaryCurrent	G	$\partial\mathcal{D}_w$

3.4.5 LS-CFEM-PN

The angular discretization agnostic weak form is given in Section 3.4.5. Most notation used for the LS-PN weak form can be found in Section 3.4.3 or Appendix B.

3.4.5.1 Steady-state CTE

Streaming kernel:

$$\begin{aligned}
& (\mathbf{W}\mathbb{L}_1\Psi^*, \mathbb{L}_1\Psi) + (\mathbf{W}\mathbb{L}_2\Psi^*, \mathbb{L}_1\Psi) + (\mathbf{W}\mathbb{L}_1\Psi^*, \mathbb{L}_2\Psi) \\
&= \sum_{g=1}^G \int_S d\Omega \left(W_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \vec{\Omega} \cdot \vec{\nabla} \Psi_g \right)_D + \int_S d\Omega \left(W_g \Sigma_{t,g} \Psi_g^*, \vec{\Omega} \cdot \vec{\nabla} \Psi_g \right)_D + \int_S d\Omega \left(W_g \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \Sigma_{t,g} \Psi_g \right)_D \\
&= \sum_{g=1}^G \left(W_g \vec{\nabla} \Phi_g^*, \vec{H} \cdot \vec{\nabla} \Phi_g \right)_D + \left(W_g \Sigma_{t,g} \Phi_g^*, \vec{D} \cdot \vec{\nabla} \Phi_g \right)_D + \left(W_g \vec{D} \cdot \vec{\nabla} \Phi_g^*, \Sigma_{t,g} \Phi_g \right)_D.
\end{aligned} \tag{188}$$

Collision kernel:

$$(\mathbf{W}\mathbb{L}_2\Psi^*, \mathbb{L}_2\Psi) = \sum_{g=1}^G \left(W_g \Sigma_{t,g} \Phi_g^*, \Sigma_{t,g} \mathbf{P} \Phi_g \right)_D. \tag{189}$$

Scattering kernel:

$$\begin{aligned}
(\mathbb{L}\Psi^*, \mathbf{W}\mathbf{S}\Psi) &= \sum_{g=1}^G \int_S d\Omega \left(\Sigma_{t,g} \Psi_g^* + \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, W_g \sum_{g'=1}^G \sum_{\ell=0}^L \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g',\ell,m} R_{\ell,m} \right)_D \\
&= \sum_{g=1}^G \left(\Sigma_{t,g} \mathbf{P} \Phi_g^* + \vec{D} \cdot \vec{\nabla} \Phi_g^*, W_g \Sigma_{s,g' \rightarrow g} \Phi_{g'} \right)_D.
\end{aligned} \tag{190}$$

Source kernel:

$$(\mathbb{L}\Psi^*, \mathbf{W}\mathbf{Q}^{\text{ext}}) = \sum_{g=1}^G \left(\Sigma_{t,g} \mathbf{P} \Phi_g^* + \vec{D} \cdot \vec{\nabla} \Phi_g^*, W_g \mathbf{Q}_g \right)_D. \tag{191}$$

3.4.5.2 Boundary conditions

The boundary conditions are also fairly close to those presented in Section 3.4.3. The main difference is that the boundary terms are weighted by a variable c as shown in Section 3.3.3. As a reminder, the boundary term is given by:

$$\left\langle \mathbb{C}\Psi^*, \Psi - \Psi^{\text{inc}} - \mathbf{B}\Psi \right\rangle^-, \quad (192)$$

where the various options for c are given in Table 3. For the LS-CFEM-PN scheme, the default value is $c_g = \tau_g^{-1}$ which was shown to be consistent with the SAAF scheme with a void treatment [54].

Vacuum boundary conditions:

$$\left\langle \mathbb{C}\Psi^*, \Psi - \Psi^{\text{inc}} \right\rangle_{\partial\mathcal{D}_s}^- = \left\langle \mathbb{C}\Phi^*, \mathbf{L}^-(\Phi - \Phi^{\text{inc}}) \right\rangle_{\partial\mathcal{D}_s} = \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left\langle c_g \Phi_{g,\ell,m'}^* L_{p,p'}^-(\Phi_{g,\ell',m'} - \Phi_{g,\ell',m'}^{\text{inc}}) \right\rangle_{\partial\mathcal{D}_s}. \quad (193)$$

Several options are available for \mathbf{L}^- , as described in Section B.3.

Reflecting boundary conditions:

$$\left\langle \mathbb{C}\Psi^*, \Psi - \Psi_r \right\rangle_{\partial\mathcal{D}_r}^- = \left\langle \Phi^*, \mathbb{C}\mathbf{N}^-\Phi \right\rangle_{\partial\mathcal{D}_r} = \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left\langle \Phi_{g,\ell,m'}^* c_g N_{p,p'}^-\Phi_{g,\ell',m'} \right\rangle_{\partial\mathcal{D}_r}. \quad (194)$$

Several options are available for \mathbf{N}^- , as described in Section B.3

White boundary conditions are currently not available.

3.4.5.3 Transient kernel

This weak form currently does not support time-dependent problems.

3.4.5.4 Neutron

Prompt fission kernel:

$$\begin{aligned} & \sum_{g=1}^G \int_S d\Omega \left(\Sigma_{t,g} \Psi_g^* + \vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \frac{W_g}{4\pi} \frac{\chi_{p,g}}{k} \sum_{g'=1}^G \nu \Sigma_{f,g'} \int_S \Psi_{g'}(\vec{\Omega}') d\Omega' \right) \\ &= \sum_{g=1}^G \left(\Sigma_{t,g} \mathbf{P}\Phi_g^* + \vec{\mathbf{D}} \cdot \vec{\nabla} \Phi_g^*, \frac{W_g}{4\pi} \frac{\chi_{p,g}}{k} F \right)_{\mathcal{D}}, \end{aligned} \quad (195)$$

where F is defined by Eq. (138). Delayed neutron kernels are not implemented, which is required for transient calculations.

3.4.5.5 Thermal radiation

This weak form currently does not support thermal radiation.

3.4.5.6 R-spherical coordinate

This weak form currently does not support R-spherical coordinate.

3.4.5.7 RZ-cylindrical coordinate

This weak form currently does not support RZ cylindrical coordinate.

3.4.5.8 Mathematical adjoint

This weak form currently does not support math adjoint.

3.4.5.9 Summary of MOOSE objects

The MOOSE objects used by LS-CFEM-SN are listed in Table 21, Table 22, Table 23, Table 24, Table 25 and Table 26.

Table 21 LS-CFEM-PN kernels.

Kernel	Type	
$\left(\vec{\nabla} \Phi_{g,\ell,m}^*, W_g \vec{H}_{p,p} \cdot \vec{\nabla} \Phi_{g,\ell,m} \right)_{\mathcal{D}}$	LSPNStreaming ^a	Y
$\left(W_g \vec{\nabla} \Phi_{g,\ell,m}^*, \vec{H}_{p,p'} \cdot \vec{\nabla} \Phi_{g,\ell',m'} \right)_{\mathcal{D}} +$ $\left(W_g \Sigma_{t,g} \Phi_{g,\ell,m}^*, \vec{D}_{p,p'} \cdot \vec{\nabla} \Phi_{g,\ell',m'} \right)_{\mathcal{D}} +$ $\left(W_g \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, \Sigma_{t,g} \Phi_{g,\ell',m'} \right)_{\mathcal{D}}$	LSPNCouplingStreaming ^b	Y
$\left(\Sigma_{t,g} \Phi_{g,\ell,m}^*, W_g \Sigma_{t,g} P_{p,p} \Phi_{g,\ell,m} \right)_{\mathcal{D}}$	LSPNRemoval	Y
$\left(\Sigma_{t,g} P_{p',p} \Phi_{g,\ell,m}^* + \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, W_g \sum_{g'=1}^G \Sigma_{s,\ell'}^{g' \rightarrow g} \Phi_{g',\ell',m'} \right)_{\mathcal{D}}$	LSPNScattering	Y
$\left(\Sigma_{t,g} P_{p',p} \Phi_{g,\ell,m}^* + \vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, W_g Q_{g,\ell',m'} \right)_{\mathcal{D}}$	LSPNSource	N/A
$\left(\Sigma_{t,g} P_{0,p} \Phi_{g,\ell,m}^* + \vec{D}_{0,p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, \frac{W_g}{4\pi} \frac{\lambda_{p,g}}{k} F \right)_{\mathcal{D}}$	LSPNFission	D ^c

^a Only diagonal terms ($p = p'$ i.e. $(\ell, m) = (\ell', m')$).

^b Only off-diagonal terms ($p \neq p'$ i.e. $(\ell, m) \neq (\ell', m')$).

^c Only the diagonal part (within-group) is assembled

Table 22 LS-CFEM-PN kernels (continued).

Type	Condition	Number of kernels
LSPNStreaming	$\Sigma_{t,g}$ ^a	$\mathcal{N} \times G$
LSPNCouplingStreaming	$\Sigma_{t,g}$	$\mathcal{N}_1^b \times G$
LSPNRemoval	$\Sigma_{t,g}$	$\mathcal{N} \times G$
LSPNScattering	$S_{g,\ell,m}$	$\mathcal{N} \times G \times (L + 1)$
LSPNSource	Q_g^{ext}	$\mathcal{N}^c \times G$
LSPNFission	$\tilde{\chi}_g$	$\mathcal{N} \times G$

^a Means where $\Sigma_{t,g}$ is available.

^b Number of non-zero elements in the set $\{(p, p') \in \{1, \dots, \mathcal{N}\}^2; \|\mathbf{H}_{p,p'}\| > 0 \text{ or } \|\vec{D}_{p,p'}\| > 0\}$.

^c At the moment, only isotropic sources are available.

Table 23 LS-CFEM-PN boundary conditions.

Equation	Type	Number of kernels	Jacobian
$\left\langle c_g \Phi_{g,\ell,m'} \sum_{p'} \mathbf{L}_{p,p'}^- (\Phi_{g,\ell',m'} - \Phi_{g,\ell',m'}^{\text{inc}}) \right\rangle_{\partial \mathcal{D}_s}$	LSPNVacuumBC	$\mathcal{N} \times G \times N_s^a$	D ^b
$\left\langle c_g \Phi_{g,\ell,m'} \sum_{p'} \mathbf{N}_{p,p'}^- \Phi_{g,\ell',m'} \right\rangle_{\partial \mathcal{D}_r}$	LSPNReflectingBC	$\mathcal{N} \times G$	D

^a N_s is the number of source transport functions. If there are no source functions, it is equal to 1.

^b D means that only the diagonal part (within-group) is assembled.

Table 24 LS-CFEM-PN auxiliary kernels.

Name	Equation	Type	Number of kernels	Condition
Fission source (F)	Eq. (138)	MaterialFissionSource	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$

Table 25 LS-CFEM-PN auxiliary materials.

Name	Type	Number of kernels	Condition
$S_{g,\ell,m}$	SsrcMaterial	$\mathcal{N}(L) \times G$	$\Sigma_{s,\ell}^{g' \rightarrow g}, g' = 1, \dots, G; g = 1, \dots, G$
F	FsrcMaterial	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$

Table 26 LS-CFEM-PN userobjects. Refer to Section B.3 for the matrix definitions.

Content	Type
\mathbf{L}^-	LMinusMatrixOnSide
\mathbf{N}^-	NMinusMatrixOnSide

3.4.6 DFEM-SN

This section covers all kernels for the discontinuous FEM discretization of the first order S_N equations.

3.4.6.1 The transport sweep

The DFEM- S_N equations allow the matrix-free inversion of the streaming and collision operator given by the terms $(\mathbb{L}^* \Psi^*, \Psi) + \langle \Psi^*, \Psi \rangle^+ - \langle \llbracket \Psi^* \rrbracket, \Psi_- \rangle_{\Gamma_{\text{int}}}$ using a transport sweep. It is recognized that the unknowns for a given direction are not coupled to any other direction via the streaming and collision operator. In addition, elements' unknowns for the same angular direction are only coupled to their upwind neighbors, i.e. neighbors in the direction of $-\vec{\Omega}$. Therefore, one can find a sequence in which elements can be swept such that for each element only a small linear system of the size of the local degrees of freedom has to be solved. Performing a sweep through all elements and angular directions completes a single transport sweep. A transport sweep is an extremely memory and computing-time efficient algorithm for inverting the streaming and collision operator.

The element sweep order for each angular direction is determined using a breadth first search algorithm. The algorithm relies on separating the faces of an element into inflow and outflow faces depending on the inner product of the normal vector and $\vec{\Omega}$. Rattlesnake allows the elements' faces to be curved such that the normal varies from quadrature point to quadrature point creating the potential for a face to be both inflow and outflow face. For obtaining a unique ordering, Rattlesnake therefore uses the average norm over faces to determine the transport sweep orders. Note that if there is a face that has both inflow and outflow quadrature point, the dependency graph of the sweep is cyclic and no ordering exists that allows exact sweeping. Using the average normal over faces then constitutes an approximation lagging fluxes at inflow quadrature points located on an outflow face [because the neighboring element has not been swept yet]. Even in the absence of curved faces the sweep dependency graph can be cyclic. In this case, the number of element's dependencies currently on top of the queue in the breadth first

search algorithm is set to zero. This means it is queued for solution despite some of its upstream neighbors not being swept yet. In the Rattlesnake sweep implementation, significant angular fluxes including inflow quadrature points on mixed inflow/outflow edges, reflective boundary conditions, and fluxes on severed dependency faces are not saved between iterations. Before commencing the sweeps all angular fluxes are set to zero. In addition, reflective boundary influxes are zeroed out. In the particular Rattlesnake Richardson implementation, this guarantees convergence to the right answer.

The sweep that is currently implemented assembles the local linear systems using quadrature rules. This allows an exact treatment of curved surfaces and spatially varying cross sections. However, the current grind time, especially in three-dimensional geometries precludes deployment for production calculations. The sweep can be used in a Richardson type iteration scheme for solving fixed source problem, in the DFEM- S_N based nonlinear diffusion acceleration method using a sweep update executioner derived from Richardson executioner, and finally it can be used to precondition a direct GMRES solve of the DFEM- S_N problem.

The transport sweep can be executed in parallel. Currently spatial domain decomposition using a block Jacobi iteration is implemented. The partition is identical to the partition provided by the selected MOOSE partitioner.

3.4.6.2 Steady-state CTE

Streaming and collision kernels:

$$(-\mathbb{L}_1 \Psi^*, \Psi) = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(-\vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \Psi_{g,d} \right)_{\mathcal{D}}. \quad (196)$$

Collision kernels:

$$(+\mathbb{L}_2 \Psi^*, \Psi) = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Sigma_{t,g} \Psi_{g,d}^*, \Psi_{g,d} \right)_{\mathcal{D}}. \quad (197)$$

The scattering term:

$$\begin{aligned} (\Psi^*, \mathbb{S}\Psi) &= \sum_{g=1}^G \sum_{g'=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^*, \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \Sigma_{s,\ell}^{g' \rightarrow g} \Phi_{g',\ell,m} R_{\ell,m}(\vec{\Omega}_d) \right)_{\mathcal{D}} \\ &= \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^*, \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} S_{g,\ell,m} R_{\ell,m}(\vec{\Omega}_d) \right)_{\mathcal{D}}, \end{aligned} \quad (198)$$

where $\Phi_{g,\ell,m}$ and $S_{g,\ell,m}$ are defined in Eq. (123) and Eq. (124), respectively.

External source:

$$(\Psi^*, \mathbf{Q}) = \sum_{g=1}^G \sum_{n=1}^N w_n \left(\Psi_{g,n}^*, Q_g(\vec{\Omega}_n) \right)_{\mathcal{D}}. \quad (199)$$

3.4.6.3 Interior face terms (DGKernel)

The interior face term:

$$\langle \llbracket \Psi^* \rrbracket, \Psi_- \rangle_{\Gamma_i} = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left\langle \llbracket \Psi_{g,d}^* \rrbracket, \Psi_{g,d,-} \right\rangle_{\Gamma_i}, \quad (200)$$

where Γ_i is the set of interior faces and $\llbracket f \rrbracket = f_+ - f_-$ is the jump across element faces with the upwind and downwind sides being defined by $\vec{\Omega}_d$.

3.4.6.4 Transient

The time derivative kernel:

$$\left(\Psi^*, \frac{1}{v} \frac{\partial \Psi}{\partial t} \right) = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^*, \frac{1}{v_g} \frac{\partial \Psi_{g,d}}{\partial t} \right)_{\mathcal{D}}. \quad (201)$$

3.4.6.5 Neutron

The prompt fission kernel:

$$(\Psi^*, \mathbb{F} \Psi) = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^*, \frac{\chi_{p,g}}{4\pi k} F \right)_{\mathcal{D}}. \quad (202)$$

where F is defined by Eq. (138).

Delayed neutron kernel:

$$(\Psi^*, \mathbf{DC}) = \sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\Psi_{g,d}^*, \frac{1}{4\pi k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i \right)_{\mathcal{D}}, \quad (203)$$

where the DNP concentrations C_i are evaluated using Eq. (140a).

3.4.6.6 Boundary Conditions

DFEM- S_N boundary conditions are identical to the SAAF- S_N -CFEM boundary condition.

3.4.6.7 Thermal radiation

DFEM- S_N does not support thermal radiation.

3.4.6.8 R-spherical coordinate

DFEM- S_N does not support R-spherical coordinate.

3.4.6.9 RZ-cylindrical coordinate

DFEM- S_N does not support RZ-cylindrical coordinate.

3.4.6.10 Mathematical adjoint

For the DFEM- S_N equations physical and mathematical adjoint are identical.

3.4.6.11 Summary of MOOSE objects

The MOOSE objects used by DFEM-SN are listed in Table 27 through Table 31.

Table 27 DFEM- S_N kernels and DG kernels.

Kernel	Type	Jacobian
$\left(-\vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d}^*, \Psi_{g,d}\right)_D$	SNStreaming	Y
$\left(\Sigma_{t,g} \Psi_{g,d}^*, \Psi_{g,d}\right)_D$	Reaction	Y
$\left(\Psi_{g,d}^* \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} \Sigma_{s,\ell}^{g' \rightarrow g} \Phi_{g',\ell,m} R_{\ell,m}(\vec{\Omega}_d)\right)_D$	SNScattering ^a	N
$\left(\Psi_{g,d}^* \sum_{\ell=0}^L \sum_{m=-\ell}^{\ell} S_{g,\ell,m} R_{\ell,m}(\vec{\Omega}_d)\right)_D$	SNGroupScattering	N
$\left(\Psi_{g,d}^* Q_g(\vec{\Omega}_d)\right)_D$	SNSource	N/A
$\left\langle \llbracket \Psi_{g,d}^* \rrbracket, \Psi_{g,d,-} \right\rangle_{\Gamma_i}$	Upwind	Y
$\left(\Psi_{g,d}^* \frac{1}{v_g} \frac{\partial \Psi_{g,d}}{\partial t}\right)_D$	SpeedTimeDerivative	Y
$\left(\Psi_{g,d}^* \frac{\chi_{p,g}}{4\pi k} F\right)_D$	SNFission	Y
$\left(\Psi_{g,d}^* \frac{1}{4\pi k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i\right)_D$	SNDelayedNeutron ^b	N

^a SNScattering is implemented for flux moments to be provided as AuxVariables or Materials (SNScatteringMat).

^b SNDelayedNeutronMat is identical except that it uses the precursors computed as material.

Table 28 DFEM- S_N kernels (continued).

Type	Condition	Number of kernels
SNStreaming	$\Sigma_{t,g}$	$N \times G$
Reaction	$\Sigma_{t,g}$	$N \times G$
SNScattering	$\Sigma_{s,\ell}^{g' \rightarrow g}, \ell = 0, \dots, L$	$N \times NNZ^a$
SNGroupScattering	$S_{g,\ell,m}$	$N \times G$
SNSource	Q_g	$N \times G$
Upwind	none	$N \times G$
SpeedTimeDerivative	v_g	$N \times G$
SNFission	$\tilde{\chi}_g$	$N \times G$
SNDelayedNeutron	$\chi_{d,g,i}, i = 1, \dots, I$	$N \times G$

^a Number of non-zeros in the scattering matrix, i.e. all (g', g) s whose $\Sigma_{s,\ell}^{g' \rightarrow g} = \left\{ \Sigma_{s,\ell}^{g' \rightarrow g}, \forall \ell = 0, \dots, L \right\} \neq 0$.

Table 29 DFEM-SN boundary conditions.

Type	Equation	Number of BCs	Jacobian
SNVacuum	$\left(\Psi_{g,n}^*, \vec{\Omega} \cdot \vec{n}_b \Psi_{g,n}\right)_{\partial \mathcal{D}_s}$	$N_s^a \times N_d \times G$	Y
SNReflecting	$\left(\Psi_{g,n}^*, \vec{\Omega} \cdot \vec{n}_b \Psi_{g,n}\right)_{\partial \mathcal{D}_r}$	$N_d \times G$	Y
SNWhite	$\sum_{\vec{\Omega} \cdot \vec{n}_b > 0} w_n \left(\Psi_{g,n}^*, \vec{\Omega} \cdot \vec{n}_b \Psi_{g,n}\right)_{\partial \mathcal{D}_w}$	$N_d \times G$	Y

^a N_s is the number of surface source transport functions. If there are no source functions, it is equal to 1.

Table 30 DFEM-SN auxiliary kernels.

Name	Equation	Type	Number of kernels	Condition
Angular flux moments ($\Phi_{g,\ell,m}$)	Eq. (123)	FluxMoment	$\mathcal{N}(L_{\text{input}})^3 \times G$	\mathcal{D}
Fission source (F)	Eq. (138)	MaterialFissionSource	1	$\nu\Sigma_{f,g}, g = 1, \dots, G$
DNP ($C_i, i = 1, \dots, I$)	Eq. (140a)	DelayedNeutronPrecursor	I	$\nu\Sigma_{f,g}, g = 1, \dots, G$

^a L_{input} is a user input. It thus can be used to truncate the scattering order when $L_{\text{input}} < L$ or ask Rattlesnake to evaluate more flux moments when $L_{\text{input}} > L$.

Table 31 DFEM-SN auxiliary materials.

Name	Equation	Type	Number of materials	Condition
$S_{g,\ell,m}$	Eq. (124)	SsrcMaterial	$\mathcal{N}(L) \times G$	$\Sigma_{s,\ell}^{g' \rightarrow g}, g' = 1, \dots, G; g = 1, \dots, G$
F	Eq. (138)	FsrcMaterial	1	$\nu\Sigma_{f,g}, g = 1, \dots, G$
C_i	Eq. (140a)	DNPMaterial	I	$\nu\Sigma_{f,g}, g = 1, \dots, G$
J_{out}	Eq. (133)	BoundaryCurrent	G	$\partial\mathcal{D}_w$

3.4.7 DFEM-PN

There are several ways to derive the DFEM- P_N weak formulation. One of them is to apply the P_N approximation directly to the transport equation and then to apply the DFEM spatial discretization. This can be for instance found in [55, 56, 57]. Here, we follow the logic of this manual (with the spatial discretization being applied prior to the angular discretization) and will change the numerical flux appropriately to end up with the same formulation. We therefore start with Eq. (112). As detailed in Appendix B, the P_N method consists of expanding all angular dependent in terms of the spherical harmonics up to degree N . Most terms in the weak formulation are pretty straightforward to derive because of the orthogonality property of the spherical harmonics. We first describe the volumetric kernels. Second, the boundary and interior facets (or interface) are explained. Third, we describe some advanced features such as filtering capabilities and mass matrix lumping and present all the MOOSE objects specific to this scheme. Finally, we explicit what the matrices for DFEM- P_1 look like for better understanding.

3.4.7.1 Steady-state CTE

Streaming kernel:

$$(\mathbb{L}_1^* \Psi^*, \Psi) = \sum_{g=1}^G \int_{\mathcal{S}} d\Omega \left(-\vec{\Omega} \cdot \vec{\nabla} \Psi_g^*, \Psi_g \right)_{\mathcal{D}} = \sum_{g=1}^G \sum_{p,p'=1}^{\mathcal{N}} \left(-\vec{D}_{p',p} \cdot \vec{\nabla} \Phi_{g,\ell,m}^*, \Phi_{g,\ell',m'} \right)_{\mathcal{D}}. \quad (204)$$

Collision kernel:

$$(\mathbb{L}_2^* \Psi^*, \Psi) = \sum_{g=1}^G \int_{\mathcal{S}} d\Omega \left(\Sigma_{t,g} \Psi_g^*, \Psi_g \right)_{\mathcal{D}} = \sum_{g=1}^G \sum_{p=1}^{\mathcal{N}} \left(\Phi_{g,\ell,m}^*, \Sigma_{t,g} P_{p,p} \Phi_{g,\ell,m} \right)_{\mathcal{D}}. \quad (205)$$

Scattering kernel:

$$\begin{aligned} (\Psi^*, \mathbb{S} \Psi) &= \sum_{g,g'=1}^G \int_{\mathcal{S}} d\Omega \left(\Psi_{g'}^*, \sum_{g'=1}^G \sum_{\ell=0}^L \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{m=-\ell}^{\ell} \Phi_{g',\ell,m} R_{\ell,m} \right)_{\mathcal{D}} \\ &= \sum_{g,g'=1}^G \left(\mathbf{P} \Phi_{g'}^*, \Sigma_{s,g' \rightarrow g} \Phi_{g'} \right)_{\mathcal{D}} \end{aligned}$$

$$= \sum_{g,g'=1}^G \sum_{p=1}^{\mathcal{N}} \left(\Phi_{g,\ell,m}^*, P_{p,p} \Sigma_{s,\ell}^{g' \rightarrow g} \Phi_{g',\ell,m} \right)_{\mathcal{D}}. \quad (206)$$

Source kernel:

$$(\Psi^*, \mathbf{Q}^{\text{ext}}) = \sum_{g=1}^G \int_{\mathcal{S}} \left(\Psi_g^*, \mathbf{Q}_g^{\text{ext}} \right)_{\mathcal{D}} d\Omega = \sum_{g=1}^G \sum_{p=1}^{\mathcal{N}} \left(\Phi_{g,\ell,m}^*, P_{p,p} \mathbf{Q}_{g,\ell,m} \right)_{\mathcal{D}}. \quad (207)$$

3.4.7.2 Transient kernel

Transient kernel:

$$\left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) = \sum_{g=1}^G \sum_{p=1}^{\mathcal{N}} \left(\Phi_{g,\ell,m}^*, P_{p,p} \frac{\partial}{\partial t} \left(\frac{\Phi_{g,\ell,m}}{v_g} \right) \right)_{\mathcal{D}}. \quad (208)$$

If we assume $v_g, g = 1, \dots, G$ is time independent, the kernel can be simplified into

$$\sum_{g=1}^G \sum_{p=1}^{\mathcal{N}} \left(\Phi_{g,\ell,m}^*, P_{p,p} \frac{1}{v_g} \frac{\partial \Phi_{g,\ell,m'}}{\partial t} \right)_{\mathcal{D}}. \quad (209)$$

3.4.7.3 Interior facets terms (DGKernel)

First, we recall the jump notations defined with respect to the direction $\vec{\Omega}$ rather than to the normal vector \vec{n} in Eq. (111a):

$$a_{\pm}(\vec{x}, \vec{\Omega}) \equiv \lim_{s \rightarrow 0^{\pm}} a(\vec{x} + s\vec{\Omega}, \vec{\Omega}),$$

$$[[a(\vec{x}, \vec{\Omega})]] \equiv a_+(\vec{x}, \vec{\Omega}) - a_-(\vec{x}, \vec{\Omega}),$$

and define the jump operator $[[\cdot]]$ and average operator $\{\{\cdot\}\}$ on Γ_{int} as

$$[[a(\vec{x})]] \equiv a^+ - a^-, \quad (211a)$$

$$\{\{a(\vec{x})\}\} \equiv \frac{a^+ + a^-}{2}, \quad (211b)$$

$$a^{\pm}(\vec{x}) \equiv \lim_{s \rightarrow 0^{\pm}} a(\vec{x} + s\vec{n}). \quad (211c)$$

We can express the interface term as follows:

$$I \equiv - \langle [[\Psi^*]], \Psi_- \rangle_{\Gamma_{\text{int}}} = - \langle [[\Psi^*]], \Psi^- \rangle_{\Gamma_{\text{int}}}^+ + \langle [[\Psi^*]], \Psi^+ \rangle_{\Gamma_{\text{int}}}^-. \quad (212)$$

Using the P_N expansion (i.e. $\Psi \approx \mathbf{R}^T \Phi$) and integrating over angle, it becomes:

$$I = - \left([[\Phi^*]], \mathbf{L}^+ \Phi^- \right)_{\Gamma_{\text{int}}} + \left([[\Phi^*]], \mathbf{L}^- \Phi^+ \right)_{\Gamma_{\text{int}}}. \quad (213)$$

Using the identity:

$$\eta\chi - \zeta\sigma = \frac{(\eta + \zeta)(\chi - \sigma) + (\eta - \zeta)(\chi + \sigma)}{2}, \quad (214)$$

we can then show that:

$$I = \left([[\Phi^*]], (\mathbf{L}^- - \mathbf{L}^+) \{\{\Phi\}\} + \frac{\mathbf{L}^+ + \mathbf{L}^-}{2} [[\Phi]] \right)_{\Gamma_{\text{int}}}. \quad (215)$$

An upwind formulation can be obtained by using a Roe-type Riemann solver [55]. This is done by first diagonalizing $(\mathbf{L}^+ - \mathbf{L}^-) \equiv \mathbf{U} \Lambda \mathbf{U}^T$ and then replacing $(\mathbf{L}^+ + \mathbf{L}^-)$ in the previous expression with $\mathbf{U} |\Lambda| \mathbf{U}^T$. Another option resulting in a sparser matrix and bypassing the need for diagonalization is to use a Lax-Friedrich numerical flux which yields the following interface terms [56, 57]:

$$I = \left(\llbracket \Phi^* \rrbracket, -\vec{\mathbf{D}} \cdot \vec{n} \{ \Phi \} + \frac{1}{2} \|\vec{n}\|_1 \mathbf{P} \llbracket \Phi \rrbracket \right)_{\Gamma_{\text{int}}}, \quad (216)$$

where $\|\cdot\|_1$ is the L1-norm and where we have noticed $\vec{\mathbf{D}} \cdot \vec{n} = (\mathbf{L}^+ - \mathbf{L}^-)$.

The treatment for the numerical flux on the boundaries depends on the type of boundaries.

3.4.7.4 Reflecting boundary

In the case of a reflecting boundary condition, the numerical flux is very similar to Eq. (216) except that the value of Ψ_+ designates the value inside the exterior facet (i.e. Ψ) and the value of Ψ_- corresponds to the value outside the exterior facet and is thus determined by the boundary condition, which is Ψ evaluated at the reflected direction. Therefore, the 'reflecting' numerical flux is given by:

$$\mathcal{F}_r(\Phi) = -\frac{1}{2} \vec{n}_b \cdot \vec{\mathbf{D}} \Phi + \frac{\|\vec{n}_b\|_1}{2} \mathbf{P} \Phi - \frac{1}{2} \mathbf{L}_r \Phi - \frac{\|\vec{n}_b\|_1}{2} \mathbf{Q}_r \Phi. \quad (217)$$

for all $\vec{x} \in \partial \mathcal{D}_r$, where the matrix definitions can be found in Section B.2.

The first two terms correspond to the inside of the exterior facet while the last two correspond to the outside of the exterior facet. We further define:

$$\mathcal{F}_r(\Phi) = \mathcal{F}_{\text{ext}}(\Phi) + \mathcal{F}_r^{\text{BC}}(\Phi), \quad (218)$$

where

$$\mathcal{F}_{\text{ext}}(\Phi) = -\frac{1}{2} \vec{n}_b \cdot \vec{\mathbf{D}} \Phi + \frac{\|\vec{n}_b\|_1}{2} \mathbf{P} \Phi, \quad (219)$$

and

$$\mathcal{F}_r^{\text{BC}}(\Phi) = -\frac{1}{2} \mathbf{L}_r \Phi - \frac{\|\vec{n}_b\|_1}{2} \mathbf{Q}_r \Phi. \quad (220)$$

We dropped the subscript r for the exterior facet numerical flux because we will see that it is identical for a surface source (incoming Dirichlet) boundary.

3.4.7.5 Incoming Dirichlet boundary

In the case of an Incoming Dirichlet boundary, we will once again use an expression similar to Eq. (216), with the value of Ψ_+ still designating the value inside the exterior facet. The boundary condition will be used to impose Ψ_- , the value outside the exterior facet. The problem is that we can only impose the incoming data of Ψ because that is all the boundary conditions give us. For the outgoing data, we reconstruct them using the outgoing data from the inside of the exterior facet. In other words, for the outside of the exterior facet, we impose the incoming data through the boundary conditions and the outgoing data imposing continuity with the inside of the exterior facet.

We start by taking the moments of Ψ^{inc} :

$$\Psi^{\text{inc}}(\vec{x}, \vec{\Omega}) \equiv \mathbf{R}^T(\vec{\Omega}) \Phi^{\text{inc}}(\vec{x}). \quad (221)$$

Note that this expansion gives values of Ψ^{inc} for $\vec{\Omega} \cdot \vec{n}_b > 0$ but this does not matter as what follows will only evaluate this expression for $\vec{\Omega} \cdot \vec{n}_b < 0$ (in other words, its value for the positive half-sphere may change the values

of the moments but not the interface term because of the angular integration over the negative half-sphere). Then, we construct the numerical flux as follows:

$$\mathcal{F}_d(\Phi, \Psi^{\text{inc}}) = -\frac{1}{2} \left(\vec{n}_b \cdot \vec{D} \Phi + \mathbf{L}^+ \Phi - \mathbf{L}^- \Phi^{\text{inc}} \right) + \frac{\|\vec{n}_b\|_1}{2} \left(\mathbf{P} \Phi - \mathbf{Q}^+ \Phi - \mathbf{Q}^- \Phi^{\text{inc}} \right), \quad (222)$$

for all $\vec{x} \in \partial\mathcal{D}_s$, where the matrix definitions can be found in Section B.2.

Again, the terms in the first parenthesis corresponds to $\vec{D}\Phi$ where $\Phi^+ = \Phi$ and where Φ^- is chosen to be Φ if it corresponds to the outgoing data and to be Φ^{inc} if it corresponds to the incoming data. The second parenthesis was similarly derived and corresponds to $\mathbf{P}[\Phi]$. We can rearrange to get an expression similar to what we got for the reflecting BC:

$$\mathcal{F}_d(\Phi, \Psi^{\text{inc}}) = \mathcal{F}_{\text{ext}}(\Phi) + \mathcal{F}_d^{\text{BC}}(\Phi, \Psi^{\text{inc}}), \quad (223)$$

where

$$\mathcal{F}_d^{\text{BC}}(\Phi, \Psi^{\text{inc}}) = -\frac{1}{2} \left(\mathbf{L}^+ \Phi - \mathbf{L}^- \Phi^{\text{inc}} \right) - \frac{\|\vec{n}_b\|_1}{2} \left(\mathbf{Q}^+ \Phi + \mathbf{Q}^- \Phi^{\text{inc}} \right). \quad (224)$$

3.4.7.6 Filtering

The purpose of the filtering approach [58, 59] is to mitigate the negativity and oscillations inherent to the P_N approximation. Besides, this technique helps improve the conditioning of the system in the streaming limit ($\Sigma_t \rightarrow 0$) in which P_N tends to do poorly. This technique therefore speeds up the calculation in that limit.

The idea is to artificially increase the cross-section in such a way that:

- The particle/energy conservation is maintained.
- The rotational invariance of the solution is preserved.
- The filter vanishes as $N \rightarrow \infty$, which is desirable since the angular expansion is exact in that limit.

Filtering is included in the transport equation by adding an extra collision term:

$$(\Phi^*, \sigma_f(\vec{x}) \mathbf{P} \mathbf{F} \Phi), \quad (225)$$

where $\sigma_f(\vec{x})$ is the filter strength (see the guidelines on how to choose it below) and the expression for \mathbf{F} is given by:

$$\mathbf{F} = \text{diag} \left\{ -\log \sigma_{\text{filterType}} \left(\frac{\ell}{N+1} \right), m = -\ell, \dots, \ell; \ell = 0, \dots, N \right\}, \quad (226)$$

Several filter types can be considered:

$$\sigma_{\text{Lanczos}}(\zeta) = \frac{\sin \zeta}{\zeta}; \quad \sigma_{\text{SSpline}}(\zeta) = \frac{1}{1 + \zeta^4}; \quad \sigma_{\text{exp}}(\zeta) = \exp(c \zeta^\alpha). \quad (227)$$

In the exponential filter, $\alpha \in \mathbb{N}$ and $c = \log(\epsilon_M)$ where ϵ_M is the machine accuracy. The Lanczos and Spherical Spline filters are used in [59] while the exponential filter is considered in [60]. The main difference between the filters is their order. Roughly speaking, the higher the order, the less the filter will give smooth solutions (think of an unfiltered calculation as a filter of order ∞). On the other hand, the convergence of the solution with respect to N is bounded by the order of the filter. The Lanczos, Spherical Spline and Exponential filters are respectively of order 2, 4 and α . The Lanczos filter is the default filter.

From experience, it works best when the filtering contribution is comparable to that of the reaction terms in the regions where it is needed. The typical way to determine the ideal $\sigma_f(\vec{x})$ is:

- Run a calculation without filter and localize where the scalar flux becomes negative.

- Activate the filter in that ‘negative’ region² as well as in an upstream region of a comparable size (it will smooth out the scalar flux in both regions usually making it positive again). For the other regions of the problem, the filter can typically be set to zero or to a smaller value.
- The choice of the value of $\sigma_f(\vec{x})$ is a little trickier: although its units are those of a macroscopic cross-section, its scaling makes it not directly comparable to a cross-section. As a rule of thumb, if you know that the unfiltered calculation has problems up to a given N_0 ,³ then a good scaling is usually obtained if you choose: $\sigma_f(\vec{x}) = -\sigma_t(\vec{x}) / \log \sigma_{\text{filterType}}(1/(N_0 + 1))$ in the regions where you need the filter, σ_t being the total macroscopic cross-section in the domain⁴.

More details can be found in [56, 57]. Another way to tackle this is to test out a few different values of σ_f and choose the one that seems the best. Note also that it is also possible to have a group-dependent filter strength. A study of such filters was done in [61].

3.4.7.7 Thermal radiation

The option of coupling the DFEM-PN scheme with the temperature equation without conduction (see Section 2.4.2) is currently available. Since conduction is not allowed at this point, the spatial discretization of the temperature equation does not present any particular difficulty. The only additional kernel that needed to be implemented for this P_N scheme was the Planckian reemission in the RTE (see Table 32).

Planckian kernel:

$$\left(\Psi^*, \frac{1}{4\pi} \mathbf{S}_{\text{BB}}(T) \right) = \sum_{g=1}^G \int_{\mathcal{S}} d\Omega \left(\Psi_{g'}^*, \frac{1}{4\pi} S_{\text{BB},g} \right)_{\mathcal{D}} = \sum_{g=1}^G \left(\Phi_{g,0,0}^*, P_{0,0} S_{\text{BB},g} \right)_{\mathcal{D}}. \quad (228)$$

$S_{\text{BB},g}$ is defined by Eq. (62).

3.4.7.8 Neutron

This weak form currently does not support neutrons.

3.4.7.9 R-spherical coordinate

This weak form currently does not support R-spherical coordinate.

3.4.7.10 RZ-cylindrical coordinate

This weak form currently does not support RZ-cylindrical coordinate.

3.4.7.11 Mass matrix lumping

Furthermore, for robustness in optically thick region, lumping the mass matrix for the removal terms can be necessary in the context of discontinuous Galerkin discretizations of discrete ordinate equations [62]. In practice, lumping a matrix is done by replacing it by a diagonal matrix whose i -th term is the sum of the elements on the i -th row of the original matrix.

²If the problem is uniform, then activate the filter everywhere.

³Recall that the unfiltered solution should become more and more accurate as N goes to ∞ . Therefore, there is an N_0 for which the unfiltered solution can be considered ‘good enough’.

⁴It can be tempting to set σ_f to $-\sigma_t / \log \sigma_{\text{filterType}}(1/(N + 1))$ but this prevents the filtered solution to converge to the unfiltered solution (and thus to the analytical solution) as $N \rightarrow \infty$.

For instance, $(u^*, u)_{\mathcal{D}}$ is computed as:

$$\sum_i u_i^* \left(\int_{\mathcal{D}} \phi_i^* u \, dx \right). \quad (229)$$

Expanding u with the spatial test functions: $u = \sum_j u_j \phi_j$, this term becomes:

$$\sum_i \sum_j u_i^* \left(\int_{\mathcal{D}} \phi_i^* \phi_j \, dx \right) u_j. \quad (230)$$

Lumping this term can be done by summing these terms but adding them to the diagonal contribution, i.e.:

$$\sum_i u_i^* \left(\int_{\mathcal{D}} \phi_i^* \sum_j \phi_j \, dx \right) u_i. \quad (231)$$

Table 32 DFEM-PN kernels.

Kernel	Type	Jacobian
$-\left(\vec{\nabla} \Phi_{g,\ell,m}^*, \vec{D}_{p,p'} \Phi_{g,\ell',m'}\right)_{\mathcal{D}}$	PNDFEMStreamingVol	Y
$\left(\Phi_{g,\ell,m}^*, \Sigma_{t,g} P_{p,p} \Phi_{g,\ell,m}\right)_{\mathcal{D}}$	PNDFEMRemoval	Y
$\left(\Phi_{g,\ell,m}^*, \Sigma_{t,g} P_{p,p} \Phi_{g,\ell,m}\right)_{\mathcal{D}}^a$	PNDFEMLumpedRemoval	Y
$\left(\Phi_{g,\ell,m}^*, \Sigma_{t,g} P_{p,p} F_{p,p} \Phi_{g,\ell,m}\right)_{\mathcal{D}}$	PNDFEMFilter	Y
$\left(\Phi_{g,\ell,m}^*, P_{p,p} \sum_{g'=1}^G \Sigma_{s,\ell}^{g' \rightarrow g} \Phi_{g',\ell,m}\right)_{\mathcal{D}}$	PNDFEMScattering	D ^b
$\left(\Phi_{g,\ell,m}^*, P_{p,p} \frac{1}{v_g} \frac{\partial \Phi_{g,\ell',m'}}{\partial t}\right)_{\mathcal{D}}$	PNDFEMTimeDerivative	Y
$\left(\Phi_{g,0,0}^*, P_{0,0} S_{\text{BB},g}\right)_{\mathcal{D}}$	PNDFEMPlanck	Y
$\left(\Phi_{g,\ell,m}^*, P_{p,p} Q_{g,\ell,m}\right)_{\mathcal{D}}$	PNDFEMSource	N/A

^a Lumped terms

^b D means that only the diagonal part (within-group) is assembled.

Table 33 DFEM-PN kernels (continued).

Type	Condition	Number of kernels
PNDFEMStreamingVol	$\Sigma_{t,g}^a$	$\mathcal{N}_1^b \times G$
PNDFEMRemoval	$\Sigma_{t,g}$, no removal lumping	$\mathcal{N} \times G$
PNDFEMLumpedRemoval	$\Sigma_{t,g}$, removal lumping	$\mathcal{N} \times G$
PNDFEMFilter	filtering	$(\mathcal{N} - 1) \times G$
PNDFEMScattering	$S_{g,\ell,m}$	$\mathcal{N}(L_{\text{input}})^c \times G$
PNDFEMTimeDerivative	v_g	$\mathcal{N} \times G$
PNDFEMPlanck	thermal	G
PNDFEMSource	Q_g^{ext}	$\mathcal{N}(L_s)^d \times G$

^a Means where $\Sigma_{t,g}$ is available.

^b Number of non-zero elements in the set $\{\vec{D}_{p,p'}; p, p' \in \{1, \dots, \mathcal{N}\}, p \neq p'\}$.

^c L_s is a user input corresponding to the maximum order of the scattering source. By default, it is set to 0 (isotropic scattering).

^d L_{input} is a user input corresponding to the maximum order of the volumetric source. By default, it is set to 0 to only evaluate isotropic sources.

Table 34 DFEM-PN dgkernels.

Kernel	Type	Jacobian
$\left\langle \llbracket \Phi_{g,\ell,m}^* \rrbracket, -\vec{D}_{p,p'} \cdot \vec{n} \llbracket \Phi_{g,\ell',m'} \rrbracket + \frac{1}{2} \ \vec{n}\ _1 P_{p,p'} \llbracket \Phi_{g,\ell',m'} \rrbracket \right\rangle_{\Gamma_{\text{int}}}$	PNDFEMStreamingInterior	D ^a

^a D means that only the diagonal part (within-group) is assembled.

Table 35 DFEM-PN dgkernels (continued).

Type	Condition	Number of dgkernels
PNDFEMStreamingInterior	$\Sigma_{t,g}$	$\mathcal{N}_1^{\text{a}} \times G$

^b Number of non-zero elements in the set $\{\vec{D}_{p,p'}; p, p' \in \{1, \dots, \mathcal{N}\}, p \neq p'\}$ plus \mathcal{N} .

Table 36 DFEM-PN userobjects. Refer to Section B.3 and derivations above for the matrix definitions.

Content	Type
\mathbf{Q}^+	QPlusMatrixOnSide
\mathbf{Q}^-	QMinusMatrixOnSide
\mathbf{L}_r	LrMatrixOnSide
\mathbf{Q}_r	QrMatrixOnSide

Table 37 DFEM-PN auxiliary materials.

Name	Type	Number of materials	Condition
$\ \vec{n}\ _1$	L1NormNormalMaterial	1	always true
$S_{g,\ell,m}$	SsrcMaterial	$\mathcal{N}(L) \times G$	$\Sigma_{s,\ell}^{g' \rightarrow g}, g' = 1, \dots, G; g = 1, \dots, G$

Table 38 DFEM-PN boundary conditions.

Type	Equation	Number of BCs	Jacobian
PNDFEMStreamingExterior	$\frac{1}{2} \left\langle \Phi_{g,\ell,m}^*, -\vec{n}_b \cdot \vec{D}_{p,p'} \Phi_{g,\ell',m'} + \ \vec{n}_b\ _1 P_{p,p'} \Phi_{g,\ell',m'} \right\rangle_{\partial \mathcal{D}}$	$\mathcal{N} \times G$	D ^a
PNDFEMReflectingBC	$\frac{1}{2} \left\langle \Phi_{g,\ell,m}^*, -L_{r,p,p'} \Phi_{g,\ell',m'} - \ \vec{n}_b\ _1 Q_{r,p,p'} \Phi_{g,\ell',m'} \right\rangle_{\partial \mathcal{D}_r}$	$\mathcal{N} \times G$	D
PNDFEMIncomingDirichletBC_bilinear	$\frac{1}{2} \left\langle \Phi_{\ell,m}^*, -L_{p,p'}^+ \Phi_{g,\ell',m'} - \ \vec{n}_b\ _1 Q_{p,p'}^+ \Phi_{g,\ell',m'} \right\rangle_{\partial \mathcal{D}_s}$	$\mathcal{N} \times G$	D
PNDFEMIncomingDirichletBC_linear	$-\frac{1}{2} \left\langle \Phi_{g,\ell,m}^*, -L_{p,p'}^- \Phi_{g,\ell',m'}^{\text{inc}} + \ \vec{n}_b\ _1 Q_{p,p'}^- \Phi_{g,\ell',m'}^{\text{inc}} \right\rangle_{\partial \mathcal{D}_s}$	$\mathcal{N}_s \times G$	N/A

^a Only the diagonal part (within-group) is assembled.

^b \mathcal{N}_s is the number of source transport functions (excluding the zero function). If there are no source functions, it is equal to 0.

3.4.7.12 DFEM- P_1

In this subsection, we detail the weak formulation in the particular P_1 case with one energy group. We do not assume any particular dimension and define:

$$\phi \equiv \Phi_{0,0} \quad , \quad \vec{J} \equiv \{\Phi_{1,0}, \Phi_{1,1}, \Phi_{1,-1}\} \quad (232)$$

If the spatial dimension is less than 3, we drop the last components of that vector accordingly so that it contains only the non-trivial currents. We first show what each volumetric, interface and boundary terms become and then summarize it by giving the entire P_1 weak formulation.

Streaming kernel:

$$\begin{aligned} (\mathbb{L}_2^* \Psi^*, \Psi) &= \left(-\vec{D} \cdot \vec{\nabla} \Phi^*, \Phi \right)_{\mathcal{D}} \\ &= \left(-\vec{\nabla} \phi^*, \vec{J} \right)_{\mathcal{D}} - \left(\vec{\nabla} \cdot \vec{J}^*, \phi \right)_{\mathcal{D}} . \end{aligned} \quad (233)$$

Collision kernel:

$$\begin{aligned} (\mathbb{L}_2^* \Psi^*, \Psi) &= (\Phi^*, \Sigma_t \mathbf{P} \Phi)_{\mathcal{D}} \\ &= (\phi^*, \Sigma_t \phi)_{\mathcal{D}} + \left(\vec{J}^*, 3 \Sigma_t \vec{J} \right)_{\mathcal{D}} . \end{aligned} \quad (234)$$

Scattering kernel:

$$\begin{aligned} (\Psi^*, \mathbb{S} \Psi) &= (\mathbf{P} \Phi^*, \Sigma_s \Phi)_{\mathcal{D}} \\ &= (\phi^*, \Sigma_{s,0} \phi)_{\mathcal{D}} + \left(\vec{J}^*, 3 \Sigma_{s,1} \vec{J} \right)_{\mathcal{D}} . \end{aligned} \quad (235)$$

Source kernel:

$$(\Psi^*, \mathbf{Q}^{\text{ext}}) = (\phi^*, Q_0)_{\mathcal{D}} + \left(\vec{J}^*, 3 Q_1 \right)_{\mathcal{D}} . \quad (236)$$

Filtering kernel:

$$(\Psi^*, \mathbf{Q}^{\text{ext}}) = (\Phi^*, \sigma_f(\vec{x}) \mathbf{P} \mathbf{F} \Phi) \quad (237)$$

$$= - \left(\vec{J}^*, 3 \sigma_f(\vec{x}) \log \sigma_{\text{filterType}}(0.5) \vec{J} \right)_{\mathcal{D}} . \quad (238)$$

Transient kernel:

$$\left(\Psi^*, \frac{\partial}{\partial t} \left(\frac{\Psi}{\mathbf{v}} \right) \right) = \left(\phi^*, \frac{1}{v} \frac{\partial \phi}{\partial t} \right)_{\mathcal{D}} + \left(\vec{J}^*, 3 \frac{1}{v} \frac{\partial \vec{J}}{\partial t} \right)_{\mathcal{D}} . \quad (239)$$

Planckian kernel:

$$(\Psi^*, \mathbf{B}) = (\phi^*, B)_{\mathcal{D}} . \quad (240)$$

Interior facet terms:

$$I = \left\langle \llbracket \Phi^* \rrbracket, -\vec{D} \cdot \vec{n} \llbracket \Phi \rrbracket + \frac{1}{2} \|\vec{n}\|_1 \mathbf{P} \llbracket \Phi \rrbracket \right\rangle_{\Gamma_{\text{int}}} , \quad (241)$$

$$= - \left\langle \llbracket \vec{J}^* \cdot \vec{n} \rrbracket, \llbracket \phi \rrbracket \right\rangle_{\Gamma_{\text{int}}} - \left\langle \llbracket \phi^* \rrbracket, \llbracket \vec{J} \cdot \vec{n} \rrbracket \right\rangle_{\Gamma_{\text{int}}} \quad (242)$$

$$+ \left\langle \frac{\|\vec{n}\|_1}{2} \llbracket \phi^* \rrbracket, \llbracket \phi \rrbracket \right\rangle_{\Gamma_{\text{int}}} + \left\langle \frac{3 \|\vec{n}\|_1}{2} \llbracket \vec{J}^* \rrbracket, \llbracket \vec{J} \rrbracket \right\rangle_{\Gamma_{\text{int}}} . \quad (243)$$

Exterior facet terms are given by:

$$\frac{1}{2} \left\langle \Phi_{g,\ell,m}^*, -\vec{n}_b \cdot \vec{D}_{p,p'} \Phi_{g,\ell',m'} + \|\vec{n}_b\|_1 P_{p,p'} \Phi_{g,\ell',m'} \right\rangle_{\partial\mathcal{D}} = -\frac{1}{2} \left\langle \vec{J}^* \cdot \vec{n}_b, \phi \right\rangle_{\partial\mathcal{D}} - \frac{1}{2} \left\langle \phi^*, \vec{J} \cdot \vec{n}_b \right\rangle_{\partial\mathcal{D}} \quad (244)$$

$$+ \left\langle \frac{\|\vec{n}_b\|_1}{2} \phi^*, \phi \right\rangle_{\partial\mathcal{D}} + \left\langle \frac{3\|\vec{n}_b\|_1}{2} \vec{J}^*, \vec{J} \right\rangle_{\partial\mathcal{D}}. \quad (245)$$

To explicit the remaining boundary terms, we first give the expression of the P_1 matrices from Table 36:

$$\begin{aligned} \mathbf{Q}^\pm &= \int_{\pm\vec{\Omega} \cdot \vec{n}_b > 0} \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}) d\Omega = \frac{1}{w^2} \int_{\pm\vec{\Omega} \cdot \vec{n}_b > 0} \begin{pmatrix} 1 & 3\vec{\Omega} \\ 3\vec{\Omega} & 9\vec{\Omega}\vec{\Omega} \end{pmatrix} d\Omega \\ &= \frac{1}{2w} \begin{pmatrix} 1 & \pm\frac{3}{2}\vec{n}_b \\ \pm\frac{3}{2}\vec{n}_b & 3\mathbb{I} \end{pmatrix}, \end{aligned} \quad (246)$$

$$\begin{aligned} \mathbf{L}^\pm &= \int_{\pm\vec{\Omega} \cdot \vec{n}_b > 0} \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}) |\vec{\Omega} \cdot \vec{n}_b| d\Omega = \frac{1}{w^2} \int_{\pm\vec{\Omega} \cdot \vec{n}_b > 0} \begin{pmatrix} 1 & 3\vec{\Omega} \\ 3\vec{\Omega} & 9\vec{\Omega}\vec{\Omega} \end{pmatrix} |\vec{\Omega} \cdot \vec{n}_b| d\Omega \\ &= \frac{1}{4w} \begin{pmatrix} 1 & \pm 2\vec{n}_b \\ \pm 2\vec{n}_b & \frac{9}{4}\mathbb{I} + \frac{9}{4}\vec{n}_b\vec{n}_b \end{pmatrix}, \end{aligned} \quad (247)$$

$$\begin{aligned} \mathbf{Q}_r &= \int_S \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}_r) d\Omega = \frac{1}{w^2} \int_S \begin{pmatrix} 1 & 3(\vec{\Omega} - 2(\vec{\Omega} \cdot \vec{n}_b)\vec{n}_b) \\ 3\vec{\Omega} & 9\vec{\Omega}(\vec{\Omega} - 2(\vec{\Omega} \cdot \vec{n}_b)\vec{n}_b) \end{pmatrix} d\Omega \\ &= \frac{1}{w} \begin{pmatrix} 1 & 0 \\ 0 & 3\mathbb{I} - 6\vec{n}_b\vec{n}_b \end{pmatrix}, \end{aligned} \quad (248)$$

$$\begin{aligned} \mathbf{L}_r &= \int_S \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}_r) |\vec{\Omega} \cdot \vec{n}_b| d\Omega = \frac{1}{w^2} \int_S \begin{pmatrix} 1 & 3(\vec{\Omega} - 2(\vec{\Omega} \cdot \vec{n}_b)\vec{n}_b) \\ 3\vec{\Omega} & 9\vec{\Omega}(\vec{\Omega} - 2(\vec{\Omega} \cdot \vec{n}_b)\vec{n}_b) \end{pmatrix} |\vec{\Omega} \cdot \vec{n}_b| d\Omega \\ &= \frac{1}{w} \begin{pmatrix} 0 & -\vec{n}_b \\ \vec{n}_b & 0 \end{pmatrix}. \end{aligned} \quad (249)$$

The reflecting boundary terms are then given by:

$$\begin{aligned} \frac{1}{2} \left\langle \Phi_{g,\ell,m}^*, -L_{r,p,p'} \Phi_{g,\ell',m'} - \|\vec{n}_b\|_1 Q_{r,p,p'} \Phi_{g,\ell',m'} \right\rangle_{\partial\mathcal{D}_r} &= -\frac{1}{2} \left\langle \vec{J}^* \cdot \vec{n}_b, \phi \right\rangle_{\partial\mathcal{D}_r} + \frac{1}{2} \left\langle \phi^*, \vec{J} \cdot \vec{n}_b \right\rangle_{\partial\mathcal{D}_r} \\ &\quad - \frac{1}{2} \left\langle \phi^*, \|\vec{n}_b\|_1 \phi \right\rangle_{\partial\mathcal{D}_r} - \frac{3}{2} \left\langle \vec{J}^*, \|\vec{n}_b\|_1 \vec{J} \right\rangle_{\partial\mathcal{D}_r} + 3 \left\langle \vec{J}^* \cdot \vec{n}_b, \|\vec{n}_b\|_1 \vec{J} \cdot \vec{n}_b \right\rangle_{\partial\mathcal{D}_r}. \end{aligned} \quad (250)$$

The bilinear contribution of the incoming Dirichlet boundary is given by:

$$\begin{aligned} \frac{1}{2} \left\langle \Phi_{\ell,m}^*, -L_{p,p'}^+ \Phi_{g,\ell',m'} - \|\vec{n}_b\|_1 Q_{p,p'}^+ \Phi_{g,\ell',m'} \right\rangle_{\partial\mathcal{D}_s} &= -\frac{1}{8} \left\langle \phi^*, \phi \right\rangle_{\partial\mathcal{D}_s} - \frac{1}{4} \left\langle \vec{J}^* \cdot \vec{n}_b, \phi \right\rangle_{\partial\mathcal{D}_s} - \frac{1}{4} \left\langle \phi^*, \vec{J} \cdot \vec{n}_b \right\rangle_{\partial\mathcal{D}_s} - \frac{9}{32} \left\langle \vec{J}^*, \vec{J} \right\rangle_{\partial\mathcal{D}_s} - \frac{9}{32} \left\langle \vec{J}^* \cdot \vec{n}_b, \vec{J} \cdot \vec{n}_b \right\rangle_{\partial\mathcal{D}_s} \\ &\quad - \frac{1}{4} \left\langle \phi^*, \|\vec{n}_b\|_1 \phi \right\rangle_{\partial\mathcal{D}_s} - \frac{3}{8} \left\langle \vec{J}^* \cdot \vec{n}_b, \|\vec{n}_b\|_1 \phi \right\rangle_{\partial\mathcal{D}_s} - \frac{3}{8} \left\langle \phi^*, \|\vec{n}_b\|_1 \vec{J} \cdot \vec{n}_b \right\rangle_{\partial\mathcal{D}_s} - \frac{3}{4} \left\langle \vec{J}^*, \|\vec{n}_b\|_1 \vec{J} \right\rangle_{\partial\mathcal{D}_s}, \end{aligned} \quad (251)$$

and the linear contribution of the incoming Dirichlet boundary (no minus sign because on the right hand side) is given by:

$$\frac{1}{2} \left\langle \Phi_{g,\ell,m}^*, -L_{p,p'}^- \Phi_{g,\ell',m'}^{\text{inc}} + \|\vec{n}_b\|_1 Q_{p,p'}^- \Phi_{g,\ell',m'}^{\text{inc}} \right\rangle_{\partial\mathcal{D}_s} \quad (252)$$

$$\begin{aligned}
&= -\frac{1}{8} \langle \phi^*, \phi^{\text{inc}} \rangle_{\partial \mathcal{D}_s} + \frac{1}{4} \langle \vec{J}^* \cdot \vec{n}_b, \phi^{\text{inc}} \rangle_{\partial \mathcal{D}_s} + \frac{1}{4} \langle \phi^*, \vec{J}^{\text{inc}} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_s} - \frac{9}{32} \langle \vec{J}^*, \vec{J}^{\text{inc}} \rangle_{\partial \mathcal{D}_s} - \frac{9}{32} \langle \vec{J}^* \cdot \vec{n}_b, \vec{J}^{\text{inc}} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_s} \\
&+ \frac{1}{4} \langle \phi^*, ||\vec{n}_b||_1 \phi^{\text{inc}} \rangle_{\partial \mathcal{D}_s} - \frac{3}{8} \langle \vec{J}^* \cdot \vec{n}_b, ||\vec{n}_b||_1 \phi^{\text{inc}} \rangle_{\partial \mathcal{D}_s} - \frac{3}{8} \langle \phi^*, ||\vec{n}_b||_1 \vec{J}^{\text{inc}} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_s} + \frac{3}{4} \langle \vec{J}^*, ||\vec{n}_b||_1 \vec{J}^{\text{inc}} \rangle_{\partial \mathcal{D}_s},
\end{aligned}$$

In summary, the bilinear form of the DFEM- P_1 method with one energy group is given by:

$$\begin{aligned}
b_{\text{DFEM-}P_1} &= \left(-\vec{\nabla} \phi^*, \vec{J} \right)_{\mathcal{D}} - \left(\vec{\nabla} \cdot \vec{J}^*, \phi \right)_{\mathcal{D}} + (\phi^*, \Sigma_a \phi)_{\mathcal{D}} + \left(\vec{J}^*, 3\Sigma_{\text{tr}} \vec{J} \right)_{\mathcal{D}} \\
&+ \left(\phi^*, \frac{1}{v} \frac{\partial \phi}{\partial t} \right)_{\mathcal{D}} + \left(\vec{J}^*, 3\frac{1}{v} \frac{\partial \vec{J}}{\partial t} \right)_{\mathcal{D}} + (\phi^*, B)_{\mathcal{D}} - \left(\vec{J}^*, 3\sigma_f(\vec{x}) \log \sigma_{\text{filterType}}(0.5) \vec{J} \right)_{\mathcal{D}} \\
&- \left\langle \llbracket \vec{J}^* \cdot \vec{n} \rrbracket, \llbracket \phi \rrbracket \right\rangle_{\Gamma_{\text{int}}} - \left\langle \llbracket \phi^* \rrbracket, \llbracket \vec{J} \cdot \vec{n} \rrbracket \right\rangle_{\Gamma_{\text{int}}} + \left\langle \frac{||\vec{n}||_1}{2} \llbracket \phi^* \rrbracket, \llbracket \phi \rrbracket \right\rangle_{\Gamma_{\text{int}}} + \left\langle \frac{3||\vec{n}||_1}{2} \llbracket \vec{J}^* \rrbracket, \llbracket \vec{J} \rrbracket \right\rangle_{\Gamma_{\text{int}}} \\
&- \frac{1}{2} \langle \vec{J}^* \cdot \vec{n}_b, \phi \rangle_{\partial \mathcal{D}} - \frac{1}{2} \langle \phi^*, \vec{J} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}} + \left\langle \frac{||\vec{n}_b||_1}{2} \phi^*, \phi \right\rangle_{\partial \mathcal{D}} + \left\langle \frac{3||\vec{n}_b||_1}{2} \vec{J}^*, \vec{J} \right\rangle_{\partial \mathcal{D}} \tag{253} \\
&- \frac{1}{2} \langle \vec{J}^* \cdot \vec{n}_b, \phi \rangle_{\partial \mathcal{D}_r} + \frac{1}{2} \langle \phi^*, \vec{J} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_r} - \frac{1}{2} \langle \phi^*, ||\vec{n}_b||_1 \phi \rangle_{\partial \mathcal{D}_r} - \frac{3}{2} \langle \vec{J}^*, ||\vec{n}_b||_1 \vec{J} \rangle_{\partial \mathcal{D}_r} + 3 \langle \vec{J}^* \cdot \vec{n}_b, ||\vec{n}_b||_1 \vec{J} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_r} \\
&- \frac{1}{8} \langle \phi^*, \phi \rangle_{\partial \mathcal{D}_s} - \frac{1}{4} \langle \vec{J}^* \cdot \vec{n}_b, \phi \rangle_{\partial \mathcal{D}_s} - \frac{1}{4} \langle \phi^*, \vec{J} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_s} - \frac{9}{32} \langle \vec{J}^*, \vec{J} \rangle_{\partial \mathcal{D}_s} - \frac{9}{32} \langle \vec{J}^* \cdot \vec{n}_b, \vec{J} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_s} \\
&- \frac{1}{4} \langle \phi^*, ||\vec{n}_b||_1 \phi \rangle_{\partial \mathcal{D}_s} - \frac{3}{8} \langle \vec{J}^* \cdot \vec{n}_b, ||\vec{n}_b||_1 \phi \rangle_{\partial \mathcal{D}_s} - \frac{3}{8} \langle \phi^*, ||\vec{n}_b||_1 \vec{J} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_s} - \frac{3}{4} \langle \vec{J}^*, ||\vec{n}_b||_1 \vec{J} \rangle_{\partial \mathcal{D}_s},
\end{aligned}$$

and the linear form of the DFEM- P_1 method with one energy group reads:

$$\begin{aligned}
l_{\text{DFEM-}P_1} &= (\phi^*, Q_0)_{\mathcal{D}} + \left(\vec{J}^*, 3Q_1 \right)_{\mathcal{D}} \tag{254} \\
&- \frac{1}{8} \langle \phi^*, \phi^{\text{inc}} \rangle_{\partial \mathcal{D}_s} + \frac{1}{4} \langle \vec{J}^* \cdot \vec{n}_b, \phi^{\text{inc}} \rangle_{\partial \mathcal{D}_s} + \frac{1}{4} \langle \phi^*, \vec{J}^{\text{inc}} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_s} - \frac{9}{32} \langle \vec{J}^*, \vec{J}^{\text{inc}} \rangle_{\partial \mathcal{D}_s} - \frac{9}{32} \langle \vec{J}^* \cdot \vec{n}_b, \vec{J}^{\text{inc}} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_s} \\
&+ \frac{1}{4} \langle \phi^*, ||\vec{n}_b||_1 \phi^{\text{inc}} \rangle_{\partial \mathcal{D}_s} - \frac{3}{8} \langle \vec{J}^* \cdot \vec{n}_b, ||\vec{n}_b||_1 \phi^{\text{inc}} \rangle_{\partial \mathcal{D}_s} - \frac{3}{8} \langle \phi^*, ||\vec{n}_b||_1 \vec{J}^{\text{inc}} \cdot \vec{n}_b \rangle_{\partial \mathcal{D}_s} + \frac{3}{4} \langle \vec{J}^*, ||\vec{n}_b||_1 \vec{J}^{\text{inc}} \rangle_{\partial \mathcal{D}_s}.
\end{aligned}$$

Some differences with the P1C method can be found (defined in [63]). First of all, the streaming terms are slightly different because the volumetric term containing $\vec{\nabla} \phi$ is integrated by parts in [63]. Second of all, some of the interface terms have different coefficients, mainly because the penalty term in the DFEM- P_N method has been modified using a Lax-Friedrich flux to reduce the number of terms for large values of N and thus the cost of the method.

3.5 Diffusion

Diffusion weak forms are derived for CDE. We will use the same FEM notation as in Section 3.3.1. CFEM schemes are given in the first subsection Section 3.5.1, followed by the DFEM scheme in Section 3.5.2. While CFEM diffusion schemes have a smaller number of unknowns for achieving a given accuracy than DFEM diffusion schemes, DFEM schemes allows the discontinuity of the solution on element sides. Connections between transport schemes and diffusion schemes are given in Section 3.6.

3.5.1 CFEM-Diffusion

We multiply the test function Φ^* , which are zero at the Dirichlet boundary, with Eq. (27) and integrate over the phase space

$$\left(\Phi^*, \frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}}\right)\right)_{\mathcal{D}} - \left(\Phi^*, \vec{\nabla} \cdot \mathbb{D} \vec{\nabla} \Phi\right)_{\mathcal{D}} + (\Phi^*, \mathbb{L}_3 \Phi)_{\mathcal{D}} = (\Phi^*, \mathbf{Q}_0^{\text{ext}})_{\mathcal{D}} + (\Phi^*, \mathbb{S}_d \Phi),$$

and apply the integration by parts and Gauss theorem on the diffusion term:

$$\left(\Phi^*, \frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}}\right)\right)_{\mathcal{D}} + \left(\vec{\nabla} \Phi^*, \mathbb{D} \vec{\nabla} \Phi\right)_{\mathcal{D}} - \left(\Phi^*, \mathbb{D} \vec{\nabla} \Phi \cdot \vec{n}_b\right)_{\partial \mathcal{D}_2} - \left(\Phi^*, \mathbb{D} \vec{\nabla} \Phi \cdot \vec{n}_b\right)_{\partial \mathcal{D}_3} + (\Phi^*, \mathbb{L}_3 \Phi)_{\mathcal{D}} = (\Phi^*, \mathbf{Q}_0^{\text{ext}})_{\mathcal{D}} + (\Phi^*, \mathbb{S}_d \Phi).$$

We then substitute the boundary conditions, Eq. (27c) and Eq. (27d) into the boundary terms and obtain

$$\left(\Phi^*, \frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}}\right)\right)_{\mathcal{D}} + \left(\vec{\nabla} \Phi^*, \mathbb{D} \vec{\nabla} \Phi\right)_{\mathcal{D}} - (\Phi^*, \mathbf{J}^{\text{net}})_{\partial \mathcal{D}_2} + \left(\Phi^*, \frac{1}{2} \mathbf{e} \Phi\right)_{\partial \mathcal{D}_3} - (\Phi^*, 2\mathbf{e} \mathbf{J}^{\text{inc}})_{\partial \mathcal{D}_3} + (\Phi^*, \mathbb{L}_3 \Phi)_{\mathcal{D}} = (\Phi^*, \mathbf{Q}_0^{\text{ext}})_{\mathcal{D}} + (\Phi^*, \mathbb{S}_d \Phi).$$

Defining

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi) \equiv \left(\Phi^*, \frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}}\right)\right)_{\mathcal{D}} + \left(\vec{\nabla} \Phi^*, \mathbb{D} \vec{\nabla} \Phi\right)_{\mathcal{D}} + \left(\Phi^*, \frac{1}{2} \mathbf{e} \Phi\right)_{\partial \mathcal{D}_3} + (\Phi^*, \mathbb{L}_3 \Phi)_{\mathcal{D}} - (\Phi^*, \mathbb{S}_d \Phi), \quad (255)$$

$$\mathbb{l}_{\text{diff}}(\Phi^*) \equiv (\Phi^*, \mathbf{J}^{\text{net}})_{\partial \mathcal{D}_2} + (\Phi^*, 2\mathbf{e} \mathbf{J}^{\text{inc}})_{\partial \mathcal{D}_3} + (\Phi^*, \mathbf{Q}_0^{\text{ext}})_{\mathcal{D}}, \quad (256)$$

the weak form is: find a solution Φ in a function space, where all functions satisfy the Dirichlet boundary, such that

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi) = \mathbb{l}_{\text{diff}}(\Phi^*), \forall \Phi^*. \quad (257)$$

We can do the same procedure for the adjoint equation and end up with: Find a solution Φ^* in a function space V , such that

$$\mathbb{b}_{\text{diff}}^*(\Phi^*, \Phi) = \mathbb{l}_{\text{diff}}^*(\Phi), \forall \Phi, \quad (258)$$

where

$$\mathbb{b}_{\text{diff}}^*(\Phi^*, \Phi) \equiv - \left(\frac{1}{\mathbf{v}} \frac{\partial}{\partial t} \Phi^*, \Phi\right)_{\mathcal{D}} + \left(\mathbb{D} \vec{\nabla} \Phi^*, \vec{\nabla} \Phi\right)_{\mathcal{D}} + \left(\frac{1}{2} \mathbf{e} \Phi^*, \Phi\right)_{\partial \mathcal{D}_3} + (\mathbb{L}_3 \Phi^*, \Phi)_{\mathcal{D}} - (\mathbb{S}_d^* \Phi^*, \Phi), \quad (259)$$

$$\mathbb{l}_{\text{diff}}^*(\Phi) \equiv (\Phi, \mathbf{J}^{\text{net}*})_{\partial \mathcal{D}_2} + (\Phi, 2\mathbf{e} \mathbf{J}^{\text{inc}*})_{\partial \mathcal{D}_3} + (\Phi, \mathbf{Q}_0^{\text{ext}*})_{\mathcal{D}}. \quad (260)$$

It can be seen that with zero initial and final conditions or steady-state, $\mathbb{b}_{\text{diff}}^*(\Phi^*, \Phi) = \mathbb{b}_{\text{diff}}(\Phi^*, \Phi)$. The same bilinear form for the primal equation can be applied to the adjoint equation. We can apply FEM for discretizing the weak form. As described in Section 3.3.1, we can still use the same notation for the discretized weak form. We expand all the kernels in Eq. (257) and extra kernels for various radiation types in order to make a direct correspondence to the code.

3.5.1.1 Steady-state

Diffusion kernel:

$$\left(\vec{\nabla}\Phi^*, \mathbb{D}\vec{\nabla}\Phi\right)_{\mathcal{D}} = \sum_{g=1}^G \left(\vec{\nabla}\Phi_g^*, D_g \vec{\nabla}\Phi_g\right)_{\mathcal{D}}. \quad (261)$$

Removal kernel:

$$\left(\Phi^*, \mathbb{L}_3\Phi\right)_{\mathcal{D}} = \sum_{g=1}^G \left(\vec{\nabla}\Phi_g^*, \Sigma_{r,g} \vec{\nabla}\Phi_g\right)_{\mathcal{D}}. \quad (262)$$

Scattering kernel:

$$\left(\Phi^*, \mathbb{S}_d\Phi\right) = \sum_{g=1}^G \sum_{g' \neq g} \left(\Phi_{g'}^*, \Sigma_{s,g' \rightarrow g} \Phi_{g'}\right)_{\mathcal{D}}, \quad (263)$$

$$= \sum_{g=1}^G \left(\Phi_g^*, Q_g\right)_{\mathcal{D}}. \quad (264)$$

External source kernel:

$$\left(\Phi^*, Q_0^{\text{ext}}\right)_{\mathcal{D}} = \sum_{g=1}^G \left(\Phi_g^*, Q_{g,0}^{\text{ext}}\right)_{\mathcal{D}}. \quad (265)$$

3.5.1.2 Boundary conditions

The Dirichlet condition is applied on the nodes of boundary $\partial\mathcal{D}_1$.

Neumann boundary condition:

$$\left(\Phi^*, \mathbf{J}^{\text{net}}\right)_{\partial\mathcal{D}_2} = \sum_{g=1}^G \left(\Phi_g^*, J_g^{\text{net}}\right)_{\partial\mathcal{D}_2}. \quad (266)$$

When $J_g^{\text{net}} = 0$, no boundary condition is needed. The homogenous Neumann boundary condition is naturally imposed. Non-homogeneous Neumann boundary condition is currently not implemented. Robin boundary condition:

$$\left(\Phi^*, \frac{1}{2}\mathbf{e}\Phi - 2\mathbf{e}\mathbf{J}^{\text{inc}}\right)_{\partial\mathcal{D}_3} = \sum_{g=1}^G \left(\Phi_g^*, \frac{1}{2}e_g\Phi_g - 2e_gJ_g^{\text{inc}}\right)_{\partial\mathcal{D}_3}. \quad (267)$$

3.5.1.3 Transient kernel

$$\left(\Phi^*, \frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}}\right)\right)_{\mathcal{D}} = \sum_{g=1}^G \left(\Phi_g^*, \frac{\partial}{\partial t} \left(\frac{\Phi_g}{v_g}\right)\right)_{\mathcal{D}}. \quad (268)$$

We currently assume v_g is time independent and have

$$\left(\Phi^*, \frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}}\right)\right)_{\mathcal{D}} = \sum_{g=1}^G \left(\Phi_g^*, \frac{1}{v_g} \frac{\partial \Phi_g}{\partial t}\right)_{\mathcal{D}}. \quad (269)$$

3.5.1.4 Neutron

Prompt fission kernel:

$$(\Phi^*, \mathbb{F}_0 \Phi) = \sum_{g=1}^G \left(\Phi_g^*, \frac{\bar{\chi}_g}{k} F \right)_{\mathcal{D}}, \quad (270)$$

where F is the fission source as in Eq. (138). It is noted that for transients $\bar{\chi}_g = \chi_{p,g}(1 - \beta)$ and in steady-state cases $\bar{\chi}_g = \chi_g$ as in the fission kernels of other schemes.

Delayed neutron kernel:

$$(\Phi^*, \mathbb{C}_0 \mathbf{C}) = \sum_{g=1}^G \left(\Phi_g^*, \frac{1}{k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i \right)_{\mathcal{D}}. \quad (271)$$

The DNP concentrations are evaluated similarly as in Eq. (140a).

3.5.1.5 Thermal radiation

Planckian emission kernel:

$$(\Phi^*, \mathbf{S}_{\text{BB}}(T)) = \sum_{g=1}^G \left(\Phi_g^*, S_{\text{BB},g} \right)_{\mathcal{D}}. \quad (272)$$

$S_{\text{BB},g}$ is defined by Eq. (62).

3.5.1.6 R-spherical coordinate

No extra kernel is needed in R-spherical coordinate for the diffusion equation.

3.5.1.7 RZ-axial symmetric coordinate

No extra kernel is needed in RZ-axial symmetric coordinate for the diffusion equation.

3.5.1.8 Mathematical adjoint

Mathematical adjoint is equal to physical adjoint for the diffusion equation.

3.5.1.9 Summary of MOOSE objects

The MOOSE objects used by CFEM-Diffusion are listed in Table 39, Table 40, Table 41, Table 42 and Table 43.

Table 39 CFEM-Diffusion kernels.

Kernel	Type	Jacobian
$\left(\vec{\nabla}\Phi_g^*, D_g \vec{\nabla}\Phi_g\right)_{\mathcal{D}}$	DiffusionApproximation	Y
$\left(\vec{\nabla}\Phi_g^*, \Sigma_{r,g} \vec{\nabla}\Phi_g\right)_{\mathcal{D}}$	Removal	Y
$\left(\Phi_{g'}^*, \Sigma_{s,g' \rightarrow g} \Phi_{g'}\right)_{\mathcal{D}}$	DiffusionScattering	Y
$\left(\Phi_{g'}^*, Q_g\right)_{\mathcal{D}}$	DiffusionGroupScattering	N
$\left(\Phi_{g'}^*, Q_{g,0}^{\text{ext}}\right)_{\mathcal{D}}$	CDSOURCE	N/A
$\left(\Phi_{g'}^*, \bar{\chi}_g F\right)_{\mathcal{D}}$	DiffusionFission	Y
$\left(\Phi_{g'}^*, \frac{1}{k} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i\right)_{\mathcal{D}}$	CDDelayedNeutron	Y ^a
$\left(\Phi_{g'}^*, S_{\text{BB},g}\right)_{\mathcal{D}}$	DiffusionPlanck	N
$\left(\Phi_{g'}^*, \frac{1}{v_g} \frac{\partial \Phi_g}{\partial t}\right)_{\mathcal{D}}$	CDTimeDerivative	Y

^a Jacobian is assembled only when DNPs are treated as material properties.

Table 40 CFEM-Diffusion kernels (continued).

Type	Condition	Number of kernels
DiffusionApproximation	D_g^{a}	G
Removal	$\Sigma_{r,g}$	G
DiffusionScattering	$\Sigma_{s,0}^{g' \rightarrow g}$	NNZ^{b}
DiffusionGroupScattering	Q_g	G
CDSOURCE	Q_g^{ext}	$N_s^{\text{c}} \times G$
DiffusionFission	$\bar{\chi}_g$	G
CDDelayedNeutron	$\chi_{d,g,i}, i = 1, \dots, I$	G
DiffusionPlanck	\mathcal{D}	G
CDTimeDerivative	v_g	G

^a Means where D_g is available.

^b Number of non-zeros in the scattering matrix, i.e. all (g', g) s whose $\Sigma_{s,0}^{g' \rightarrow g} = \left\{ \Sigma_{s,\ell}^{g' \rightarrow g}, \forall \ell = 0, \dots, L \right\} \neq 0$.

^c Number of external source functions.

Table 41 CFEM-Diffusion boundary conditions.

Type	Equation	Number of BCs	Jacobian
CDPartialCurrentBC	$\left(\Phi_{g'}^*, \frac{1}{2} e_g \Phi_g - 2e_g J_g^{\text{inc}}\right)_{\partial \mathcal{D}_3}$	$N_s^{\text{a}} \times G$	Y

^a N_s is the number of surface source transport functions. If there are no source functions, it is equal to 1.

Table 42 CFEM-Diffusion auxiliary kernels.

Name	Equation	Type	Number of kernels	Condition
Fission source (F)	Eq. (138)	MaterialFissionSource	1	$\nu \Sigma_{f,g'}, g = 1, \dots, G$
DNP ($C_i, i = 1, \dots, I$)	Eq. (140a)	DelayedNeutronPrecursor	I	$\nu \Sigma_{f,g'}, g = 1, \dots, G$

Table 43 CFEM-Diffusion auxiliary materials.

Name	Equation	Type	Number of materials	Condition
Q_g	Eq. (23)	SsrcMaterial	G	$\Sigma_{s,0}^{g' \rightarrow g}, g' = 1, \dots, G; g = 1, \dots, G$
F	Eq. (138)	FsrcMaterial	1	$\nu \Sigma_{f,g}, g = 1, \dots, G$
$C_i, i = 1, \dots, I$	Eq. (140a)	DNPMaterial	I	$\nu \Sigma_{f,g}, g = 1, \dots, G$

3.5.2 DFEM-Diffusion

We directly give the weak form suitable for DFEM. The derivation can be found in literature [64]. We will need a few terms defined on the interior sides of the mesh. It is typical to impose the Dirichlet boundary condition weakly.

$$\begin{aligned} \mathbb{b}_{\text{diff-DFEM}}(\Phi^*, \Phi) = & \mathbb{b}_{\text{diff}}(\Phi^*, \Phi) + \\ & (\llbracket \Phi^* \rrbracket, \llbracket \kappa \Phi \rrbracket)_{\Gamma_{\text{int}}} + \left(\llbracket \Phi^* \rrbracket, \llbracket \mathbb{D} \vec{\nabla} \Phi \cdot \vec{n} \rrbracket \right)_{\Gamma_{\text{int}}} + \epsilon \left(\llbracket \mathbb{D} \vec{\nabla} \Phi^* \cdot \vec{n} \rrbracket, \llbracket \Phi \rrbracket \right)_{\Gamma_{\text{int}}} + \\ & (\Phi^*, \kappa \Phi)_{\partial \mathcal{D}_1} - \left(\Phi^*, \mathbb{D} \vec{\nabla} \Phi \cdot \vec{n}_b \right)_{\partial \mathcal{D}_1} - \epsilon \left(\mathbb{D} \vec{\nabla} \Phi^* \cdot \vec{n}_b, \Phi \right)_{\partial \mathcal{D}_1}, \end{aligned} \quad (273)$$

$$\mathbb{l}_{\text{diff-DFEM}}(\Phi^*) = \mathbb{l}_{\text{diff}}(\Phi^*) + (\Phi^*, \kappa \Phi_b)_{\partial \mathcal{D}_1} - \epsilon \left(\mathbb{D} \vec{\nabla} \Phi^* \cdot \vec{n}_b, \Phi_b \right)_{\partial \mathcal{D}_1}. \quad (274)$$

We recall the jump operator $\llbracket \cdot \rrbracket$ and average operator $\llbracket \cdot \rrbracket$ on Γ_{int} defined in Eq. (211a):

$$\begin{aligned} \llbracket a(\vec{x}) \rrbracket & \equiv a^+ - a^-, \\ \llbracket a(\vec{x}) \rrbracket & \equiv \frac{a^+ + a^-}{2}, \\ a^\pm(\vec{x}) & \equiv \lim_{s \rightarrow 0^\pm} a(\vec{x} + s\vec{n}). \end{aligned}$$

Operator κ is defined as

$$\kappa \equiv \begin{bmatrix} \kappa_1 \\ \kappa_2 \\ \vdots \\ \kappa_G \end{bmatrix}, \quad (276)$$

where the penalty coefficients $\kappa_g, g = 1, \dots, G$ are evaluated with the following formulation:

$$\kappa_g = \begin{cases} \llbracket c(p) \frac{D_g}{h_\perp} \rrbracket, & \vec{x} \in \Gamma_{\text{int}} \\ c(p) \frac{D_g}{h_\perp}, & \vec{x} \in \partial \mathcal{D}_1 \end{cases}, \quad (277)$$

$$c(p) = 2p(p+1), \quad (278)$$

where p is the local polynomial order of the elements; h_\perp is the length of the element orthogonal to the side. Switching parameter ϵ can be 1, 0, or -1, corresponding to symmetric, incomplete, or non-symmetric penalty method (SIP, IIP and NIP, respectively). When $\epsilon = 1$, the bilinear form is SPD with sufficiently large penalty coefficients. Those extra terms are consistent. It is noted that the weak form is indeed independent of the orientation of the interior normal.

3.5.2.1 Steady-state

All volumetric kernels are identical to those for CFEM-Diffusion.

We have the extra DGKernel:

$$\begin{aligned} & (\llbracket \Phi^* \rrbracket, \llbracket \kappa \Phi \rrbracket)_{\Gamma_{\text{int}}} + \left(\llbracket \Phi^* \rrbracket, \llbracket \mathbb{D} \vec{\nabla} \Phi \cdot \vec{n} \rrbracket \right)_{\Gamma_{\text{int}}} + \epsilon \left(\llbracket \mathbb{D} \vec{\nabla} \Phi^* \cdot \vec{n} \rrbracket, \llbracket \Phi \rrbracket \right)_{\Gamma_{\text{int}}} = \\ & \sum_{g=1}^G \left(\left(\llbracket \Phi_g^* \rrbracket, \llbracket \kappa_g \Phi_g \rrbracket \right)_{\Gamma_{\text{int}}} + \left(\llbracket \Phi_g^* \rrbracket, \llbracket D_g \vec{\nabla} \Phi_g \cdot \vec{n} \rrbracket \right)_{\Gamma_{\text{int}}} + \epsilon \left(\llbracket D_g \vec{\nabla} \Phi_g^* \cdot \vec{n} \rrbracket, \llbracket \Phi_g \rrbracket \right)_{\Gamma_{\text{int}}} \right). \end{aligned} \quad (279)$$

3.5.2.2 Boundary conditions

Dirichlet boundary condition:

$$\begin{aligned}
 & (\Phi^*, \kappa(\Phi - \Phi_b))_{\partial\mathcal{D}_1} - (\Phi^*, \mathbb{D} \vec{\nabla} \Phi \cdot \vec{n}_b)_{\partial\mathcal{D}_1} - \epsilon (\mathbb{D} \vec{\nabla} \Phi^* \cdot \vec{n}_b, (\Phi - \Phi_b))_{\partial\mathcal{D}_1} = \\
 & \sum_{g=1}^G \left((\Phi_g^*, \kappa_g(\Phi_g - \Phi_{b,g}))_{\partial\mathcal{D}_1} - (\Phi_g^*, D_g \vec{\nabla} \Phi_g \cdot \vec{n}_b)_{\partial\mathcal{D}_1} - \epsilon (D_g \vec{\nabla} \Phi_g^* \cdot \vec{n}_b, (\Phi_g - \Phi_{b,g}))_{\partial\mathcal{D}_1} \right). \quad (280)
 \end{aligned}$$

Currently only homogeneous Dirichlet boundary condition, i.e. $\Phi_b = 0$, is supported.

Neumann and Robin boundary conditions are identical to those for CFEM-Diffusion.

3.5.2.3 Transient kernel

All transient kernels are identical to those for CFEM-Diffusion.

3.5.2.4 Neutron

All neutron kernels are identical to those for CFEM-Diffusion.

3.5.2.5 Thermal radiation

All thermal radiation kernels are identical to those for CFEM-Diffusion.

3.5.2.6 R-spherical coordinate

No extra kernel is needed in R-spherical coordinate for the diffusion equation.

3.5.2.7 RZ-axial symmetric coordinate

No extra kernel is needed in RZ-axial symmetric coordinate for the diffusion equation.

3.5.2.8 Mathematical adjoint

The mathematical adjoint is equal to the physical adjoint when parameter ϵ in Eq. (273) is one, i.e. symmetric penalty method. We should not evaluate adjoint with incomplete and asymmetric penalty methods.

3.5.2.9 Summary of MOOSE objects

The MOOSE objects used by DFEM-Diffusion are almost identical to the objects of CFEM-Diffusion. It only has the DG kernel and the weakly imposed Dirichlet boundary condition, which is summarized below.

Table 44 DFEM-Diffusion DG kernel and Dirichlet boundary condition.

Type	Equation	Number of BCs	Jacobian
DGDiffusion ^a	$\begin{aligned} & \left(\llbracket \Phi_g^* \rrbracket, \llbracket \kappa_g \Phi_g \rrbracket \right)_{\Gamma_{\text{int}}} + \\ & \left(\llbracket \Phi_g^* \rrbracket, \llbracket D_g \vec{\nabla} \Phi_g \cdot \vec{n} \rrbracket \right)_{\Gamma_{\text{int}}} + \\ & \epsilon \left(\llbracket D_g \vec{\nabla} \Phi_g^* \cdot \vec{n} \rrbracket, \llbracket \Phi_g \rrbracket \right)_{\Gamma_{\text{int}}} \end{aligned}$	G	Y
DGDiffusionDirichletBC ^b	$\begin{aligned} & \left(\Phi_g^*, \kappa_g (\Phi_g - \Phi_{b,g}) \right)_{\partial \mathcal{D}_1} - \\ & \left(\Phi_g^*, D_g \vec{\nabla} \Phi_g \cdot \vec{n}_b \right)_{\partial \mathcal{D}_1} - \\ & \epsilon \left(D_g \vec{\nabla} \Phi_g^* \cdot \vec{n}_b, (\Phi_g - \Phi_{b,g}) \right)_{\partial \mathcal{D}_1} \end{aligned}$	G	Y

^a DG diffusion kernel is on Γ_{int} .

^b DG diffusion Dirichlet BC is on $\partial \mathcal{D}_1$.

3.6 Nonlinear diffusion acceleration

Nonlinear diffusion acceleration (NDA) is an effective angular acceleration scheme for the solution of the multi-group transport equations discretized with S_N . It introduces a low-order diffusion equation with additional nonlinear closure terms so that, at convergence, the low-order scalar fluxes become identical to the projection of the high order solution into the low order space. NDA is a generalization of the well-known coarse-mesh finite difference (CMFD) [65, 66] in the sense that instead of a simple low order balance equation, a full low order diffusion problem is solved. The closure terms are updated iteratively with a limited number of high-order transport updates of the angular fluxes. As a result, the majority of the work can be carried out in the low-order equation featuring a significantly smaller number of unknowns. Because the required number of transport updates is much smaller than the number of transport updates when solving the high-order equation directly the computational cost can be significantly reduced given that the transport sweeps account for the majority of the memory and CPU-time consumption. Transport updates can generally be done quite efficiently because only the streaming-collision operator needs be inverted. For example, updates can be conducted with the well-known matrix-free sweeping algorithm with DFEM S_N .

The NDA high order and low order equations constitute a nonlinear system of equations that is typically solved by fixed-point or Picard iteration, i.e. the iteration procedure alternates between diffusion solves and transport updates until convergence. In the first subsection, we describe the Picard iteration process using abstracted weak forms of the high-order and low-order equations. Stability including the effect of the choice of diffusion coefficients is briefly discussed. Fourier analysis for the convergence of the Picard iteration can be found in Ref. [67, 46, 68] and is not included in this manual. In Rattlesnake, we do not consider NDA solved by a Newton-type methods, that was suggested as a remedy to restore stability [69][70] because (1) it generally requires more transport updates and is hence more expensive and (2) is more difficult to implement. Current discussions are limited to *steady-state* equations for *common/neutron* transport although extensions to transient with other types of particles are possible. The next three subsections detail the weak forms for [SAAF-CFEM-SN](#), [LS-CFEM-SN](#) and [DFEM-SN](#) respectively. Discussions on the important implementation details are also included.

3.6.1 Picard iteration

The low-order weak form can be written as

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi) - \mathbb{F}_0(\Phi^*, \Phi) + \mathbb{C}[\Psi](\Phi^*, \Phi) = \mathbb{I}(\Phi^*), \quad (281)$$

where \mathbb{b}_{diff} is the properly defined diffusion bilinear form, \mathbb{F}_0 is the diffusion fission form and $\mathbb{C}[\Psi]$ is the closure bilinear form depending on the high-order solution Ψ . Besides, \mathbb{I} is the high-order linear operator multiplied by an isotropic (angular independent) test function with $\Psi_g^* = \Phi_g^*, g = 1, \dots, G$. For the CDE, we do not have a fission term; when solving k -eigenvalue problems, we do not have the linear form and the fission bilinear form is divided

by k -effective:

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi) - \frac{1}{k} \mathbb{F}_0(\Phi^*, \Phi) + \mathbb{C}[\Psi](\Phi^*, \Phi) = 0. \quad (282)$$

It should be noted that k -effective is an unknown and not simply a constant in eigenvalue calculations. The high-order weak form is

$$\bar{\mathbb{L}}(\Psi^*, \Psi) - \mathbb{S}_0(\Psi^*, \Phi) + (\mathbb{L} - \bar{\mathbb{L}})(\Psi^*, \Psi) - (\mathbb{S} - \mathbb{S}_0)(\Psi^*, \Psi) - \mathbb{F}(\Psi^*, \Phi) = \mathbb{I}(\Psi^*). \quad (283)$$

The streaming and collision operator \mathbb{L} is split into two parts: an approximation $\bar{\mathbb{L}}$ that is easy to invert and a remainder that should be small in a suitable operator norm. For example, we can perform the transport sweep on an approximate first order mesh [featuring planar faces] created from the second order mesh by dropping all nodes on faces leading to $\mathbb{L} - \bar{\mathbb{L}} \neq 0$. In addition, with DFEM discretizations we want to make $\bar{\mathbb{L}}$ strictly lower triangular and hence, whenever cyclic dependencies in the sweeping order or re-entrant faces are present the corresponding terms are separated out into the remainder resulting in $\mathbb{L} - \bar{\mathbb{L}} \neq 0$. We also split the scattering operator \mathbb{S} into the zero-th moment part \mathbb{S}_0 and the high order scattering terms $\mathbb{S} - \mathbb{S}_0$, which is zero when scattering is isotropic. Nonzero $\mathbb{S} - \mathbb{S}_0$ and/or $\mathbb{L} - \bar{\mathbb{L}}$ imply that the high-order system cannot be reduced to a problem formulated only in terms of the scalar fluxes, because the solution depends on significant angular fluxes or higher order angular flux moments. In this case the high-order system will hold a state during the Picard iteration, i.e. it contains information that goes beyond what the low order system saves. We use Φ in \mathbb{S}_0 to indicate that the zero-th scattering moment only depends on the zero-th moment of the angular flux. Similarly we use Φ to indicate that fission only depends on the zero-th moment of the angular flux because fission cross sections are isotropic. In a k -eigenvalue problem, we have

$$\bar{\mathbb{L}}(\Psi^*, \Psi) - \mathbb{S}_0(\Psi^*, \Phi) + (\mathbb{L} - \bar{\mathbb{L}})(\Psi^*, \Psi) - (\mathbb{S} - \mathbb{S}_0)(\Psi^*, \Psi) - \frac{1}{k} \mathbb{F}(\Psi^*, \Phi) = 0. \quad (284)$$

The Picard iteration algorithm for the steady-state source problem is illustrated in Algorithm 1. Similarly, The

Algorithm 1 Nonlinear diffusion acceleration algorithm for the steady-state source problem.

- 1: Initial solve of the low-order equation: $\mathbb{b}_{\text{diff}}(\Phi^*, \Phi^{(0)}) - \mathbb{F}_0(\Phi^*, \Phi^{(0)}) = \mathbb{I}(\Phi^*)$.
- 2: $n \leftarrow 0$.
- 3: **while** $\epsilon_1 > \text{tol}$ **do**
- 4: Perform a transport update:

$$\bar{\mathbb{L}}(\Psi^*, \Psi^{(n+1)}) - \mathbb{S}_0(\Psi^*, \Phi^{(n)}) + (\mathbb{L} - \bar{\mathbb{L}})(\Psi^*, \Psi^{(n)}) - (\mathbb{S} - \mathbb{S}_0)(\Psi^*, \Psi^{(n)}) - \mathbb{F}(\Psi^*, \Phi^{(n)}) = \mathbb{I}(\Psi^*). \quad (285)$$

- 5: Solve the low-order equation with closures:

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi^{(n+1)}) - \mathbb{F}_0(\Phi^*, \Phi^{(n+1)}) + \mathbb{C}[\Psi^{(n+1)}](\Phi^*, \Phi^{(n+1)}) = \mathbb{I}(\Phi^*). \quad (286)$$

- 6: evaluate $\epsilon_1 = \left| \Phi^{(n+1)} - \Phi^{(n)} \right|$ and $\epsilon_2 = \left| \int_S \Psi^{(n+1)} d\Omega - \Phi^{(n)} \right|$
 - 7: $n \leftarrow n + 1$.
 - 8: **end while**
-

Picard iteration algorithm for the k -eigenvalue problem is given in Algorithm 2. It should be pointed out that both algorithms start with a full low-order solve in order to provide a reasonable initial guess of the scattering and fission sources before the first transport update, thus potentially saving one costly transport update. We use the absolute error to determine the convergence for illustration purposes. In reality we can use both absolute and relative errors for detecting the convergence and we can also use various norms of the solution. Evaluation of ϵ_2 is not necessary but we found it is extremely useful for detecting any inconsistency between low-order and high-order solutions.

The stability of Picard iteration is affected by several factors: the choice of the diffusion coefficients in the low-order system, the difference of the scalar flux space between high-order and low-order systems, the consistency between the high-order and low-order systems, and the initial guess. As the initial guess is essentially provided by the diffusion problem without high order correction, the difference of the stand-alone diffusion and transport solutions may cause non-convergence of the Picard iterations. We have evidence [46, 68] that if the local diffusion coefficients

Algorithm 2 Nonlinear diffusion acceleration algorithm for the k -eigenvalue problem.

- 1: Initial solve of the low-order equation: $\mathbb{b}_{\text{diff}}(\Phi^*, \Phi^{(0)}) = \frac{1}{k^{(0)}} \mathbb{F}_0(\Phi^*, \Phi^{(0)})$.
- 2: $n \leftarrow 0$.
- 3: **while** $\epsilon_1 > \text{tol}$ **do**
- 4: Perform a transport update:

$$\bar{\mathbb{L}}(\Psi^*, \Psi^{(n+1)}) - \mathbb{S}_0(\Psi^*, \Phi^{(n)}) + (\mathbb{L} - \bar{\mathbb{L}})(\Psi^*, \Psi^{(n)}) - (\mathbb{S} - \mathbb{S}_0)(\Psi^*, \Psi^{(n)}) = \frac{1}{k^{(n)}} \mathbb{F}(\Psi^*, \Phi^{(n)}). \quad (287)$$

- 5: Solve the low-order equation with closures:

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi^{(n+1)}) + \mathbb{C}[\Psi^{(n+1)}](\Phi^*, \Phi^{(n+1)}) = \frac{1}{k^{(n+1)}} \mathbb{F}_0(\Phi^*, \Phi^{(n+1)}). \quad (288)$$

- 6: evaluate $\epsilon_1 = \|\Phi^{(n+1)} - \Phi^{(n)}\|$ and $\epsilon_2 = \left| \int_{\mathcal{S}} \Psi^{(n+1)} d\Omega - \Phi^{(n)} \right|$
- 7: $n \leftarrow n + 1$.
- 8: **end while**

are derived from the transport cross section, the FE function spaces used to represent the solutions are identical, scattering is only mildly anisotropic and the material is only mildly heterogeneous the Picard iteration procedure is unconditionally stable. The more deviation from these conditions, the higher the likelihood that the Picard iteration algorithm becomes unstable.

The diffusion coefficient is essentially a free parameter in NDA schemes because in order to achieve consistency in a projective sense the diffusion terms present in the bilinear form \mathbb{b}_{diff} and \mathbb{C} must cancel each other at convergence. Hence, the choice of the diffusion coefficient does not affect consistency, but it does affect iterative convergence properties and stability. Rattlesnake offers three options for evaluating the diffusion coefficient:

1. Compute the transport cross section at each required point $\Sigma_{\text{tr},g}(\vec{r})$ with Eq. (20) from the total and P_1 scattering cross sections and compute the diffusion coefficient D_g as $1/(3\Sigma_{\text{tr},g})$. Note that in the general case of differing transport and diffusion meshes, proper projection operators are required to compute the transport cross section. Ref. [68] can be consulted for more details.
2. Like 1., but limit the value of the diffusion coefficient to a maximum value $D_g = \min(1/(3\Sigma_{\text{tr},g}), D_{\text{max}})$. This is a simple measure the allow deployment of NDA in problems containing voids or near-voids.
3. Finally, the nonlocal diffusion tensor $\vec{\vec{D}}_g(\vec{r})$ discussed in Section 2.2 can be used. Note, that in contrast to straight diffusion problems using the NLTDC, NDA usually uses the pointwise NLTDC, i.e. $\vec{\vec{D}}_g(\vec{r})$ continuously depends on space. NLTDC used with NDA are discussed in [71, 49] for first order S_N and second order least-squares transport equations, respectively.

It should be mentioned for completeness that work outside Rattlesnake was performed studying optimal choices of the diffusion coefficients for nonlinear diffusion acceleration. Most notably, [72] describes an optimally diffusive CMFD method where the diffusion coefficient is selected based on Fourier analysis leading to a CMFD algorithm that in all selected test problems outperforms CMFD and pCMFD; [73] searched for the optimal diffusion coefficient using a golden section search finding that the optimal diffusion coefficient performs very close to the optimally diffusive CMFD method described in [72].

3.6.2 SAAF-CFEM-SN

We start deriving the consistent low-order weak form by first copying the balance equation for CTE with SAAF (Eq. (93)) here:

$$\left(\Phi^*, \frac{\partial}{\partial t} \left(\frac{\Phi}{\mathbf{v}} \right) \right)_{\mathcal{D}} - \left(\vec{\nabla} \Phi^*, \vec{\mathbf{j}} \right)_{\mathcal{D}} + (\Phi^*, \mathbb{L}_2 \Phi)_{\mathcal{D}} - (\Phi^*, \mathbb{S}_0 \Phi)_{\mathcal{D}} + (\Phi^*, (\mathbf{I} - \alpha) \mathbf{J}^0)_{\partial \mathcal{D}} +$$

$$\left(\tau \vec{\nabla} \Phi^*, \boxed{\frac{\partial}{\partial t} \left(\frac{\vec{J}}{\mathbf{v}} \right) + \vec{\nabla} \cdot (\vec{\mathbf{E}} \Phi) + \mathbb{L}_2 \vec{J} - \mathbb{S}_1 \vec{J} - \vec{Q}_1^{\text{ext}}} \right)_{\mathcal{D}} = (\Phi^*, \mathbf{J}^{\text{inc}})_{\partial \mathcal{D}} + (\Phi^*, \mathbf{Q}_0^{\text{ext}})_{\mathcal{D}}.$$

We will denote the boxed momentum residual as

$$\vec{r}_1 \equiv \frac{\partial}{\partial t} \left(\frac{\vec{J}}{\mathbf{v}} \right) + \vec{\nabla} \cdot (\vec{\mathbf{E}} \Phi) + \mathbb{L}_2 \vec{J} - \mathbb{S}_1 \vec{J} - \vec{Q}_1^{\text{ext}}. \quad (289)$$

We remove the time-derivative terms and add the fission term for neutron transport. We also apply $\mathbb{L}_2 - \mathbb{S}_0 = \mathbb{L}_3 - \mathbb{S}_d$ and end up with the steady-state neutron balance equation with SAAF:

$$- \left(\vec{\nabla} \Phi^*, \vec{J} \right)_{\mathcal{D}} + (\Phi^*, \mathbb{L}_3 \Phi)_{\mathcal{D}} - (\Phi^*, \mathbb{S}_d \Phi)_{\mathcal{D}} + (\Phi^*, (\mathbf{I} - \alpha) \mathbf{J}^o)_{\partial \mathcal{D}} + \left(\vec{\nabla} \Phi^*, \tau \vec{r}_1 \right)_{\mathcal{D}} - (\Phi^*, \mathbb{F}_0 \Phi)_{\mathcal{D}} = (\Phi^*, \mathbf{J}^{\text{inc}})_{\partial \mathcal{D}} + (\Phi^*, \mathbf{Q}_0^{\text{ext}})_{\mathcal{D}}. \quad (290)$$

On the reflecting boundary $\partial \mathcal{D}_r$, $\mathbf{I} - \alpha = \mathbf{0}$ and on the surface source boundary $\partial \mathcal{D}_v$, $\mathbf{I} - \alpha = \mathbf{I}$. If we assume $\partial \mathcal{D} = \partial \mathcal{D}_r \cup \partial \mathcal{D}_v$ and assume that the Robin boundary $\partial \mathcal{D}_3$ corresponds to the surface source boundary and Newmann boundary $\partial \mathcal{D}_2$ corresponds to the reflecting boundary, we have

$$- \left(\vec{\nabla} \Phi^*, \vec{J} \right)_{\mathcal{D}} + (\Phi^*, \mathbb{L}_3 \Phi)_{\mathcal{D}} - (\Phi^*, \mathbb{S}_d \Phi)_{\mathcal{D}} + (\Phi^*, \mathbf{J}^o)_{\partial \mathcal{D}_3} + \left(\vec{\nabla} \Phi^*, \tau \vec{r}_1 \right)_{\mathcal{D}} - (\Phi^*, \mathbb{F}_0 \Phi)_{\mathcal{D}} = (\Phi^*, \mathbf{J}^{\text{inc}})_{\partial \mathcal{D}_3} + (\Phi^*, \mathbf{Q}_0^{\text{ext}})_{\mathcal{D}}. \quad (291)$$

The steady-state neutron diffusion weak form with $\mathbf{J}^{\text{net}} = \mathbf{0}$ and $e = \frac{1}{2}$ and without Dirichlet boundary condition is then

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi) \equiv \left(\vec{\nabla} \Phi^*, \mathbb{D} \vec{\nabla} \Phi \right)_{\mathcal{D}} + \left(\Phi^*, \frac{1}{4} \Phi \right)_{\partial \mathcal{D}_3} + (\Phi^*, \mathbb{L}_3 \Phi)_{\mathcal{D}} - (\Phi^*, \mathbb{S}_d \Phi)_{\mathcal{D}} - (\Phi^*, \mathbb{F}_0 \Phi)_{\mathcal{D}}, \quad (292)$$

$$\mathbb{l}_{\text{diff}}(\Phi^*) \equiv (\Phi^*, \mathbf{J}^{\text{inc}})_{\partial \mathcal{D}_3} + (\Phi^*, \mathbf{Q}_0^{\text{ext}})_{\mathcal{D}}. \quad (293)$$

We can rearrange this as

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi) + \left(\vec{\nabla} \Phi^*, \tau \vec{r}_1 - \vec{J} - \mathbb{D} \vec{\nabla} \Phi \right)_{\mathcal{D}} + \left(\Phi^*, \mathbf{J}^o - \frac{1}{4} \Phi \right)_{\partial \mathcal{D}_3} = \mathbb{l}_{\text{diff}}(\Phi^*). \quad (294)$$

We then define the drift vector $\vec{\mathbf{D}}$ as

$$\vec{\mathbf{D}} \equiv \frac{\tau \vec{r}_1 - \vec{J} - \mathbb{D} \vec{\nabla} \Phi}{\Phi}, \quad (295)$$

and the boundary coefficient γ as

$$\gamma \equiv \frac{\mathbf{J}^o}{\Phi} - \frac{1}{4} \mathbf{I}. \quad (296)$$

Their expanded forms are

$$\vec{D}_g = \frac{\int_{\mathcal{S}} \left(\tau_g \vec{\Omega} \vec{\Omega} \cdot \vec{\nabla} \Psi_g + (\tau_g \Sigma_{t,g} - 1) \vec{\Omega} \Psi_g - \tau_g \sum_{g'=1}^G \Sigma_{s,1}^{g' \rightarrow g} \vec{\Omega} \Psi_{g'} - \tau_g \vec{\Omega} Q_g^{\text{ext}} - D_g \vec{\nabla} \Psi_g \right) d\Omega}{\int_{\mathcal{S}} \Psi_g d\Omega}, \quad (297a)$$

$$\gamma_g = \frac{\int_{\vec{\Omega} \cdot \vec{n}_b > 0} \left| \vec{\Omega} \cdot \vec{n}_b \right| \Psi_g d\Omega}{\int_{\mathcal{S}} \Psi_g d\Omega}, g = 1, \dots, G. \quad (297b)$$

To avoid confusion, all variables in Eq. (297) are expanded with the angular fluxes, the solution of the transport equation. It is noted that when there is no anisotropic scattering, no anisotropic external source and $\tau_g = \frac{1}{\Sigma_{t,g}}$, drift vector is simplified into

$$\vec{D}_g = \frac{\int_S \left(\frac{1}{\Sigma_{t,g}} \vec{\Omega} \vec{\Omega} \cdot \vec{\nabla} \Psi_g - D_g \vec{\nabla} \Psi_g \right) d\Omega}{\int_S \Psi_g d\Omega}. \quad (298)$$

It is also noted that the drift vector is well-defined in void regions. With S_N , the above angular integrations are evaluated with the angular quadrature as

$$\vec{D}_g = \frac{\sum_{d=1}^{N_d} w_d \left(\tau_g \vec{\Omega}_d \vec{\Omega}_d \cdot \vec{\nabla} \Psi_{g,d} + (\tau_g \Sigma_{t,g} - 1) \vec{\Omega}_d \Psi_{g,d} - \tau_g \sum_{g'=1}^G \Sigma_{s,1}^{g' \rightarrow g} \vec{\Omega}_d \Psi_{g',d} - \tau_g \vec{\Omega}_d Q_{g,d}^{\text{ext}} - D_g \vec{\nabla} \Psi_{g,d} \right)}{\sum_{d=1}^{N_d} w_d \Psi_{g,d}}, \quad (299)$$

$$\gamma_g = \frac{\sum_{\vec{\Omega}_d \cdot \vec{n}_b > 0} w_d \left| \vec{\Omega}_d \cdot \vec{n}_b \right| \Psi_{g,d}}{\sum_{d=1}^{N_d} w_d \Psi_{g,d}}, g = 1, \dots, G. \quad (300)$$

It is worth mentioning that \mathbf{J}^{inc} needs to be evaluated with angular quadrature as well because it is typically not equal to the analytical integrated one. The above bilinear form becomes

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi) + \left(\vec{\nabla} \Phi^*, \vec{D} \Phi \right)_D + (\Phi^*, \gamma \Phi)_{\partial D} = \mathbb{l}_{\text{diff}}(\Phi^*). \quad (301)$$

So if we know the transport solution, we can evaluate the drift vector and the boundary coefficient, with which a diffusion calculation reproduces the scalar flux of the transport solution. The drift term plus the additional boundary term are the closure terms of the diffusion weak form. To fit our nonlinear iteration algorithm, we define

$$\mathbb{C}_{\text{SAAF}}[\Psi](\Phi^*, \Phi) \equiv \left(\vec{\nabla} \Phi^*, \vec{D} \Phi \right)_D + (\Phi^*, \gamma \Phi)_{\partial D}. \quad (302)$$

With SAAF, we have an option if we want to include $\langle \Psi^*, \mathbb{B} \Psi \rangle^-$ in the $\bar{\mathbb{L}}$ operator. It is noted that in SAAF-CFEM-SN NDA, low-order diffusion must currently be defined on the same mesh with the same shape functions as the angular fluxes.

3.6.3 LS-CFEM-SN

Although the consistent diffusion scheme can be derived for LS-CFEM-SN as for SAAF-CFEM-SN, the resultant diffusion scheme does not have a physical meaning. Also because LS-CFEM-SN does not have global conservation, it is thus believed that the diffusion scheme directly from the diffusion equation should be used for NDA. The low-order weak form with the LS closure should be in the same form as SAAF:

$$\mathbb{b}_{\text{diff}}(\Phi^*, \Phi) + \left(\vec{\nabla} \Phi^*, \vec{D} \Phi \right)_D + (\Phi^*, \kappa \Phi)_{\partial D} = \mathbb{l}_{\text{diff}}(\Phi^*). \quad (303)$$

This form is called *conservative*. Numerical experiments shows that LS-NDA with this conservative low-order form is stable and gives better convergence on k -effective. The drift vector in SAAF can be rearranged a little as:

$$\vec{D}_g = \frac{\int_S \left(\tau_g \Sigma_{\text{tr},g} \left(\frac{1}{\Sigma_{\text{tr},g}} \left(\vec{\Omega} \vec{\Omega} \cdot \vec{\nabla} \Psi_g - \sum_{g' \neq g} \Sigma_{s,1}^{g' \rightarrow g} \vec{\Omega} \Psi_{g'} - \vec{\Omega} Q_g^{\text{ext}} \right) \right) + (1 - \tau_g \Sigma_{\text{tr},g}) (-\vec{\Omega} \Psi_g) - D_g \vec{\nabla} \Psi_g \right) d\Omega}{\int_S \Psi_g d\Omega},$$

where the boxed terms appear as the weighted summation of two contributions: Eddington form $-\int_S \frac{1}{\Sigma_{\text{tr},g}} (\vec{\Omega} \vec{\Omega} \cdot \vec{\nabla} \Psi_g - \sum_{g' \neq g} \Sigma_{s,1}^{g' \rightarrow g} \vec{\Omega} \Psi_{g'} - \vec{\Omega} Q_g^{\text{ext}}) d\Omega$ and the current form $\int_S \vec{\Omega} \Psi_g d\Omega$. When $\tau_g = \frac{1}{\Sigma_{\text{tr},g}}$, we only have the Eddington form. When $\tau_g = 0$, we only have the current form. Based on this observation, Hans [49] proposed a modified drift vector formulation for LS NDA:

$$\vec{D}_g = \begin{cases} \frac{\int_S \left(\frac{1}{\Sigma_{\text{tr},g}} \left(\vec{\Omega} \vec{\Omega} \cdot \vec{\nabla} \Psi_g - \sum_{g' \neq g} \Sigma_{s,1}^{g' \rightarrow g} \vec{\Omega} \Psi_{g'} - \vec{\Omega} Q_g^{\text{ext}} \right) - D_g \vec{\nabla} \Psi_g \right) d\Omega}{\int_S \Psi_g d\Omega}, & \Sigma_{\text{tr},g} h \geq 0.01 \\ \frac{\int_S (-\vec{\Omega} \Psi_g - D_g \vec{\nabla} \Psi_g) d\Omega}{\int_S \Psi_g d\Omega}, & \Sigma_{\text{tr},g} h < 0.01 \end{cases}, \quad (304)$$

where h is the characteristic length of the local element, typically the maximum vertex separation.

We also have an option if we want to include $\langle \Psi^*, \mathbb{B}\Psi \rangle^-$ in the $\bar{\mathbb{L}}$ operator as SAAF. Low-order diffusion must currently be defined on the same mesh with the same shape functions as the angular fluxes as well.

3.6.4 DFEM-SN

The DFEM-SN NDA method allows for the low order problem to be discretized using fewer energy groups, using a coarser mesh and using different FEM shape functions. Therefore, even at convergence, the scalar fluxes obtained from the transport and diffusion system cannot be equal. The equality condition at convergence is generalized to the requirement that at convergence the projection of the transport scalar flux into the diffusion solution space is equal to the diffusion scalar flux. To distinguish transport and diffusion quantities, we use capital letters for the former and lower case symbols for the latter. Defining a projection operator $\mathbb{P}[\cdot]$, the consistency requirement becomes:

$$\phi = \mathbb{P}[\Phi], \quad (305)$$

where $\mathbb{P}[\cdot]$ can be obtained by requiring the left hand side of the above equation to be equal to the right hand side when taking an inner product of both sides with respect to an arbitrary test function selected from the low order space. The prolongation operator is denoted by \mathbb{P}^{-1} and is defined equivalently to \mathbb{P} . A more exhaustive discussion of the projection and prolongation operators can be found in [68].

By using an angularly isotropic test function, the angular balance equation of the first order S_N weak form is derived:

$$\begin{aligned} & - \left(\vec{\nabla} \Phi^*, \vec{\mathbf{j}} \right) + (\Phi^*, \mathbb{L}_2 \Phi) + ([\Phi^*], [\mathbf{J}^0])_{\Gamma_{\text{int}}} + (\Phi^*, \mathbf{J}^0)_{\partial \mathcal{D}_s} - (\Phi^*, \mathbb{S}_0 \Phi) - (\Phi^*, \mathbb{F}_0 \Phi) \\ & = (\Phi^*, \mathbf{J}^{\text{inc}})_{\partial \mathcal{D}_s} + (\Phi^*, \mathbf{Q}_0^{\text{ext}}). \end{aligned} \quad (306)$$

$\mathbf{J}^{0,\pm}$ on Γ_{int} are the partial out-going currents with respect to the two neighboring elements. Note that we derive the weak forms for the source-driven, steady-state case. The consistent diffusion weak form is a modified interior penalty (MIP) discontinuous FEM diffusion weak form augmented with closure terms and using $e = \frac{1}{2}$:

$$\begin{aligned} & \mathbb{b}_{\text{diff}}(\phi^*, \phi) + ([\phi^*], [\kappa \phi])_{\Gamma_{\text{int}}} + ([\phi^*], [\mathbb{D} \vec{\nabla} \phi \cdot \vec{n}])_{\Gamma_{\text{int}}} + \left(\phi^*, \frac{\phi}{4} \right)_{\partial \mathcal{D}_s} + \mathbb{C}[\Psi](\phi^*, \phi), \\ & = (\phi^*, \mathbb{P}[\mathbf{J}^{\text{inc}}])_{\partial \mathcal{D}_s} + (\phi^*, \mathbb{P}[\mathbf{Q}_0^{\text{ext}}]). \end{aligned} \quad (307)$$

where the closure is given by:

$$\mathbb{C}[\Psi](\phi^*, \phi) = (\vec{\nabla} \phi^*, \vec{\mathbf{D}} \phi) + ([\phi^*], [\hat{\kappa} \phi])_{\Gamma_{\text{int}}} + (\phi, \gamma \phi)_{\partial \mathcal{D}_s}. \quad (308)$$

The volume, interior face and boundary closure coefficients are computed by:

$$\vec{\mathbf{D}} = - \frac{\mathbf{D} \vec{\nabla} \mathbb{P}[\Phi] + \mathbb{P}[\vec{\mathbf{j}}]}{\mathbb{P}[\Phi]}, \quad (309a)$$

$$\begin{aligned} \hat{\kappa}^{\pm} &= \frac{\mathbb{P}[\mathbf{J}^{0,\pm}]}{\mathbb{P}[\Phi^{\pm}]} \mp \frac{[\kappa \mathbb{P}[\Phi]]}{2\mathbb{P}[\Phi^{\pm}]} - \frac{1}{\mathbb{P}[\Phi^{\pm}]} \left(\frac{1}{4} \phi_s \pm \frac{1}{2} \{ \vec{n} \cdot \mathbf{D} \vec{\nabla} \mathbb{P}[\Phi] \} \right) \\ \gamma &= \frac{\mathbb{P}[\mathbf{J}^0]}{\mathbb{P}[\Phi]} - \frac{1}{4} \mathbf{I}, \end{aligned} \quad (309b)$$

where the quantity $\mathbf{J}^{0,\pm}$ is defined as:

$$\mathbf{J}^{0,\pm} = \sum_{\pm \vec{\Omega}_n \cdot \vec{n}_b < 0} w_n |\vec{\Omega}_n \cdot \vec{n}_b| \Psi_n^{\pm}, \quad (309c)$$

and the face scalar flux is defined by:

$$\phi_s = \frac{\{ \mathbb{P}[\Phi] \mathbf{D} / h \}}{\{ \mathbf{D} / h \}}. \quad (309d)$$

Recall that the normal in this situation points from the upstream to the downstream face, i.e. from $-$ to $+$. We compute the cross sections for the diffusion problem via flux weighted averages under the projection operator \mathbb{P} . To distinguish the diffusion from the transport cross section, we add a $\hat{\cdot}$ to the diffusion cross sections; the diffusion groups are indexed with p .

- Removal cross section: $\hat{\Sigma}_{r,p} = \frac{\mathbb{P}[\Sigma_{t,g}\Phi_g] - \mathbb{P}[\Sigma_{s,0}^{g \rightarrow g}\Phi_g]}{\mathbb{P}[\Phi_g]},$
- Scattering cross section: $\hat{\Sigma}_{s,0}^{p' \rightarrow p} = \frac{\mathbb{P}\left[\sum_{g \in p} \Sigma_{s,0}^{g' \rightarrow g}\Phi_{g'}\right]}{\mathbb{P}[\Phi_{g'}]},$
- Fission production cross section: $\nu\hat{\Sigma}_{f,p} = \frac{\mathbb{P}[\nu\Sigma_{f,g}\Phi_g]}{\mathbb{P}[\Phi_g]},$
- Fission spectrum: $\hat{\chi}_p = \frac{\mathbb{P}\left[\chi_g\left(\sum_{g'} \nu\Sigma_{f,g'}\Phi_{g'}\right)\right]}{\mathbb{P}\left[\sum_{g'} \nu\Sigma_{f,g'}\Phi_{g'}\right]}.$

It is important to avoid introducing spurious information when updating the transport scalar fluxes after a diffusion solve. An update formula must ensure that only the fraction of the transport FEM solution that is within the range of the diffusion FEM space is altered. We use the following update formula:

$$\Phi \leftarrow \Phi + \mathbb{P}^{-1} [\phi - \mathbb{P}[\Phi]]. \quad (310)$$

3.7 Multi-scheme interface conditions

Multi-resolution transport is a concept in which the spatial solution domain is divided into several subdomains where various discretization schemes with different angular, space, energy resolutions can be applied [74, 75, 76, 77, 78, 79, 80]. Multi-resolution is motivated by the recognition that uniform fine resolution over the full solution domain is unfeasible for realistic problems in a foreseeable future. Uniform fine resolution is also seldom required because of the existing homogenization methods. On the other hand, fine-scale resolution in places where interesting multiphysics phenomena take place is desired, either because of unavailability of accurate homogenization or the loss of information due to homogenization. The computational resources can be much more efficiently managed by concentrating the main efforts (and resolution) on the most important subdomains while still taking into account the coupling with other less important regions of the domain. Thus, it makes sense to apply different resolutions suitable for subdomains and solve them as a whole without iterations over the interface quantities.

In practice, various discretization schemes with possible different meshes, angular orders and/or number of groups are simultaneously handled in Rattlesnake. To enable multi-resolution or multi-scheme capability, we need to handle the interface coupling in space, angle and energy properly. The interface conditions in the weak forms will be presented. In all of these conditions, we assume embedded energy structures across two neighboring resolutions, i.e. any group in these two scales is either a subset or completely outside of another group. Spatial coupling are implemented via the general mortar FEM and the interface kernel capability available in MOOSE and are out of the scope of this manual. Although users are allowed to amend the interface conditions with the properties defined on interfaces, such as discontinuity factors from the generalized homogenization theory [81], this section will only talk about the nominal conditions.

Discrete ordinates methods (S_N) are suitable for handling heterogeneous problems due to the decoupling of angular variables in the streaming operators, while the spherical harmonics expansion methods (P_N) typically generate more accurate results with the same number of unknowns for problems with significant homogenization. It is believed that regions of different levels of homogenization can be treated most efficiently with hybrid S_N - P_N calculations. Multi-scheme transport is thus typically implemented in the way of hybrid S_N - P_N calculations. Hybrid S_N - P_N calculations have been studied in the past [74, 75, 76, 77, 78, 79, 33]. Among them [74, 75, 78, 79] looked at the one-dimensional case and [76, 77] focused on the coupling between VNM (variational nodal method) with P_N and SNM (S_N nodal method), both of which are locally conservative. [33] presents a hybrid S_N - P_1 method in two-dimension

with the upwinding scheme for coupling diffusion to transport, while the interface condition for P_1 is naturally imposed with the nonlinear iteration scheme.

We propose two distinct methods for imposing the interface condition within the domain decomposition framework: (1) Lagrange multipliers and (2) classical upwinding scheme [82]. The Lagrange multiplier method enforces the continuity of angular flux moments across the interface by introducing the Lagrange multipliers defined over the interface. This approach creates a saddle point problem and is considered only for CFEM schemes, such as LS and SAAF. The upwinding method has been studied in the past at the element level in the discontinuous FEM for radiation transport [83]. It has also been applied previously for hybrid transport algorithms. The upwinding method solves the equations for each subdomain imposing surface source boundary conditions provided from their neighbors. It preserves the causality inherent to the hyperbolic nature of particle transport on the subdomain interface at the price of introducing asymmetry in the streaming operator. It is noted that the upwinding method can be applied to transport schemes including first-order discontinuous schemes, etc., other than SAAF and LS.

3.7.1 Lagrange multiplier method

We directly give the interface condition in the weak form with Lagrange multiplier method:

$$- \int_{\mathcal{S}} [(\llbracket \Psi \rrbracket, \Lambda^*)_{\Gamma} + (\llbracket \Psi^* \rrbracket, \Lambda)_{\Gamma}] d\Omega, \quad (311)$$

where Γ is the interface. This condition is added on top of the bilinear forms of all subdomains. It is noted that the function space of solutions and test functions in each individual bilinear forms are defined only over the corresponding subdomains. Thus the solutions such as the scalar fluxes across interfaces are allowed to be discontinuous. For instance in Fig. 3, there are three bilinear forms for three subdomains, and three interface conditions are required for three domain interfaces. The jump operator $\llbracket \Psi \rrbracket$ is defined with respect to the interface norm \vec{n} , whose orientation

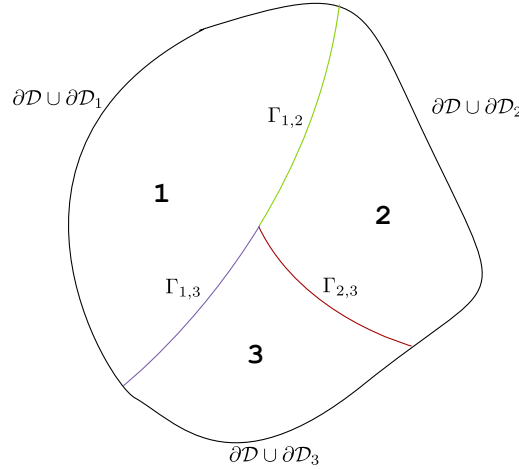


Figure 3 Domain decomposition.

can be absorbed in the Lagrange multipliers Λ and its test functions Λ^* thus do not affect the calculation,

$$\llbracket \Psi \rrbracket \equiv \Psi^+ - \Psi^-, \quad (312)$$

$$\Psi^\pm \equiv \lim_{s \rightarrow 0^+} \Psi(\vec{x} + s\vec{n}). \quad (313)$$

Lagrange multipliers Λ have the same physical meaning as $\vec{\Omega} \cdot \vec{n} \Psi$. So $\int_{\mathcal{S}} \Lambda d\Omega = \vec{J} \cdot \vec{n}$, which can be used to prove the subdomain-wise balance. Lagrange multipliers are defined over Γ .

3.7.1.1 SN-PN

The two neighboring subdomains are denoted as \mathcal{D}_{SN} and \mathcal{D}_{PN} . On \mathcal{D}_{SN} , we solve the equation in a given set of directions specified by an angular quadrature $\{\vec{\Omega}_d, w_d, d = 1, \dots, N_d\}$. On \mathcal{D}_{PN} , we let Ψ be expanded using

spherical harmonics up to order N . The angular flux in this space can be represented as $\Psi = \vec{\Phi}(\vec{x})\vec{R}(\vec{\Omega})$, where $\vec{\Phi}$ contains the angular flux moments of all groups and \vec{R} is the spherical harmonics \vec{Y} times the normalization factors, i.e.

$$\vec{R} = \mathbf{P}\vec{Y}, \quad (314)$$

where $\mathbf{P} = \left(\int_{\mathcal{S}} \vec{Y}\vec{Y}^T d\Omega \right)^{-1}$ is a diagonal matrix containing the normalization factors characteristic of the selected spherical harmonics functions. Since the dimension of the P_N angular space is typically smaller than M , we fix the S_N side as the master and make the unit normal vector on the interface \vec{n} always point towards the S_N side.

We select the Λ and Λ^* from the same angular space as the S_N solution. On the S_N side this yields,

$$\sum_{g=1}^G \sum_{d=1}^{N_d} w_d \left(\left(\Psi_{g,d}, \Lambda_{g,d}^* \right)_{\Gamma} + \left(\Psi_{g,d}^*, \Lambda_{g,d} \right)_{\Gamma} \right), \quad (315)$$

while on the P_N side we obtain:

$$- \sum_{g=1}^G \left(\sum_{d=1}^{N_d} w_d \left(\vec{\Phi}_g \vec{R}(\vec{\Omega}_d), \Lambda_d^* \right)_{\Gamma} - \left(\vec{\Phi}_g^*, \mathbf{P} \vec{\Lambda}_g \right)_{\Gamma} \right), \quad (316)$$

where $\vec{\Lambda}_g$ is the Lagrange multiplier moments of the energy group g . It is noted that if the angular quadrature can integrate spherical harmonics up to order $N_{SN} > N$, the first term in Eq. (315) and Eq. (316) will ensure the continuity of the angular flux moments. It also will cause all the flux moments evaluated on the S_N side with order being greater than N and less than or equal to N_{SN} to be zero. However, the above interface condition creates non-smooth P_N solutions close to the interface based on observations in numerical experiments. Numerical evidence shows that the Lagrange multiplier moments $\vec{\Lambda}$ needs to be evaluated with the inverse of the moment-to-direction matrix \mathbf{M}^{-1} instead of the above direction-to-moment matrix \mathbf{D} . Refer to Appendix E.1 for more information. This interface condition is implemented in *SNPNConstraint*. The total number of constrains is $M \times G$.

3.7.1.2 PN-PN

P_N schemes are applied on both neighboring subdomains denoted as 1 and 2. P_N orders of two subdomains L_1 and L_2 may be different but we always assume $L_1 \geq L_2$. We fix the side of subdomain 1 as the master and make the unit normal vector on the interface \vec{n} always point towards the master side. We select Λ and Λ^* from the angular space as the P_N solution with order L_1 . Because of the spherical harmonics orthogonality, this results into

$$\sum_{g=1}^G \left(\sum_{\ell=0}^{L_2} \sum_{m=-\ell}^{\ell} \left(\left(\Phi_{g,\ell,m}, \Lambda_{g,\ell,m}^* \right)_{\Gamma} + \left(\Phi_{g,\ell,m}^*, \Lambda_{g,\ell,m} \right)_{\Gamma} \right) + \sum_{\ell=L_2+1}^{L_1} \sum_{m=-\ell}^{\ell} \left(\left(\Phi_{g,\ell,m}^-, \Lambda_{g,\ell,m}^* \right)_{\Gamma} + \left(\Phi_{g,\ell,m}^{*-}, \Lambda_{g,\ell,m} \right)_{\Gamma} \right) \right) \quad (317)$$

The first part of the interface condition is *EqualValueConstraint*, which is part of MOOSE. The second part of the condition is implemented in *ZeroValueConstraint*, which is more like a boundary condition. The total number of constraints is the number of angular moments with order L_1 times the number of energy groups G . Diffusion schemes can be treated as P0.

3.7.1.3 SN-SN

We currently do not allow different orders of SN to interface with each other. The same SN order results into $M \times G$ *EqualValueConstraint* constraints.

$$\sum_{g=1}^G \sum_{d=1}^{N_d} \left[\left(\Psi_{g,d}, \Lambda_{g,d}^* \right)_{\Gamma} + \left(\Psi_{g,d}^*, \Lambda_{g,d} \right)_{\Gamma} \right]. \quad (318)$$

The master side can be fixed on either side.

3.7.2 Upwinding

The classical upwinding scheme basically uses the angular flux of the neighboring subdomain as the surface source for the current domain. As the result, we have an interface term similar to the DG term in [DFEM](#): $-\langle \llbracket \Psi^* \rrbracket, \Psi_- \rangle_\Gamma$. In order to use the jump notation with the surface normal \vec{n} for the interface condition, we transform it into

$$\begin{aligned}
 & \langle \llbracket \Psi^* \rrbracket, \Psi_- \rangle_\Gamma \\
 &= \int_S \left(\left| \vec{\Omega} \cdot \vec{n} \right| \llbracket \Psi^* \rrbracket, \Psi_- \right)_\Gamma d\Omega && \text{pull angular integration} \\
 &= \int_S \left(\left| \vec{\Omega} \cdot \vec{n} \right| \llbracket \Psi^* \rrbracket, \Psi^- \right)_{\Gamma^+} d\Omega - \int_S \left(\left| \vec{\Omega} \cdot \vec{n} \right| \llbracket \Psi^* \rrbracket, \Psi^+ \right)_{\Gamma^-} d\Omega && \text{split interface} \\
 &= \langle \llbracket \Psi^* \rrbracket, \Psi^- \rangle_\Gamma^+ - \langle \llbracket \Psi^* \rrbracket, \Psi^+ \rangle_\Gamma^- && \text{half angular integration notation}
 \end{aligned}$$

So the interface condition with upwinding is

$$-\langle \llbracket \Psi^* \rrbracket, \Psi^- \rangle_\Gamma^+ + \langle \llbracket \Psi^* \rrbracket, \Psi^+ \rangle_\Gamma^- . \quad (319)$$

This condition preserves the causality inherent to the hyperbolic nature of particle transport on the subdomain interface at the price of introducing asymmetry in the streaming operator. No variable defined over the interface is required with upwinding. It is recommended to use an *odd* P_N order for the upwind coupling.

3.7.2.1 SN-PN

We apply the same S_N and P_N discretizations as before and make the unit normal vector on the interface \vec{n} always point towards the S_N side. On the S_N side we obtain:

$$\sum_{g=1}^G \left(\sum_{\vec{\Omega}_d \cdot \vec{n} < 0} w_d \left(\left| \vec{\Omega}_d \cdot \vec{n} \right| \Psi_{g,d}^*, \Psi_{g,d} \right)_\Gamma - \sum_{\vec{\Omega}_d \cdot \vec{n} > 0} w_d \left(\left| \vec{\Omega}_d \cdot \vec{n} \right| \Psi_{g,d}^*, \vec{\Phi}_g \vec{R}(\vec{\Omega}_d) \right)_\Gamma \right). \quad (320)$$

and on the P_N side we get

$$\sum_{g=1}^G \left(\left(\vec{\Phi}_g^*, \mathbf{L}(\vec{n}) \vec{\Phi}_g \right)_\Gamma - \left(\vec{\Phi}_g^*, \vec{j}_g \right)_\Gamma \right), \quad (321)$$

where

$$\mathbf{L}(\vec{n}) \equiv \int_{\vec{\Omega} \cdot \vec{n} > 0} \vec{R}(\vec{\Omega}) \vec{R}^T(\vec{\Omega}) \left| \vec{\Omega} \cdot \vec{n} \right| d\Omega, \quad (322)$$

$$\vec{j}_g \equiv \sum_{\vec{\Omega}_d \cdot \vec{n} < 0} w_d \left| \vec{\Omega}_d \cdot \vec{n} \right| \vec{R}(\vec{\Omega}_d) \Psi_{g,d}. \quad (323)$$

As opposed to the Lagrange multiplier method, numerical results indicate that angular smoothing is not required with upwinding. On the other hand, numerical tests show that the discrete version of \mathbf{L}

$$\bar{\mathbf{L}}(\vec{n}) \equiv \sum_{\vec{\Omega}_d \cdot \vec{n} > 0} w_d \vec{R}(\vec{\Omega}_d) \vec{R}^T(\vec{\Omega}_d) \left| \vec{\Omega}_d \cdot \vec{n} \right|, \quad (324)$$

gives better results. It is noted that both $\mathbf{L}(\vec{n})$ and $\bar{\mathbf{L}}(\vec{n})$ are symmetric and depend on \vec{n} . However, their eigenstructures do not depend on \vec{n} and thus are rotationally invariant. The direct meaning of Eq. (320) and Eq. (321) is that the coupling from P_N to S_N is performed by sampling the streaming directions with the angular quadrature from the P_N angular moments, while the coupling from S_N to P_N is the Marshak boundary condition with the partial angular moments obtained from the S_N out-going angular fluxes. This interface condition is implemented in the interface kernel *InterfaceSNPNUpwind*.

3.7.2.2 SN-SN

We solve the equation in a given set of directions specified by an angular quadrature $\{\vec{\Omega}_d, w_d, d = 1, \dots, N_d\}$ with S_N . If the angular quadratures on two adjacent subdomains are the same, the discretized version of the upwinding interface condition can simply be:

$$\sum_{g=1}^G \left(- \sum_{\vec{\Omega}_d \cdot \vec{n} > 0} w_d \left(|\vec{\Omega}_d \cdot \vec{n}| \llbracket \Psi_{g,d}^* \rrbracket, \Psi_{g,d}^- \right)_{\Gamma} + \sum_{\vec{\Omega}_d \cdot \vec{n} < 0} w_d \left(|\vec{\Omega}_d \cdot \vec{n}| \llbracket \Psi_{g,d}^* \rrbracket, \Psi_{g,d}^+ \right)_{\Gamma} \right). \quad (325)$$

This condition is the traditional upwinding scheme for the DFEM transport [83]. This interface condition is implemented in the interface kernel *InterfaceUpwind*.

3.7.2.3 PN-PN

If we plug $\Psi = \vec{\Phi} \vec{R}(\vec{\Omega})$ into Eq. (319) we can see that

$$\begin{aligned} & - \left\langle \llbracket \vec{\Phi}^* \vec{R}(\vec{\Omega}) \rrbracket, \vec{\Phi}^- \vec{R}(\vec{\Omega}) \right\rangle_{\Gamma}^+ + \left\langle \llbracket \vec{\Phi}^* \vec{R}(\vec{\Omega}) \rrbracket, \vec{\Phi}^+ \vec{R}(\vec{\Omega}) \right\rangle_{\Gamma}^- \\ &= - \left(\llbracket \vec{\Phi}^* \rrbracket, \mathbf{L}^+(\vec{n}) \vec{\Phi}^- \right)_{\Gamma} + \left(\llbracket \vec{\Phi}^* \rrbracket, \mathbf{L}^-(\vec{n}) \vec{\Phi}^+ \right)_{\Gamma} \\ &= \left(\llbracket \vec{\Phi}^* \rrbracket, \frac{\mathbf{L}^+ + \mathbf{L}^-}{2} \llbracket \vec{\Phi} \rrbracket - (\mathbf{L}^+ - \mathbf{L}^-) \{ \vec{\Phi} \} \right)_{\Gamma} \end{aligned} \quad (326)$$

We show that the above formulation is closely related to the upwinding in P_N -DFEM. P_N -DFEM upwinding is based on the eigenstructure of streaming matrices. The interface weak form can be rewritten as:

$$\begin{aligned} & - \int_{\mathcal{S}} \left(\vec{\Omega} \cdot \vec{n} \llbracket \Psi^* \rrbracket, \Psi_n \right)_{\Gamma} d\Omega \\ &= \left(\llbracket \vec{\Phi}^* \rrbracket, -(\mathbf{L}^+ - \mathbf{L}^-) \vec{\Phi}_n \right)_{\Gamma} && \text{PN expansion} \\ &= \left(\llbracket \vec{\Phi}^* \rrbracket, -(\mathbf{M}^+ - \mathbf{M}^-) \vec{\Phi}_n \right)_{\Gamma} && \text{Eigen decomposition} \\ &= \left(\llbracket \vec{\Phi}^* \rrbracket, -(\mathbf{M}^+ \vec{\Phi}^- - \mathbf{M}^- \vec{\Phi}^+) \right)_{\Gamma} && \text{PN upwinding} \\ &= \left(\llbracket \vec{\Phi}^* \rrbracket, \frac{\mathbf{M}^+ + \mathbf{M}^-}{2} \llbracket \vec{\Phi} \rrbracket - (\mathbf{M}^+ - \mathbf{M}^-) \{ \vec{\Phi} \} \right)_{\Gamma} && \text{Rearrangement} \\ &= \left(\llbracket \vec{\Phi}^* \rrbracket, \frac{\mathbf{M}^+ + \mathbf{M}^-}{2} \llbracket \vec{\Phi} \rrbracket - (\mathbf{L}^+ - \mathbf{L}^-) \{ \vec{\Phi} \} \right)_{\Gamma} \end{aligned} \quad (327)$$

where Ψ_n is the numerical flux to be determined by the upwinding scheme. \mathbf{M}^{\pm} is constructed with the eigenvalue decomposition of \mathbf{L} . We first let

$$\mathbf{L}^+ - \mathbf{L}^- = \mathbf{U} \Lambda \mathbf{U}^T, \quad (328)$$

where Λ is the diagonal matrix whose diagonal elements are the eigenvalues of $\mathbf{L}^+ - \mathbf{L}^-$. \mathbf{U} is a matrix whose columns are the eigenvectors of $\mathbf{L}^+ - \mathbf{L}^-$. We separate Λ into

$$\Lambda = \Lambda^+ - \Lambda^-, \quad (329)$$

where Λ^+ contains only the positive eigenvalues of Λ and Λ^- contains the magnitude of the negative eigenvalues. We then define

$$\mathbf{M}^{\pm} = \mathbf{U} \Lambda^{\pm} \mathbf{U}^T. \quad (330)$$

It is clear that

$$\mathbf{L}^+ - \mathbf{L}^- = \mathbf{M}^+ - \mathbf{M}^-. \quad (331)$$

This form of interface condition is also referred to as Roe-type Riemann solver. Both Eq. (326) and Eq. (327) are implemented in the interface kernel *InterfacePNUUpwind*.

3.8 Reduced order model for NTE

It is often desired to create a reduced order model (ROM) from the FEM weak forms with a function space in which the solution is sought having a reduced dimensionality. Because the dimensionality can be significantly reduced for particular applications, computing efforts with ROM can be drastically decreased. ROM is particularly useful when repeated calculations are required, such as uncertainty quantification, online monitoring (inverse problem), acceleration for the existing calculations, etc. The reduced function space can be generated with snapshots of calculations of a scope of training models with the assumption that solutions of new models can be interpolated well with the snapshots. The reduced function space can also be generated with eigen modes or with perturbation theory at a particular configuration.

To perform the derivation, we first give the abstract weak form for the transient NTE as

$$\frac{d}{dt} \mathbb{T}(\Psi^*, \Psi) = \mathbb{F}(\Psi^*, \Psi) - \mathbb{L}(\Psi^*, \Psi) + \mathbb{S}(\Psi^*, \Psi) - \sum_{i=1}^I \mathbb{F}_{d,i}(\Psi^*, \Psi) + \sum_{i=1}^I \mathbb{S}_{d,i}(\Psi^*, C_i), \quad (332)$$

$$\frac{d}{dt} (C_i^*, C_i)_{\mathcal{D}} = (C_i^*, \mathbb{P}_i \Phi)_{\mathcal{D}} - (C_i^*, \lambda_i C_i)_{\mathcal{D}}, \quad i = 1, \dots, I, \quad (333)$$

where \mathbb{T} , \mathbb{F} , \mathbb{L} , \mathbb{S} , $\mathbb{F}_{d,i}$, $\mathbb{S}_{d,i}$ are the time, fission, loss due to streaming and collision, scattering, delayed fission and delayed source kernels of the flux Ψ . We split the weak form into flux Ψ^* and I -group delayed neutron precursors C_i^* , $i = 1, \dots, I$. It is noted that because the test function is arbitrary in the function space, this split does not change the weak form. We make the test functions time-independent.

In particular, for SAAF, we have:

$$\mathbb{T}(\Psi^*, \Psi) \equiv \left((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \frac{\Psi}{\mathbf{v}} \right), \quad (334a)$$

$$\mathbb{F}(\Psi^*, \Psi) \equiv ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{F} \Psi), \quad (334b)$$

$$\mathbb{L}(\Psi^*, \Psi) \equiv -((\mathbb{I} + \tau \mathbb{L}_1) \mathbb{L}_1 \Psi^*, \Psi) + ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{L}_2 \Psi) + \langle \Psi^*, \Psi \rangle^+ - \langle \Psi^*, \mathbb{B} \Psi \rangle^- - \left(\tau' \mathbb{L}_1 \Psi^*, \frac{\Psi}{\mathbf{v}} \right), \quad (334c)$$

$$\mathbb{S}(\Psi^*, \Psi) \equiv ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{S} \Psi), \quad (334d)$$

$$\mathbb{F}_{d,i}(\Psi^*, \Psi) \equiv ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, \mathbb{F}_{d,i} \Psi), \quad (334e)$$

$$\mathbb{S}_{d,i}(\Psi^*, C_i) \equiv ((\mathbb{I} + \tau \mathbb{L}_1) \Psi^*, C_i C_i). \quad (334f)$$

It is noted that with SAAF, the steady-state adjoint DNP concentration is $C_i^* = \chi_{d,i}^T (\Phi^* + \tau \vec{\nabla} \cdot \vec{\mathbf{J}}^*)$ but not $\chi_{d,i}^T \Phi^*$. If $\Psi^* = \mathbf{1}$, isotropic and constant in space, $\vec{\mathbf{J}}^* = \mathbf{0}$, $\mathbf{C}^* = \chi_{d,i}^T \Phi^* = 1$, consistent with the analytical adjoint DNPs. Likewise, with CFEM-Diffusion, the abstract weak form becomes:

$$\mathbb{T}(\Phi^*, \Phi) \equiv \left(\Phi^*, \frac{\Phi}{\mathbf{v}} \right)_{\mathcal{D}}, \quad (335a)$$

$$\mathbb{F}(\Phi^*, \Phi) \equiv (\Phi^*, \mathbb{F}_0 \Phi)_{\mathcal{D}}, \quad (335b)$$

$$\mathbb{L}(\Phi^*, \Phi) \equiv \left(\vec{\nabla} \Phi^*, \mathbb{D} \vec{\nabla} \Phi \right)_{\mathcal{D}} + \left(\Phi^*, \frac{1}{2} \mathbf{e} \Phi \right)_{\partial \mathcal{D}_3} + (\Phi^*, \mathbb{L}_3 \Phi)_{\mathcal{D}}, \quad (335c)$$

$$\mathbb{S}(\Phi^*, \Phi) \equiv (\Phi^*, \mathbb{S}_d \Phi)_{\mathcal{D}}, \quad (335d)$$

$$\mathbb{F}_{d,i}(\Phi^*, \Phi) \equiv (\Phi^*, \mathbb{F}_{d,i,0} \Phi)_{\mathcal{D}}, \quad (335e)$$

$$\mathbb{S}_{d,i}(\Phi^*, C_i) \equiv (\Phi^*, \mathbb{C}_i C_i)_{\mathcal{D}}. \quad (335f)$$

The primal variable are the scalar fluxes for the diffusion equation, but we can still use Ψ as the notation of the solution and Ψ^* as the test function. All weak forms of the NTE can be written into this abstract form.

Now we can assume the solution can be expanded with $\Psi = \vec{\mathbf{B}} \mathbf{n}$, where $\vec{\mathbf{B}} = \{\mathbf{b}_1, \dots, \mathbf{b}_N\}$ and N is the dimension of ROM. \mathbf{n} contains the solution coefficients with dimension N . All \mathbf{b} s are in the same function space as Ψ of the weak form. N is much smaller than the dimension of the original weak form function space. Similarly we also have a test function space $\vec{\mathbf{B}}^* = \{\mathbf{b}_1^*, \dots, \mathbf{b}_N^*\}$ with the the same dimension. Any test function in the space can be expressed as $\Psi^* = \vec{\mathbf{B}}^* \mathbf{n}^*$. The solution and test spaces with the reduced dimentionality are not necessarily identical.

We also expand the delayed neutron precursors as $C_i = \vec{C}_i \mathbf{c}_i$ and $C_i^* = \vec{C}_i^* \mathbf{c}^*$, where $\vec{C}_i = \{C_{i,1}, C_{i,2}, \dots, C_{i,N}\}$ and $\vec{C}_i^* = \{C_{i,1}^*, C_{i,2}^*, \dots, C_{i,N}^*\}$. We typically let the dimension of the DNP spaces be the same as the flux dimension N . It is typical to let the DNP basis of the test space be the function of the basis of Ψ^* , for example, $C_{i,j}^* = \chi_{d,i}^T \int_S (\mathbf{b}_j^* + \tau \mathbb{L}_1 \mathbf{b}_j^*) d\Omega$ with SAAF and $C_{i,j}^* = \chi_{d,i}^T \mathbf{b}_j^*$ with CFEM-Diffusion. Thus we have

$$\mathbb{S}_{d,i}(\Psi^*, C_i) = (C_i^*, \lambda_i C_i)_{\mathcal{D}}, \quad (336a)$$

$$\mathbb{F}_{d,i}(\Psi^*, \Psi) = (C_i^*, \mathbb{P}_i \Phi)_{\mathcal{D}}. \quad (336b)$$

Then we substitute the solution and the test function in the weak form and obtain the following ODE:

$$\begin{aligned} \frac{d}{dt} \left(\mathbb{T}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}) \mathbf{n} \right) &= \left((\mathbb{F} - \mathbb{L} - \mathbb{S})(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}) - \sum_{i=1}^I \mathbb{F}_{d,i}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}) \right) \mathbf{n} + \sum_{i=1}^I \mathbb{S}_{d,i}(\vec{\mathbf{B}}^*, \vec{C}_i) \mathbf{c}_i, \\ \frac{d}{dt} \left((\vec{C}_i^*, \vec{C}_i)_{\mathcal{D}} \mathbf{c}_i \right) &= (\vec{C}_i^*, \mathbb{P}_i \mathbf{B}) \mathbf{n} - (\vec{C}_i^*, \lambda_i \vec{C}_i)_{\mathcal{D}} \mathbf{c}_i, \quad i = 1, \dots, I. \end{aligned}$$

If we define $\bar{\mathbf{n}} = \mathbb{T}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}) \mathbf{n}$ and $\bar{\mathbf{c}}_i = (\vec{C}_i^*, \vec{C}_i)_{\mathcal{D}} \mathbf{c}_i$, after some arrangement, we obtain

$$\frac{d\bar{\mathbf{n}}(t)}{dt} = (\mathbf{A}(t) - \sum_{i=1}^I \mathbf{B}_i(t)) \mathbf{T}^{-1}(t) \bar{\mathbf{n}}(t) + \sum_{i=1}^I \mathbf{C}_i(t) \bar{\mathbf{c}}_i(t), \quad (337a)$$

$$\frac{d\bar{\mathbf{c}}_i(t)}{dt} = \mathbf{B}_i(t) \mathbf{T}^{-1}(t) \bar{\mathbf{n}}(t) - \mathbf{C}_i(t) \bar{\mathbf{c}}_i(t), \quad i = 1, \dots, I, \quad (337b)$$

where

$$\begin{aligned} \mathbf{A}(t) &= (\mathbb{F} - \mathbb{L} - \mathbb{S})(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}), \\ \mathbf{B}_i(t) &= \mathbb{F}_{d,i}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}), \\ \mathbf{T}(t) &= \mathbb{T}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}), \\ \mathbf{C}(t) &= (\vec{C}_i^*, \lambda_i \vec{C}_i)_{\mathcal{D}} (\vec{C}_i^*, \vec{C}_i)_{\mathcal{D}}^{-1}. \end{aligned} \quad (337c)$$

Once the solution and test spaces are constructed, all of the coefficient matrices can be evaluated at any time t . We further re-define \mathbf{A} , \mathbf{B} and \mathbf{T} with a factoring matrix to give them a better physical meaning:

$$\mathbf{A}(t) = (\mathbb{F} - \mathbb{L} - \mathbb{S})(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}) \mathbb{F}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}})^{-1}, \quad (337d)$$

$$\mathbf{B}_i(t) = \mathbb{F}_{d,i}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}) \mathbb{F}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}})^{-1}, \quad (337e)$$

$$\mathbf{T}(t) = \mathbb{T}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}}) \mathbb{F}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}})^{-1}. \quad (337f)$$

This completes the derivation of ROM for NTE. The matrices \mathbf{A} , \mathbf{B}_i , \mathbf{T} and \mathbf{C} depends on cross sections. If the cross sections can be expressed as analytical function of time, these matrices can be analytical as well.

When $N = 1$, we end up with the so-called PKE (point-kinetics equation) parameters. \mathbf{A} becomes the reactivity and $\mathbf{B}_i, i = 1, \dots, I$ are the effective fractions for all DNP groups, \mathbf{T} becomes the mean generation time and \mathbf{C} is the effective decay constant. If $\mathbb{T}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}})$ is time independent, for example with diffusion, both $\vec{\mathbf{B}}$ and velocities are time-independent, we can scale the above PDE with $\mathbb{T}^{-1}(\vec{\mathbf{B}}^*, \vec{\mathbf{B}})$ to obtain:

$$\frac{d\mathbf{n}}{dt} = \mathbf{T}^{-1} (\mathbf{A} - \sum_{i=1}^I \mathbf{B}_i) \mathbf{n} + \sum_{i=1}^I \bar{\mathbf{C}}_i \mathbf{c}_i, \quad (337g)$$

$$\frac{d\mathbf{c}_i}{dt} = \mathbf{T}^{-1} \mathbf{B}_i \mathbf{n} - \bar{\mathbf{C}}_i \mathbf{c}_i, \quad i = 1, \dots, I, \quad (337h)$$

where \mathbf{c}_i is redefined as $\mathbf{T}^{-1}(\vec{C}_i^*, \vec{C}_i)_{\mathcal{D}} \mathbf{c}_i$ and $\bar{\mathbf{C}}_i \equiv \mathbf{T}^{-1} \mathbf{C}_i \mathbf{T}$. If we do not expand the delayed neutron precursors, the equations are

$$\frac{d\bar{\mathbf{n}}}{dt} = (\mathbf{A} - \sum_{i=1}^I \mathbf{B}_i) \mathbf{T}^{-1} \bar{\mathbf{n}} + \sum_{i=1}^I \mathbb{S}_{d,i}(\vec{\mathbf{B}}^*, C_i), \quad (338a)$$

$$\frac{d}{dt} (C_i^*, C_i)_{\mathcal{D}} = (C_i^*, \mathbb{P}_i \vec{\mathbf{B}} \mathbf{T}^{-1} \bar{\mathbf{n}})_{\mathcal{D}} - (C_i^*, \lambda_i C_i)_{\mathcal{D}}, \quad i = 1, \dots, I. \quad (338b)$$

Thus we only apply ROM on part of the system. A few applications of ROM are given in the following subsections.

3.8.1 Dumping the PKE parameters during a spatial kinetics calculation

In this case, we simply set the dimension of flux to one. We use the adjoint steady-state solution as the test function and let the actual solution normalized with total fission production n as the reduced function space. We define

$$n(t) = \mathbb{F}(\bar{\Psi}^*, \Psi) \quad (339a)$$

$$c_i(t) = (C_i^*, C_i)_{\mathcal{D}}. \quad (339b)$$

$\bar{\Psi}^*$ does not have to be Ψ^* . If it is set to \mathbf{I} , $n(t)$ will be the total fission source. We also set the normalized DNP concentrations as the reduced DNP function space. The reduced PKE is

$$\frac{d(\Lambda n)}{dt} = (\rho - \beta_{\text{eff}}) n + \sum_{i=1}^I \lambda_i c_i, \quad (339c)$$

$$\frac{dc_i}{dt} = \beta_{\text{eff},i} n - \lambda_i c_i, i = 1, \dots, I, \quad (339d)$$

where

$$\rho = \frac{(\mathbb{F} - \mathbb{L} - \mathbb{S})(\Psi^*, \Psi)}{\mathbb{F}(\bar{\Psi}^*, \Psi)}, \quad (339e)$$

$$\beta_{\text{eff},i} = \frac{\mathbb{F}_{d,i}(\Psi^*, \Psi)}{\mathbb{F}(\bar{\Psi}^*, \Psi)}, \quad (339f)$$

$$\Lambda = \frac{\mathbb{T}(\Psi^*, \Psi)}{\mathbb{F}(\bar{\Psi}^*, \Psi)}, \quad (339g)$$

$$\lambda = \frac{(C_i^*, \lambda C_i)_{\mathcal{D}}}{(C_i^*, C_i)_{\mathcal{D}}}, \quad (339h)$$

$$\beta_{\text{eff}} = \sum_{i=1}^I \beta_{\text{eff},i}. \quad (339i)$$

All of these quantities $n, \rho, \Lambda, (c_i, \beta_{\text{eff},i}, \lambda, i = 1, \dots, I)$ can be evaluated once the solution Ψ is obtained at every time step with a given fixed Ψ^* . If the same time integration as the spatial kinetics calculation is used for solving the PDE with the dumped parameters, the same scalar neutron population and DNP concentrations are expected.

3.8.2 Multi-region PKE parameters

The original development of this section was motivated by generating the PKE parameters for coupled reactors [84]. We reproduce the derivation here with minor notation improvements. The idea can be generalized into the ROM framework.

We split the entire domain into N non-overlapping subdomains and let $\cup_{m=1}^N \mathcal{D}_m = \mathcal{D}$. We define subdomain bilinear forms using the diffusion fission bilinear form as an example

$$\mathbb{F}_m(\Psi^*, \Psi) \equiv (\Psi^*, \mathbb{F}\Psi)_{\mathcal{D}_m}. \quad (340)$$

We can define similarly $\mathbb{S}_{d,m,i}(\Psi^*, C_i)$ and $\mathbb{F}_{d,m,i}(\Psi^*, \Psi)$. We then evaluate Ψ_m given the solution Ψ and \mathbf{C} such that $\forall \Psi^* \in W_0$

$$\boxed{\frac{d}{dt} \mathbb{T}(\Psi^*, \Psi_m)} + \mathbb{L}(\Psi^*, \Psi_m) - \mathbb{S}(\Psi^*, \Psi_m) = \mathbb{F}_m(\Psi^*, \Psi) - \boxed{\sum_{i=1}^I \mathbb{F}_{d,m,i}(\Psi^*, \Psi) + \sum_{i=1}^I \mathbb{S}_{d,m,i}(\Psi^*, C_i)}, \quad (341)$$

where the Ψ_m are referred to as the partial fluxes from the fission source in sub-region m . We can easily see that $\Psi = \sum_{m=1}^N \Psi_m$. We can obtain the same partial fluxes for adjoint solutions.

If we assume equilibrium, i.e. the cross sections do not deviate from those of the initial steady state, we have:

$$\mathbb{F}_m(\Psi_n^*, \Psi) = \mathbb{L}(\Psi_n^*, \Psi_m) - \mathbb{S}(\Psi_n^*, \Psi_m) = \mathbb{F}_n(\Psi^*, \Psi_m), \quad (342)$$

where Ψ and Ψ_m are the forward solutions and Ψ^* and Ψ_n^* are the adjoint solutions. Eq. (342) states that the inner product over region m of the full forward flux and the partial adjoint flux of region n is equal to the inner product over region n of the full adjoint flux and the partial forward flux of region m . We refer to this equation as reciprocity relation.

We can apply the partial adjoint solution Ψ_n^* in Eq. (341) as the test function and assume the equilibrium condition:

$$\begin{aligned} \frac{d}{dt} \mathbb{T}(\Psi_n^*, \Psi_m) &= \mathbb{F}_m(\Psi_n^*, \Psi) \\ &- \sum_{i=1}^I \mathbb{F}_{d,m,i}(\Psi_n^*, \Psi) - \mathbb{F}_n(\Psi^*, \Psi_m) + \sum_{i=1}^I \mathbb{S}_{d,m,i}(\Psi_n^*, C_i). \end{aligned} \quad (343)$$

The streaming and scattering bilinear forms \mathbb{L} and \mathbb{S} do not appear in the equilibrium condition, which makes the final equation much simpler. We will have $N \times N$ of these equations. If we define

$$n_n \equiv \mathbb{F}_n(\mathbf{1}, \Psi), \quad (344)$$

$$\bar{\Psi}_n^* \equiv \frac{\mathbb{F}_n(\Psi^*, \Psi)}{n_n}, \quad (345)$$

$$\Lambda_{n,m} \equiv \frac{\mathbb{T}(\Psi_n^*, \Psi_m)}{\mathbb{F}_n(\Psi^*, \Psi_m)}, \quad (346)$$

$$n_{n,m} \equiv \frac{\mathbb{F}_n(\Psi^*, \Psi_m)}{\bar{\Psi}_n^*}, \quad (347)$$

$$\rho_{n,m,j} \equiv \frac{\mathbb{F}_m(\Psi_n^*, \Psi_j)}{\mathbb{F}_m(\Psi^*, \Psi_j)} \cdot \frac{\bar{\Psi}_m^*}{\bar{\Psi}_n^*}, \quad (348)$$

where n_n is the total neutron production rate in sub-region n ; $\bar{\Psi}_n^*$ is the averaged importance of region n ; $\Lambda_{n,m}$, $\rho_{n,m,j}$ are the coupled-reactor point kinetics parameters; $n_{n,m}$ is the fractional neutron production rate, and we define

$$c_{i,n,m} \equiv \frac{(C_{i,n}^*, C_i) \mathcal{D}_m}{\bar{\Psi}_n^*}, \quad (349)$$

$$\beta_{\text{eff},i,n,m} \equiv \frac{\mathbb{F}_{d,m,i}(\Psi_n^*, \Psi)}{\mathbb{F}_m(\Psi_n^*, \Psi)}, \quad (350)$$

$$\lambda_{i,n,m} \equiv \frac{\mathbb{F}_{d,m,i}(\Psi_n^*, C_i)}{\mathbb{C}_{m,i}(\Psi_n^*, C_i)}, \quad (351)$$

$$\beta_{\text{eff},n,m} \equiv \sum_{i=1}^I \beta_{\text{eff},i,n,m}, \quad (352)$$

where $c_{i,n,m}$ is the partial n adjoint weighted DNP concentration in sub-region m , or the inner product over \mathcal{D}_m of the partial adjoint DNP concentrations of n and the forward DNP concentration of group i . $\beta_{\text{eff},i,n,m}$ and $\lambda_{i,n,m}$ are weighted fractions and decay constants of DNPs and we assume $\bar{\Psi}_n^*$ to be time-independent, we have

$$\begin{aligned} \frac{d}{dt} (\Lambda_{n,m} n_{n,m}) &= \sum_{j=1}^N \rho_{n,m,j} n_{m,j} - \beta_{\text{eff},n,m} n_{n,m} - n_{n,m} \\ &+ \sum_{i=1}^I \lambda_{i,n,m} c_{i,n,m}, \end{aligned} \quad (353)$$

$$\begin{aligned} \frac{dc_{i,n,m}}{dt} &= \beta_{\text{eff},i,n,m} n_{n,m} - \lambda_{i,n,m} c_{i,n,m}, \\ n, m &= 1, \dots, N; i = 1, \dots, I. \end{aligned} \quad (354)$$

The reactivity equation in Ref. [85] is given by

$$\rho_{n,m} \equiv \frac{n_{n,m}}{n_m} = \frac{n_n}{n_m} \cdot \frac{\mathbb{F}_n(\Psi^*, \Psi_m)}{\mathbb{F}_n(\Psi^*, \Psi)}, \quad (355)$$

which is different from our formulations derived from the weak form. They provide more characterizations for the coupled reactor. However, we can prove that

$$\sum_{j=1}^N \rho_{n,m,j} n_{m,j} = \rho_{n,m} \sum_{j=1}^N n_{m,j}, \quad (356)$$

i.e. $\rho_{n,m}$ is a lumped value of $\rho_{n,m,j}$, $j = 1, \dots, N$. It is noted that the reactivities do not depend on the cross sections other than the fission cross section because of the equilibrium condition we applied in Eq. (343). Delayed neutron fractions and DNP decay constants are assumed as constant in Ref. [85], so we do not compare with them. We can lump the DNP equations to

$$\frac{dc_{i,m}}{dt} = \beta_{\text{eff},i,m} n_m - \lambda_{i,m} c_{i,m}, \quad i = 1, \dots, I, \quad (357)$$

with the following definitions:

$$c_{i,m} \equiv \sum_{n=1}^N \frac{\Psi_n^*}{\Psi_m^*} c_{i,n,m}, \quad (358)$$

$$\beta_{\text{eff},i,m} \equiv \frac{\mathbb{F}_{d,m,i}(\Psi^*, \Psi)}{\mathbb{F}_m(\Psi^*, \Psi)}, \quad (359)$$

$$\lambda_{i,m} \equiv \frac{\mathbb{D}_{m,i}(\Psi^*, C_i)}{\mathbb{D}_{m,i}(\Psi^*, C_i)}. \quad (360)$$

If we assume

$$\frac{c_{i,n,m}}{c_{i,m}} = \frac{n_{n,m}}{n_m}, \text{ or } \frac{\mathbb{F}_m(\Psi_n^*, \Psi)}{\mathbb{F}_m(\Psi^*, \Psi)} = \frac{\mathbb{F}_{d,m,i}(\Psi_n^*, \Psi)}{\mathbb{F}_{d,m,i}(\Psi^*, \Psi)}, \quad (361)$$

which is true when delayed spectrum is the same as the prompt spectrum, Eq. (353) can be rewritten into

$$\begin{aligned} \frac{d}{dt} (\Lambda_{n,m} n_{n,m}) &= \rho_{n,m} (1 - \beta_{\text{eff},m}) \sum_{j=1}^N n_{m,j} - n_{n,m} \\ &+ \rho_{n,m} \sum_{i=1}^I \lambda_{i,m} c_{i,m}, \end{aligned} \quad (362)$$

Eq. (362) and Eq. (357) is the same as the equations in Ref. [85].

3.8.3 Improved quasi-static

Improved quasi-static (IQS) [86, 87] is a spatial kinetics time-integration method that involves factorizing the flux solution into a time-dependent amplitude and a space- and time-dependent shape. It is believed that time dependence of the shape is weaker than that of the flux itself, thus we can use a much smaller time step size for amplitude and a relatively large step size for shape to achieve a comparable accuracy of solving directly with the smaller time step size. We refer to the smaller time step as the micro step and to the shape evaluation time step as the macro step. Because the cost of solving the PKE – a small ODE system – for the amplitude is trivial comparing to solving for the shape, the overall computing time is expected to drop significantly. The two-time scale solution process, a micro scale for the amplitude and a macro scale for the shape, is illustrated in Fig. 4.

Based on this idea, the solution is factored as

$$\Psi(\vec{x}, t) = n(t) \psi(\vec{x}, t), \quad (363a)$$

where

$$n = \frac{\mathbb{T}(\psi^*, \Psi)}{\mathbb{T}(\psi^*, \Psi^0)}, \quad (363b)$$

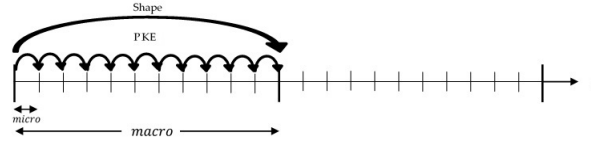


Figure 4 IQS method visualization.

$$\psi = \frac{\mathbb{T}(\psi^*, \Psi^0)}{\mathbb{T}(\psi^*, \Psi)} \Psi, \quad (363c)$$

where ψ^* is a time-independent weighting function satisfying equilibrium adjoint DNP condition. Thus

$$\mathbb{T}(\psi^*, \psi) = \mathbb{T}(\psi^*, \Psi^0). \quad (364)$$

This condition also makes the factorization unique. We define

$$c_i = \frac{(C_i^*, C_i)}{\mathbb{T}(\psi^*, \psi)}, i = 1, \dots, I. \quad (365)$$

Substituting Eq. (363a) and Eq. (365), the PKE equation with the weighting function ψ^* reads

$$\frac{dn}{dt} = \frac{\rho - \beta_{\text{eff}}}{\Lambda} n + \sum_{i=1}^I \lambda_i c_i, \quad (366a)$$

$$\frac{dc_i}{dt} = \frac{\beta_{\text{eff},i}}{\Lambda} n - \lambda_i c_i, i = 1, \dots, I, \quad (366b)$$

with the PKE parameters evaluated as

$$\rho = \frac{(\mathbb{F} - \mathbb{L} - \mathbb{S})(\psi^*, \Psi)}{\mathbb{F}(\psi^*, \Psi)}, \quad (366c)$$

$$\beta_{\text{eff},i} = \frac{\mathbb{F}_{d,i}(\psi^*, \Psi)}{\mathbb{F}(\psi^*, \Psi)}, \quad (366d)$$

$$\Lambda = \frac{\mathbb{T}(\psi^*, \Psi)}{\mathbb{F}(\psi^*, \Psi)}, \quad (366e)$$

$$\lambda_i = \frac{(C_i^*, \lambda_i C_i)_{\mathcal{D}}}{(C_i^*, C_i)_{\mathcal{D}}}, \quad (366f)$$

$$\beta_{\text{eff}} \equiv \sum_{i=1}^I \beta_{\text{eff},i}. \quad (366g)$$

It is noted that we can directly use the full solution Ψ to evaluate $\rho, \beta_i, i = 1, \dots, I$ and Λ instead of ψ because they differ from each other only by a scalar normalization factor n . It is also noted that we dropped $\frac{1}{\mathbb{T}(\psi^*, \psi)} \frac{d\mathbb{T}(\psi^*, \psi)}{dt} n$ because of the normalization condition in Eq. (364). The initial condition is

$$n(t = t_0) = 1, \quad (366h)$$

$$c_i(t = t_0) = \frac{(C_i^*, C_i^0)}{\mathbb{T}(\psi^*, \Psi^0)} = \frac{\beta_{\text{eff},i}}{\lambda_i \Lambda} \Big|_{t=t_0}. \quad (366i)$$

The shape function satisfies

$$\frac{d}{dt} \mathbb{T}(\Psi^*, \psi) = \mathbb{F}(\Psi^*, \psi) - \mathbb{L}(\Psi^*, \psi) + \mathbb{S}(\Psi^*, \psi) - \sum_{i=1}^I \mathbb{F}_{d,i}(\Psi^*, \psi) + \sum_{i=1}^I \mathbb{S}_{d,i}(\Psi^*, \frac{1}{n} C_i) - \frac{1}{n} \frac{dn}{dt} \mathbb{T}(\Psi^*, \psi), \quad (367a)$$

$$\frac{d}{dt} (C_i^*, C_i)_{\mathcal{D}} = (C_i^*, n \mathbb{P}_i \phi)_{\mathcal{D}} - (C_i^*, \lambda_i C_i)_{\mathcal{D}}, \quad i = 1, \dots, I, \quad (367b)$$

with the original initial condition. If we let the test function be the weighting function, we see

$$\begin{aligned}
& \frac{1}{n} \frac{dn}{dt} \mathbb{T}(\boldsymbol{\psi}^*, \boldsymbol{\psi}) \\
&= \left(\frac{\rho - \beta_{\text{eff}}}{\Lambda} + \sum_{i=1}^I \lambda_i \frac{c_i}{n} \right) \mathbb{T}(\boldsymbol{\psi}^*, \boldsymbol{\psi}) \\
&= \frac{(\mathbb{F} - \mathbb{L} + \mathbb{S})(\boldsymbol{\psi}^*, \boldsymbol{\Psi}) - \sum_{i=1}^I \mathbb{F}_{d,i}(\boldsymbol{\psi}^*, \boldsymbol{\Psi})}{\mathbb{T}(\boldsymbol{\psi}^*, \boldsymbol{\Psi})} \mathbb{T}(\boldsymbol{\psi}^*, \boldsymbol{\psi}) + \sum_{i=1}^I \mathbb{S}_{d,i}(\boldsymbol{\psi}^*, \frac{1}{n} C_i) \\
&= \mathbb{F}(\boldsymbol{\psi}^*, \boldsymbol{\psi}) - \mathbb{L}(\boldsymbol{\psi}^*, \boldsymbol{\psi}) + \mathbb{S}(\boldsymbol{\psi}^*, \boldsymbol{\psi}) - \sum_{i=1}^I \mathbb{F}_{d,i}(\boldsymbol{\psi}^*, \boldsymbol{\psi}) + \sum_{i=1}^I \mathbb{S}_{d,i}(\boldsymbol{\psi}^*, \frac{1}{n} C_i).
\end{aligned}$$

Substituting it into Eq. (367a), we obtain

$$\frac{d}{dt} \mathbb{T}(\boldsymbol{\psi}^*, \boldsymbol{\psi}) = 0, \quad (368)$$

which means that $\mathbb{T}(\boldsymbol{\psi}^*, \boldsymbol{\psi})$ does not change with time. We also can see that if we include $\frac{1}{\mathbb{T}(\boldsymbol{\psi}^*, \boldsymbol{\psi})} \frac{d\mathbb{T}(\boldsymbol{\psi}^*, \boldsymbol{\psi})}{dt} n$ in the PKE equation, we end up with a singular flux system. n is like a Lagrange multiplier enforcing the condition Eq. (368). The equations of flux shape $\boldsymbol{\psi}$ and DNPs $C_i, i = 1, \dots, I$ and amplitude equations Eq. (366) of $n, C_i, i = 1, \dots, I$ are coupled nonlinearly.

It has been found that evaluation of DNPs without considering the amplitude change at micro steps results in a less accurate solution. An analytical representation of the precursors is proposed as follows. Using an exponential operator, the precursor equation can be analytically solved as:

$$C_i(t) = C_i(t_{\text{old}}) e^{-\lambda_i(t-t_{\text{old}})} + \int_{t_{\text{old}}}^t n(t') \mathbb{P}_i \boldsymbol{\phi}(t') e^{-\lambda_i(t-t')} dt', \quad i = 1, \dots, I,$$

where

$$n(t') \mathbb{P}_i \boldsymbol{\phi}(t') = n(t') \beta_i(t') f(t').$$

If we assume β_i and the fission source shape f are linearly varying in time as

$$n(t') \mathbb{P}_i \boldsymbol{\phi}(t') = n(t') \left(\frac{t' - t_{\text{old}}}{t - t_{\text{old}}} \beta_i(t) + \frac{t - t'}{t - t_{\text{old}}} \beta_i(t_{\text{old}}) \right) \left(\frac{t' - t_{\text{old}}}{t - t_{\text{old}}} f(t) + \frac{t - t'}{t - t_{\text{old}}} f(t_{\text{old}}) \right),$$

we have the analytical expression for $C_i(t)$:

$$C_i(t) = C_i(t_{\text{old}}) e^{-\lambda_i \Delta t} + (a_3 \beta_i(t) + a_2 \beta_i(t_{\text{old}})) f(t) + (a_2 \beta_i(t) + a_1 \beta_i(t_{\text{old}})) f(t_{\text{old}}), \quad (369a)$$

where the integration coefficients are defined as:

$$a_1 = \int_{t_{\text{old}}}^t \left(\frac{t - t'}{\Delta t} \right)^2 n(t') e^{-\lambda(t-t')} dt', \quad (369b)$$

$$a_2 = \int_{t_{\text{old}}}^t \frac{(t' - t_{\text{old}})(t - t')}{(\Delta t)^2} n(t') e^{-\lambda(t-t')} dt', \quad (369c)$$

$$a_3 = \int_{t_{\text{old}}}^t \left(\frac{t' - t_{\text{old}}}{\Delta t} \right)^2 n(t') e^{-\lambda(t-t')} dt'. \quad (369d)$$

The amplitude $n(t)$ can be taken as piecewise linear within micro steps. If number of micro steps is one, $n(t')$ and the linear shape of fission source can be merged and we retrieve back the original equation Eq. (140a) with linear power and exponential integration.

We can use Picard iterations to solve the coupled equations as in Algorithm 3. It has been found that the simple predictor-corrector approach as detailed in Algorithm 4 can yield good results.

Algorithm 3 Nonlinear Picard iteration for IQS at a time step t .

-
- 1: $iter = 0$.
 - 2: **while** $\epsilon > tol$ or $iter < max_{iter}$ **do**
 - 3: Evaluate PKE parameters at t with Eq. (366c) to Eq. (366f) with $\Psi(t = t_{old})$.
 - 4: Perform micro PKE solves with the the *linearly* interpolated PKE parameters.
 - 5: Solve for flux equation Eq. (367a) and DNPs with Eq. (369).
 - 6: Evaluate the error

$$\epsilon = \frac{\mathbb{T}(\Psi^*, \Psi) - \mathbb{T}(\Psi^*, \Psi_0)}{\mathbb{T}(\Psi^*, \Psi_0)} \quad (370)$$

- 7: $iter = iter + 1$.
 - 8: **end while**
-

Algorithm 4 Predictor-corrector for IQS at a time step t .

-
- 1: Solve the equation Eq. (332).
 - 2: Normalize the solution to obtain the shape $\psi(t)$ with Eq. (364).
 - 3: Evaluate PKE parameters at t with Eq. (366c) to Eq. (366f).
 - 4: Perform micro PKE solves with the the *linearly* interpolated PKE parameters.
 - 5: Obtain the corrected solution as $\Psi = n(t)\psi(t)$ and evaluate DNPs with Eq. (369).
-

In multiphysics calculations [88], other physical quantities, such as temperature, are affected by reaction rates and subsequently affect the operators of the flux equations. For IQS, this feedback affects both the shape equation and the parameters of the PRKE; thus, it is an additional nonlinear component to the already non-linear shape-amplitude equations. Each of these components have different temporal behaviors; so it may be beneficial for efficiency to evaluate them on different time scales. The amplitude is more rapidly varying than the shape which is computationally expensive to obtain and is evaluated only on macro-time steps. In multiphysics simulations, one may take advantage of a fine-scale power distribution in the coupled physics components (temperature). We have an intermediate step to update other physical quantities and the affected PKE parameters.

4 Rattlesnake Design

4.1 Relation to libMesh/MOOSE

Rattlesnake is a MOOSE-based radiation transport application, while MOOSE provides a multiphysics modeling and simulation (M&S) environment bridging physics and the underlying mesh framework provided by libMesh. These dependencies are reflected by the repository structures, where MOOSE is a required submodule of Rattlesnake and libMesh is a required submodule of MOOSE. The roles of each of them from bottom to top are described below.

The libMesh library is a C++ FEM framework. It offers mesh related operations, including build-in mesh generation, mesh loading with various mesh formats including ExodusII, GMV, Gmsh, VTK, etc., mesh partitioning with Metis, Parmetis, Hilbert, etc., and various mesh modification utilities. libMesh supports mixed types of 1D, 2D and 3D elements. It provides dimension agnostic definitions of finite element families and quadratures for assembling the algebraic system. It also interacts with PETSc, Trilinos and SLEPSc for sparse linear algebra. All of these capabilities are enabled on serial and parallel platforms.

MOOSE is a multiphysics framework primarily developed by Idaho National Laboratory (INL). It provides a high-level interface to some of the most sophisticated nonlinear solver technology on the planet. MOOSE presents a straightforward API that aligns well with the real-world problems scientists and engineers (who are not necessarily advanced mathematicians or computer scientists) need to tackle. Every detail about how an engineer interacts with MOOSE has been thought through, from the installation process through running simulations on state of the art supercomputers. MOOSE integrates all the pluggable interfaces, including kernels, auxkernels, postprocessors, materials, etc., which can be specific for each physics. MOOSE provides generic tools for application developers to describe the system and interact with libMesh and the underlying solvers.

Rattlesnake is a radiation transport solver for the multigroup linear Boltzmann radiation transport equation. It models interactions of radiation particles such as neutron, thermal photon, etc., with background materials. It provides various discretization schemes, including the self-adjoint angular flux (SAAF) and least squares (LS) formulation with continuous finite element method (FEM) and the first-order formulation with discontinuous FEM. The angular discretization can be applied with the discrete ordinates method (S_N), the spherical harmonics expansion method (P_N) or using the diffusion approximation. All these schemes are leveraged with the multi-scheme capability by allowing the assignment of more suitable schemes on various subdomains with different levels of resolution in order to optimize the use of computing resources. Rattlesnake solves steady-state, transient and eigenvalue problems with arbitrary order of scattering anisotropy. It also inherits all the features provided by the MOOSE framework, including unstructured higher-order meshes, massive parallelization, dimension agnosticism, etc. Rattlesnake is designed for multiphysics simulations. Its applications to the fully-coupled multiphysics simulation and the tightly-coupled multiphysics simulations with data transfer have been both successfully demonstrated. It interacts with YAKXS, a multigroup cross section library management toolkit, for incorporating the cross sections generated with lattice physics codes. Rattlesnake relies on the material properties describing how radiation particles interact with background media and field variables like the temperature and the material density to compute various quantities of interests, including the radiation flux distribution, reaction rates, radiation-induced power profile, etc. These can be provided to the analyst or to other physics for multiphysics simulations.

This design enables an efficient and powerful development environment. Developments in MOOSE/libMesh can benefit all physics. On the other hand, particular capabilities required by one physics can typically enhance the simulation of other physics. For example, mortar FEM capabilities can be used both by solid mechanics for contact problems and by radiation transport for multi-scheme applications. While MOOSE and libMesh developers are typically working on their specific needs, it does not prevent them from benefiting other areas. It has been commonly seen that application developers send patches to the framework for fixing defects or adding new capabilities and vice-versa. This pattern enables large projects otherwise infeasible by a few developers. The code structure also encourages the modularity of code development, resulting in much more readable and maintainable codes.

4.2 Transport system actions

Rattlesnake is faced with a unique challenge not present in other physics such as solid mechanics, fluid dynamics, etc. In multigroup (or multiband) radiation transport, because of the seven-dimensional phase space, the number of variables is typically significantly larger than in these other physics, sometimes thousands of times larger. Consequently, adding all the variables, kernels, auxiliary variables, postprocessors with the MOOSE syntax⁵ would be too tedious and error prone to be a valid option. It becomes crucial – if not indispensable – to use the action system provided by MOOSE for adding variables and MOOSE objects automatically with a simplified and dedicated input syntax.

There are also other reasons for using the action system in Rattlesnake. In particular, we can provide additional options for controlling how the MOOSE objects are added. For example, we can have an option to switch the discretization schemes used for solving the transport equation, so that users do not have to do this manually. As a result, chances of input errors can be significantly reduced. MOOSE objects frequently interact with materials and executioners (or solvers). Action systems can be used to automatically set up necessary parameters for materials and executioners based on the options in use for setting up the transport system. Rattlesnake’s action system provides defaults for many advanced parameter so the user does not need to have expert knowledge of radiation transport algorithms.

The desired user inputs for setting up the transport system are now explained. As showed in Section 2, the input file should describe

1. what particle the transport system is for;
2. what the equation type (e.g. steady-state source or eigenvalue, or transient) is;
3. the material properties with transport materials;
4. the volumetric external source with mesh blocks if any;
5. boundary conditions (type and the non-homogeneous part if any) with mesh side sets.

All the weak forms were detailed in Section 3. Rattlesnake’s transport action system also allows the discretization schemes to be defined on sub-domains, thus allowing the user to specify different discretization methods on different parts of the domain. In such cases, the sub-domain boundary with side sets need to be specified and appropriate boundary conditions need to be applied thereon. We could also have multiple schemes on the sub-domains. When two sub-domains with different schemes share an interface, proper interface conditions for these two schemes need to be imposed.

The Rattlesnake input syntax follows the hierarchical structure of the particle transport problem. One or more *TransportSystems* are defined, each of which contains one or more discretization blocks. A discretization block is associated with one or more subdomains. The *TransportSystems* block contains input parameters pertinent to all the discretization blocks, such as the equations type, the number of energy groups, etc.; in contrast a discretization block contains parameters that pertain to the discretization itself: the quadrature type and order for S_N methods, the P_N order, the FE family and order etc. The following listing includes the commonly used parameters. The complete list of parameters can be found in Rattlesnake user manual.

```
[TransportSystems]
particle = |common|neutron|thermal|...|
equation_type = |steady-state|transient|eigenvalue|
G = <integer number greater than 0>
VolumetricSourceBlock = <vector of block names>
VolumetricSource = <vector of real numbers>
VolumetricSourceFunc = <vector of transport solution functions>
PointSourceValue = <real number>
PointSourceLocation = <x, y and z coordinates of the point source>
VacuumBoundary = <vector of side set names>
```

⁵The MOOSE syntax is the input syntax with which each individual variable either primal or auxiliary, user object, postprocessor, kernel either primal or auxiliary, boundary condition, etc. can be added.

```

SurfaceSource = <vector of transport solution functions>
ReflectingBoundary = <vector of side set names>
WhiteBoundary = <vector of side set names>
DirichletBoundary = <vector of side set names>
DirichletFunction = <vector of transport solution functions>
for_adjoint = <logical>
for_math_adjoint = <logical>

[./scheme1]
  scheme = SAAF-SN-CFEM
  block = <vector of block names>
  verbose = |0|1|2|3|
  prefix = <string>
  prefix_variables = <string>
  material_prop_namesapce = <string>
  group_collapsing = <vector of integer numbers>
  family = |LAGRANGE|MONOMIAL|...|
  order = |CONSTANT|FIRST|SECOND|...|
  group_weights = <vector of real numbers>
  collapse_scattering = <logical>
  balance_table = <logical>
  balance_table_on = |linear|timestep_end|...|
  fixed_jacobian = <logical>
[../]
[./scheme2]
  [All parameters for the scheme defined on another set of blocks]
[../]
[]

```

[We use this scheme as an example]
 [The blocks these scheme is applied on]
 [Control the screen print-out]
 [Name prefix of all added MOOSE objects]
 [Name prefix of all added MOOSE variables]
 [Name prefix of all material properties in use]
 [The coarse group index of all fine groups]
 [Shape function type]
 [Shape function order]
 [Optional group weightings]
 [To evaluate scattering sources for kernels]
 [True to add postprocessors for the balance table]
 [When balance table should be evaluated]
 [If Jacobian is changing during Newton iterations]

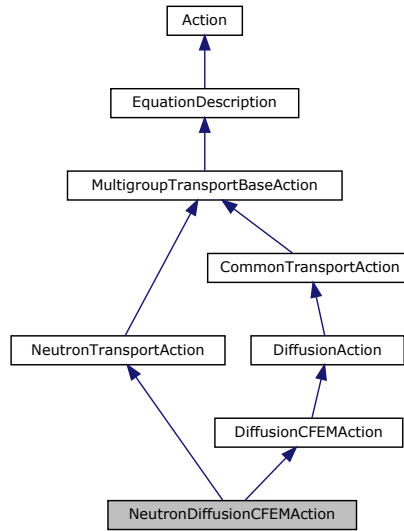


Figure 5 Action for setting up neutron diffusion with CFEM.

Fig. 5 shows the six-level inheritance graph of the action system for setting up a neutron diffusion system with CFEM. First of all, at the root of this graph is the MOOSE Action. Any action setting up a transport system, or transport system action (TSA), is a MOOSE Action. It can be invoked or driven with a defined input syntax. And it can be (and it is) invoked multiple times by multiple tasks. The action can know which task is invoking it by checking its member variable `_current_task`. All TSAs can be invoked by the following tasks: *after_mesh_setup*, *before_add_variable*, *add_variable*, *add_user_object*, *add_postprocessor*, *add_kernel*, *add_bc*, *add_aux_material*, *set_coupling_matrix*, and *after_everything*. Among them, *add_variable*, *add_user_object*, *add_postprocessor*, *add_kernel*, *add_bc* are the existing MOOSE tasks. The other ones are specifically registered by Rattlesnake. *after_mesh_setup* is used for some mesh related setups. *before_add_variable* is used for adding necessary objects before adding variables. For example, adding the angular quadrature should happen in this task because Rattlesnake needs the number of directions in the quadrature to add the appropriate angular fluxes variables. *add_aux_material* is used to add some auxiliary materials after the

normal materials are added. For instance, the stabilization parameter used by SAAF-SN for each material is treated as an auxiliary material and added at this stage. *set_coupling_matrix* is used to specify a desired pattern in the off-diagonal terms of the Jacobian for preconditioning purposes. *after_everything* is the final task before the beginning of the solve. We currently use it to print the summary of the action. The detailed ordering of all tasks including the above mentioned can be checked with the task dependency graph in the source code, or using the *show_actions* option in the *Debug* block in the input files.

There is a total of six levels of actions above MOOSE action. One level above it is the *EquationDescription* action, whose valid parameters are the parameters in the *TransportSystems* block of the input. These parameters describe the transport equation and are passed into the derived actions. Operations for setting up the global system – for example setting some parameters for the executioner, adding the postprocessor for eigenvalue, etc. – can be implemented here. The interaction with Executioner happens in the first level of the previous 6-level action system. Interactions with the material system to set the proper material parameters for simplifying the input syntax are performed at this level. It is also where the multi-scheme interface conditions are added.

Immediately on top of *EquationDescription* is the *MultigroupTransportBaseAction*. Its valid parameters contain those from *EquationDescription* and some more basic parameters for setting up a transport system. It contains the block where the system is defined on and the number of energy groups to be added by the action. It provides some utility functions to consistently enforcing some naming conventions or the verbose level. It also defines functions to retrieve information such as all the materials, finding the blocks and side sets restricted to subdomains and to each transport system, etc. It acts based on two sets of virtual function calls: one for CTE and another for particle related terms, which need to be implemented in the specific derived (and otherwise empty) classes. Some information describing what the action has added is printed on the screen at the end of the action, depending on the verbose level.

Fig. 6 (the doxygen page with the focus on *CommonTransportAction*) gives some insight regarding the purpose of *CommonTransportAction*. It provides the material property names for CTE. It specifies if we want to evaluate the ingroup scattering sources and use them in the kernels. This level of action also specifies what shape functions will be used for the primal variables. It states if we want to create the balance table for our simulations. All of the CTE operators shared by all different discretization schemes should go here.

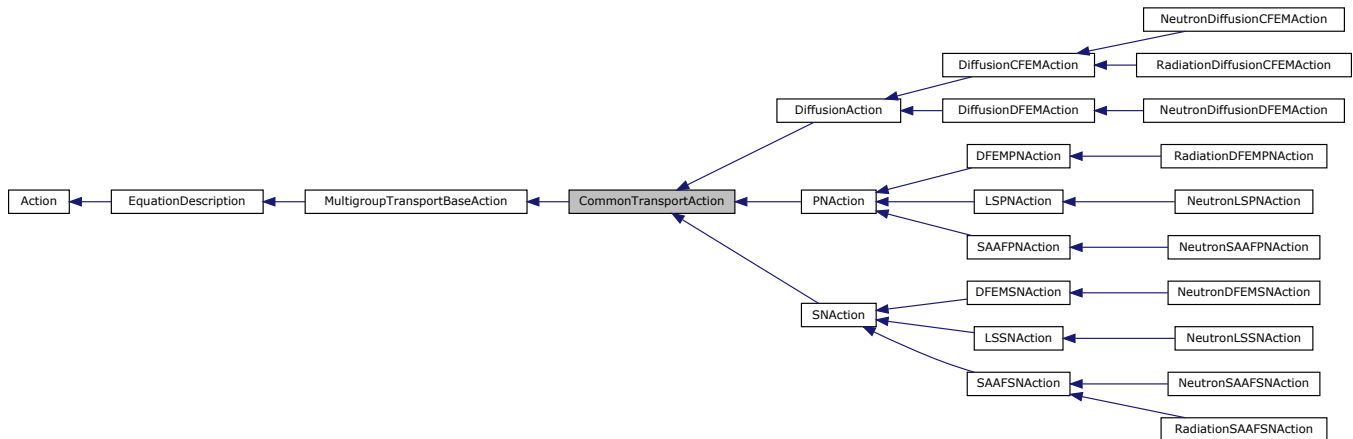


Figure 6 Common transport action.

The next level of inheritance is for the angular discretization as shown in Fig. 5 and Fig. 6. Three are currently supported: diffusion, P_N and S_N . CFEM diffusion and DFEM diffusion share a lot of kernels and boundary conditions. This level of action is the perfect place for these common kernels and BCs. The next level is the combination of angular and spatial discretizations. This level is complete for setting up a CTE.

NeutronTransportAction is directly derived from *MultigroupTransportBaseAction* because it requires utilities provided in *MultigroupTransportBaseAction*. It handles fission related objects for NTE. However it is not complete by itself for setting up a NTE. It has to be inherited by the six levels of action in Fig. 5 for setting up NTE with a particular discretization scheme. At first glance, Fig. 5 seems to show a diamond inheritance. This situation is avoided by using C++ virtual inheritance. When an object of *NeutronDiffusionCFEMAction* is created, virtual inheritance of

MultigroupTransportBaseAction ensures that only one *MultigroupTransportBaseAction* object will be created and that all derived objects are pointing to it.

The full picture of the currently implemented actions is given in Fig. 7. We only have neutron, and thermal radiation

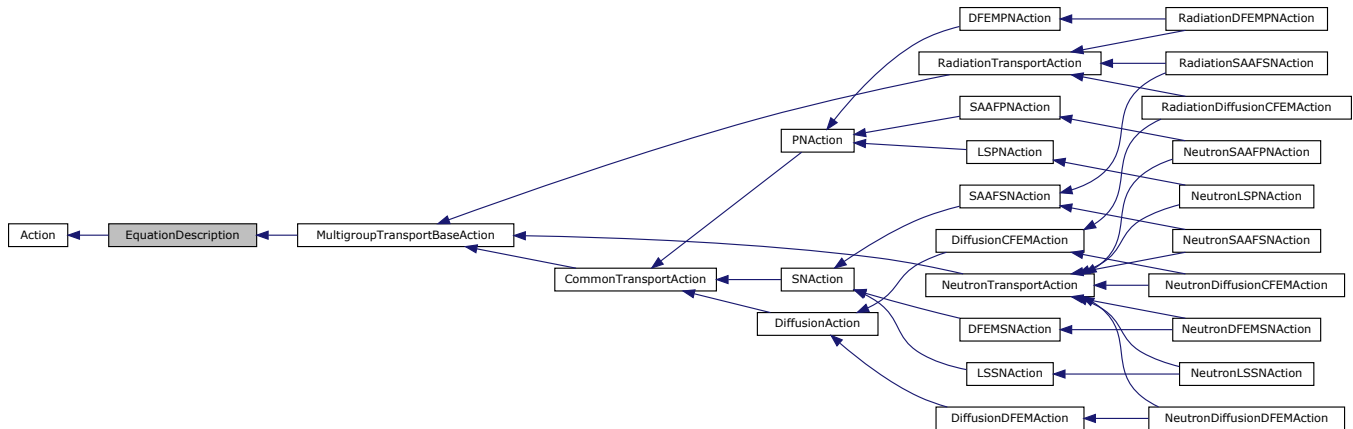


Figure 7 Full picture of all action for setting up transport systems.

transport at this moment and thermal radiation transport is only working with diffusion, SAAF-SN and DFEM-PN.

One last question would be to know how the sub-blocks of *TransportSystems* in the input add the required TSA. For each syntax, MOOSE creates an action which is statically associated with the syntax. We cannot associate our TSAs like the *NeutronDiffusionCFEMAction* with the syntax, otherwise multiple TSAs would be added and we would have to guard the action so that they can act only when the scheme parameter is in agreement with the action type. In Rattlesnake, we implemented something called action of actions or meta action.

We have another meta action, *AddTransportSystemAction*, which is the action associated with the sub-blocks of *TransportSystems*. It will check the parameter *scheme* and import the parameters in *TransportSystems* block to generate the proper TSA into the action warehouse on the fly. To do so, all valid TSAs need to be registered in it. It is noted that this action should not be used for interacting with Executioner, because Executioner is a global object, while this action is block restricted. All the valid parameters for a particular scheme can be dumped with the discretization name, such as SAAF-CFEM-SN, DFEM-SN, etc.

For code developing, we should always think first where we should put your codes. We should always protect your codes with *mooseErrors* when it is not supposed to work with some optional parameters or when some capabilities have not been implemented.

4.3 Extendable neutronic materials

Rattlesnake creates a base material *MultigroupTransportMaterial* for all transport materials. The class accepts 11 private parameters to indicate how the material properties should be declared and evaluated. These private parameters are set by the transport systems. For this reason, a transport material can only be used for one transport system. These private parameters are:

- *ngroup* - number of energy groups or frequency bands.
- *forDiffusion* - a logical parameter to ask the derived material to either evaluate diffusion coefficients and removal cross sections or total cross sections.
- *for_transport_update* - a logical parameter valid only when *forDiffusion* is false. It tells the derived material to evaluate the diffusion coefficients and transport cross sections, which are to be used for evaluating the drift vectors.
- *diffusion_coefficient_scheme* - an enumeration which controls how the diffusion coefficients are to be evaluated.

- *forAdjoint* - a logical parameter to control if the scattering matrix should be transposed.
- *allowVacuum* - a logical parameter to tell the derived material if a check to ensure non-zero total cross section is needed.
- *computingNA* - cut-off order of scattering anisotropy.
- *forPN* - a logical parameter to tell the derived material to evaluate extra properties on sides for P_N methods.
- *forLS* - a logical parameter to tell the derived material to evaluate extra properties on sides for LS methods.
- *namespace* - a prefix of all material property names to avoid chances of conflict (since two objects cannot be given the same name).

It also have 6 normal parameters:

- *speed_required* - a logical parameter to tell the derived material that particle speed properties should be declared and evaluated.
- *isMeter* - a logical parameter to tell the derived material if the mesh is in unit of meters (as opposed to centimeters).
- *disp_x* - the nodal variable for the x-displacement. It is valid for Cartesian coordinate only.
- *disp_y* - the nodal variable for the y-displacement. It is valid for Cartesian coordinate only with a dimension greater than 1.
- *disp_z* - the nodal variable for the z-displacement. It is valid for Cartesian coordinate with a dimension greater than 2 and RZ coordinate.
- *disp_r* - the nodal variable for the r-displacement. It is valid for R spherical coordinate and RZ coordinate.

The density ratio with respect to the original density on the undisplaced mesh on the quadrature points are evaluated by the function *computeDensityRatio()* of this base material. It also declares a real property named after the material name to facilitate getting the blocks where this material is defined on. Immediately derived from this material are *NeutronicsMaterial* and *TRMaterial*. The inheritance graph can be seen in Fig. 8.

All neutronics materials declare a set of material properties, which are listed in Table 45. This list could be useful for advanced users who want to utilize those properties for postprocessing. They are declared on the blocks where the materials are defined. Diffusion coefficients are evaluated on element faces for diffusion calculations. Total cross sections are evaluated on element faces for P_N and LS. *sigma_scattering_n<1>* and neutron velocities are evaluated for P_N .

YAKXS is a general toolkit for managing the multigroup cross sections for neutron transport calculations developed along with Rattlesnake. It provides an XML cross section library format for storing multigroup cross sections. YAKXS provides operations like cross section interpolation, mixing, fitting, collapsing, etc. Users and developers are insulated from the complexity of these operations and interact with YAKXS' high-level interfaces. YAKXS contains ten C++ classes and one *Utility* sub-namespace:

1. *Utility* sub-namespace collects a set of useful data structures and functions for cross section processing, like converting an isotope name to its material number, getting the default class of an isotope, etc.
2. *MultigroupLibrary* holds the raw data loaded from a library in YAKXS format. It comes along with manipulators, accessors and writers. *MultigroupLibrary* can be outputted into and be constructed from a data file in the YAKXS XML or binary format.
3. *MixingTable* regulates the raw data for the mixing operation. It holds the microscopic cross sections at a particular state. *MixingTable* can be constructed from *MultigroupLibrary* through a selection or interpolation operation.

Table 45 Neutronics material properties.

Name	Notation	Type	Declared if
neutron_speed_g<g>	v_g	Real	Transient ^a
neutron_speed	$v_g, g = 1, \dots, G$	std::vector<Real>	Transient
diffusion_coefficient_g<g>	D_g	Real	Diffusion ^b
diffusion_coefficient	$D_g, g = 1, \dots, G$	std::vector<Real>	Diffusion
vector_diffusion_coefficient_g<g>	D_g	RealVector	Diffusion
vector_diffusion_coefficient	$D_g, g = 1, \dots, G$	std::vector<RealVector>	Diffusion
tensor_diffusion_coefficient_g<g>	D_g	RealTensor	Diffusion
tensor_diffusion_coefficient	$D_g, g = 1, \dots, G$	std::vector<RealTensor>	Diffusion
sigma_removal_g<g>	$\sigma_{r,g}$	Real	Diffusion
sigma_removal	$\sigma_{r,g}, g = 1, \dots, G$	std::vector<Real>	Diffusion
sigma_total_g<g>	$\sigma_{t,g}$	Real	NotDiffusion ^c
sigma_total	$\sigma_{t,g}, g = 1, \dots, G$	std::vector<Real>	NotDiffusion
sigma_scattering_g<p>_g<g>	$\sigma_{s,l}^{p \rightarrow g}, l = 0, \dots, L$	std::vector<Real>	ScatteringPattern ^d
sigma_scattering	$\sigma_{s,l}^{p \rightarrow g}, p, g = 1, \dots, G;$ $l = 0, \dots, L$	std::vector<CSR<Real>>	NonzeroScattering ^e
phase_function_g<p>_g<g>	$p_{p \rightarrow g}(\mu)$	PhaseFunction pointer	UsingPhaseFunction ^f
sigma_scattering_for_ phase_func_g<p>_g<g>	$\sigma_{s,phase}^{p \rightarrow g}$	Real	UsingPhaseFunction
nu_sigma_fission	$\nu\sigma_{f,g}, g = 1, \dots, G$	std::vector<Real>	Fissile ^g
nu_sigma_fission_g<g>	$\nu\sigma_{f,g}$	Real	Fissile
kappa_sigma_fission	$\kappa\sigma_{f,g}, g = 1, \dots, G$	std::vector<Real>	Fissile & Plus ^h
kappa_sigma_fission_g<g>	$\kappa\sigma_{f,g}$	Real	Fissile & Plus
sigma_fission	$\sigma_{f,g}, g = 1, \dots, G$	std::vector<Real>	Fissile & Plus
sigma_fission_g<g>	$\sigma_{f,g}$	Real	Fissile & Plus
fission_spectrum	$\chi_{p,g}, g = 1, \dots, G$	std::vector<Real>	Fissile & Plus
chi_g<g>	χ_g	Real	Fissile & FissionPattern ⁱ
beta_i<i>	β_i	Real	Fissile & $I > 0$ ^j
lambda_i<i>	λ_i	Real	Fissile & $I > 0$
chi_delay_i<i>	$\chi_{g,i}, g = 1, \dots, G$	std::vector<Real>	Fissile & Plus
chi_delay_g<g>	$\chi_{g,i}, i = 1, \dots, I$	std::vector<Real>	Fissile & FissionPattern
sigma_nalpha_g<g>	$\sigma_{a,g}$	Real	Provided ^k
sigma_nalpha	$\sigma_{a,g}, g = 1, \dots, G$	std::vector<Real>	Provided
sigma_capture_g<g>	$\sigma_{c,g}$	Real	Provided
sigma_capture	$\sigma_{c,g}, g = 1, \dots, G$	std::vector<Real>	Provided
sigma_absorption_g<g>	$\sigma_{a,g}$	Real	Plus
sigma_absorption	$\sigma_{a,g}, g = 1, \dots, G$	std::vector<Real>	Plus

^a Transient means that the problem is transient;^b Diffusion means that scheme is equal to CFEM-Diffusion or DFEM-Diffusion;^c NotDiffusion means that scheme is equal to any other than CFEM-Diffusion and DFEM-Diffusion;^d ScatteringPattern means that the property will be declared for the non-zero scattering entries;^e NonzeroScattering means that there is scattering for this material. For diffusion calculations, this means that there is cross-group scatterings because in-group scatterings are merged into total as removal cross sections and thus never used;^f If scattering phase functions are used;^g Fissile means that the material is fissile;^h Plus means that additional properties are required;ⁱ FissionPattern means that the property will be declared for the group with non-zero fission neutron yields;^j $I > 0$ means that I is greater than 0; Note that the material has to be fissile to provide delayed neutron data;^k Provided means that the cross section data is provided by the input;

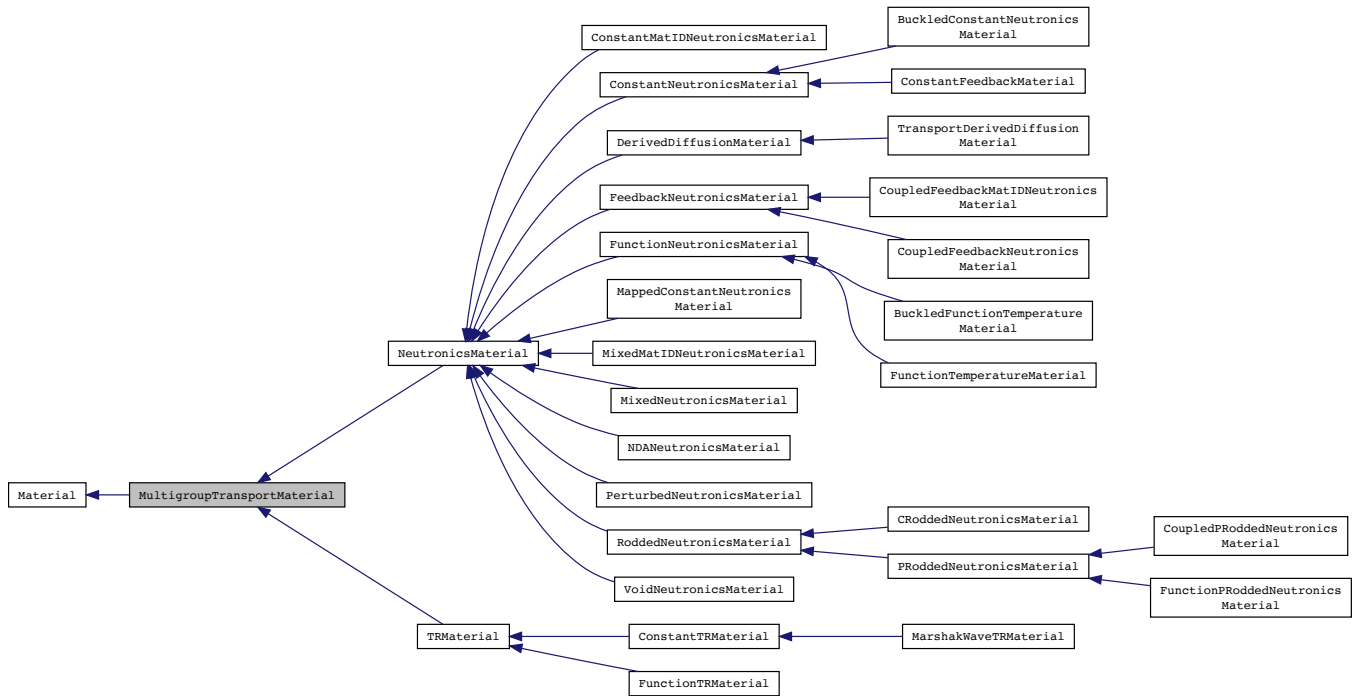


Figure 8 Full picture of all transport materials.

4. *Mixture* contains the macroscopic cross sections, which can be directly used by the transport solvers. *Mixture* can be outputted into a data file in the INSTANT XML format. However, *mixture* is not constructed from a data file in the INSTANT XML format.
5. *MixedMultigroupLibrary* contains the macroscopic cross sections of all state points in *MultigroupLibrary*. *MixedMultigroupLibrary* can be obtained from *MultigroupLibrary* through mixing or folding operations. If a folding operation is performed, new variables will be introduced and the atomic density dependency on these new variables is folded into the generated library. *MixedMultigroupLibrary* can be outputted into and constructed from a data file in the YAKXS XML or binary format.
6. *InputXS* is one of the macroscopic cross section holders. *InputXS* provides constant macroscopic cross sections to transport solvers. *InputXS* can be constructed from *Mixture*. *InputXS* can be outputted into a data file in the INSTANT XML format and can be constructed from a data file in the INSTANT XML format.
7. *PerturbedInputXS* is one of the macroscopic cross section holders. *PerturbedInputXS* provides to transport solvers a simple model for cross sections with various feedback effects. *PerturbedInputXS* can be constructed from a RELAP-5 input file containing the cross sections data with all the perturbation coefficients or from a *MixedMultigroupLibrary* through fitting.
8. *FunctionInputXS* is one of the macroscopic cross section holders. Every cross section can be a function varying in space and time.
9. *YAKXScreator* is designed for cross section generators to create the multigroup library in YAKXS format.
10. *TransmutationLibrary* holds the raw data loaded from a transmutation library.
11. *WorkingTransmutationLibrary* contains the preprocessed data from *TransmutationLibrary* which can be used for transmutation calculations directly. The number of isotopes in the *WorkingTransmutationLibrary* does not have to be the full list of isotopes in the *TransmutationLibrary*, from which the working library are generated.

All of them are included in YAKXS *namespace*. Cross section processing capabilities are built with the classes and utility functions. The relations among these classes are illustrated in Fig. 9. Rattlesnake neutronics materials interact with YAKXS for loading the cross sections and evaluate the material properties required for neutron transport calculations.

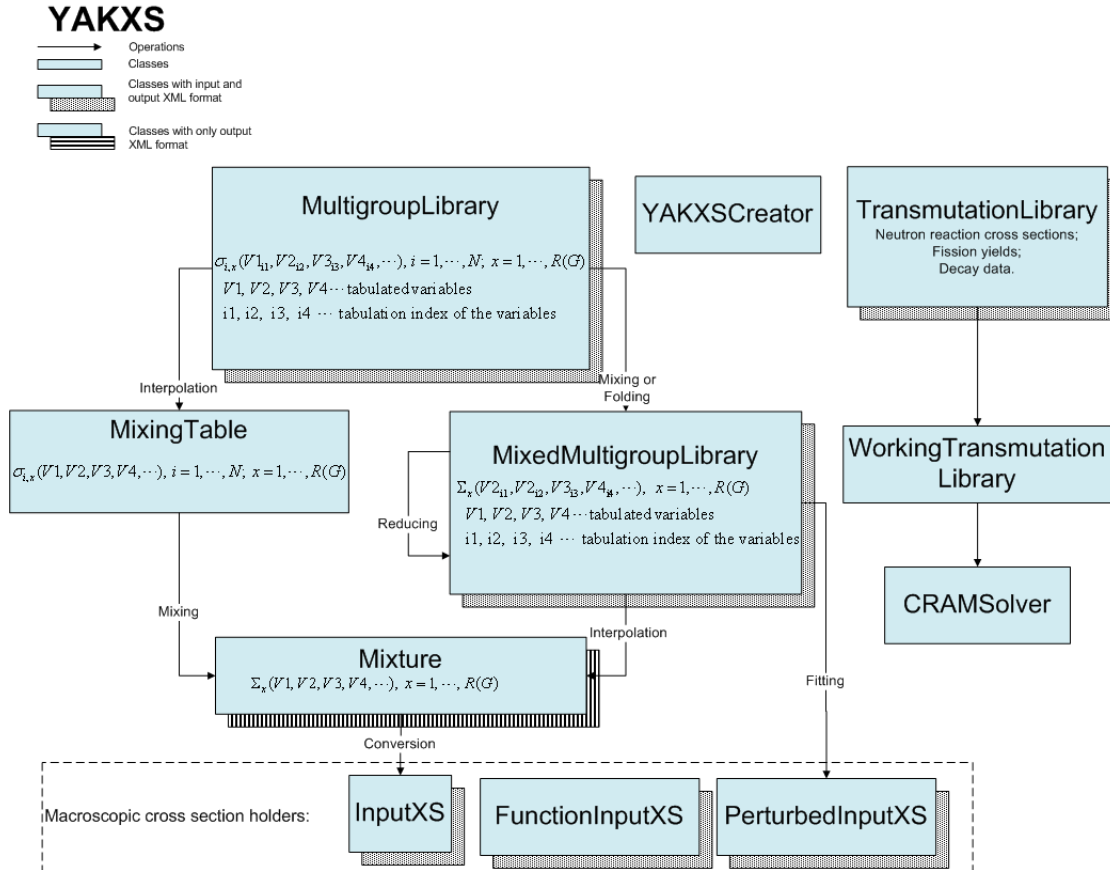


Figure 9 Graphical view of YAKXS.

The derived class then interacts with YAKXS to bring the nuclear data in and evaluate the properties on quadrature points based on them. For example of `ConstantNeutronicsMaterial`, which can load a macroscopic cross section file in INSTANT XML format and construct a YAKXS object in type of `InputXS`. `ConstantNeutronicsMaterial` then passes the request from the base class to `InputXS` object to obtain the macroscopic cross sections. Another example is `MixedNeutronicsMaterial`, which loads a multigroup library in YAKXS XML format and construct an object in type of `MultigroupLibrary`. This neutronics material is then simply do the interpolation and mixing with a fixed set of grid values and atomic densities and generate an object in type of `Mixture`. `MixedNeutronicsMaterial` then passes the request from the base class to `Mixture` object to obtain the macroscopic cross sections. A more sophisticated material is `CoupledFeedbackNeutronicsMaterial`, which loads a multigroup library in YAKXS XML format and construct an object in type of `MultigroupLibrary` similarly as `MixedNeutronicsMaterial`. It, however, can couple other variables in and use their values on quadrature points to generate the `Mixture` on the fly. Thus the variable dependent cross sections can be evaluated. `RoddedNeutronicsMaterial` can be used to handle the control-rod movements. Material names with `MatID` in it are for the materials who can load multiple data as a whole and branch the data with the local material ID of elements. The materials are highly extendable for particular usage. For instance, `BuckledConstantNeutronicsMaterial` is almost identical to `ConstantNeutronicsMaterial` but have an extra input parameter, `buckling`, to alter the removal or total cross sections in `ConstantNeutronicsMaterial`.

4.3.1 Flux-Corrected On-The-Fly Group Condensation

With the multigroup approximation, fine-group cross sections are typically pre-condensed using a fixed fine spectrum within each energy group. For clarity, we will refer to the energy groups for multigroup calculations as the coarse energy groups. It is noted that fine and coarse here are relative, i.e. fine-group does not mean hundreds of groups in lattice physics and coarse-group is not necessarily few groups. In real settings, the fine spectrum might differ significantly from the spectrum used for pre-condensation, for example near a moving strong neutron absorber, making the coarse group cross sections and solutions less accurate. If we assume that the *change* in fine spectrum is smooth, and it thus can be represented by interpolating fluxes across coarse groups, we can construct a fine spectrum better representing reality. This better spectrum can then be used to condense the group cross sections, which in turn produce improved coarse group solutions. We call this concept on-the-fly group condensation. On-the-fly group condensation can potentially produce significantly more accurate solutions than the coarse group solutions without costly fine group calculations. It can be seen that on-the-fly group condensation introduce a nonlinear effect of the fluxes to the residual, but this effect is typically not strong enough to adversely affect the convergence of the employed PJFNK solver in Rattlesnake. On-the-fly group condensation is handled by the base class `MultigroupTransportMaterial`. The required information for group condensation such as the fine spectrum, group structure, etc. are provided by the derived materials. The key of the on-the-fly group condensation is how to properly prolong the coarse group fluxes and use them to correct the fine spectrum. We adapted the same procedure in Ref. [89] for modifying the fine spectrum as:

$$f_{\text{corrected},g} = f_g \frac{(\Pi^{-1}\Phi)_g}{(\Pi^{-1}\Pi f)_g}, g = 1, \dots, G, \quad (371)$$

where G is the number of fine groups. Π is the projection operation to project fine group solutions onto coarse groups

$$\Pi(f_g, g = 1, \dots, G) = \sum_{g \in p} f_g, p = 1, \dots, P. \quad (372)$$

P is the number of coarse groups. $g \in p$ means that energy range of fine group g is within the coarse energy group p . It is noted that the fine groups are embedded in coarse groups, i.e. energy range of any fine group does not extend to more than one coarse energy ranges. Π^{-1} is the prolongation operation to prolong coarse groups solutions to fine groups solutions. We currently implemented two schemes for the prolongation. The first scheme assumes the fine spectrum is linear with respect to energy E in each coarse groups:

$$f(E) = \frac{F_{p-1/2} + F_{p+1/2}}{2} + (F_{p-1/2} - F_{p+1/2}) \frac{2(E - \bar{E}_p)}{\Delta E_p}, E \in [E_{p-1/2}, E_{p+1/2}] \quad (373a)$$

$$\bar{E}_p = \frac{E_{p-1/2} + E_{p+1/2}}{2}, \quad (373b)$$

$$\Delta E_p = E_{p-1/2} + E_{p+1/2}, \quad (373c)$$

$$F_{p+1/2} = \begin{cases} 3\frac{F_1}{\Delta E_1} - \frac{F_2}{\Delta E_2}, & p = 0 \\ \frac{\frac{F_p}{\Delta E_p} + \frac{F_{p+1}}{\Delta E_{p+1}}}{2}, & 1 \leq p < P, \quad p = 0, \dots, P, \\ 3\frac{F_p}{\Delta E_p} - \frac{F_{p+1}}{\Delta E_{p+1}}, & p = P \end{cases} \quad (373d)$$

then the fine group spectrum is evaluated as

$$f_g = \frac{f(E_{g-1/2}) + f(E_{g+1/2})}{2} \Delta E_g. \quad (373e)$$

We use F to denote the quantity to be prolonged. Rattlesnake refers this scheme as *LinearESimple*. This scheme is identical to the *Piecewise Linear $\phi(E)$* scheme in Ref. [89]. The second scheme uses piece-wise second order functions with respect to lethargy $u = \log E$ to fit the spectrum within each coarse group

$$f(u) = \frac{F_p}{\Delta u_p} + b_p \frac{2(u - \bar{u}_p)}{\Delta u_p} + c_p \frac{1}{2} \left(3 \left(\frac{2(u - \bar{u}_p)}{\Delta u_p} \right)^2 - 1 \right), u \in [u_{p-1/2}, u_{p+1/2}], \quad (374a)$$

$$\bar{u}_p = \frac{u_{p-1/2} + u_{p+1/2}}{2}, \quad (374b)$$

$$\Delta u_p = u_{p-1/2} + u_{p+1/2}. \quad (374c)$$

We choose to use lethargy instead of energy for fitting because the typical energy ranges are more evenly distributed in lethargy. The $2P$ unknown coefficients $b_p, c_p, p = 1, \dots, P$ are evaluated with the following $2P$ constraints:

$$c_1 = c_P = 0, \quad (374d)$$

$$f^+(u_{p+1/2}) = f^-(u_{p+1/2}), p = 1, \dots, P-1, \quad (374e)$$

$$\partial_u f^+(u_{p+1/2}) = \partial_u f^-(u_{p+1/2}), p = 1, \dots, P-1. \quad (374f)$$

Superscript $+$ and $-$ mean that the quantity is evaluated with Eq. (374a) in the plus and minus energy range. Once the coefficients are obtained, fine groups spectrum is evaluated as

$$f_g = \int_{u_{g-1/2}}^{u_{g+1/2}} f(u) du. \quad (374g)$$

Rattlesnake refers this scheme as *LogESpline*. It relies on users to provide the fine spectrum and to choose the prolongation scheme for obtaining accurate solutions.

4.3.2 Cusping Treatment

Rattlesnake provides *RoddedNeutronicsMaterial* for simulating control rod movement by defining different cross section regions in a subdomain along the direction of control rod motion. The most challenging part of this capability is to handle the cusping effect. Cusping arises from control rods that are partially inserted into a mesh element. In the following it is shown that the effective cross section in the element should be computed as polynomial projection. If simple volume weighting of the cross sections is selected, the problem eigenvalue plotted vs. the element's control rod fraction exhibits cusps giving the phenomenon its name. Within Rattlesnake's FEM framework, the terms in the weak form that contain a cross section are of the form:

$$(b_l(\vec{x}), \Sigma(\vec{x})f(\vec{x})) = \sum_{k=1}^K f_k \int_s b_l(\vec{x}) \Sigma(\vec{x}) b_k(\vec{x}) dx, \quad l = 1, \dots, K \quad (375)$$

where b_l is the l -th FEM shape function, and f is a quantity (scalar flux, angular flux, or angular flux moment) that is expanded into the set of shape functions by its expansion coefficients f_k , and s is the volume of an element containing a partially inserted control rod. K is the number of local degrees of freedom. The integral in Eq. (375) can be obtained exactly on segments of an element with different cross section regions of the control rod, but splitting element and performing integrals accordingly is computationally inefficient.

The material composition over each subsets of the element denoted by s_j is uniform, but state variables that the cross section depends on, e.g. temperature and density, can change over the extent of s_j , i.e. $\Sigma_j(T(\vec{x}), \dots)$. However, we *average* the state variables over each s_j :

$$T_j = \frac{1}{s_j} \int_{s_j} T(\vec{x}) dx, \quad (376)$$

and evaluate the cross section at the averaged state variable values:

$$\Sigma_j = \Sigma_j(T_j, \dots). \quad (377)$$

Now we can write Eq. (375) as:

$$\sum_{k=1}^K f_k \int_s b_l(\vec{x}) \Sigma(\vec{x}) b_k(\vec{x}) dx = \sum_{k=1}^K f_k \sum_{j=1}^J \Sigma_j \int_{s_j} b_l(\vec{x}) b_k(\vec{x}) dx. \quad (378)$$

In common reactor geometries, the control rods are inserted in a direction that aligns with either of the three axes x, y, z and hence the control rod position can be described by the position of its tip only. In addition, we assume that the element containing the partially inserted control rod is *extruded* in the direction of the control rod axis. For convenience, we refer to the direction of control rod motion as the x_3 coordinate, while the x_1 and x_2 coordinates span a plane orthogonal to the control rod motion. Given the aforementioned assumptions it follows that the shape functions can be written as products of two functions: a function $g(x_1, x_2)$ varying only in the plane perpendicular to the control rod motion and another function $h(x_3)$ varying only along x_3 :

$$b_l(\vec{x}) = g_l(x_1, x_2) h_l(x_3) \quad (379)$$

Using Eq. (378), Eq. (378) becomes:

$$\sum_{k=1}^K f_k \sum_{j=1}^J \Sigma_j \int_{s_j} b_l(\vec{x}) b_k(\vec{x}) dx = \sum_{k=1}^K f_k \left[\int_a g_l g_k dx_1 dx_2 \right] \left[\sum_{j=1}^J \Sigma_j \int_{x_{3,j-1}}^{x_{3,j}} h_l h_k dx_3 \right], \quad (380)$$

where a is the extend of element s in the x_1 - x_2 direction. Formally, we can define a weighted cross section $\Sigma_{l,k}$

$$\Sigma_{l,k} = \frac{\sum_{j=1}^J \Sigma_j \int_{x_{3,j-1}}^{x_{3,j}} h_l h_k dx_3}{\int_{x_{3,min}}^{x_{3,max}} h_l h_k dx_3}. \quad (381)$$

This allows us to rewrite Eq. (375) as:

$$\sum_{k=1}^K f_k \int_s b_l(\vec{x}) \Sigma_{l,k} b_k(\vec{x}) dx, \quad (382)$$

which treats the phenomenon of cusping without any approximations. However, l, k dependence of cross sections comes with the price of more complicated materials and kernels in Rattlesnake and lower efficiency on evaluating the residual of the kernels in Rattlesnake. Therefore, a different approach must be selected. Rattlesnake explicitly supports cross sections varying within an element. Let us define a cross sections that depends on x_3 , $\bar{\Sigma}(x_3)$. Substituting this cross section in Eq. (375) and repeating the steps performed previously leads to:

$$\sum_{k=1}^K f_k \left[\int_a g_l g_k dx_1 dx_2 \right] \left[\int_{x_{3,min}}^{x_{3,max}} \bar{\Sigma}(x_3) h_l h_k dx_3 \right], \quad l, k = 1, \dots, K. \quad (383)$$

Requiring Eq. (380) and Eq. (383) to be equal leads to the following condition on $\bar{\Sigma}(x_3)$:

$$\int_{x_{3,min}}^{x_{3,max}} \bar{\Sigma}(x_3) h_l h_k dx_3 = \sum_{j=1}^J \Sigma_j \int_{x_{3,j-1}}^{x_{3,j}} h_l h_k dx_3. \quad (384)$$

The cross section $\bar{\Sigma}(x_3)$ is now expanded into a series of Legendre polynomials P_i :

$$\bar{\Sigma}(x_3) = \sum_{i=0}^I \bar{\Sigma}_i P_i(x_3), \quad (385)$$

leading to

$$\sum_{i=0}^I \left[\int_{x_{3,min}}^{x_{3,max}} P_i h_l h_k dx_3 \right] \bar{\Sigma}_i = \sum_{j=1}^I \Sigma_j \int_{x_{3,j-1}}^{x_{3,j}} h_l h_k dx_3. \quad (386)$$

The expansion order of $\bar{\Sigma}_i$ must be chosen to satisfy Eq. (384) for all unique combinations of $h_l \cdot h_k$. The required order is given by $I = 2p$ if p is the FEM shape function order. As h_l and h_k are polynomials, the product $H_m = h_l h_k$ is also a polynomial. Hence, Eq. (386) is simply the polynomial projection of Σ_j onto the set H_m . The polynomial H_m can be expressed by a linear combination of Legendre polynomials up to order I :

$$H_m = \sum_{q=0}^I \eta_{m,q} P_q. \quad (387)$$

Introducing Eq. (387) into Eq. (386) gives:

$$\sum_{i=0}^I \sum_{q=0}^I \eta_{m,q} \left[\int_{x_{3,min}}^{x_{3,max}} P_i P_q dx_3 \right] \bar{\Sigma}_i = \sum_{q=0}^I \eta_{m,q} \sum_{j=1}^I \Sigma_j \int_{x_{3,j-1}}^{x_{3,j}} P_q dx_3. \quad (388)$$

Performing the integral on the left hand side, bringing the expression on the left-hand side to the right hand side and enforcing the condition that the sum is zero if each term of the sum is zero leads to the following set of equations:

$$\bar{\Sigma}_i = \frac{2i+1}{x_{3,max} - x_{3,min}} \sum_{j=1}^I \Sigma_j \int_{x_{3,j-1}}^{x_{3,j}} P_i dx_3, \quad i = 1, \dots, I. \quad (389)$$

This implies that the expansion coefficients can be obtained explicitly by projecting Σ_j onto the set of Legendre Polynomials.

In diffusion calculations, the diffusion coefficient multiplies the stiffness matrix and its cusping treatment differs from the method described above. The derivation of cusping treatment for the diffusion coefficient starts from the weak form:

$$(\nabla \phi^*, D(\vec{x}) \nabla f(\vec{x})) = \sum_{p=1}^d \sum_{k=1}^K f_k \int_s b_l^{(p)}(\vec{x}) D(\vec{x}) b_k^{(p)}(\vec{x}) dx, \quad (390)$$

where d is the dimensionality of the problem and $b_l^{(p)}$ is the derivative of b_l with respect to x_p , $\partial b_l / \partial x_p$. Using steps totally equivalent to the treatment of the cross section terms, we arrive at expressions similar to Eq. (384) for the x_1 - x_2 direction and x_3 direction that are not identical. Hence, we define a diagonal tensor diffusion coefficient $\vec{\vec{D}}(x_3)$ that allows us to separately satisfy the resulting constraints. We denote the elements on the diagonal as D_p noting that $D_1 = D_2 \neq D_3$. Then we expand each component into a series of Legendre polynomials:

$$D_p(x_3) = \sum_{i=0}^I D_{p,i} P_i(x_3). \quad (391)$$

For the components of the diffusion coefficient in the x_1 and x_2 direction, this is identical to the Eq. (386):

$$\sum_{i=0}^I \left[\int_{x_{3,min}}^{x_{3,max}} P_i h_l h_k dx_3 \right] D_{p,i} = \sum_{j=1}^I D_j \int_{x_{3,j-1}}^{x_{3,j}} h_l h_k dx_3. \quad (392)$$

For the x_3 direction, we have:

$$\sum_{i=0}^I \left[\int_{x_{3,min}}^{x_{3,max}} P_i \frac{\partial h_l}{\partial x_3} \frac{\partial h_k}{\partial x_3} dx_3 \right] D_{p,i} = \sum_{j=1}^I D_j \int_{x_{3,j-1}}^{x_{3,j}} \frac{\partial h_l}{\partial x_3} \frac{\partial h_k}{\partial x_3} dx_3. \quad (393)$$

For the x_3 calculation it should be noted that $\frac{\partial h_l}{\partial x_3} \frac{\partial h_k}{\partial x_3}$ is a polynomial of at most order $2(p-1)$, $p > 1$. Hence it can be treated exactly like the previously described case except that the I is now given by $I = 2(p-1)$. Note that this cusping treatment requires the use of a diagonal tensor diffusion coefficient with $D_1 = D_2 \neq D_3$.

4.4 Transport functions

Transport functions are designed to simplify the specifications of energy, spatially, time and angularly dependent functions that are frequently encountered in transport problems. A number of advantages result from that design, among which are:

- Simplification of the input file, in particular to specify incoming source boundary conditions or volumetric source by taking full advantage of the action system and automatically add functions needed to describe the problem but not defined in detail by the user. More specifically, if the boundary or volumetric source is constant in each group and block, numbers can be directly specified in the input file instead of individually creating all the necessary functions.
- Automatic normalization independently of the (spatial) dimensionality of the problem. Indeed, if the problem is only dependent upon one or two spatial dimensions, it is desired to keep the moments unchanged compared to a 3-D calculation with the appropriate symmetries. This requires to integrate the source terms and practically multiply the boundary and volumetric source by 2 for a 2-D problem and by 2π for a 1-D problem.
- The angular moments of S_N functions (i.e. described along specific directions) or – in return – the value at specific directions of P_N functions (i.e. described by their angular moments) can be automatically computed. More specifically, if the values f_n of a function are given along N_d particular directions, the moments may be reconstructed using the angular weights w_n and the values of the spherical harmonics along those particular directions⁶:

$$f_{\ell,m} = \sum_{n=1}^{N_d} f_n w_n Y_{\ell,m}(\vec{\Omega}_n) \quad (394)$$

Likewise, a function described by its moments $f_{\ell,m}$ can be evaluated at any direction $\vec{\Omega}$ using its spherical harmonics expansion:

$$f(\vec{\Omega}) = \sum_{0 \leq |m| \leq \ell \leq N} f_{\ell,m} R_{\ell,m}(\vec{\Omega}) \quad (395)$$

(see Appendix B for more details on the spherical harmonics definitions and Appendix E on the direction-to-moment and moment-to-direction matrices)

- Creating a transport function using the solution of a previous calculation and compute the error norm of P_N functions. In particular, for transient calculations, the initial condition is reconstructed from the exodus file using Eq. (394) or Eq. (395) using a transport function.
- Possibility of evaluating functions in the vicinity of interfaces, rather than on the actual interface. In particular, if the function comes from a previous calculation and is discontinuous on the surface, it can be desired to use the value immediately on the outside of the interface, which can be done by evaluating the function at:

$$\vec{x}^+ = \vec{x} + \epsilon(\vec{x} - \vec{x}_e) \quad (396)$$

rather than at \vec{x} (on the interface), where ϵ is a small positive number and \vec{x}_e is the position of the centroid of the current element.

4.5 Method of manufactured solution

The Method of Manufactured Solutions (MMS) is a powerful tool for computer code verification (CCV). Typically, the system of PDEs solved by a computer code is not amenable to analytical solutions and therefore the only way to obtain an analytical solution is using the MMS. Assume that the system of PDEs is given by

$$\mathbb{L}\mathbf{u} = \mathbf{f}, \quad (397)$$

where \mathbb{L} is an integro-differential operator, \mathbf{u} is the function we are trying to solve for and \mathbf{f} is a forcing term. The MMS reverses the solution process by selecting a function \mathbf{u}^{MMS} and computing the forcing term such that \mathbf{u}^{MMS} solves the resulting "manufactured" PDEs:

$$\mathbf{f}^{h,\text{MMS}} = \mathbb{L}^h \mathbf{u}^{\text{MMS}}, \quad (398)$$

⁶This way is not unique as detailed in Appendix E.

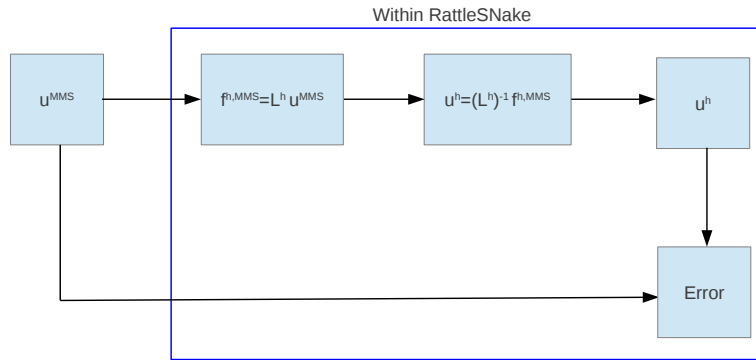


Figure 10 Schematic design of the implementation of the MMS in Rattlesnake.

where the h denotes that the respective quantity is evaluated on a grid with mesh size h (in fact h refers to both the spatial and angular refinement level).

The MMS can then be used to facilitate computer code verification via the order of convergence test. The code is used to solve the problem:

$$\mathbb{L}^h \mathbf{u}^h = \mathbf{f}^{h,MMS}, \quad (399)$$

Using a sequence of successively refined meshes, a set of solutions \mathbf{u}^{h_j} is obtained whose error e_j with respect to the true solution \mathbf{u}^{MMS} is measured in some norm $\|\cdot\|_p$:

$$e_j = \|\mathbf{u}^{h_j} - \mathbf{u}^{MMS}\|_p, \quad (400)$$

From the sequence of errors e_j the observed order of accuracy r_j can be computed as:

$$r_j = \frac{\ln \frac{e_j}{e_{j-1}}}{\ln \frac{h_j}{h_{j-1}}}. \quad (401)$$

For a successful verification the observed order is required to approach the theoretical order of accuracy (inherent to the discretization method), i.e.

$$r_j \rightarrow r_F \text{ for } j \rightarrow \infty. \quad (402)$$

The MMS benchmark capability is currently implemented for the S_N angular discretization and harnesses the flexible MOOSE framework to allow adding additional manufactured solutions easily. The user only defines a manufactured solution by deriving a new class from the appropriate *SNSolutionFunction* class and overriding the required member functions. *SNSolutionFunction* is a derived class of *TransportFunction*. The conceptual design of the MMS in Rattlesnake is depicted in Fig. 10.

5 Solvers

The multigroup transport partial differential equations have been presented in Section 2 and various discretization schemes with continuous or discontinuous FEM in space and S_N , P_N or diffusion in angle have been detailed in Section 3. This section focuses on how the discretized algebraical system is solved numerically. To facilitate the presentation, three types of problems, steady-state source, steady-state eigenvalue and transient, are respectively abstracted into a simpler notation:

$$\mathcal{F}(\mathbf{x}) = 0, \quad (403)$$

$$\frac{\partial \mathcal{T}(t, \mathbf{x})}{\partial t} = \mathcal{R}(t, \mathbf{x}), \quad \mathbf{x}(t = t_0) = \mathbf{x}_0 \quad (404)$$

$$\mathcal{A}(\mathbf{x}) = \frac{1}{\lambda} \mathcal{B}(\mathbf{x}), \quad (405)$$

where \mathbf{x} is the finite dimensional solution vector (its size being noted N in this section) which contains the coefficients corresponding to the shape functions of the angular fluxes for S_N , of the angular flux moments for P_N and of the scalar fluxes for diffusion. \mathcal{F} is the generic time independent nonlinear operator on \mathbf{x} , which could depend on the material properties and/or the solution vector itself directly or indirectly. $\mathcal{F}(\mathbf{x})$ can be represented by a matrix-vector production if \mathcal{F} is a linear operator. \mathcal{R} is used to denote the generic time dependent operator on \mathbf{x} with an explicit time variable t in its argument list. \mathcal{T} is for the generic time derivative term. We use λ for the eigenvalues and \mathcal{A} and \mathcal{B} for the two operators on the left and right hand sides, respectively.

By default, Rattlesnake uses the PJFNK (Preconditioned Jacobian-Free Newton Krylov) method to solve the steady-state source problem. The method is detailed in Section 5.1. Rattlesnake provides the Jacobian-free residual evaluation and the preconditioning matrix, sometimes referred to as Jacobian, to PETSc for the underlining solve. If the preconditioning matrix is the exact Jacobian, Jacobian-free residual evaluation is not needed during linear solves and the method is then the preconditioned Newton method. It is noted that this is rarely practical for transport problem due to the extremely large dimension N and the variable coupling through scattering and fission events. It can however be very useful for debugging purposes. Transport discretization schemes with continuous FEM and discontinuous FEM are fundamentally different in terms of the matrix structure. The former results into a diffusion-like matrix for the streaming plus collision operator, which is suitable for multigrid (MG) method. MG method can be completely mesh free if the preconditioning matrix is assembled and stored. Various other PETSc preconditioning techniques can be utilized out of the box. The latter may result⁷ into an almost low-block-triangular matrix for the streaming plus collision operator, in which case it can be inverted efficiently with a custom matrix-free sweeping algorithm. It is also noted that PETSc itself is not aware of mesh adaptation, multiphysics coupling, etc., so MOOSE provides extra outer iterations wrapping the Newton iteration for mesh refinement cycles, multiphysics couplings, etc. For example, NDA calculations are performed with the steady-state source or eigenvalue solvers with a wrapping Picard iteration.

The method of lines is exclusively used for solving transient problems due to its simplicity. MOOSE provides the time stepping and time integration for transient calculations. Time stepping and integration schemes and their documentations can be found in MOOSE. We briefly discuss them in Section 5.2. At each time step, the PJFNK or Newton method can be used along with outer iterations as in the steady-state source problem.

Rattlesnake interacts with SLEPc [90] for eigenvalue calculations. The most common approach is based on PJFNK, where the original eigenvalue problem is cast into a nonlinear problem with the eigenvalue evaluated from the solution. This approach requires a good initial guess to guarantee convergence to the fundamental mode, which is typically our interest. This good initial guess can be obtained using a few inverse power iterations. For certain problems whose dominant ratio is not close to one, inverse power iterations converge fast and can be a good option. Both methods are detailed in Section 5.3.

Source iteration is a very popular concept in radiation transport. It uses the solution of the current iteration to construct the source and then inverts the streaming plus collision operator to update the solution. If the external source is excluded from the source, the solution at a given iteration is the partial solution corresponding to the particles having undergone a number of collisions equal to the iteration index. If the problem is not very diffusive, source iteration thus converges fast. The convergence is guaranteed if there is no multiplication in the system.

⁷Unfortunately, this is not true for P_N methods.

Source iteration can be viewed as a Richardson iteration. This solver is discussed in Section 5.4. In Rattlesnake, it is mainly used as transport updates for nonlinear diffusion accelerations.

5.1 PJFNK

Newton iterations start with an initial guess of the solution $\mathbf{x}^{(0)}$. At each Newton iteration we solve the following linear problem:

$$\mathbf{J}(\mathbf{x}^{(n-1)})\delta\mathbf{x}^{(n)} = \mathcal{F}(\mathbf{x}^{(n-1)}), \quad (406)$$

where

$$\mathbf{J}(\mathbf{x}^{(n-1)}) \equiv \mathcal{F}_{\mathbf{x}}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^{(n-1)}} \quad (407)$$

is the *Jacobian* matrix evaluated at $\mathbf{x}^{(n-1)}$. It depends on $\mathbf{x}^{(n-1)}$ for general nonlinear problems but is constant for linear problems. The right hand side of Eq. (406) is called the *residual* at $\mathbf{x}^{(n-1)}$. The Krylov methods are employed for solving the above linear equation, which requires only the evaluation of the matrix-vector product $\mathbf{J}(\mathbf{x}^{(n-1)})\mathbf{y}$. The Jacobian-free method is used to approximate matrix vector products by a finite-difference like approximation:

$$\mathbf{J}(\mathbf{x}^{(n-1)})\mathbf{y} \approx \frac{\mathcal{F}(\mathbf{x}^{(n-1)} + \epsilon\mathbf{y}) - \mathcal{F}(\mathbf{x}^{(n-1)})}{\epsilon}, \quad (408)$$

where the scalar value ϵ is chosen by PETSc automatically to approximate $\mathbf{J}(\mathbf{x}^{(n-1)})\mathbf{y}$ accurately for the linear solve. It is noted that for a linear problem for which $\mathcal{F}(\mathbf{x})$ can be expressed as $\mathbf{Ax} - \mathbf{b}$, where the matrix \mathbf{A} is the Jacobian (independent of \mathbf{x}) and \mathbf{b} is the right-hand-side vector, the right hand side of Eq. (408) is independent of ϵ . Section 5.5 of the PETSc user's manual on matrix-free methods details the algorithms for choosing the value of ϵ .

Krylov methods typically also require an approximation of the actual Jacobian $\mathbf{P}(\mathbf{x}^{(n-1)}) \approx \mathbf{J}(\mathbf{x}^{(n-1)})$ for preconditioning the Krylov solution at each linear iteration to improve the convergence. It is noted that the preconditioning matrix is seldom the exact Jacobian \mathbf{J} because it could require too much computational time and memory and, in some cases, is simply impossible to compute. By default the type of Krylov methods in use is the Generalized Minimum Residual (GMRES) method because it does not have assumptions on the properties of the underlying Jacobian. If the Jacobian is symmetric positive definite (SPD), the conjugate gradient (CG) method should be used. The initial guess for each linear solve is always set to zero, which implies that the initial linear residual is the same as the nonlinear residual. The residual norm at each linear iteration is evaluated by PETSc, for instance, while updating the Hessenberg matrix if the GMRES method is used. At the conclusion of the linear solve, the solution is updated as follows

$$\mathbf{x}^n = \mathbf{x}^{n-1} + \alpha\delta\mathbf{x}^{(n)} \quad (409)$$

where α is determined by the line-search algorithm. At each nonlinear or Newton iteration, we need to update the preconditioning matrix and evaluate the residual with the updated solution. At each linear iteration, we simply need a residual evaluation and the operation of the preconditioner built from the preconditioning matrix. It is noted that the default preconditioner type⁸ depends on the number of processors and also depends on the assembled preconditioning matrix \mathbf{P} . Typically incomplete LU (PCILU) is the default type with one processor and block Jacobi (PCBJACOBI) is the default with multiple processors. Consequently, the same convergence may differ depending on the number of processors. Various preconditioning techniques are available through PETSc, which can be found in the PETSc user manual.

The convergence can be determined by either one of the following:

$$\|\mathcal{F}(\mathbf{x}^n)\| < \epsilon_{\text{abs}}, \quad (410)$$

$$\|\mathcal{F}(\mathbf{x}^n)\| < \epsilon_{\text{rel}} \|\mathcal{F}(\mathbf{x}^0)\|, \quad (411)$$

$$\|\delta\mathbf{x}^{(n)}\| < \epsilon_{\text{step}}. \quad (412)$$

where ϵ denotes a given tolerance.

⁸In PETSc the preconditioner type refers to the method to obtain an approximation of the inverse of \mathbf{P} .

5.2 Transient

This executioner uses the method of lines to treat the time independent variable. Using the implicit Euler scheme as an example, we are solving, at every time step,

$$\frac{\mathcal{T}(t, \mathbf{x}) - \mathcal{T}(t^{\text{old}}, \mathbf{x}^{\text{old}})}{\Delta t} = \mathcal{R}(t, \mathbf{x}), \quad (413)$$

where $\Delta t = t - t^{\text{old}}$, t and t^{old} being the current and previous time step, respectively. If we define $\mathcal{F}(\mathbf{x})$ as

$$\mathcal{F}(\mathbf{x}) \equiv \mathcal{R}(t, \mathbf{x}) + \frac{1}{\Delta t} \mathcal{T}(t, \mathbf{x}) - \frac{1}{\Delta t} \mathcal{T}(t^{\text{old}}, \mathbf{x}^{\text{old}}), \quad (414)$$

where t , t^{old} and \mathbf{x}^{old} are given, then, at every time step, we are solving a finite-dimensional steady-state nonlinear or linear problem as given in Eq. (413). A PJFNK solver is used for this steady-state solve. Time stepping is controlled by MOOSE time steppers. It can be as simple as a constant time step size or complicated as using time adaptation. How the steady-state problem is formed at every time step is controlled by MOOSE time integrators. The available integrators are implicit Euler, explicit Euler, Crank-Nicolson, BDF2, Runge-Kutta second order, diagonal implicit Runge-Kutta, explicit TVD Runge-Kutta second order, etc.

5.3 Eigenvalue

The operator form of the steady-state eigenvalue problem of the neutron transport equation is given by Eq. (405). The solution \mathbf{x} has the following properties: (1) the trivial solution $\mathbf{x} = 0$ is usually discarded, and (2) if \mathbf{x} is a solution, for any real number a , $a\mathbf{x}$ is also a solution. Typically we are only interested in the absolute minimum eigenvalue and the corresponding eigenvector of the system, referred to as the fundamental mode in the remainder of this document. The inverse power iteration algorithm is given in Algorithm 5. We notice immediately that $\frac{|\mathcal{B}(\mathbf{x})|}{\lambda}$ remains constant during the iteration, so if $\frac{|\mathcal{B}(\mathbf{x}^{(0)})|}{\lambda^{(0)}}$ is set to one, the algorithm can be simplified significantly into Algorithm 6. The choice of norm $|\mathcal{B}(\mathbf{x})|$ is arbitrary and does not affect the convergence of the inverse power method.

Also, in this simplified algorithm, the solution is automatically normalized enforcing $|\mathbf{B}(\mathbf{x})| = k$. Post-processing is used to normalize the solution so that $|\mathbf{x}| = c$, where $|\cdot|$ can be any norm and c is a scalar constant. If the dominance ratio, defined as the ratio between the smallest and the second smallest eigenvalue, is close to one, inverse power iterations converge very slowly. We can apply accelerations, such as Chebyshev acceleration [91], based on the on-the-fly estimation of the dominance ratio. PJFNK can still be applied at each iteration to update the solution.

From the simplified inverse power method, if we let

$$\lambda = |\mathcal{B}(\mathbf{x})|, \quad (417)$$

the eigenvalue problem can be viewed as the following nonlinear problem

$$\mathbf{A}(\mathbf{x}) = \frac{1}{|\mathcal{B}(\mathbf{x})|} \mathbf{B}(\mathbf{x}). \quad (418)$$

This new viewpoint opens the door of using PJFNK method to solve the eigenvalue problem. However, as the eigenvalue problems has an infinite number of solutions different from the fundamental mode, we have to ensure that the nonlinear solution does not converge to any of the higher harmonics but to the fundamental mode. To this end we need to ensure that the initial guess supplied to the PJFNK solver is sufficiently close to the fundamental mode. This can be achieved by performing several free inverse power iterations before starting the Newton iteration. It is noted that λ does need to be part of the solution vector. This PJFNK eigensolver was proposed in Ref. [69].

Occasionally, it is desired to determine a parameter p , such as the soluble boron concentration, control rod bank position, etc., (which is affecting the operators $\mathcal{A}(\mathbf{x})$ and/or $\mathcal{B}(\mathbf{x})$), to force the eigenvalue λ in Eq. (405) to assume a specific value, usually $\lambda = 1$. In such a case, the system is referred to as critical and one is algorithm the targeted eigenvalue. For this reason this kind of calculation is often called criticality search. We can think of the eigenvalue

Algorithm 5 Inverse power iteration

1: Initialization:

$$\begin{aligned}\lambda^{(0)} &= \lambda_0 \\ \mathbf{x}^{(0)} &= \mathbf{x}_0.\end{aligned}$$

2: Update \mathbf{x} by performing a steady-state solve[‡] and update λ :

$$\begin{aligned}\mathcal{A}(\mathbf{x}^{(n)}) &= \frac{1}{\lambda^{(n-1)}} \mathcal{B}(\mathbf{x}^{(n-1)}) \\ \lambda^{(n)} &= \lambda^{(n-1)} \frac{|\mathcal{B}(\mathbf{x}^{(n)})|}{|\mathcal{B}(\mathbf{x}^{(n-1)})|}\end{aligned}\tag{415}$$

3: Check the convergence

$$\frac{|\lambda^{(n)} - \lambda^{(n-1)}|}{|\lambda^{(n)}|} < \epsilon_\lambda$$

and

$$\frac{|\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}|}{|\mathbf{x}^{(n)}|} < \epsilon_x.\tag{416}$$

When either of them is not true, return Step 2, otherwise exit.

[‡] Note that we typically do not need to have a full solve. The currently code will only perform one Newton iteration.

Algorithm 6 Modified inverse power iteration

1: Initialization:

$$\begin{aligned}\lambda^{(0)} &= \lambda_0 \\ \mathbf{x}^{(0)} &= \lambda_0 \frac{\mathbf{x}_0}{|\mathcal{B}(\mathbf{x}_0)|}.\end{aligned}$$

2: Update \mathbf{x} by performing a steady-state solve and update k :

$$\begin{aligned}\mathcal{A}(\mathbf{x}^{(n)}) &= \frac{1}{\lambda^{(n-1)}} \mathcal{B}(\mathbf{x}^{(n-1)}) \\ \lambda^{(n)} &= |\mathcal{B}(\mathbf{x}^{(n)})|\end{aligned}$$

3: Check the convergence

$$\frac{|\lambda^{(n)} - \lambda^{(n-1)}|}{|\lambda^{(n)}|} < \epsilon_\lambda$$

and

$$\frac{|\mathbf{x}^{(n)} - \mathbf{x}^{(n-1)}|}{|\mathbf{x}^{(n)}|} < \epsilon_x.$$

When either of them is not true, return Step 2, otherwise exit.

$\lambda(p)$ as a function of p . If the parameter p is close to the target parameter, a Newton search is performed for $\lambda(p) - \lambda_{target} = 0$ with

$$p^{(n+1)} = p^{(n)} - \left(\frac{d\lambda}{dp} \Big|_{p=p^{(n)}} \right)^{-1} (\lambda(p^{(n)}) - \lambda_{target}), \quad (419)$$

where $\frac{d\lambda}{dp} \Big|_{p=p^{(n)}}$ is approximated by

$$\frac{d\lambda}{dp} \Big|_{p=p^{(n)}} = \frac{\lambda(p^{(n)}(1 + \epsilon)) - \lambda(p^{(n)})}{\epsilon p^{(n)}}, \quad (420)$$

where ϵ is a small scalar value used to perturb the parameter for evaluating the λ -derivative with respect to p . For every Newton step n , we thus perform two eigenvalue calculations: one with the current parameter $p^{(n)}$ and another with the perturbed parameter $p^{(n)}(1 + \epsilon)$ with PJFNK. It is noted that the perturbation with ϵ should be large enough so that no numerical issue arise while evaluating $k(p^{(n)}(1 + \epsilon)) - k(p^{(n)})$ and the search is stable. The smaller $\frac{dk}{dp}$ is, the larger ϵ should be. On the other hand, if a large ϵ is used, we end up with the secant method, which converges more slowly than Newton's method. Newton's method requires a fairly close initial guess of p if the dependence of k on p is complicated. In such a case, we can perform few bisection evaluations to obtain an initial guess.

5.4 Richardson iteration

In some cases, it is advantageous to split the general nonlinear problem Eq. (403) as

$$\mathcal{L}(\mathbf{x}) = \mathcal{R}(\mathbf{x}). \quad (421)$$

and to perform the following iteration scheme

$$\mathcal{L}(\mathbf{x}^{(n)}) = \mathcal{R}(\mathbf{x}^{(n-1)}). \quad (422)$$

The scattering source iteration, well-known in radiation transport, is a special case of Richardson iteration. Although the above equation can be solved with PJFNK, this form can lead to a faster converging scheme.. For example, when \mathcal{L} only has the streaming plus collision operator in S_N transport, the problem can be solved with AMG for CFEM and with sweeper for DFEM directly. The Diffusion Synthetic Acceleration (DSA) is a well-known technique to improve convergence of the source iteration scheme for diffusive problems and can be recast as an efficient preconditioner for S_N schemes [92, 93].

Appendices

A Vector Notations

We use vectors throughout this manual to simplify notation for the multigroup transport or diffusion equations. These vectors are different from the vector in the three-dimensional space or in the two-dimensional angular domain, where \vec{v} is used. We use \mathbf{v} to denote a quantity with multiple components, such as the angular fluxes for all groups Ψ . These vectors could be combined with the traditional vector notations, for example, $\vec{\mathbf{J}}$ as the current of all groups and the traditional tensor notations, such as $\vec{\vec{\mathbf{D}}}$ for the diffusion coefficient tensor of all energy groups or $\vec{\vec{\mathbf{E}}}$ for the Eddington tensor of all groups. We also adopt an implicit multiplication operator of any two vectors of the same size as $\mathbf{v}_1 \mathbf{v}_2$, which represents a resultant vector whose components are simply the multiplication of the corresponding components of the two vectors.

We give a special notation for PDE operators, for example \mathbb{L} for the streaming plus collision operator. Operation of a PDE operator on a vector results into a vector of the operator on each elements. We also bold font to denote matrices (e.g. \mathbf{M}), especially for P_N methods. Matrix-vector product is interpreted in the usual way. Each element of a matrix can be a traditional vector or tensor, as $\vec{\mathbf{D}}$ and $\vec{\vec{\mathbf{H}}}$ in Section B.2. Although generally a PDE operator can be written in a matrix form, we use a separate notation for them for clarity. To differentiate the PDE operator and the operator form of the weak form, we use \mathbb{b} to represent the bilinear or linear forms appearing in the weak forms. Bilinear or linear operators return a scalar value which is defined as the summation of bilinear or linear operation of all components.

We summarize all vectors and PDE operator notation appearing in this manual in Table 46.

Table 46 Vector and operator notations.

Name	Definition	Page number
Angular flux	$\Psi \equiv \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \vdots \\ \Psi_G \end{bmatrix}$	12
Adjoint angular flux	$\Psi^* = \begin{bmatrix} \Psi_1^* \\ \Psi_2^* \\ \vdots \\ \Psi_G^* \end{bmatrix}$	14
Angular flux divided by speed	$\frac{\Psi}{\mathbf{v}} \equiv \begin{bmatrix} \frac{\Psi_1}{v_1} \\ \frac{\Psi_2}{v_2} \\ \vdots \\ \frac{\Psi_G}{v_G} \end{bmatrix}$	12
Boundary surface source	$\Psi^{\text{inc}} \equiv \begin{bmatrix} \Psi_1^{\text{inc}} \\ \Psi_2^{\text{inc}} \\ \vdots \\ \Psi_G^{\text{inc}} \end{bmatrix}$	13
Boundary adjoint surface source	$\Psi_b^{*\text{out}} = \begin{bmatrix} \Psi_1^{*\text{out}} \\ \Psi_2^{*\text{out}} \\ \vdots \\ \Psi_G^{*\text{out}} \end{bmatrix}$	14
Initial angular flux	$\Psi^0 \equiv \begin{bmatrix} \Psi_1^0 \\ \Psi_2^0 \\ \vdots \\ \Psi_G^0 \end{bmatrix}$	13

Final adjoint angular flux	$\Psi_1^* = \begin{bmatrix} \Psi_1^{*1} \\ \Psi_2^{*1} \\ \vdots \\ \Psi_G^{*1} \end{bmatrix}$	14
External angular source	$\mathbf{Q}^{\text{ext}} \equiv \begin{bmatrix} Q_1^{\text{ext}} \\ Q_2^{\text{ext}} \\ \vdots \\ Q_G^{\text{ext}} \end{bmatrix}$	12
Zero-th moment of external source	$\mathbf{Q}_0^{\text{ext}} \equiv \begin{bmatrix} \int_S Q_1^{\text{ext}} d\Omega \\ \int_S Q_2^{\text{ext}} d\Omega \\ \vdots \\ \int_S Q_G^{\text{ext}} d\Omega \end{bmatrix}$	15
Adjoint external angular source	$\mathbf{Q}^{*\text{ext}} = \begin{bmatrix} Q_1^{*\text{ext}} \\ Q_2^{*\text{ext}} \\ \vdots \\ Q_G^{*\text{ext}} \end{bmatrix}$	14
Scalar flux	$\Phi \equiv \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_G \end{bmatrix}$	15
Adjoint scalar flux	$\Phi^* \equiv \begin{bmatrix} \Phi_1^* \\ \Phi_2^* \\ \vdots \\ \Phi_G^* \end{bmatrix}$	19
Scalar flux divided by speed	$\frac{\Phi}{\mathbf{v}} \equiv \begin{bmatrix} \frac{\Phi_1}{v_1} \\ \frac{\Phi_2}{v_2} \\ \vdots \\ \frac{\Phi_G}{v_G} \end{bmatrix}$	15
Current	$\vec{\mathbf{J}} \equiv \begin{bmatrix} \vec{J}_1 \\ \vec{J}_2 \\ \vdots \\ \vec{J}_G \end{bmatrix}$	15
Boundary partial incoming current	$\mathbf{J}^{\text{i}} \equiv \begin{bmatrix} \int_{ \vec{\Omega} \cdot \vec{n}_b < 0} \vec{\Omega} \cdot \vec{n}_b \Psi_1 d\Omega \\ \int_{ \vec{\Omega} \cdot \vec{n}_b < 0} \vec{\Omega} \cdot \vec{n}_b \Psi_2 d\Omega \\ \vdots \\ \int_{ \vec{\Omega} \cdot \vec{n}_b < 0} \vec{\Omega} \cdot \vec{n}_b \Psi_G d\Omega \end{bmatrix}$	16
Boundary partial out-going current	$\mathbf{J}^{\text{o}} \equiv \begin{bmatrix} \int_{ \vec{\Omega} \cdot \vec{n}_b > 0} \vec{\Omega} \cdot \vec{n}_b \Psi_1 d\Omega \\ \int_{ \vec{\Omega} \cdot \vec{n}_b > 0} \vec{\Omega} \cdot \vec{n}_b \Psi_2 d\Omega \\ \vdots \\ \int_{ \vec{\Omega} \cdot \vec{n}_b > 0} \vec{\Omega} \cdot \vec{n}_b \Psi_G d\Omega \end{bmatrix}$	16

Boundary incoming source	$\mathbf{J}^{\text{inc}} \equiv \begin{bmatrix} \int_{ \vec{\Omega} \cdot \vec{n}_b < 0} \vec{\Omega} \cdot \vec{n}_b \Psi_1^{\text{inc}} d\Omega \\ \int_{ \vec{\Omega} \cdot \vec{n}_b < 0} \vec{\Omega} \cdot \vec{n}_b \Psi_2^{\text{inc}} d\Omega \\ \vdots \\ \int_{ \vec{\Omega} \cdot \vec{n}_b < 0} \vec{\Omega} \cdot \vec{n}_b \Psi_G^{\text{inc}} d\Omega \end{bmatrix}$	16
Boundary reflectivity	$\boldsymbol{\alpha} \equiv \begin{bmatrix} \alpha_1^s + \alpha_1^d \\ \alpha_2^s + \alpha_2^d \\ \vdots \\ \alpha_G^s + \alpha_G^d \end{bmatrix}$	16
Extrapolation factor	$\mathbf{e} \equiv \begin{bmatrix} \frac{1}{e_1} \\ \frac{1}{e_2} \\ \vdots \\ \frac{1}{e_G} \end{bmatrix}$	19
Eddington tensor	$\vec{\mathbf{E}} \equiv \begin{bmatrix} \vec{E}_1 \\ \vec{E}_2 \\ \vdots \\ \vec{E}_G \end{bmatrix}$	16
Delayed neutron precursor	$\mathbf{C} \equiv \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_I \end{bmatrix}$	24
DNP decay constant	$\boldsymbol{\lambda} \equiv \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_I \end{bmatrix}$	24
SAAF stabilization	$\boldsymbol{\tau} \equiv \begin{bmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_G \end{bmatrix}$	38
LS weighting	$\mathbf{W} \equiv \begin{bmatrix} W_1(\vec{x}, t) \\ W_2(\vec{x}, t) \\ \vdots \\ W_G(\vec{x}, t) \end{bmatrix}$	41
Scattering	$\mathbb{S}\boldsymbol{\Psi} \equiv \begin{bmatrix} \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow 1} f_{g' \rightarrow 1} \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow 2} f_{g' \rightarrow 2} \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \vdots \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow G} f_{g' \rightarrow G} \Psi_{g'}(\vec{\Omega}') d\Omega' \end{bmatrix}$	13
Minus scattering	$\bar{\mathbb{S}}\boldsymbol{\Psi} \equiv \begin{bmatrix} \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow 1} f_{g' \rightarrow 1} (-\vec{\Omega} \cdot \vec{\Omega}) \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow 2} f_{g' \rightarrow 2} (-\vec{\Omega} \cdot \vec{\Omega}) \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \vdots \\ \sum_{g'=1}^G \int_{\mathcal{S}} \Sigma_{s,g' \rightarrow G} f_{g' \rightarrow G} (-\vec{\Omega} \cdot \vec{\Omega}) \Psi_{g'}(\vec{\Omega}') d\Omega' \end{bmatrix}$	40

Adjoint scattering operator	$\mathbb{S}^* \Psi \equiv \begin{bmatrix} \sum_{g'=1}^G \int_S \Sigma_{s,1 \rightarrow g'} f_{1 \rightarrow g'} \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \sum_{g'=1}^G \int_S \Sigma_{s,2 \rightarrow g'} f_{2 \rightarrow g'} \Psi_{g'}(\vec{\Omega}') d\Omega' \\ \vdots \\ \sum_{g'=1}^G \int_S \Sigma_{s,G \rightarrow g'} f_{G \rightarrow g'} \Psi_{g'}(\vec{\Omega}') d\Omega' \end{bmatrix}$	15
Isotropic scattering	$\mathbb{S}_0 \Phi \equiv \begin{bmatrix} \sum_{g'=1}^G \Sigma_{s,g' \rightarrow 1} \Phi_{g'} \\ \sum_{g'=1}^G \Sigma_{s,g' \rightarrow 2} \Phi_{g'} \\ \vdots \\ \sum_{g'=1}^G \Sigma_{s,g' \rightarrow G} \Phi_{g'} \end{bmatrix}$	15
Linearly anisotropic scattering	$\mathbb{S}_1 \vec{J} \equiv \begin{bmatrix} \sum_{g'=1}^G \Sigma_{s,g' \rightarrow 1,1} \vec{J}_{g'} \\ \sum_{g'=1}^G \Sigma_{s,g' \rightarrow 2,1} \vec{J}_{g'} \\ \vdots \\ \sum_{g'=1}^G \Sigma_{s,g' \rightarrow G,1} \vec{J}_{g'} \end{bmatrix}$	16
Out-scattering	$\mathbb{S}_d \Phi \equiv \begin{bmatrix} \sum_{g' \neq 1} \Sigma_{s,g' \rightarrow 1} \Phi_{g'} \\ \sum_{g' \neq 2} \Sigma_{s,g' \rightarrow 2} \Phi_{g'} \\ \vdots \\ \sum_{g' \neq G} \Sigma_{s,g' \rightarrow G} \Phi_{g'} \end{bmatrix}$	16
Adjoint out-scattering	$\mathbb{S}_d^* \Phi^* \equiv \begin{bmatrix} \sum_{g' \neq 1} \Sigma_{s,1 \rightarrow g'} \Phi_{g'}^* \\ \sum_{g' \neq 2} \Sigma_{s,2 \rightarrow g'} \Phi_{g'}^* \\ \vdots \\ \sum_{g' \neq G} \Sigma_{s,G \rightarrow g'} \Phi_{g'}^* \end{bmatrix}$	19
Boundary reflecting	$\mathbb{B} \Psi \equiv \begin{bmatrix} \alpha_1^s \Psi_1(\vec{\Omega}_r) + \alpha_1^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} \vec{\Omega}' \cdot \vec{n}_b \Psi_1(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} \vec{\Omega}' \cdot \vec{n}_b d\Omega'} \\ \alpha_2^s \Psi_2(\vec{\Omega}_r) + \alpha_2^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} \vec{\Omega}' \cdot \vec{n}_b \Psi_2(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} \vec{\Omega}' \cdot \vec{n}_b d\Omega'} \\ \vdots \\ \alpha_G^s \Psi_G(\vec{\Omega}_r) + \alpha_G^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} \vec{\Omega}' \cdot \vec{n}_b \Psi_G(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b > 0} \vec{\Omega}' \cdot \vec{n}_b d\Omega'} \end{bmatrix}$	13
Adjoint boundary reflecting	$\mathbb{B}^* \Psi \equiv \begin{bmatrix} \alpha_1^s \Psi_1(\vec{\Omega}_r) + \alpha_1^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} \vec{\Omega}' \cdot \vec{n}_b \Psi_1(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} \vec{\Omega}' \cdot \vec{n}_b d\Omega'} \\ \alpha_2^s \Psi_2(\vec{\Omega}_r) + \alpha_2^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} \vec{\Omega}' \cdot \vec{n}_b \Psi_2(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} \vec{\Omega}' \cdot \vec{n}_b d\Omega'} \\ \vdots \\ \alpha_G^s \Psi_G(\vec{\Omega}_r) + \alpha_G^d \frac{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} \vec{\Omega}' \cdot \vec{n}_b \Psi_G(\vec{\Omega}') d\Omega'}{\int_{\vec{\Omega}' \cdot \vec{n}_b < 0} \vec{\Omega}' \cdot \vec{n}_b d\Omega'} \end{bmatrix}$	15
Streaming+collision operator	$\mathbb{L} \Psi \equiv \begin{bmatrix} \left(\vec{\nabla} \cdot \vec{\Omega} + \Sigma_{t,1} \right) \Psi_1 \\ \left(\vec{\nabla} \cdot \vec{\Omega} + \Sigma_{t,2} \right) \Psi_2 \\ \vdots \\ \left(\vec{\nabla} \cdot \vec{\Omega} + \Sigma_{t,G} \right) \Psi_G \end{bmatrix}$	13
Adjoint streaming+collision operator	$\mathbb{L}^* \Psi \equiv \begin{bmatrix} \left(-\vec{\Omega} \cdot \vec{\nabla} + \Sigma_{t,1} \right) \Psi_1 \\ \left(-\vec{\Omega} \cdot \vec{\nabla} + \Sigma_{t,2} \right) \Psi_2 \\ \vdots \\ \left(-\vec{\Omega} \cdot \vec{\nabla} + \Sigma_{t,G} \right) \Psi_G \end{bmatrix}$	14

Streaming	$\mathbb{L}_1 \Psi \equiv$	$\begin{bmatrix} \vec{\Omega} \cdot \vec{\nabla} \Psi_1 \\ \vec{\Omega} \cdot \vec{\nabla} \Psi_2 \\ \vdots \\ \vec{\Omega} \cdot \vec{\nabla} \Psi_G \end{bmatrix}$	38
Curvilinear streaming	$\mathbb{E} \Psi \equiv$	$\begin{bmatrix} E \Psi_1 \\ E \Psi_2 \\ \vdots \\ E \Psi_G \end{bmatrix}$	33
Second curvilinear streaming	$\mathbb{E}_2 \Psi \equiv$	$\begin{bmatrix} E_2 \Psi_1 \\ E_2 \Psi_2 \\ \vdots \\ E_2 \Psi_G \end{bmatrix}$	33
Collision	$\mathbb{L}_2 \Phi \equiv$	$\begin{bmatrix} \Sigma_{t,1} \Phi_1 \\ \Sigma_{t,2} \Phi_2 \\ \vdots \\ \Sigma_{t,G} \Phi_G \end{bmatrix}$	15
Inverse collision	$\mathbb{L}_2^{-1} \Psi \equiv$	$\begin{bmatrix} \Psi_1 \\ \Sigma_{t,1}^{-1} \Psi_1 \\ \Psi_2 \\ \Sigma_{t,2}^{-1} \Psi_2 \\ \vdots \\ \Psi_G \\ \Sigma_{t,G}^{-1} \Psi_G \end{bmatrix}$	38
Removal	$\mathbb{L}_3 \Phi \equiv$	$\begin{bmatrix} \Sigma_{r,1} \Phi_1 \\ \Sigma_{r,2} \Phi_2 \\ \vdots \\ \Sigma_{r,G} \Phi_G \end{bmatrix}$	16
Transport	$\mathbb{L}_{tr} \vec{J} \equiv$	$\begin{bmatrix} \Sigma_{tr,1} \vec{J}_1 \\ \Sigma_{tr,2} \vec{J}_2 \\ \vdots \\ \Sigma_{tr,G} \vec{J}_G \end{bmatrix}$	17
Diffusion	$\mathbb{D} \vec{J} \equiv$	$\begin{bmatrix} D_1 \vec{J}_1 \\ D_2 \vec{J}_2 \\ \vdots \\ D_G \vec{J}_G \end{bmatrix}$	19
Prompt fission	$\mathbb{F}_{p,0} \Phi \equiv$	$\begin{bmatrix} (1-\beta) \chi_{p,1} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ (1-\beta) \chi_{p,2} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ (1-\beta) \chi_{p,G} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}$	24
Adjoint prompt fission	$\mathbb{F}_{p,0}^* \Phi^* \equiv$	$\begin{bmatrix} (1-\beta) \nu \Sigma_{f,1} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ (1-\beta) \nu \Sigma_{f,2} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ \vdots \\ (1-\beta) \nu \Sigma_{f,G} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \end{bmatrix}$	26
Angular prompt fission	$\mathbb{F}_p \Psi \equiv \frac{1}{4\pi} \mathbb{F}_{p,0} \Phi = \frac{1}{4\pi}$	$\begin{bmatrix} (1-\beta) \chi_{p,1} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ (1-\beta) \chi_{p,2} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ (1-\beta) \chi_{p,G} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}$	24

Adjoint angular prompt fission	$\mathbb{F}_p^* \Psi^* \equiv \frac{1}{4\pi} \mathbb{F}_{p,0}^* \Phi^* =$	$\begin{bmatrix} \frac{1}{4\pi} (1 - \beta) \nu \Sigma_{f,1} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ \frac{1}{4\pi} (1 - \beta) \nu \Sigma_{f,2} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ \vdots \\ \frac{1}{4\pi} (1 - \beta) \nu \Sigma_{f,G} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \end{bmatrix}$	26
Delayed fission	$\mathbb{F}_{d,0} \Phi \equiv$	$\begin{bmatrix} \sum_{i=1}^I \beta_i \chi_{d,i,1} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \sum_{i=1}^I \beta_i \chi_{d,i,2} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \sum_{i=1}^I \beta_i \chi_{d,i,G} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}$	24
Adjoint delayed fission	$\mathbb{F}_{d,0}^* \Phi^* \equiv$	$\begin{bmatrix} \nu \Sigma_{f,1} \sum_{g'=1}^G \sum_{i=1}^I \beta_i \chi_{d,i,g'} \Phi_{g'}^* \\ \nu \Sigma_{f,2} \sum_{g'=1}^G \sum_{i=1}^I \beta_i \chi_{d,i,g'} \Phi_{g'}^* \\ \vdots \\ \nu \Sigma_{f,G} \sum_{g'=1}^G \sum_{i=1}^I \beta_i \chi_{d,i,g'} \Phi_{g'}^* \end{bmatrix}$	26
Angular delayed fission	$\mathbb{F}_d \Psi \equiv \frac{1}{4\pi} \mathbb{F}_{d,0} \Phi =$	$\begin{bmatrix} \frac{1}{4\pi} \sum_{i=1}^I \beta_i \chi_{d,i,1} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \frac{1}{4\pi} \sum_{i=1}^I \beta_i \chi_{d,i,2} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \frac{1}{4\pi} \sum_{i=1}^I \beta_i \chi_{d,i,G} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}$	24
Adjoint angular delayed fission	$\mathbb{F}_d^* \Psi^* \equiv \frac{1}{4\pi} \mathbb{F}_{d,0}^* \Phi^* =$	$\begin{bmatrix} \frac{1}{4\pi} \nu \Sigma_{f,1} \sum_{g'=1}^G \sum_{i=1}^I \beta_i \chi_{d,i,g'} \Phi_{g'}^* \\ \frac{1}{4\pi} \nu \Sigma_{f,2} \sum_{g'=1}^G \sum_{i=1}^I \beta_i \chi_{d,i,g'} \Phi_{g'}^* \\ \vdots \\ \frac{1}{4\pi} \nu \Sigma_{f,G} \sum_{g'=1}^G \sum_{i=1}^I \beta_i \chi_{d,i,g'} \Phi_{g'}^* \end{bmatrix}$	26
Partial delayed fission	$\mathbb{F}_{d,i,0} \Phi \equiv$	$\begin{bmatrix} \beta_i \chi_{d,i,1} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \beta_i \chi_{d,i,2} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \beta_i \chi_{d,i,G} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}$	25
Partial angular delayed fission	$\mathbb{F}_{d,i} \Phi \equiv \frac{1}{4\pi}$	$\begin{bmatrix} \beta_i \chi_{d,i,1} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \beta_i \chi_{d,i,2} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \beta_i \chi_{d,i,G} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}$	25
Fission	$\mathbb{F}_0 \Phi =$	$\begin{bmatrix} \chi_1 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \chi_2 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \chi_G \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}$	24
Adjoint fission	$\mathbb{F}_0^* \Phi^* \equiv$	$\begin{bmatrix} \nu \Sigma_{f,1} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ \nu \Sigma_{f,2} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ \vdots \\ \nu \Sigma_{f,G} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \end{bmatrix}$	26
Angular fission	$\mathbb{F} \Psi \equiv \frac{1}{4\pi} \mathbb{F}_0 \Phi =$	$\begin{bmatrix} \frac{1}{4\pi} \chi_1 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \frac{1}{4\pi} \chi_2 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \frac{1}{4\pi} \chi_G \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}$	24

Adjoint angular fission	$\mathbb{F}^* \Psi^* \equiv \frac{1}{4\pi} \mathbb{F}_0^* \Phi^* =$	$\begin{bmatrix} \frac{1}{4\pi} \nu \Sigma_{f,1} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ \frac{1}{4\pi} \nu \Sigma_{f,2} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \\ \vdots \\ \frac{1}{4\pi} \nu \Sigma_{f,G} \sum_{g'=1}^G \chi_{p,g'} \Phi_{g'}^* \end{bmatrix}$	26
Delayed source	$\mathbb{C}_0 \mathbf{C} =$	$\begin{bmatrix} \sum_{i=1}^I \chi_{d,i,1} \lambda_i C_i \\ \sum_{i=1}^I \chi_{d,i,2} \lambda_i C_i \\ \vdots \\ \sum_{i=1}^I \chi_{d,i,G} \lambda_i C_i \end{bmatrix}$	24
Adjoint delayed source	$\mathbb{P}_0^* \mathbf{C}^* \equiv$	$\begin{bmatrix} \nu \Sigma_{f,1} \sum_{i=1}^I \beta_i C_i^* \\ \nu \Sigma_{f,2} \sum_{i=1}^I \beta_i C_i^* \\ \vdots \\ \nu \Sigma_{f,G} \sum_{i=1}^I \beta_i C_i^* \end{bmatrix}$	26
Angular delayed source	$\mathbb{C} \mathbf{C} \equiv \frac{1}{4\pi} \mathbb{C}_0 \mathbf{C} =$	$\begin{bmatrix} \frac{1}{4\pi} \sum_{i=1}^I \chi_{d,i,1} \lambda_i C_i \\ \frac{1}{4\pi} \sum_{i=1}^I \chi_{d,i,2} \lambda_i C_i \\ \vdots \\ \frac{1}{4\pi} \sum_{i=1}^I \chi_{d,i,G} \lambda_i C_i \end{bmatrix}$	24
Adjoint angular delayed source	$\mathbb{P}^* \mathbf{C}^* \equiv \frac{1}{4\pi} \mathbb{P}_0^* \mathbf{C}^* =$	$\begin{bmatrix} \frac{1}{4\pi} \nu \Sigma_{f,1} \sum_{i=1}^I \beta_i C_i^* \\ \frac{1}{4\pi} \nu \Sigma_{f,2} \sum_{i=1}^I \beta_i C_i^* \\ \vdots \\ \frac{1}{4\pi} \nu \Sigma_{f,G} \sum_{i=1}^I \beta_i C_i^* \end{bmatrix}$	26
Partial delayed source	$\mathbb{C}_{i,0} C_i =$	$\begin{bmatrix} \chi_{d,i,1} \lambda_i C_i \\ \chi_{d,i,2} \lambda_i C_i \\ \vdots \\ \chi_{d,i,G} \lambda_i C_i \end{bmatrix}$	25
Partial angular delayed source	$\mathbb{C}_i C_i = \frac{1}{4\pi}$	$\begin{bmatrix} \chi_{d,i,1} \lambda_i C_i \\ \chi_{d,i,2} \lambda_i C_i \\ \vdots \\ \chi_{d,i,G} \lambda_i C_i \end{bmatrix}$	25
DNP production	$\mathbb{P} \Phi \equiv$	$\begin{bmatrix} \beta_1 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \beta_2 \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \\ \vdots \\ \beta_I \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'} \end{bmatrix}$	24
Partial DNP production	$\mathbb{P}_i \Phi \equiv \beta_i \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'}$		25
Adjoint DNP production	$\mathbb{C}_0^* \Phi^* \equiv$	$\begin{bmatrix} \lambda_1 \sum_{g'=1}^G \chi_{d,1,g'} \Phi_{g'}^* \\ \lambda_2 \sum_{g'=1}^G \chi_{d,2,g'} \Phi_{g'}^* \\ \vdots \\ \lambda_I \sum_{g'=1}^G \chi_{d,I,g'} \Phi_{g'}^* \end{bmatrix}$	26
Black body emission	$\mathbf{S}_{\text{BB}} \equiv$	$\begin{bmatrix} S_{\text{BB},1} \\ S_{\text{BB},2} \\ \vdots \\ S_{\text{BB},G} \end{bmatrix}$	31

B Spherical Harmonics

B.1 Basics

The flying direction $\vec{\Omega}$ can be fully characterized with μ , the cosine of the polar angle, and ω , the azimuthal angle. For any integers ℓ, m such that $|m| \leq \ell$, the real spherical harmonics of degree ℓ and order m is defined as:

$$Y_{\ell,m}(\mu, \omega) = \begin{cases} \sqrt{2} C_{\ell}^{|m|} P_{\ell}^{|m|}(\mu) \cos(|m|\omega), & 0 < m \leq \ell \\ C_{\ell}^0 P_{\ell}^0(\mu), & m = 0 \\ \sqrt{2} C_{\ell}^{|m|} P_{\ell}^{|m|}(\mu) \sin(|m|\omega), & -\ell \leq m < 0 \end{cases}, 0 \leq \ell < \infty, \quad (423)$$

where $C_{\ell}^{|m|} = \sqrt{\frac{(\ell-|m|)!}{(\ell+|m|)!}}$ is a normalization constant and P_{ℓ}^m designates the associated Legendre polynomial of degree ℓ and order m and is defined as (for $0 \leq m \leq \ell$):

$$P_{\ell}^m(\mu) = (-1)^m (1 - \mu^2)^{m/2} \frac{d^m}{d\mu^m} P_{\ell}(\mu) = \frac{(-1)^m}{2^{\ell} \ell!} (1 - \mu^2)^{m/2} \frac{d^{\ell+m}}{d\mu^{\ell+m}} ((\mu^2 - 1)^{\ell}) \quad (424)$$

where P_{ℓ} is the Legendre polynomial of degree ℓ . The $(-1)^m$ term in blue is the Condon-Shortley phase and may or may not be included in the definition of P_{ℓ}^m . We add $|\cdot|$ around m even when m is positive to avoid ambiguity which will be clear once we give the classic definition of spherical harmonics:

$$Y_{\ell}^m(\mu, \omega) = \sqrt{\frac{2\ell+1}{4\pi} \frac{(\ell-m)!}{(\ell+m)!}} P_{\ell}^m(\mu) e^{im\omega}, m = -\ell, \dots, \ell; \ell = 0, \dots, \infty, \quad (425)$$

with the associated Legendre polynomial with negative m being related with the one with positive m as

$$P_{\ell}^{-m}(\mu) = (-1)^m \frac{(\ell-m)!}{(\ell+m)!} P_{\ell}^m(\mu), 0 \leq m \leq \ell, \quad (426)$$

or

$$Y_{\ell}^m(\mu, \omega) = \sqrt{\frac{2\ell+1}{4\pi} \frac{(\ell-|m|)!}{(\ell+|m|)!}} P_{\ell}^{|m|}(\mu) (-1)^{\frac{|m|-m}{2}} e^{im\omega}, m = -\ell, \dots, \ell; \ell = 0, \dots, \infty. \quad (427)$$

It can be observed that

$$\bar{Y}_{\ell}^m(\mu, \omega) = Y_{\ell}^m(\mu, -\omega) = (-1)^m Y_{\ell}^{-m}(\mu, \omega). \quad (428)$$

It is worth giving the conversion equations between the real spherical harmonics and the classic spherical harmonics:

$$Y_{\ell,m} = \begin{cases} \frac{1}{\sqrt{2}} \sqrt{\frac{4\pi}{2\ell+1}} (Y_{\ell}^m + (-1)^m Y_{\ell}^{-m}), & 0 < m \leq \ell \\ \sqrt{\frac{4\pi}{2\ell+1}} Y_{\ell}^0, & m = 0 \\ -i \frac{1}{\sqrt{2}} \sqrt{\frac{4\pi}{2\ell+1}} (Y_{\ell}^{-m} - (-1)^m Y_{\ell}^m), & -\ell \leq m < 0 \end{cases} \quad (429)$$

and

$$Y_{\ell}^m = \begin{cases} \frac{1}{\sqrt{2}} \sqrt{\frac{2\ell+1}{4\pi}} (Y_{\ell,m} + i Y_{\ell,-m}), & 0 < m \leq \ell \\ \sqrt{\frac{2\ell+1}{4\pi}} Y_{\ell,0}, & m = 0 \\ (-1)^m \frac{1}{\sqrt{2}} \sqrt{\frac{2\ell+1}{4\pi}} (Y_{\ell,-m} - i Y_{\ell,m}), & -\ell \leq m < 0 \end{cases} \quad (430)$$

Rattlesnake uses the real spherical harmonics without the Condon-Shortley phase by default.

Spherical harmonics are related with the solution to Laplace's equation in spherical coordinates. The Laplacian in spherical coordinates on a generic function $f(r, \mu, \omega)$ is

$$\Delta f = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial f}{\partial r} \right) + \frac{1}{r^2} \left[\frac{\partial}{\partial \mu} \left((1 - \mu^2) \frac{\partial f}{\partial \mu} \right) + \frac{1}{1 - \mu^2} \frac{\partial^2 f}{\partial \omega^2} \right], \quad (431)$$

where μ is the cosine of the polar angle φ ; ω is the azimuthal angle. The Laplacian is also often denoted as $\vec{\nabla}^2$ because it is equal to $\vec{\nabla} \cdot \vec{\nabla}$, where $\vec{\nabla}$ is the gradient operator

$$\vec{\nabla} f = \frac{\partial f}{\partial r} \vec{e}_r + \frac{1}{r} \left[-\sqrt{1 - \mu^2} \frac{\partial f}{\partial \mu} \vec{e}_\varphi + \frac{1}{\sqrt{1 - \mu^2}} \frac{\partial f}{\partial \omega} \vec{e}_\omega \right], \quad (432)$$

and $\vec{\nabla} \cdot$ is the divergence operator

$$\vec{\nabla} \cdot \vec{f} = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 f_r \right) + \frac{1}{r} \left[-\frac{\partial}{\partial \mu} \left(\sqrt{1 - \mu^2} f_\varphi \right) + \frac{1}{\sqrt{1 - \mu^2}} \frac{\partial f_\omega}{\partial \omega} \right]. \quad (433)$$

The operator in square bracket of the Laplacian is also known as the spherical Laplacian, i.e. the Laplacian on the unit sphere:

$$\Delta_S f = \frac{\partial}{\partial \mu} \left((1 - \mu^2) \frac{\partial f}{\partial \mu} \right) + \frac{1}{1 - \mu^2} \frac{\partial^2 f}{\partial \omega^2}. \quad (434)$$

Spherical harmonics are eigenfunctions of the spherical Laplacian:

$$\Delta_S Y_{\ell, m} = -\ell(\ell + 1) Y_{\ell, m}. \quad (435)$$

Similarly we can have the spherical gradient and spherical divergence:

$$\vec{\nabla}_S f = -\sqrt{1 - \mu^2} \frac{\partial f}{\partial \mu} \vec{e}_\varphi + \frac{1}{\sqrt{1 - \mu^2}} \frac{\partial f}{\partial \omega} \vec{e}_\omega \quad (436)$$

$$\vec{\nabla}_S \cdot \vec{f} = -\frac{\partial}{\partial \mu} \left(\sqrt{1 - \mu^2} f_\varphi \right) + \frac{1}{\sqrt{1 - \mu^2}} \frac{\partial f_\omega}{\partial \omega}. \quad (437)$$

Because on the unit sphere, unit vectors are

$$\vec{e}_r = \sqrt{1 - \mu^2} \cos \omega \vec{e}_x + \sqrt{1 - \mu^2} \sin \omega \vec{e}_y + \mu \vec{e}_z, \quad (438)$$

$$\vec{e}_\varphi = \mu \cos \omega \vec{e}_x + \mu \sin \omega \vec{e}_y - \sqrt{1 - \mu^2} \vec{e}_z, \quad (439)$$

$$\vec{e}_\omega = -\sin \omega \vec{e}_x + \cos \omega \vec{e}_y, \quad (440)$$

$$\begin{aligned} \vec{\nabla}_S f = & \left[-\mu \sqrt{1 - \mu^2} \cos \omega \frac{\partial f}{\partial \mu} - \frac{1}{\sqrt{1 - \mu^2}} \sin \omega \frac{\partial f}{\partial \omega} \right] \vec{e}_x + \left[-\mu \sqrt{1 - \mu^2} \sin \omega \frac{\partial f}{\partial \mu} + \frac{1}{\sqrt{1 - \mu^2}} \cos \omega \frac{\partial f}{\partial \omega} \right] \vec{e}_y \\ & + \left[(1 - \mu^2) \frac{\partial f}{\partial \mu} \right] \vec{e}_z. \end{aligned} \quad (441)$$

It is noted that $\vec{e}_r = \vec{\Omega}$. The angular momentum operator L in quantum mechanics is defined as $-i\vec{\Omega} \times \vec{\nabla}_S$. It is noted that $L^2 = \Delta_S$ and $L_z = -i \frac{\partial}{\partial \omega}$. Two related operators are $L_\pm = L_x \pm iL_y$, also known as ladder operators. Angular momentum operators do not change the spherical harmonics degree. An important property for analyzing the streaming term in radiation transport is

$$\vec{\Omega} Y_{\ell, m} = \frac{1}{2\ell + 1} \left(\vec{\nabla}_S + \ell \vec{\Omega} \right) Y_{\ell, m} - \frac{1}{2\ell + 1} \left(\vec{\nabla}_S - (\ell + 1) \vec{\Omega} \right) Y_{\ell, m}, \quad (442)$$

where the first and second parts on the right hand side contain only spherical harmonics of degree $(\ell - 1)$ and $(\ell + 1)$, respectively. This formula can be proved with solid spherical harmonics [94].

The spherical harmonics addition theorem is

$$2\pi P_\ell(\vec{\Omega} \cdot \vec{\Omega}') = P_\ell(\mu_0) = \sum_{m=-\ell}^{\ell} Y_{\ell,m}(\vec{\Omega}) Y_{\ell,m}(\vec{\Omega}'). \quad (443)$$

The moments of the angular flux Ψ_g , and more generally of any function, are defined as follows:

$$\Phi_{g,\ell,m}(\vec{x}) \equiv \int_S Y_{\ell,m}(\vec{\Omega}) \Psi_g(\vec{x}, \vec{\Omega}) d\Omega, \quad (444)$$

allowing to expand Ψ_g using its moments:

$$\Psi_g(\vec{x}, \vec{\Omega}) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \Phi_{g,\ell,m}(\vec{x}) R_{\ell,m}(\vec{\Omega}). \quad (445)$$

It is noted that because $Y_{0,0} = 1$ and $\{Y_{1,1}, Y_{1,-1}, Y_{1,0}\}^T = \vec{\Omega}$, the zeroth order and the first order moments are scalar flux and currents. The P_N approximation consists of truncating this expansion at degree N for any angular dependent quantity. In particular, the angular flux is approximated as:

$$\Psi_g(\vec{x}, \vec{\Omega}) \approx \mathbf{R}^T(\vec{\Omega}) \Phi_g(\vec{x}) = \sum_{\ell=0}^N \sum_{m=-\ell}^{\ell} \Phi_{g,\ell,m}(\vec{x}) R_{\ell,m}(\vec{\Omega}), \quad (446)$$

where the paired spherical harmonics are defined as:

$$R_{\ell,m} \equiv \frac{2\ell+1}{w} Y_{\ell,m}, \quad (447)$$

with w being the total angular weight ($w = 2$ in 1D, $w = 2\pi$ in 2D, $w = 4\pi$ in 3D; a similar definition is used for the unit sphere S , so much that $\int_S d\Omega = w$). This implies:

$$\int_S Y_{\ell,m}(\vec{\Omega}) R_{\ell',m'}(\vec{\Omega}) d\Omega = \delta_{\ell\ell'} \delta_{mm'}. \quad (448)$$

Furthermore we have defined:

$$\mathbf{R} \equiv \{R_{\ell,m}, m = -\ell, \dots, \ell; \ell = 0, \dots, N\} \quad (449)$$

We use E_ℓ to denote the subspace of all spherical harmonics of degree ℓ , whose dimension is 1, $\ell+1$ and $2\ell+1$ for 1D, 2D and 3D respectively. We use \mathcal{E}_ℓ as the subspace of all spherical harmonics of degree less than or equal to ℓ

$$\mathcal{E}_\ell = \oplus_{\ell'=0}^{\ell} E_{\ell'}, \quad (450)$$

whose dimension is $\ell+1$, $\frac{(\ell+1)(\ell+2)}{2}$ and $(\ell+1)^2$ for 1D, 2D and 3D respectively. The order of the subspace \mathcal{E}_ℓ is then said to be ℓ . For a given set of spherical harmonics denoted as a vector \mathbf{Y} , we call the minimum order of subspace not contained by \mathbf{Y} minus one as the order of the subspace denoted by $L(\mathbf{Y})$. For example, $\{Y_{0,0}, Y_{1,0}, Y_{1,1}, Y_{1,-1}, Y_{2,0}\}$ has order 1.

B.2 Matrices

Most P_N kernels rely on the evaluation of angular matrices. In practice, they are computed using quadrature rules which yield exact integration (to machine precision). We define them in this subsection:

$$\vec{\mathbf{D}} \equiv \int_S \vec{\Omega} \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}) d\Omega, \quad (451)$$

$$\mathbf{P} \equiv \int_S \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}) d\Omega = \text{diag} \left\{ \frac{2\ell+1}{w}, m = -\ell, \dots, \ell; \ell = 0, \dots, N \right\}, \quad (452)$$

$$\vec{\mathbf{H}} \equiv \int_S (\vec{\Omega} \vec{\Omega}^T) \otimes \mathbf{R} \mathbf{R}^T d\Omega, \quad (453)$$

where \otimes is the tensor product. In particular, for all $u, v \in \{1, \dots, d\}$, we have:

$$\mathbf{H}_{u,v} \equiv \int_S \Omega_u \Omega_v \mathbf{R} \mathbf{R}^T d\Omega. \quad (454)$$

For the boundary conditions, we further define

$$\mathbf{L}^\pm(\vec{n}) \equiv \int_{\vec{\Omega} \cdot \vec{n} \gtrless 0} |\vec{\Omega} \cdot \vec{n}| \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}) d\Omega, \quad (455)$$

and:

$$\mathbf{C} \equiv \text{diag}\left\{(-1)^\ell, m = -\ell, \dots, \ell; \ell = 0, \dots, N\right\}. \quad (456)$$

which implies:

$$\mathbf{L}^- = \mathbf{C} \mathbf{L}^+ \mathbf{C}. \quad (457)$$

In addition,

$$\mathbf{Q}^\pm \equiv \int_{\pm \vec{\Omega} \cdot \vec{n}_b > 0} \mathbf{R} \mathbf{R}^T d\Omega. \quad (458)$$

Besides, we define

$$\mathbf{N}(\vec{n}) \equiv \int_{\vec{\Omega} \cdot \vec{n}_b < 0} |\vec{\Omega} \cdot \vec{n}_b| \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}_r) d\Omega, \quad (459)$$

$$\mathbf{L}_r \equiv \int_S \vec{\Omega} \cdot \vec{n}_b \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}_r) d\Omega, \quad \mathbf{Q}_r \equiv \int_S \mathbf{R}(\vec{\Omega}) \mathbf{R}^T(\vec{\Omega}_r) d\Omega \quad (460)$$

with $\vec{\Omega}_r = \vec{\Omega} - 2(\vec{\Omega} \cdot \vec{n}_b) \vec{n}_b$. In practice, the matrices \mathbf{L}^+ and \mathbf{N} are evaluated for a particular direction $\vec{n}_b = \vec{e}_x$ (we then refer to them as \mathbf{L}_0 and \mathbf{N}_0 , respectively) and we use rotation matrices to compute them for the desired normal vector. The rotation matrix has the following definition:

$$\mathcal{R}(\vec{n}) \equiv \int_S \mathbf{Y}_n \mathbf{R}^T d\Omega \quad (461)$$

where \mathbf{Y}_n is the spherical harmonics evaluated in the rotated coordinate (with the polar angle aligned with \vec{n}) with a given direction $\vec{\Omega}$. These matrices can be analytically evaluated [95, 96] or numerically evaluated with Gauss-Chebyshev angular quadrature in Section E.3 in Rattlesnake. It is noted that \mathbf{L}^+ is symmetric positive definite and that rotating a matrix does not change its eigenvalues. Also, for the specific direction $\vec{n}_b = \vec{e}_z$, $(\mathbf{L}_r)_{\ell, m, \ell', m'} = (-1)^{(\ell' + m')} \bar{D}_{\ell, m, \ell', m'} \cdot \vec{e}_z$ and $(\mathbf{Q}_r)_{\ell, m, \ell', m'} = (-1)^{(\ell' + m')} P_{\ell, m, \ell', m'}$.

We detail the 2-D matrices in the P_1 case for the interested reader.

The \mathbf{P} matrix and $\vec{\mathbf{D}}$ matrices for each spatial dimensions are given by:

$$\mathbf{P} = \begin{pmatrix} \frac{1}{2\pi} & 0 & 0 \\ 0 & \frac{3}{2\pi} & 0 \\ 0 & 0 & \frac{3}{2\pi} \end{pmatrix}, \quad \mathbf{D}_x = \begin{pmatrix} 0 & \frac{1}{2\pi} & 0 \\ \frac{1}{2\pi} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{D}_y = \begin{pmatrix} 0 & 0 & \frac{1}{2\pi} \\ 0 & 0 & 0 \\ \frac{1}{2\pi} & 0 & 0 \end{pmatrix} \quad (462)$$

Also, for the particular direction $\vec{n}_b = \vec{e}_x$, we have:

$$\mathbf{Q}_r = \begin{pmatrix} \frac{1}{2\pi} & 0 & 0 \\ 0 & -\frac{3}{2\pi} & 0 \\ 0 & 0 & \frac{3}{2\pi} \end{pmatrix}, \quad \mathbf{L}_r = \begin{pmatrix} 0 & -\frac{1}{2\pi} & 0 \\ \frac{1}{2\pi} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (463)$$

$$\mathbf{Q}^{\pm} = \begin{pmatrix} \frac{1}{4\pi} & \pm \frac{3}{8\pi} & 0 \\ \pm \frac{3}{8\pi} & \frac{1}{4\pi} & 0 \\ 0 & 0 & \frac{3}{4\pi} \end{pmatrix}, \quad \mathbf{L}^{\pm} = \begin{pmatrix} \frac{1}{8\pi} & \pm \frac{1}{4\pi} & 0 \\ \pm \frac{1}{4\pi} & \frac{1}{16\pi} & 0 \\ 0 & 0 & \frac{9}{32\pi} \end{pmatrix} \quad (464)$$

B.3 Boundary types

The previous half-range angular matrices – namely \mathbf{L}^+ , \mathbf{L}^- and \mathbf{N} – are used to evaluate boundary terms. Although we have shown the most common definition for these matrices, other options also exist, which are detailed here.

These originate from different ways to treat vacuum boundary terms [50]:

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi^*, \Psi^{\text{inc}} \rangle_{\partial \mathcal{D}_s}^- = \begin{cases} \langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi^*, \Psi^{\text{inc}} \rangle_{\partial \mathcal{D}_s}^- & , \text{ SAAF} \\ \left\langle \Psi^*, \Psi + \Psi(-\vec{\Omega}) - \Psi^{\text{inc}}(-\vec{\Omega}) \right\rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi^*, \Psi^{\text{inc}} \rangle_{\partial \mathcal{D}_s}^- & , \text{ EVEN} \\ \left\langle \Psi^*, \Psi - \Psi(-\vec{\Omega}) + \Psi^{\text{inc}}(-\vec{\Omega}) \right\rangle_{\partial \mathcal{D}_s}^+ - \langle \Psi^*, \Psi^{\text{inc}} \rangle_{\partial \mathcal{D}_s}^- & , \text{ ODD} \\ \langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_s}^+ + \langle \Psi^*, \Psi - 2\Psi^{\text{inc}} \rangle_{\partial \mathcal{D}_s}^- & , \text{ PE} \end{cases} \quad (465)$$

After performing the P_N expansion, we obtain:

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_s}^+ = \begin{cases} (\Phi^*, \mathcal{R}^T \mathbf{L}_0 \mathcal{R} \Phi - \mathcal{R}^T (\mathbf{C} \mathbf{L}_0 \mathbf{C}) \mathcal{R} \Phi^{\text{inc}})_{\partial \mathcal{D}_s} & , \text{ SAAF} \\ (\Phi^*, \mathcal{R}^T (\mathbf{L}_0 + \mathbf{L}_0 \mathbf{C}) \mathcal{R} \Phi - \mathcal{R}^T (\mathbf{C} \mathbf{L}_0 \mathbf{C} + \mathbf{L}_0 \mathbf{C}) \mathcal{R} \Phi^{\text{inc}})_{\partial \mathcal{D}_s} & , \text{ EVEN} \\ (\Phi^*, \mathcal{R}^T (\mathbf{L}_0 - \mathbf{L}_0 \mathbf{C}) \mathcal{R} \Phi - \mathcal{R}^T (\mathbf{C} \mathbf{L}_0 \mathbf{C} - \mathbf{L}_0 \mathbf{C}) \mathcal{R} \Phi^{\text{inc}})_{\partial \mathcal{D}_s} & , \text{ ODD} \\ (\Phi^*, \mathcal{R}^T (\mathbf{L}_0 + \mathbf{C} \mathbf{L}_0 \mathbf{C}) \mathcal{R} \Phi - 2\mathcal{R}^T (\mathbf{C} \mathbf{L}_0 \mathbf{C}) \mathcal{R} \Phi^{\text{inc}})_{\partial \mathcal{D}_s} & , \text{ PE} \end{cases} \quad (466)$$

where Φ^{inc} is defined using an expansion of $\Psi^{\text{inc}} \equiv \mathbf{R}^T \Phi^{\text{inc}}$ (although the expansion is not unique as the values of Ψ^{inc} for $\vec{\Omega} \cdot \vec{n}_b > 0$ are arbitrary, the previous expressions are independent of that choice, as all terms are integrated over $\vec{\Omega} \cdot \vec{n}_b < 0$).

Therefore, the more general forms of the \mathbf{L}^+ and \mathbf{L}^- matrices are given in Table 47.

Table 47 Matrices for vacuum boundary conditions available in Rattlesnake.

Type	\mathbf{L}^+	\mathbf{L}^-
SAAF	$\mathcal{R}^T \mathbf{L}_0 \mathcal{R}$	$\mathcal{R}^T (\mathbf{C} \mathbf{L}_0 \mathbf{C}) \mathcal{R}$
EVEN	$\mathcal{R}^T (\mathbf{L}_0 + \mathbf{L}_0 \mathbf{C}) \mathcal{R}$	$\mathcal{R}^T (\mathbf{C} \mathbf{L}_0 \mathbf{C} + \mathbf{L}_0 \mathbf{C}) \mathcal{R}$
ODD	$\mathcal{R}^T (\mathbf{L}_0 - \mathbf{L}_0 \mathbf{C}) \mathcal{R}$	$\mathcal{R}^T (\mathbf{C} \mathbf{L}_0 \mathbf{C} - \mathbf{L}_0 \mathbf{C}) \mathcal{R}$
PE	$\mathcal{R}^T (\mathbf{L}_0 + \mathbf{C} \mathbf{L}_0 \mathbf{C}) \mathcal{R}$	$2\mathcal{R}^T (\mathbf{C} \mathbf{L}_0 \mathbf{C}) \mathcal{R}$

Similarly, there are several ways to treat reflecting boundary terms [50]:

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \langle \Psi^*, \Psi_r \rangle_{\partial \mathcal{D}_r}^- = \begin{cases} \langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \langle \Psi^*, \Psi_r \rangle_{\partial \mathcal{D}_r}^- & , \text{ SAAF} \\ \langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \left\langle \Psi^*, \frac{\Psi + \Psi_r}{2} \right\rangle_{\partial \mathcal{D}_r}^- - \left\langle \Psi^*, \frac{-\Psi + \Psi_r}{2} \right\rangle_{\partial \mathcal{D}_r}^+ & , \text{ SYM} \\ \langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \left\langle \Psi^*, \frac{\Psi + \Psi_r}{2} \right\rangle_{\partial \mathcal{D}_r}^- - \left\langle \Psi^*, \frac{\Psi - \Psi_r}{2} \right\rangle_{\partial \mathcal{D}_r}^+ & , \text{ NSYM} \end{cases} \quad (467)$$

After performing the P_N expansion, we obtain:

$$\langle \Psi^*, \Psi \rangle_{\partial \mathcal{D}_r}^+ - \langle \Psi^*, \Psi_r \rangle_{\partial \mathcal{D}_r}^- = \begin{cases} (\Phi^*, \mathcal{R}^T (\mathbf{L}_0 - \mathbf{N}_0) \mathcal{R} \Phi)_{\partial \mathcal{D}_r} & , \text{ SAAF} \\ \left(\Phi^*, \mathcal{R}^T \left(\mathbf{L}_0 - \frac{\mathbf{C} \mathbf{L}_0 \mathbf{C} + \mathbf{N}_0}{2} + \frac{\mathbf{L}_0 - \mathbf{N}_0^T}{2} \right) \mathcal{R} \Phi \right)_{\partial \mathcal{D}_r} & , \text{ SYM} \\ \left(\Phi^*, \mathcal{R}^T \left(\mathbf{L}_0 - \frac{\mathbf{C} \mathbf{L}_0 \mathbf{C} + \mathbf{N}_0}{2} - \frac{\mathbf{L}_0 - \mathbf{N}_0^T}{2} \right) \mathcal{R} \Phi \right)_{\partial \mathcal{D}_r} & , \text{ NSYM} \end{cases} \quad (468)$$

Similar expressions are obtained for the LS scheme (see Section 3.3.3):

$$\langle \Psi^*, \Psi - \Psi_r \rangle_{\partial \mathcal{D}_r}^- = \begin{cases} (\Phi^*, \mathcal{R}^T(\mathbf{C}\mathbf{L}_0\mathbf{C} - \mathbf{N}_0)\mathcal{R}\Phi)_{\partial \mathcal{D}_r} & , \text{ SAAF} \\ \left(\Phi^*, \mathcal{R}^T\left(\frac{\mathbf{C}\mathbf{L}_0\mathbf{C} - \mathbf{N}_0}{2} + \frac{\mathbf{L}_0 - \mathbf{N}_0^T}{2}\right)\mathcal{R}\Phi \right)_{\partial \mathcal{D}_r} & , \text{ SYM} \\ \left(\Phi^*, \mathcal{R}^T\left(\frac{\mathbf{C}\mathbf{L}_0\mathbf{C} - \mathbf{N}_0}{2} - \frac{\mathbf{L}_0 - \mathbf{N}_0^T}{2}\right)\mathcal{R}\Phi \right)_{\partial \mathcal{D}_r} & , \text{ NSYM} \end{cases} \quad (469)$$

In the code, the matrices are defined \mathbf{N}^+ and \mathbf{N}^- with their definitions given in Table 48.

Table 48 Matrices for reflecting boundary conditions available in Rattlesnake.

Type	\mathbf{N}^+	\mathbf{N}^-
SAAF	$\mathcal{R}^T(\mathbf{L}_0 - \mathbf{N}_0)\mathcal{R}$	$\mathcal{R}^T(\mathbf{C}\mathbf{L}_0\mathbf{C} - \mathbf{N}_0)\mathcal{R}$
SYM	$\mathcal{R}^T(\mathbf{L}_0 - \frac{\mathbf{C}\mathbf{L}_0\mathbf{C} + \mathbf{N}_0}{2} + \frac{\mathbf{L}_0 - \mathbf{N}_0^T}{2})\mathcal{R}$	$\mathcal{R}^T(\frac{\mathbf{C}\mathbf{L}_0\mathbf{C} - \mathbf{N}_0}{2} + \frac{\mathbf{L}_0 - \mathbf{N}_0^T}{2})\mathcal{R}$
NSYM	$\mathcal{R}^T(\mathbf{L}_0 - \frac{\mathbf{C}\mathbf{L}_0\mathbf{C} + \mathbf{N}_0}{2} - \frac{\mathbf{L}_0 - \mathbf{N}_0^T}{2})\mathcal{R}$	$\mathcal{R}^T(\frac{\mathbf{C}\mathbf{L}_0\mathbf{C} - \mathbf{N}_0}{2} - \frac{\mathbf{L}_0 - \mathbf{N}_0^T}{2})\mathcal{R}$

C Streaming in Curvilinear Coordinates

Because the streaming operator has different forms in curvilinear coordinates than in Cartesian coordinates, we present the streaming operator in curvilinear coordinates here. The detailed derivation of the streaming forms can be found in Ref. [3]. We simply copy three tables in Ref. [3] for Cartesian coordinate, spherical coordinate and cylindrical coordinate with the modified notations consistent with this manual. Streaming in Cartesian coordinate is presented in Table 49 for completeness. Conservation forms in curvilinear coordinates are used in Table 50 and Table 51. Volume differential elements dx are presented in the last column for reference. We often find using $\vec{\Omega} \cdot \vec{\nabla} \Psi$ as the streaming operator is confusing in curvilinear coordinates, so we use a new notation $\partial_{\vec{\Omega}} \Psi$. In Cartesian coordinate, $\partial_{\vec{\Omega}} \Psi = \vec{\Omega} \cdot \vec{\nabla} \Psi = \vec{\nabla} \cdot (\vec{\Omega} \Psi)$. But none of these three notations is equivalent in curvilinear coordinates. At the end, we give the curvilinear part of the streaming operator.

Cartesian coordinate:

Table 49 Streaming operator for Cartesian coordinate.

Dim ^a	Spatial variables	Angular variables	$\partial_{\vec{\Omega}} \Psi$	dx
1	x	μ	$\Omega_x \frac{\partial \Psi}{\partial x}$	dx
2	x, y	μ, ω	$\Omega_x \frac{\partial \Psi}{\partial x} + \Omega_y \frac{\partial \Psi}{\partial y}$	$dx dy$
3	x, y, z	μ, ω	$\Omega_x \frac{\partial \Psi}{\partial x} + \Omega_y \frac{\partial \Psi}{\partial y} + \Omega_z \frac{\partial \Psi}{\partial z}$	$dx dy dz$

^a Dimensions.

The polar angle $\varphi \in [0, \pi]$ is with respect to the fixed direction \hat{e}_x at every spatial point. It is typically use the cosine of the polar angle $\mu = \cos \varphi$ as the polar variable. And the azimuthal angle ω is measured in the plane of \hat{e}_y, \hat{e}_z starting from \hat{e}_y . Thus

$$\Omega_x = \mu, \quad (470a)$$

$$\Omega_y = \sqrt{1 - \mu^2} \cos \omega, \quad (470b)$$

$$\Omega_z = \sqrt{1 - \mu^2} \sin \omega, \quad (470c)$$

$$\vec{\Omega} \equiv (\Omega_x, \Omega_y, \Omega_z). \quad (470d)$$

Spherical coordinate:

Table 50 Streaming operator for spherical coordinate in conservation form.

Dim	Spatial variables	Angular variables	$\partial_{\vec{\Omega}} \Psi$	dx
1	r	μ	$\frac{\Omega_r}{r^2} \frac{\partial}{\partial r} (r^2 \Psi) + \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi]$	$4\pi r^2 dr$
2	r, ϕ	μ, ω	$\frac{\Omega_r}{r^2} \frac{\partial}{\partial r} (r^2 \Psi) + \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi] + \frac{\Omega_\phi}{r \sin \phi} \frac{\partial}{\partial \phi} (\sin \phi \Psi) - \frac{\cos \phi}{r \sin \phi} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi)$	$2\pi r^2 \sin \phi dr d\phi$
2	r, θ	μ, ω	$\frac{\Omega_r}{r^2} \frac{\partial}{\partial r} (r^2 \Psi) + \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi] - \frac{\cos \phi}{r \sin \phi} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi) + \frac{\Omega_\theta}{r \sin \phi} \frac{\partial \Psi}{\partial \theta}$	$r^2 \sin \phi dr d\theta$
3	r, ϕ, θ	μ, ω	$\frac{\Omega_r}{r^2} \frac{\partial}{\partial r} (r^2 \Psi) + \frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi] + \frac{\Omega_\phi}{r \sin \phi} \frac{\partial}{\partial \phi} (\sin \phi \Psi) - \frac{\cos \phi}{r \sin \phi} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi) + \frac{\Omega_\theta}{r \sin \phi} \frac{\partial \Psi}{\partial \theta}$	$r^2 \sin \phi dr d\theta d\phi$

$\theta \in [0, 2\pi]$ is the azimuthal angle in space. $\phi \in [0, \pi]$ is the polar angle in space. The polar angle $\varphi \in [0, \pi]$ is with respect to the direction \hat{e}_r at the spatial point. It is typically use the cosine of the polar angle $\mu = \cos \varphi$ as the polar variable. And the azimuthal angle ω is measured in the plane of $\hat{e}_\theta, \hat{e}_\phi$ starting from \hat{e}_θ . Thus

$$\Omega_r = \mu, \quad (471a)$$

$$\Omega_\phi = \sqrt{1 - \mu^2} \cos \omega, \quad (471b)$$

$$\Omega_\theta = \sqrt{1 - \mu^2} \sin \omega, \quad (471c)$$

$$\vec{\Omega} \equiv (\Omega_r, \Omega_\phi, \Omega_\theta). \quad (471d)$$

Cylindrical coordinate:

Table 51 Streaming operator for cylindrical coordinate in conservation form.

Dim	Spatial variables	Angular variables	$\partial_{\vec{\Omega}} \Psi$	dx
1	r	μ, ω	$\frac{\Omega_r}{r} \frac{\partial}{\partial r} (r\Psi) - \frac{1}{r} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi)$	$2\pi r dr$
2	r, θ	μ, ω	$\frac{\Omega_r}{r} \frac{\partial}{\partial r} (r\Psi) - \frac{1}{r} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi) + \frac{\Omega_\theta}{r} \frac{\partial \Psi}{\partial \theta}$	$r dr d\theta$
2	r, z	μ, ω	$\frac{\Omega_r}{r} \frac{\partial}{\partial r} (r\Psi) - \frac{1}{r} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi) + \Omega_z \frac{\partial \Psi}{\partial z}$	$2\pi r dr dz$
3	r, θ, z	μ, ω	$\frac{\Omega_r}{r} \frac{\partial}{\partial r} (r\Psi) - \frac{1}{r} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi) + \frac{\Omega_\theta}{r} \frac{\partial \Psi}{\partial \theta} + \Omega_z \frac{\partial \Psi}{\partial z}$	$r dr d\theta dz$

$\theta \in [0, 2\pi]$ is the azimuthal angle in space. The polar angle $\varphi \in [0, \pi]$ is with respect to the fixed z-direction \hat{e}_z at the spatial point. It is typically use the cosine of the polar angle $\mu = \cos \varphi$ as the polar variable. And the azimuthal angle ω is measured in the plane of $\hat{e}_r, \hat{e}_\theta$ starting from \hat{e}_r . Thus

$$\Omega_z = \mu, \quad (472a)$$

$$\Omega_r = \sqrt{1 - \mu^2} \cos \omega, \quad (472b)$$

$$\Omega_\theta = \sqrt{1 - \mu^2} \sin \omega, \quad (472c)$$

$$\vec{\Omega} = (\Omega_z, \Omega_r, \Omega_\theta). \quad (472d)$$

It is more convenient to split the streaming operator into two parts

$$\partial_{\vec{\Omega}} \Psi = \vec{\nabla} \cdot (\vec{\Omega} \Psi) + E\Psi, \quad (473)$$

where

$$E\Psi \equiv \partial_{\vec{\Omega}} \Psi - \vec{\nabla} \cdot (\vec{\Omega} \Psi) \quad (474)$$

and $\vec{\nabla} \cdot$ is the normal divergence operator in all types of coordinates. We also define

$$E_2\Psi \equiv \vec{\nabla} \cdot (\vec{\Omega} \Psi) - \vec{\Omega} \cdot \vec{\nabla} \Psi, \quad (475)$$

where $\vec{\nabla}$ is the normal gradient operator. The operators $E\Psi$ and $E_2\Psi$ for all coordinates in 3D are summarized in Table 52.

Table 52 Curvilinear part of the streaming operator.

Coordinate type	$E\Psi$	$E_2\Psi$
Cartesian	0	0
Spherical	$\frac{1}{r} \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi] - \frac{\cos \phi}{r \sin \phi} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi)$	$\left(\frac{2\Omega_r}{r} + \Omega_\phi \frac{\cos \phi}{r \sin \phi} \right) \Psi$
Cylindrical	$-\frac{1}{r} \frac{\partial}{\partial \omega} (\Omega_\theta \Psi)$	$\frac{\Omega_r}{r} \Psi$

This split is useful for understanding integration by parts and boundary conditions with divergence theorem:

$$\int_{\mathcal{D}} \Psi^* \vec{\nabla} \cdot (\vec{\Omega} \Psi) dx = \int_{\mathcal{D}} \vec{\nabla} \cdot (\vec{\Omega} \Psi \Psi^*) dx - \int_{\mathcal{D}} \Psi \vec{\Omega} \cdot \vec{\nabla} \Psi^* dx = \int_{\partial \mathcal{D}} \Psi \Psi^* \vec{\Omega} \cdot \vec{n}_b ds - \int_{\mathcal{D}} \Psi \vec{\Omega} \cdot \vec{\nabla} \Psi^* dx,$$

or with shorter notation

$$\left(\Psi^*, \vec{\nabla} \cdot (\vec{\Omega} \Psi) \right) = \langle \Psi^*, \Psi \rangle - \left(\vec{\Omega} \cdot \vec{\nabla} \Psi^*, \Psi \right). \quad (476)$$

The volume differential element dx has been provided for all coordinates. The heuristic way of obtaining the area element $\vec{n}_b ds$ is provided in the following. If the surface is defined by

$$x = x(u, v); y = y(u, v); z = z(u, v), \quad (477a)$$

$$\vec{n}_b ds = \vec{T}_u \times \vec{T}_v du dv, \quad (477b)$$

with the tangent vectors

$$\vec{T}_u = \frac{\partial x}{\partial u} \vec{e}_x + \frac{\partial y}{\partial u} \vec{e}_y + \frac{\partial z}{\partial u} \vec{e}_z, \quad (477c)$$

$$\vec{T}_v = \frac{\partial x}{\partial v} \vec{e}_x + \frac{\partial y}{\partial v} \vec{e}_y + \frac{\partial z}{\partial v} \vec{e}_z. \quad (477d)$$

The boundary of a mesh is composed of boundary side, each of which is defined by the nodes on the side. Thus, mesh framework can evaluate the outward normals and quadrature weights (area) on the quadrature points. The surface integral can be evaluated numerically. With curvilinear coordinates, normals and weights can be provided in a coordinate agnostic way. For example in R-Z coordinate, a surface can be defined as

$$\begin{aligned} x &= r(z) \cos \theta \\ y &= r(z) \sin \theta \\ z &= z. \end{aligned}$$

Surface integral of a vector $\mathbf{v} = a\hat{e}_r + b\hat{e}_z = \begin{bmatrix} a \cos \theta \\ a \sin \theta \\ b \end{bmatrix}$ over such a surface would be

$$\begin{aligned} & \int_{z_{\min}}^{z_{\max}} \int_{2\pi} \mathbf{v} \cdot \mathbf{r} \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix} \times \left(\begin{bmatrix} r_z \cos \theta \\ r_z \sin \theta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) d\theta dz \\ &= \int_{z_{\min}}^{z_{\max}} \int_{2\pi} \mathbf{v} \cdot \mathbf{r} \begin{bmatrix} \cos \theta \\ \sin \theta \\ r_z \end{bmatrix} d\theta dz \\ &= \int_{z_{\min}}^{z_{\max}} \begin{bmatrix} a \\ b \end{bmatrix} \cdot \begin{bmatrix} 1 \\ r_z \end{bmatrix} \boxed{2\pi r} dz \\ &= \int_S \mathbf{v} \cdot \vec{n}_b \boxed{2\pi r} ds, \end{aligned}$$

with $ds = \sqrt{1+r_z^2} dz$ and $\vec{n}_b = \frac{1}{\sqrt{1+r_z^2}} (\hat{e}_r + r_z \hat{e}_z)$. We also use S to denote the surface. The notation in the last step of derivation can be used for general surfaces. The boxed $2\pi r$ is the extra area element for R-Z coordinate. This extra area element corresponds to the volume differential element dx , so we can extend it for other curvilinear coordinates.

D Phase Function

D.1 Definition of phase function

A phase function $f(\vec{\Omega}, \vec{\Omega}')$ – also known as the scattering kernel – is a function for redistributing radiation from the incoming directions $\vec{\Omega}'$ to a certain outgoing direction $\vec{\Omega}$. The scattering operator $S[\Psi]$ can be written as

$$S[\Psi] \equiv \Sigma_s \int_{\mathcal{S}} f(\vec{\Omega}, \vec{\Omega}') \Psi(\vec{\Omega}') d\Omega'. \quad (478)$$

By definition, the phase function should not cause radiation extinction, i.e.

$$\int_{\mathcal{S}} f(\vec{\Omega}, \vec{\Omega}') d\Omega = 1. \quad (479)$$

For *isotropic media*, the phase function only depends on the cosine of the angle suspended by the incoming and outgoing directions, i.e. $\vec{\Omega} \cdot \vec{\Omega}'$. We thus define $\mu_0 \equiv \vec{\Omega} \cdot \vec{\Omega}'$ and also apply a factor 2π for the azimuthal symmetry:

$$f(\mu_0) \equiv 2\pi f(\vec{\Omega} \cdot \vec{\Omega}'). \quad (480)$$

If we denote

$$\vec{\Omega} = (\mu, \sqrt{1 - \mu^2} \cos \omega, \sqrt{1 - \mu^2} \sin \omega)^T, \quad (481)$$

$$\vec{\Omega}' = (\mu', \sqrt{1 - \mu'^2} \cos \omega', \sqrt{1 - \mu'^2} \sin \omega')^T, \quad (482)$$

we can see that

$$\mu_0 \equiv \vec{\Omega} \cdot \vec{\Omega}' = \mu\mu' + \sqrt{1 - \mu^2} \sqrt{1 - \mu'^2} \cos(\omega - \omega'). \quad (483)$$

Now the condition for the phase function becomes

$$\int_{-1}^1 f(\mu_0) d\mu_0 = 1, \quad (484)$$

and the scattering operator is

$$S[\Psi] \equiv \Sigma_s \int_{\mathcal{S}} \frac{1}{2\pi} f(\mu_0) \Psi(\vec{\Omega}') d\Omega'. \quad (485)$$

The phase function can be expanded with Legendre polynomials

$$f(\mu_0) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{2} f_{\ell} P_{\ell}(\mu_0), \quad (486)$$

where

$$f_{\ell} = \int_{-1}^1 f(\mu_0) P_{\ell}(\mu_0) d\mu_0. \quad (487)$$

With the addition theorem, the phase function can be expanded with spherical harmonics

$$f(\vec{\Omega} \cdot \vec{\Omega}') = \frac{1}{2\pi} f(\mu_0) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} f_{\ell} \sum_{m=-\ell}^{\ell} Y_{\ell,m}(\vec{\Omega}) Y_{\ell,m}(\vec{\Omega}'), \quad (488)$$

or written in matrix-vector product form as

$$f(\vec{\Omega} \cdot \vec{\Omega}') = \mathbf{Y}^T(\vec{\Omega}) \mathbf{F} \mathbf{Y}(\vec{\Omega}'), \quad (489)$$

where \mathbf{Y} is the column vector formed by all spherical harmonics and \mathbf{F} is the diagonal matrix containing the phase function moments with a factor $\frac{2\ell+1}{4\pi}$.

D.2 Common phase functions

Below, we list the phase functions that are available in Rattlesnake.

D.2.1 Isotropic

The simplest phase function possible does not present any anisotropy and is given by:

$$f(\mu_0) = \frac{1}{2}. \quad (490)$$

D.2.2 Rayleigh

The Rayleigh phase function is defined as:

$$f(\mu_0) = \frac{3}{8}(1 + \mu_0^2). \quad (491)$$

D.2.3 Henyey-Greenstein

Given a function $h = h(t, \vec{x})$, the Henyey-Greenstein phase function is defined as:

$$f(\mu_0) = \frac{1 - h^2}{2(1 + h^2 - 2h\mu)^{3/2}}. \quad (492)$$

D.2.4 Heaviside

The Heaviside phase function is defined as:

$$f(\mu_0) = \begin{cases} 1 & , \mu_0 > 0 \\ 0 & , \text{otherwise} \end{cases}. \quad (493)$$

D.2.5 Dirac

Given $\tilde{\mu} \in [-1, 1]$, the Dirac phase function is defined as:

$$f(\mu_0) = \delta(\mu_0 - \tilde{\mu}) \quad (494)$$

where δ represents the delta Dirac function.

In practice, only its truncated version is currently available in Rattlesnake:

$$f(\mu_0) \approx \sum_{\ell=0}^L \frac{2\ell+1}{2} P_{\ell}(\tilde{\mu}) P_{\ell}(\mu_0). \quad (495)$$

where P_{ℓ} is the Legendre polynomial of degree ℓ and L is the truncation degree.

D.3 Scattering treatments for SN

D.3.1 PN treatment

The default way in Rattlesnake is to express the scattering operator using a spherical harmonics expansion of Ψ , truncated at degree L :

$$(S\Psi)_g = \sum_{g'=1}^G \int_{4\pi} \Sigma_{s,g' \rightarrow g} \frac{f(\mu_0)}{2\pi} \Psi_{g'}(\vec{\Omega}') d\Omega' \approx \sum_{g'=1}^G \sum_{\ell=0}^L \frac{2\ell+1}{w} \Sigma_{s,\ell}^{g' \rightarrow g} \sum_{k=-\ell}^{\ell} \Phi_{g',\ell,k} Y_{\ell,k}(\vec{\Omega}). \quad (496)$$

where, as a reminder w is the total angular weight (2 in 1-D, 2π in 2-D and 4π in 3-D) with in particular $w = \sum_m w_m$. The angular moments are computed using:

$$\Phi_{g,\ell,k} = \sum_{m'=1}^M w_{m'} \Psi_{g,m'} Y_{\ell,k}(\vec{\Omega}_{m'}). \quad (497)$$

The direction-to-moment matrix is then:

$$\mathcal{D}_{(\ell,k),m} = w_m Y_{\ell,k}(\vec{\Omega}_m) \quad (498)$$

where we have defined \mathcal{D} as:

$$\Phi_g \equiv \mathcal{D} \Psi_g. \quad (499)$$

In addition, the angular flux being expressed as:

$$\Psi_g(\vec{\Omega}) \approx \sum_{\ell=0}^L \frac{2\ell+1}{w} \sum_{k=-\ell}^{\ell} \Phi_{g,\ell,k} Y_{\ell,k}(\vec{\Omega}) = \mathbf{R}^T \Phi_g \quad (500)$$

the moment-to-direction matrix is:

$$\mathcal{M}_{m,(\ell,k)} = R_{\ell,k}(\vec{\Omega}_m) \quad (501)$$

where we have defined \mathcal{M} as:

$$\Psi_g \equiv \mathcal{M} \Phi_g. \quad (502)$$

In particular, assuming that the angular quadrature was chosen such that it can integrate exactly all the spherical harmonics up to degree L , we have:

$$\mathcal{D} \mathcal{M} = I \quad (503)$$

where I is the identity matrix (of size M).

In 1-D or 2-D, if the external source is also expanded as $Q_g^{\text{ext}} = \mathbf{R}^T \mathbf{Q}_g$, nothing needs to be changed in the code: the value of w in the definition of \mathbf{R} is scaled accordingly to make sure the solution for the scalar flux is unchanged.

D.3.2 SN treatment

In Rattlesnake, we have two S_N treatments [97] to express the scattering source which we refer to as *sampling* and *patch* methods. Both rely on constructing a direction-to-direction matrix. Let us recall the expression of the scattering source:

$$(S\Psi)_g(\vec{\Omega}) = \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \int_{4\pi} \frac{f(\mu_0)}{2\pi} \Psi_{g'}(\vec{\Omega}') d\Omega', \quad (504)$$

In the *sampling* method, if the angular flux depends on all three spatial dimensions, we directly use the full-sphere angular quadrature for integration purposes:

$$(S\Psi)_{g,m} \approx \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \sum_{m'=1}^N \frac{f(\vec{\Omega}_m \cdot \vec{\Omega}_{m'})}{2\pi} \Psi_{g',m'} w_{m'}, \quad (505)$$

with N being the total number of directions, the subscript indicating the direction index and w being the angular weight. In reduced geometries, more care is required because the angular quadrature does not span the entire unit sphere.

The main difference in the *patch* method is that the unit sphere is decomposed in non-overlapping patches of the unit sphere $\mathcal{S} = \bigcup_{m'} \mathcal{S}_{m'}$ where $\mathcal{S}_{m'}$ contains the direction $\vec{\Omega}_{m'}$ and we express the scattering source as:

$$(S\Psi)_{g,m} \approx \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \sum_{m'=1}^N \int_{\mathcal{S}_{m'}} \frac{f(\vec{\Omega}_m \cdot \vec{\Omega}')}{2\pi} d\vec{\Omega}' \Psi_{g',m'}, \quad (506)$$

1-D S_N Treatment

With a problem depending only on one spatial direction, the axes can be chosen such that Ψ does not depend on ω . Then, defining:

$$g(\mu, \mu') \equiv \int_0^{2\pi} \frac{f(\mu_0)}{2\pi} d\omega', \quad (507)$$

where, as a reminder,

$$\mu_0 \equiv \vec{\Omega} \cdot \vec{\Omega}' = \mu\mu' + \sqrt{1-\mu^2}\sqrt{1-\mu'^2}\cos(\omega - \omega'), \quad (508)$$

we have:

$$\int_{4\pi} \frac{f(\mu_0)}{2\pi} \Psi(\vec{\Omega}') d\Omega' = \int_{-1}^1 g(\mu, \mu') \Psi(\mu') d\mu'. \quad (509)$$

The scattering source in the *sampling* method can then be approximated using:

$$(S\Psi)_{g,m} \approx \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \sum_{m'=1}^N g(\mu_m, \mu_{m'}) \Psi_{g',m'} w_{m'}. \quad (510)$$

The reduced function g depends on both μ and μ' .

Likewise, in the *patch* method, it is:

$$(S\Psi)_{g,m} \approx \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \sum_{m'=1}^N \int_{\mathcal{S}_{m'}} g(\mu_m, \mu') d\mu' \Psi_{g',m'}. \quad (511)$$

2-D S_N Treatment

With a problem depending on two spatial dimensions, we can choose to have Ψ be an even function of μ . With the following definition,

$$g(\mu, \mu', \mu_0) \equiv \frac{f(\mu_0) + f(\mu_0 - 2\mu\mu')}{2}, \quad (512)$$

one can show that,

$$\int_{4\pi} \frac{f(\mu_0)}{2\pi} \Psi(\vec{\Omega}') d\Omega' = \int_0^1 \int_0^{2\pi} \frac{g(\mu, \mu', \mu_0)}{\pi} \Psi(\mu', \omega') d\omega' d\mu'. \quad (513)$$

The scattering source may then computed as follows in the *sampling* method:

$$(S\Psi)_{g,m} \approx \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \sum_{m'=1}^N \frac{g(\mu_m, \mu_{m'}, \vec{\Omega}_m \cdot \vec{\Omega}_{m'})}{\pi} \Psi_{g',m'} w_{m'}. \quad (514)$$

and as:

$$(S\Psi)_{g,m} \approx \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g} \sum_{m'=1}^N \int_{\mathcal{S}_{m'}} \frac{g(\mu_m, \mu', \vec{\Omega}_m \cdot \vec{\Omega}')}{\pi} d\vec{\Omega}' \Psi_{g',m'}. \quad (515)$$

in the *patch* method. The reduced function g then depends on μ , μ' and $(\omega - \omega')$.

D.3.3 S_N - P_N equivalency

In this section, we assume that f can be expanded using a Legendre polynomials expansion of degree L :

$$f(\mu_0) = \sum_{\ell=0}^L \frac{2\ell+1}{2} f_\ell P_\ell(\mu_0), \quad (516)$$

and show that the P_N and S_N treatments are then equivalent.

First of all, in an S_N calculation, the moment $\Phi_{\ell,k}$ is evaluated using:

$$\Phi_{\ell,k} = \sum_{m'=1}^M w_{m'} \Psi_{m'} Y_{\ell,k}(\vec{\Omega}_{m'}). \quad (517)$$

Next, we consider the contribution of the scattering source to the direction $\vec{\Omega}_m$:

$$\begin{aligned} (H\Psi)_m &= \sum_{\ell=0}^L \frac{2\ell+1}{4\pi} \sigma_{s,\ell} \sum_{k=-\ell}^{\ell} \Phi_{\ell,k} Y_{\ell,k}(\vec{\Omega}_m), \\ &= \sum_{m'=1}^M w_{m'} \Psi_{m'} \sum_{\ell=0}^L \frac{2\ell+1}{4\pi} \sigma_{s,\ell} \sum_{k=-\ell}^{\ell} Y_{\ell,k}(\vec{\Omega}'_m) Y_{\ell,k}(\vec{\Omega}_m). \end{aligned} \quad (518)$$

Using the spherical harmonics addition theorem:

$$\sum_{k=-\ell}^{\ell} Y_{\ell,k}(\vec{\Omega}'_m) Y_{\ell,k}(\vec{\Omega}_m) = P_\ell(\mu_0), \quad (519)$$

we have:

$$(H\Psi)_m = \sum_{m'=1}^M w_{m'} \Psi_{m'} \sum_{\ell=0}^L \sigma_{s,\ell} \frac{2\ell+1}{4\pi} P_\ell(\mu_0). \quad (520)$$

In other words, using Eq. (516),

$$(H\Psi)_m = \sigma_s \sum_{m'=1}^M \frac{f(\mu_0)}{2\pi} w_{m'} \Psi_{m'}, \quad (521)$$

which is equal to Eq. (505).

E Angular Quadratures

E.1 General discussions

An angular quadrature $\{\vec{\Omega}_d, w_d, d = 1, \dots, N_d\}$ is designed to numerically integrate a function $\Psi(\vec{\Omega})$ over the angular domain

$$\int_S \Psi(\vec{\Omega}) d\Omega = \sum_{d=1}^{N_d} w_d \Psi(\vec{\Omega}_d). \quad (522)$$

We adopt the first element of $\vec{\Omega}$ as the cosine of the polar angle and measure the azimuthal angle from the second axis as illustrated in Fig. 11:

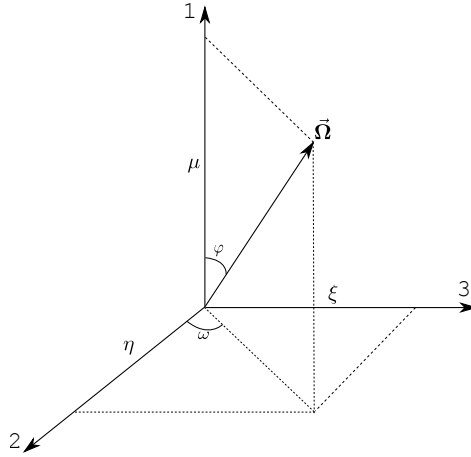


Figure 11 Angular coordinates for angular quadratures.

$$\Omega_1 = \mu, \quad (523a)$$

$$\Omega_2 = \sqrt{1 - \mu^2} \cos \omega, \quad (523b)$$

$$\Omega_3 = \sqrt{1 - \mu^2} \sin \omega, \quad (523c)$$

where μ is in $[-1, 1]$; the dependency on ω can be dropped in 1D, and ω is in $[0, \pi]$ and in $[0, 2\pi]$ for 2D and 3D respectively. It is traditional to give a shorter notation for Ω_1 , Ω_2 and Ω_3 as μ , η , and ξ respectively. Thus the three axis are called μ , η , or ξ axis. Although we typically choose $\Omega_1 = \Omega_x$, it is often desired that the angular quadrature can switch the axis with respect to the spatial coordinate because the resolution of the angular quadrature with respect to each axis can be different. For instance, in calculations with production quadratures, it is typical to let $\Omega_1 = \Omega_z$ because the heterogeneity is not as significant in the z-direction.

With the exception of biased or especially tailored quadratures, most angular quadratures have the *planar symmetries*, i.e. $-\mu$ is in the quadrature if μ is in the quadrature in 1D, $(-\mu, \pi - \omega)$ is in the quadrature if (μ, ω) is in the quadrature in 2D, $(-\mu, \text{mod}(\pi + \omega, 2\pi))$ is in the quadrature if (μ, ω) is in the quadrature in 3D. The product of two spherical harmonics $Y_{\ell, m}$ and $Y_{\ell', m'}$ can be exactly integrated except when $\ell + \ell'$ and $m + m'$ are even and $m \times m' \geq 0$. Because of this, angular quadratures are typically designed in the first octant and then extended to the other octants.

It is desired that angular quadratures can exactly integrate spherical harmonics

$$\int_S R_{\ell, m}(\vec{\Omega}) d\Omega = \sum_{d=1}^{N_d} w_d R_{\ell, m}(\vec{\Omega}_d) = \delta_{\ell, 0}. \quad (524)$$

This property is commonly referred to as the *conservation* because of its relation with conservation in the scattering operator:

$$\int_S S[\Psi] d\Omega = \sum_{\ell=0}^L \frac{2\ell+1}{4\pi} \Sigma_{s,\ell} \sum_{m=-\ell}^{\ell} \int_S Y_{\ell,m}(\vec{\Omega}) d\Omega \Phi_{\ell,m} = \Sigma_s \Phi. \quad (525)$$

This property must hold for spherical harmonics whose order ℓ is less than or equal to the scattering anisotropy L in the phase function to preserve the particle during a scattering event.

The vector of angular moments Φ of a function $\Psi(\vec{\Omega})$ is defined as the inner product of the function and the vector of spherical harmonics \mathbf{Y} over the sphere

$$\Phi \equiv (\Psi, \mathbf{Y})_S = \int_S \Psi \mathbf{Y} d\Omega. \quad (526)$$

We denote the size of the vectors Φ and \mathbf{Y} as N_ℓ . The moment-to-direction matrix $\mathbf{M}_{N_d \times N_\ell}$ can be generated by simply sampling the function over the directions in the angular quadrature. Its elements are

$$M_{i,j} = R_{\ell(j),m(j)}(\vec{\Omega}_i), \quad (527)$$

where $\ell(j), m(j)$ respectively are the spherical harmonics degree and order corresponding to the index j in the spherical harmonics vector. With the angular quadrature we can define a direction-to-moment matrix $\mathbf{D}_{N_\ell \times N_d}$ with elements

$$D_{i,j} = w_j Y_{\ell(i),m(i)}(\vec{\Omega}_j), \quad (528)$$

which is equal to $\mathbf{M}^T \mathbf{W}$, where $\mathbf{W} = \text{diag}\{w_1, \dots, w_{N_d}\}$. One desired property is that the product of the direction-to-moment and moment-to-direction matrices should not change the original function at the directions in the angular quadrature, i.e.

$$\mathbf{MD} = \mathbf{I}. \quad (529)$$

This requires at least $N_\ell \geq N_d$. We typically just make $N_\ell = N_d$. We call this property *transform invertibility*. We refer to angular quadratures having this property as so-called *Galerkin quadratures*. This property is particularly important for handling forward-peaked scattering, where the phase function is close to $\delta(\mu_0 - 1)$. To exactly integrate the scattering kernel $\delta(\mu_0 - 1)$, the angular quadrature need satisfy transform invertibility for a given set of spherical harmonics.

$$S_\delta[\Psi] = \int_S \delta(\mu_0 - 1) \Psi(\vec{\Omega}') d\Omega' = \Psi(\vec{\Omega}). \quad (530)$$

We can prove that the transform invertibility holds for Gaussian quadratures with $2N$ points and the Legendre polynomials of degree from 0 to $2N - 1$ in 1D. The property is however not valid for other 1D quadratures or for general multi-dimension angular quadratures. There are two ways of modifying the angular quadratures to maintain this property:

1. using the collocation approach to redesign the direction-to-moment matrix, which will be essentially the inversion of the moment-to-direction matrix.
2. changing the way of generating the moment-to-direction matrix from sampling to simply inverting the direction-to-moment matrix.

Either way, we need to make sure that the subset of spherical harmonics \mathbf{Y} results in an invertible moment-to-direction matrix while maximizing the subspace order $L(\mathbf{Y})$.

The collocation method first finds an approximate solution in a subspace spanned by interpolation sphere functions $f_d(\vec{\Omega}), d = 1, \dots, N_d$, which satisfy the condition

$$f_d(\vec{\Omega}_{d'}) = \delta_{dd'}, 1 \leq d, d' \leq N_d. \quad (531)$$

The continuous function $\Psi(\vec{\Omega})$ can be constructed with values on discrete directions specified by the angular quadrature $\Psi_d, d = 1, \dots, N_d$ as

$$\Psi(\vec{\Omega}) = \sum_{d=1}^{N_d} \Psi_d f_d(\vec{\Omega}). \quad (532)$$

Then we can *analytically* integrate Eq. (526) to get the direction-to-moment matrix

$$D_{i,j} = \int_{\mathcal{S}} Y_{\ell(i),m(i)}(\vec{\Omega}) f_j(\vec{\Omega}) d\Omega. \quad (533)$$

There are different ways of constructing the interpolation functions, such as finite element on sphere, spherical harmonics expansion, etc. We will explore the spherical harmonics method and show that the resulting direction-to-moment matrix is simply the inversion of the moment-to-direction matrix. We let the interpolation subspace be the same as the subspace spanned by the spherical harmonics in \mathbf{Y} . Then we have

$$f_d(\vec{\Omega}) = \mathbf{a}_d^T \mathbf{Y}, \quad (534)$$

where \mathbf{a} contains the expansion coefficients of f_d . If we define

$$\mathbf{f} \equiv \begin{bmatrix} f_1 \\ \vdots \\ f_{N_d} \end{bmatrix}, \mathbf{A} \equiv \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_{N_d}^T \end{bmatrix}, \quad (535)$$

we have $\mathbf{f} = \mathbf{A}\mathbf{Y}$. Because of the orthogonality of spherical harmonics, $\mathbf{A} = \mathbf{D}$. Based on the constraints, we eventually have $\mathbf{A}\mathbf{M} = \mathbf{D}\mathbf{M} = \mathbf{I}$. Collocation methods also provide a way of obtaining an angularly continuous function $\bar{\Psi}(\vec{\Omega})$ with $\Psi(\vec{\Omega}_d), d = 1, \dots, N_d$ for approximating $\Psi(\vec{\Omega})$. We can then use $\bar{\Psi}(\vec{\Omega})$ to evaluate values at a direction not necessarily in the quadrature, angular derivatives required by curvilinear coordinates, etc. The general interpolation capability is however not implemented for all angular quadratures currently in Rattlesnake.

Two angular derivatives need to be evaluated in curvilinear coordinates (refer to Appendix C): $\frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi]$ and $\sin \omega \frac{\partial \Psi}{\partial \omega}$. One task is to evaluate these derivatives by only knowing the function on the discrete directions of the angular quadrature $\Psi_d, d = 1, \dots, N_d$. We define two coefficient matrices α -matrix and β -matrix for evaluating the two derivatives as follows:

$$\sin \omega \frac{\partial \Psi}{\partial \omega} \Big|_{\vec{\Omega}=\vec{\Omega}_d} = \sum_{d'=1}^{N_d} \alpha_{d,d'} \Psi_{d'}, \quad (536)$$

$$\frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi] \Big|_{\vec{\Omega}=\vec{\Omega}_d} = \sum_{d'=1}^{N_d} \beta_{d,d'} \Psi_{d'}. \quad (537)$$

The angular quadrature is capable of evaluating the angular derivatives if those coefficient matrices are assembled. We currently have only implemented the μ -derivative for the 1D Gaussian quadrature and the ω -derivative for the Level-Symmetric quadrature.

E.1.1 Polar derivative

Ref. [3] provides one way to obtain these β s. We let

$$\frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi] \Big|_{\mu=\mu_d} = 2 \frac{\alpha_{d+1/2} \Psi_{d+1/2} - \alpha_{d-1/2} \Psi_{d-1/2}}{w_d}, \quad (538)$$

and obtain the constraints,

$$\alpha_{N_d+1/2} = \alpha_{1/2} = 0, \quad (539)$$

$$\alpha_{d+1/2} = \alpha_{d-1/2} - w_d \mu_d, d = 1, \dots, N_d. \quad (540)$$

We also include the approximation

$$\Psi_{d+1/2} = 2\Psi_d - \Psi_{d-1/2}, d = 1, \dots, N_d. \quad (541)$$

We need a dummy direction with $\mu = -1$ for $\Psi_{1/2}$ indexed as $d = 0$ later to start solving the angular fluxes sequentially from $d = 1$ to N_d . Eq. (536) on this dummy direction is simply equal to 2Ψ . We call this direction dummy because its angular flux does not contribute to the evaluation of the angular flux moments. It is desired to

add another dummy direction $\mu = 1$ denoted with $d = N_d + 1$ for the adjoint calculation. We propose the following equation

$$\Psi_{N_d+1} = 2\Psi_{N_d+1} - (2\Psi_{N_d} - \Psi_{N_d-1/2}). \quad (542)$$

to artificially bring in the coupling from the directions of $d = 0, \dots, N_d$ for this additional dummy direction. Finally, we obtain

$$\beta_{d,d'} = \begin{cases} (-1)^d \frac{2}{w_d} (\alpha_{d+1/2} + \alpha_{d-1/2}), & d' = 0, 0 < d \leq N_d; \\ (-1)^{d'+d} \frac{4}{w_d} (\alpha_{d+1/2} + \alpha_{d-1/2}), & 0 < d' < d, 0 < d \leq N_d; \\ \frac{4}{w_d} \alpha_{d+1/2}, & d' = d, 0 < d \leq N_d; \\ -2\mu_0 & d' = d = 0; \\ 0 & d' > d; 0 \leq d \leq N_d \\ 2, & d' = 0, d = N_d + 1; \\ (-1)^{d'} 4, & 0 < d' \leq N_d; d = N_d + 1 \\ -4 & d' = d = N_d + 1. \end{cases} \quad (543)$$

We can prove

$$\sum_{d'=0}^{N_d} \beta_{d,d'} = -2\mu_d. \quad (544)$$

This implies that the μ -derivative is exactly evaluated with isotropic Ψ . The matrix formed by the β s is lower triangular. Because this scheme enables the sequential solving of angular fluxes once the source is known, we refer to it as the sweeping scheme. This scheme is only available for production quadratures, for which the polar and azimuthal angles are constructed independently (this in particular excludes the Level-Symmetric quadrature).

We also present another new way of evaluating the μ -derivative. Numerical results show the superior angular convergence of this treatment. From the discrete angular fluxes $\Psi_d, d = 1, \dots, N_d$, we evaluate the angular flux moments up to order $N_d - 1$ with

$$\Phi_\ell = \sum_{d'=1}^{N_d} w_{d'} P_\ell(\mu_{d'}) \Psi_{d'}, \ell = 0, \dots, N_d - 1, \quad (545)$$

where P_ℓ is the Legendre polynomial of degree ℓ . Then we can construct the continuous angular flux

$$\Psi(\mu) = \sum_{\ell=0}^{N_d-1} \frac{2\ell+1}{2} P_\ell(\mu) \sum_{d'=1}^{N_d} w_{d'} P_\ell(\mu_{d'}) \Psi_{d'}, \quad (546)$$

which allows us to evaluate $\left. \frac{\partial}{\partial \mu} [(1 - \mu^2) \Psi] \right|_{\mu=\mu_d}$ as

$$\sum_{\ell=0}^{N_d-1} \frac{2\ell+1}{2} ((\ell-1)\mu_d P_\ell(\mu_d) - (\ell+1)P_{\ell+1}(\mu_d)) \sum_{d'=1}^{N_d} w_{d'} P_\ell(\mu_{d'}) \Psi_{d'}, \quad (547)$$

i.e.

$$\beta_{d,d'} = \sum_{\ell=0}^{N_d-1} \frac{2\ell+1}{2} ((\ell-1)\mu_d P_\ell(\mu_d) - (\ell+1)P_{\ell+1}(\mu_d)) w_{d'} P_\ell(\mu_{d'}). \quad (548)$$

We can prove that Eq. (544) still holds for these β s. Because $\mu_d = -\mu_{N_d+1-d}, w_d = w_{N_d+1-d}, d = 1, \dots, N_d/2$, we have

$$\beta_{N_d+1-d, N_d+1-d'} = -\beta_{d,d'}, \quad (549)$$

which is an appealing feature for adjoint calculations. With this treatment, β matrix is no longer lower triangular, meaning that we do not perform an angular sweep to invert the streaming-plus-collision operator. However, this is not necessarily considered a disadvantage with Rattlesnake's solvers. Moreover this scheme does not require dummy directions and works for all angular quadratures if the M angular flux moments can be constructed. Because this scheme involves the spherical harmonics (Legendre polynomials in 1D) expansion of the angular flux, we refer to it as the PN scheme for evaluating the angular derivatives.

E.1.2 Azimuthal derivative with P_N expansion

The purpose of this section is to show how we can evaluate the azimuthal derivative of the angular flux Ψ with a P_N expansion. This is in particular used in Section 3.4.2.7 for the SAAF-SN-RZ method. The goal is to find the α 's to evaluate the azimuthal derivative as:

$$\left[\Omega_\theta \frac{\partial \Psi}{\partial \omega} \right] \Big|_{\Omega_m} = \sum_{n=1}^{N_d} \alpha_{m,n} \Psi_n \quad (550)$$

Yet, we have shown

$$\Phi = \mathbf{A} \Psi \quad (551)$$

i.e.⁹:

$$\Phi_{\ell,k} = \sum_{n=1}^{N_d} A_{p(\ell,k),n} \Psi_n \quad (552)$$

In addition, one can show that:

$$\frac{\partial R_{\ell,k}}{\partial \omega} = -k R_{\ell,-k} \quad (553)$$

implying:

$$\Omega_\theta \frac{\partial \Psi}{\partial \omega} = - \sum_{p=1}^{N_\ell} \Phi_{\ell,k} k \Omega_\theta R_{\ell,-k} \quad (554)$$

This then gives us:

$$\alpha_{m,n} = - \sum_{p=1}^{N_\ell} k \Omega_{\theta,m} R_{\ell,-k} (\vec{\Omega}_m) A_{p(\ell,k),n}. \quad (555)$$

Three angular quadratures are implemented in Rattlesnake: LS (Level-Symmetric), GC (Gauss-Chebyshev) and BO (Bickley-3 Optimized). Their capabilities are illustrated in Table 53.

Table 53 The angular quadrature capability matrix.

Scheme	1D	2D	3D	Production	Interpolation	μ -deri	ω -deri	Galerkin
Level-Symmetric	N	Y	Y	N	N	N	Y	Y
Gauss-Chebyshev	Y	Y	Y	Y	N	Y	N	N
Bickley-3 Optimized	N	Y	N	Y	N	N	N	N

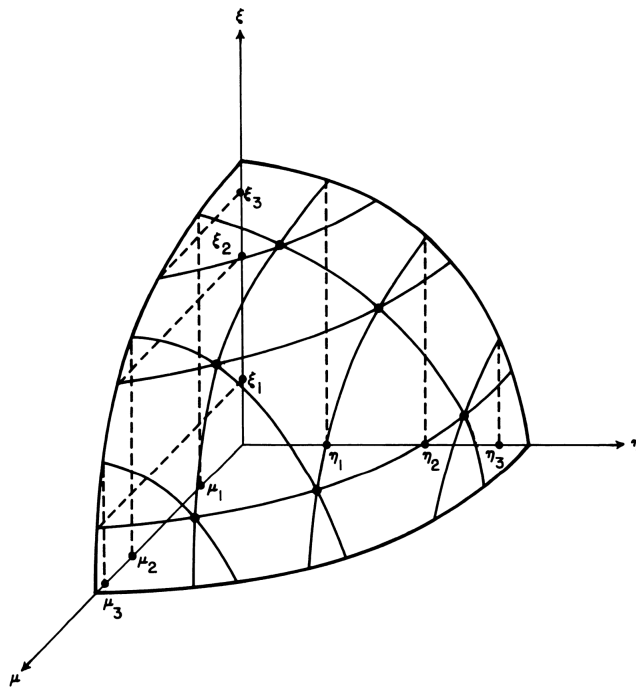
E.2 Level-Symmetric

The Level-Symmetric angular quadrature [98, 99] has the octant rotation invariance, i.e. the (μ, η, ξ) coordinates of points on the unit sphere chosen to represent $\vec{\Omega}$ are invariant under all 90-degree rotations about the μ , η , or ξ axis. Thus the set of points on each axis must be the same. For n points on each axis $[-1, 1]$, there are $\frac{n(n+2)}{8}$ points per octant where n is a positive even number, which is often called the *quadrature order*. The arrangement of quadrature points with $n = 6$ is illustrated in Fig. 12. Because of the symmetry, three indices i, j, k of the coordinates of a point on the sphere sum to $n/2 + 2$, i.e. $\mu_i^2 + \mu_j^2 + \mu_k^2 = 1$, where $k = n/2 + 2 - i - j$; $j = 1, \dots, n/2 - i + 1$; $i = 1, 2, \dots, n/2$. This relation requires that

$$\mu_i^2 = \mu_1^2 + (i-1) \frac{2(1-3\mu_1^2)}{N-2}, \quad (556)$$

where μ_1 is a free parameter controlling the location of all quadrature points. Rotational invariance also dictates that weights for points on the unit sphere should be chosen in a symmetric fashion, i.e. the directions of all permutations

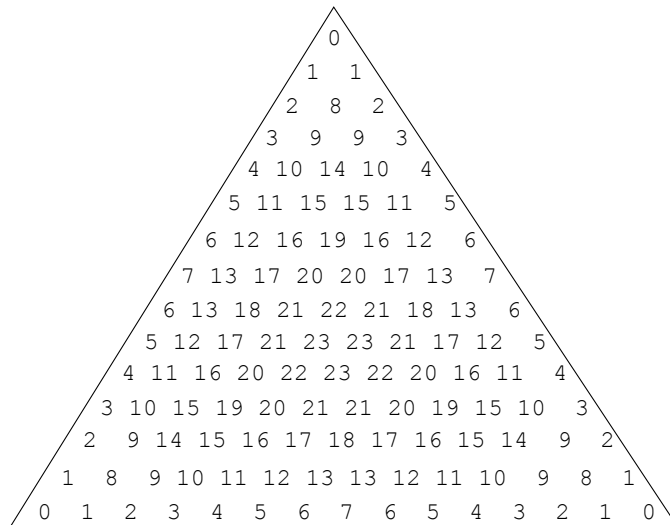
⁹ Another possibility is to use $\Phi_{\ell,k} = \sum_{n=1}^{N_d} w_n Y_{\ell,k}(\vec{\Omega}_n) \Psi_n$ which then gives $\alpha_{m,n} = - \sum_{p \in \mathcal{N}} k \Omega_{\theta,m} R_{\ell,-k}(\vec{\Omega}_m) w_n Y_{\ell,k}(\vec{\Omega}_n)$.

Figure 12 Completely symmetric point arrangement, $n = 6$.

of a given set of indices (i, j, k) must have the same weights. The number of distinct weights per octant N_w is equal to

$$N_w = 1 + \left\lfloor \frac{N(N+8) - 1}{48} \right\rfloor, \quad (557)$$

where $\lfloor \cdot \rfloor$ is the floor operator. We denote the distinct weight for a particular direction d as $w_{j(d)}$. For example, the distinct weight IDs of $n = 30$ is illustrated in Fig. 13. μ_1 and the distinct weights add up to the total $N_w + 1$

Figure 13 ID of distinct weights of $n = 30$.

free parameters. When μ_1 is given, the same number of angular functions in form of $\mu^p \eta^q$ are chosen denoted with

$(p_i, q_i), i = 0, \dots, N_w - 1$ to determine N_w weights. The following N_w equations

$$\sum_{d=1}^{N_d} w_{j(d)} \mu_d^{p_i} \eta_d^{q_i} = \int_0^1 d\mu \int_0^{\frac{\pi}{2}} d\omega \mu^{p_i} \eta^{q_i} = \frac{1}{p_i + 1} \prod_{k=1, \text{odd}}^{q_i-1} \frac{k}{p_i + k + 2}, i = 0, \dots, N_w - 1, \quad (558)$$

can thus be solved to obtain the N_w distinct weights. It is noted that (p_i, q_i) is necessary to be even because of the symmetry. We can chose another extra angular functions (p_{N_w}, q_{N_w}) to determine μ_1 by satisfying

$$f(\mu_1) = \sum_{d=1}^{N_d} w_{j(d)} \mu_d^{p_{N_w}} \eta_d^{q_{N_w}} - \frac{1}{p_{N_w} + 1} \prod_{k=1, \text{odd}}^{q_{N_w}-1} \frac{k}{p_{N_w} + k + 2} = 0. \quad (559)$$

A Newton iteration can be used to find the root for this nonlinear equation. Level-Symmetric quadratures can uniquely be determined by $(p_i, q_i), i = 0, \dots, N_w$. Table 54 summarizes the angular functions used for constructing the Level-Symmetric quadratures. μ_1 are also listed in Table 54. Distinct weights are not included in Table 54 because they can easily be evaluated with Eq. (558) given μ_1 . The maximum orders L_{\max} of spherical harmonics that the quadratures can exactly integrate are also given in Table 54. They are almost always equal to $n - 1$ with exceptions of smaller ns , such as 2, 4, 6, 8, 10, and 14. It is noted that there are negative weights starting from order 20.

Spherical harmonics are selected with the following rules to construct the interpolation sphere functions:

$$\begin{cases} \{(l, m), m = 0, \dots, l; l = 0, \dots, n - 1; (l = n, m), m = 1, 3, \dots, n - 1\}, & 2D \\ \{(l, m), m = -l, \dots, l; l = 0, \dots, n - 1; (l = n, m), m = -l + 1, -l + 3, \dots, n - 1; \\ (l = n, m), m = -l, -l + 2, \dots, -2; (l = n + 1, m), m = -l + 1, -l + 3, \dots, -1\}, & 3D \end{cases} \quad (560)$$

The numbers of spherical harmonics in the selections are indeed equal to $\frac{n(n+2)}{2}$ and $n(n+2)$ for 2D and 3D respectively.

Table 54 Level-Symmetric quadrature.

Order	N_w	Moments: $(p_i, q_i), i = 0 \cdots, N_w - 1$	(p_{N_w}, q_{N_w})	μ_1	L_{\max}
2	1	(0,0)	(2,0)	0.577350269189625764509149	3
4	1	(0,0)	(4,0)	0.350021174581540677777041	5
6	2	(0,0), (4,0)	(4,2)	0.266635401516704720331535	7
8	3	(0,0), (4,0), (6,0)	(8,0)	0.218217890235992381266097	9
10	4	(0,0), (4,0), (6,0), (8,0)	(6,4)	0.189321326478010476671494	11
12	5	(0,0), (4,0), (6,0), (8,0) (10,0)	(6,6)	0.167232497141491206371208	11
14	7	(0,0), (4,0), (6,0), (8,0), (10,0), (12,0), (6,6)	(8,6)	0.151985861461031912404799	15
16	8	(0,0), (4,0), (6,0), (8,0), (10,0), (12,0), (6,6), (14,0)	(8,8)	0.138975694612626558498101	15
18	10	(0,0), (4,0), (6,0), (8,0), (10,0), (12,0), (6,6), (14,0), (16,0), (8,8)	(10,8)	0.129348112085829872872402	17
20	12	(0,0), (4,0), (6,0), (8,0), (10,0), (12,0), (6,6), (14,0), (16,0), (8,8), (18,0), (10,8)	(10,10)	0.120620464081574425273591	19
22	14	(0,0), (4,0), (6,0), (8,0), (10,0), (12,0), (6,6), (14,0), (16,0), (8,8), (18,0), (10,8), (20,0), (10,10)	(12,10)	0.113892684099307786159500	21
24	16	(0,0), (4,0), (6,0), (8,0), (10,0), (12,0), (6,6), (14,0), (16,0), (8,8), (18,0), (10,8), (20,0), (10,10), (22,0), (12,10)	(12,12)	0.107559189018123222298175	23
26	19	(0,0), (4,0), (6,0), (8,0), (10,0), (12,0), (6,6), (14,0), (16,0), (8,8), (18,0), (10,8), (20,0), (10,10), (22,0), (12,10), (24,0), (14,10), (12,12)	(14,12)	0.102543854666975591668637	25
28	21	(0,0), (4,0), (6,0), (8,0), (10,0), (12,0), (6,6), (14,0), (16,0), (8,8), (18,0), (10,8), (20,0), (10,10), (22,0), (12,10), (24,0), (14,10), (14,12), (26,0), (16,10)	(14,14)	0.0976979229188335649373355	27
30	24	(0,0), (4,0), (6,0), (8,0), (10,0), (12,0), (6,6), (14,0), (16,0), (8,8), (18,0), (10,8), (20,0), (10,10), (22,0), (12,10), (24,0), (14,10), (14,12), (26,0), (16,10), (28,0), (16,12), (14,14)	(16,14)	0.0937869286135548109935698	29

E.3 Gauss Chebyshev

The angular quadrature was proposed in [98, 100]. The angular integral is

$$\int_S d\Omega = \int_{-1}^1 d\mu \int_0^{2\pi} d\omega. \quad (561)$$

If we let $\cos \omega = y$, the integral becomes

$$\int_S d\Omega = 2 \int_{-1}^1 d\mu \int_{-1}^1 \frac{1}{\sqrt{1-y^2}} dy. \quad (562)$$

If n_c is the order of the Chebyshev quadrature, or the number of quadrature points on the circle divided by 2, polynomials in y up to y^m , where $m \leq 2n_c - 1$, are integrated exactly. The quadrature points are given by

$$y_i = \cos \omega_i, \omega_i = \frac{\pi}{2n_c} + i \frac{\pi}{n_c}, i = 0, \dots, 2n_c - 1. \quad (563)$$

The equal weights are

$$w_{\omega,i} = \frac{\pi}{n_c}, i = 0, \dots, 2n_c - 1. \quad (564)$$

Chebyshev quadrature weights sum to $\frac{\pi}{2}$ in one octant. The abscissas for order n_g Gaussian quadrature are roots of the Legendre polynomial $P_{n_g}(\mu)$. Its weights $w_{\mu,j}, j = 1, \dots, n_g$ can be evaluated with the abscissas and sum to 2.

Gauss-Chebyshev quadrature is a *production* quadrature with the polar and azimuthal angles being determined by a Gaussian and Chebyshev quadrature, respectively. The weight is $w_{\omega,i} w_{\mu,j}, i = 0, \dots, 2n_c - 1; j = 1, \dots, n_g$. The Chebyshev quadrature of order $N + 1$ can exactly integrate the azimuthal component of the product of any two spherical harmonics $Y_{l,m}$ and $Y_{n,k}$, where $l, n \leq N$. Since the integral over ω leads to δ_{mk} , the polar component of the product $P_{l,m} P_{n,m}$ is a polynomial of degree $l + n$. If $l, n \leq N$, $P_{l,m} P_{n,m}$ can be integrated exactly by a Gaussian quadrature of order $n_g = N + 1$. Thus the Gauss-Chebyshev quadrature can exactly integrate the product of any two spherical harmonics up to degree N if $n_g = n_c = N + 1$. We typically specify the numbers of directions per octant, i.e. $n_g/2$ and $n_c/2$ to construct the Gauss-Chebyshev quadrature. Because of the different angular resolutions in polar and azimuthal, n_g and n_c are typically set differently. It is noted that the polar direction is also set to 3-rd direction or z-direction because the heterogeneity in z-direction is typically significantly smaller than in the x-y plane and n_g can be chosen to be much smaller. Construction of the interpolation sphere functions is not available for Gauss-Chebyshev quadrature currently.

E.4 Bickley-3 Optimized

Yamamoto, et al. [101] presented a polar angular quadrature whose quadrature points and weights are obtained by minimizing the following function:

$$\max_{0 \leq x \leq \infty} |Ki_3(x) - \bar{K}i_3(x)|, \quad (565)$$

where

$$Ki_n(x) \equiv \int_0^{\frac{\pi}{2}} \sin^{n-1} \theta e^{-x/\sin \theta} d\theta = \int_0^1 \sin^{n-2} \theta e^{-x/\sin \theta} d\mu, \quad (566)$$

is the Bickley-Naylor function; μ is the cosine of the polar angle θ .

$$\bar{K}i_n(x) \equiv \sum_{m=1}^M w_m \sin^{n-2} \theta_m e^{-x/\sin \theta_m}, \quad (567)$$

is the approximation of the Bickley-Naylor function with an angular quadrature set $\{\mu_m, w_m, m = 1, \dots, M\}$, whose weights have the following constraint:

$$\sum_{m=1}^M w_m = 1. \quad (568)$$

They showed that the results with this polar quadrature sets combined with the azimuthal Chebyshev quadrature, termed as Bickly3-Optimized from now on, are more accurate than those with the Gauss-Chebyshev quadrature for two-dimensional problems with isotropic scatterings. Later, Wang [102] developed a Remez algorithm to evaluate the polar quadrature set up to order 10.

Polar quadrature sets for the Bickly3-Optimized quadrature are given in Table 55.

Table 55 Bickley3 optimized polar quadrature sets.

M	μ_m	w_m
1	6.024131420387014(-1) ¹⁰	1.0000000000000000(-0)

¹⁰a - read as $6.024131420387014 \times 10^{-1}$.

2	4.360958113484749(-1) 9.314378404484403(-1)	7.871457062207388(-1) 2.128542937792612(-1)
3	3.603647303112622(-1) 8.432845122259793(-1) 9.859174682982180(-1)	6.706042790023822(-1) 2.830561603802525(-1) 4.633956061736536(-2)
4	3.141875392954668(-1) 7.737261935507366(-1) 9.565700705377187(-1) 9.964908924759165(-1)	5.942539702959482(-1) 3.081802644914400(-1) 8.583055309397002(-2) 1.173521211864178(-2)
5	2.822123937782038(-1) 7.183367242066656(-1) 9.243890357401717(-1) 9.869210137447640(-1) 9.989805234100937(-1)	5.392298516673624(-1) 3.161569124601166(-1) 1.145258445609754(-1) 2.666114347502181(-2) 3.426247836523877(-3)
6	2.583733275426284(-1) 6.731541943779271(-1) 8.928184594761741(-1) 9.739501108108027(-1) 9.956751479051810(-1) 9.996671268418812(-1)	4.971386123696218(-1) 3.165413922838222(-1) 1.351413251200945(-1) 4.114408312220697(-2) 8.913999888857199(-3) 1.120587215397415(-3)
7	2.397120885176438(-1) 6.354694937374401(-1) 8.630154208969673(-1) 9.591269243416681(-1) 9.904912203462675(-1) 9.984467262851485(-1) 9.998810138141363(-1)	4.635909412993530(-1) 3.133163597564287(-1) 1.500090812844326(-1) 5.419762400157879(-2) 1.527035073134659(-2) 3.214840635447734(-3) 4.008022914124950(-4)
8	2.245880775349724(-1) 6.034457775633614(-1) 8.352991248138116(-1) 9.434126531176009(-1) 9.838604659459459(-1) 9.963154231626841(-1) 9.994020909797952(-1) 9.999542844306081(-1)	4.360367333581454(-1) 3.083426498170013(-1) 1.607860237675117(-1) 6.561305478313453(-2) 2.187028706967403(-2) 5.957654736248668(-3) 1.239567293457643(-3) 1.540291748268793(-4)
9	2.120076965583993(-1) 5.758072692682238(-1) 8.096644076737477(-1) 9.273952812402495(-1) 9.761684644230841(-1) 9.933431249206955(-1) 9.984927536729850(-1) 9.997561022316164(-1) 9.999813665750792(-1)	4.128790659026987(-1) 3.025568594322564(-1) 1.686133933860652(-1) 7.546385961182840(-2) 2.836709480974464(-2) 9.106738601716640(-3) 2.444203919514400(-3) 5.059965031046078(-4) 6.278783307103768(-5)
10	2.013290862332992(-1) 5.516424802433549(-1) 7.859829476099630(-1) 9.114327828719222(-1) 9.677252510878877(-1) 9.896402351776381(-1) 9.971339904004026(-1) 9.993534827244780(-1) 9.998955132558777(-1) 9.999920201513501(-1)	3.930599065541667(-1) 2.964536651014043(-1) 1.742812929119517(-1) 8.391482985532173(-2) 3.457727379475643(-2) 1.248193414958536(-2) 3.937595868490171(-3) 1.049773508736626(-3) 2.168378883868098(-4) 2.689036720028376(-5)

It is noted that the polar direction is also set to 3-rd direction or z-direction because the heterogeneity in z-direction is typically significantly smaller than in the x-y plane and n_g can be chosen to be much smaller.

F Multigroup Approximation

Integrate continuous-energy neutron transport equation over number G of energy intervals or groups $[E_g, E_{g-1}]$, $g = 1, \dots, G$, we obtain

$$\begin{aligned} \left(\frac{\partial}{\partial t} \frac{1}{v_g} + \vec{\Omega} \cdot \vec{\nabla} + \Sigma_{t,g} \right) \Psi_g = Q_{\text{ext},g} + \frac{1}{4\pi} \sum_{g'=1}^G \chi_{p,g,g'} (1 - \beta_{g'}) \nu \Sigma_{f,g'} \Phi_{g'} + \sum_{g'=1}^G \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \sum_{m=-\ell}^{\ell} \Sigma_{s,\ell,m}^{g' \rightarrow g} \Phi_{g',\ell,m} Y_{\ell,m}(\vec{\Omega}) \\ + \frac{1}{4\pi} \sum_{i=1}^I \chi_{d,g,i} \lambda_i C_i, \end{aligned} \quad (569a)$$

where

$$\Psi_g(\vec{x}, \vec{\Omega}, t) \equiv \int_{E_g}^{E_{g-1}} \Psi(\vec{x}, \vec{\Omega}, E, t) dE, \quad (569b)$$

$$\Phi_g(\vec{x}, t) \equiv \int_{E_g}^{E_{g-1}} \int_{4\pi} \Psi(\vec{x}, \vec{\Omega}, E, t) d\Omega dE, \quad (569c)$$

$$\Phi_{g,\ell,m}(\vec{x}, t) \equiv \int_{E_g}^{E_{g-1}} \int_{4\pi} \Psi(\vec{x}, \vec{\Omega}, E, t) Y_{\ell,m}(\vec{\Omega}) d\Omega dE, \quad (569d)$$

$$Q_{\text{ext},g}(\vec{x}, \vec{\Omega}, t) \equiv \int_{E_g}^{E_{g-1}} Q_{\text{ext}}(\vec{x}, \vec{\Omega}, E, t) dE, \quad (569e)$$

$$\frac{1}{v_g(\vec{x}, \vec{\Omega}, t)} \equiv \frac{\int_{E_g}^{E_{g-1}} \frac{1}{v(E)} \Psi(\vec{x}, \vec{\Omega}, E, t) dE}{\int_{E_g}^{E_{g-1}} \Psi(\vec{x}, \vec{\Omega}, E, t) dE}, \quad (569f)$$

$$\Sigma_{t,g}(\vec{x}, \vec{\Omega}, t) \equiv \frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\vec{x}, E) \Psi(\vec{x}, \vec{\Omega}, E, t) dE}{\int_{E_g}^{E_{g-1}} \Psi(\vec{x}, \vec{\Omega}, E, t) dE}, \quad (569g)$$

$$\nu \Sigma_{f,g}(\vec{x}, t) \equiv \frac{\int_{E_g}^{E_{g-1}} \nu \Sigma_f(\vec{x}, E) \Phi(\vec{x}, E, t) dE}{\int_{E_g}^{E_{g-1}} \Phi(\vec{x}, E, t) dE}, \quad (569h)$$

$$\beta_{i,g}(\vec{x}, t) \equiv \frac{\int_{E_g}^{E_{g-1}} \beta_i(\vec{x}, E) \nu \Sigma_f(\vec{x}, E) \Phi(\vec{x}, E, t) dE}{\int_{E_g}^{E_{g-1}} \nu \Sigma_f(\vec{x}, E) \Phi(\vec{x}, E, t) dE}, \quad (569i)$$

$$\beta_g(\vec{x}, t) = \sum_{i=0}^I \beta_{i,g}(\vec{x}, t), \quad (569j)$$

$$\chi_{p,g,g'}(\vec{x}, t) \equiv \frac{\int_{E_g}^{E_{g-1}} \int_{E_{g'}}^{E_{g'-1}} \chi_p(\vec{x}, E, E') (1 - \beta(\vec{x}, E')) \nu \Sigma_f(E') \Phi(\vec{x}, E', t) dE' dE}{\int_{E_{g'}}^{E_{g'-1}} (1 - \beta(\vec{x}, E')) \nu \Sigma_f(\vec{x}, E') \Phi(\vec{x}, E', t) dE'}, \quad (569k)$$

$$\Sigma_{s,\ell,m}^{g' \rightarrow g}(\vec{x}, t) \equiv \frac{\int_{E_{g'}}^{E_{g'-1}} \left[\int_{E_g}^{E_{g-1}} \Sigma_{s,\ell}(\vec{x}, E' \rightarrow E) dE \right] \Phi_{\ell,m}(\vec{x}, E', t) dE'}{\int_{E_{g'}}^{E_{g'-1}} \Phi_{\ell,m}(\vec{x}, E', t) dE'}, \quad (569l)$$

and

$$\chi_{d,g,i}(\vec{x}) \equiv \int_{E_g}^{E_{g-1}} \chi_{d,i}(\vec{x}, E) dE. \quad (569m)$$

The delayed neutron precursor equation becomes

$$\frac{\partial C_i}{\partial t} = \sum_{g'=1}^G \beta_{i,g'} \nu \Sigma_{f,g'} \Phi_{g'} - \lambda_i C_i, \quad i = 1, \dots, I. \quad (569n)$$

We often generate the prompt fission spectrum χ_p without the g' dependency and delayed neutron precursor fraction β_i without the g dependency with

$$\beta_i(\vec{x}, t) \equiv \frac{\sum_{g=1}^G \int_{E_g}^{E_{g-1}} \beta_i(\vec{x}, E) \nu \Sigma_f(\vec{x}, E) \Phi(\vec{x}, E, t) dE}{\sum_{g=1}^G \int_{E_g}^{E_{g-1}} \nu \Sigma_f(\vec{x}, E) \Phi(\vec{x}, E, t) dE} = \frac{\sum_{g=1}^G \beta_{i,g} \nu \Sigma_{f,g} \Phi_g}{\sum_{g=1}^G \nu \Sigma_{f,g} \Phi_g}, \quad (569o)$$

$$\begin{aligned} \chi_{p,g}(\vec{x}, t) &\equiv \frac{\int_{E_g}^{E_{g-1}} \sum_{g'=1}^G \int_{E_{g'}}^{E_{g'-1}} \chi_p(\vec{x}, E, E') (1 - \beta(\vec{x}, E')) \nu \Sigma_f(E') \Phi(\vec{x}, E', t) dE' dE}{\sum_{g'=1}^G \int_{E_{g'}}^{E_{g'-1}} (1 - \beta(\vec{x}, E')) \nu \Sigma_f(\vec{x}, E') \Phi(\vec{x}, E', t) dE'} \\ &= \frac{\sum_{g'=1}^G \chi_{p,g,g'} (1 - \beta_{g'}(\vec{x}, t)) \nu \Sigma_{f,g'} \Phi_{g'}}{(1 - \beta) \sum_{g=1}^G \nu \Sigma_{f,g} \Phi_g}. \end{aligned} \quad (569p)$$

Thus

$$\frac{1}{4\pi} \sum_{g'=1}^G \chi_{p,g,g'} (1 - \beta_{g'}) \nu \Sigma_{f,g'} \Phi_{g'} = \frac{1}{4\pi} \chi_{p,g} (1 - \beta) \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'}. \quad (569q)$$

By definition

$$\sum_{g=1}^G \chi_{p,g,g'}(\vec{x}, t) \equiv \sum_{g=1}^G \chi_{p,g}(\vec{x}, t) \equiv 1. \quad (570)$$

Note that the group-averaged total cross sections is angular dependent. We can apply similar spherical harmonics expansion to the total collision term,

$$\begin{aligned} &\int_{E_g}^{E_{g-1}} \Sigma_t(\vec{x}, E) \Psi(\vec{x}, \vec{\Omega}, E, t) dE \\ &= \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \sum_{m=-\ell}^{\ell} \int_{E_g}^{E_{g-1}} \Sigma_t(\vec{x}, E) \Phi_{\ell,m}(\vec{x}, E, t) Y_{\ell,m}(\vec{\Omega}) dE \\ &= \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \sum_{m=-\ell}^{\ell} \Sigma_{t,g,\ell,m}(\vec{x}, t) \Phi_{g,\ell,m}(\vec{x}, t) Y_{\ell,m}(\vec{\Omega}), \end{aligned}$$

where

$$\Sigma_{t,g,\ell,m}(\vec{x}, t) \equiv \frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\vec{x}, E) \Phi_{\ell,m}(\vec{x}, E, t) dE}{\int_{E_g}^{E_{g-1}} \Phi_{\ell,m}(\vec{x}, E, t) dE}. \quad (571a)$$

Note that it is possible $\Phi_{g,\ell,m}(\vec{x}, t)$ is equal to zero. In such a case, $\Sigma_{t,g,\ell,m}(\vec{x}, t)$ is not defined and not needed. We can re-define

$$\Sigma_{t,g}(\vec{x}, t) \equiv \Sigma_{t,g,0,0}(\vec{x}, t) \quad (571b)$$

and modify the in-group scattering cross section with

$$\Sigma_{s,\ell,m}^{g' \rightarrow g}(\vec{x}, t) \leftarrow \Sigma_{s,\ell,m}^{g' \rightarrow g}(\vec{x}, t) + (\Sigma_{t,g}(\vec{x}, t) - \Sigma_{t,g,\ell,m}(\vec{x}, t)) \delta_{\ell,0} \delta_{m,0} \delta_{g,g'}. \quad (571c)$$

Although the group averaged velocity has the angular dependency, we can assume such a dependency is weak and the averaged velocity can be redefined as

$$\frac{1}{v_g(\vec{x}, \vec{\Omega}, t)} \equiv \frac{\int_{E_g}^{E_{g-1}} \frac{1}{v(E)} \Phi(\vec{x}, E, t) dE}{\int_{E_g}^{E_{g-1}} \Phi(\vec{x}, E, t) dE}. \quad (571d)$$

The above equations state the exact projection operation without homogenization in order that the multigroup equation preserves the solution of the energy-dependent transport equation. Quantities including v_g , $\Sigma_{t,g}$, $\nu\Sigma_{f,g}$, $\chi_{p,g}$, $\Sigma_{s,\ell,m}^{g' \rightarrow g}$, $\chi_{d,g,i}$ are called group constants.

The above equations also mean that we can obtain the multigroup cross sections exactly only after we know the continuous transport solution. The spectrum can be obtained by doing ultra-fine group calculations with proper resonance treatment with some small reference configurations. We assume these reference configurations can produce the spectrum sufficiently accurate for the core level simulations.

Or we can introduce the multigroup approximation:

$$\Psi(\vec{x}, \vec{\Omega}, E, t) = \hat{\Psi}(\vec{x}, \vec{\Omega}, t) f_r(E), \quad (572)$$

so

$$\Phi(\vec{x}, E, t) = \hat{\Phi}(\vec{x}, t) f_r(E), \quad (573)$$

where r is the region index under consideration. $f_r(E)$ is called as the neutron spectrum of a region r . It is noted that this approximation is consistent for energy refinement, i.e. the approximation diminishes when the energy resolution is increased. With this approximation,

$$\Sigma_{t,g}(\vec{x}) \equiv \frac{\int_{E_g}^{E_{g-1}} \Sigma_t(\vec{x}, E) f_r(E) dE}{\int_{E_g}^{E_{g-1}} f_r(E) dE}, \quad (574a)$$

$$\nu\Sigma_{f,g}(\vec{x}) \equiv \frac{\int_{E_g}^{E_{g-1}} \nu\Sigma_f(\vec{x}, E) f_r(E) dE}{\int_{E_g}^{E_{g-1}} f_r(E) dE}, \quad (574b)$$

$$\Sigma_{s,\ell}^{g' \rightarrow g}(\vec{x}) \equiv \frac{\int_{E_{g'}}^{E_{g'-1}} \left[\int_{E_g}^{E_{g-1}} \Sigma_{s,\ell}(\vec{x}, E' \rightarrow E) dE \right] f_r(E') dE'}{\int_{E_{g'}}^{E_{g'-1}} f_r(E') dE'}, \quad (574c)$$

$$\frac{1}{v_g(\vec{x})} \equiv \frac{\int_{E_g}^{E_{g-1}} \frac{1}{v(E)} f_r(E) dE}{\int_{E_g}^{E_{g-1}} f_r(E) dE}. \quad (574d)$$

It is noted that this approximation makes $\Sigma_{t,g,\ell,m} = \Sigma_{t,g,m} = -\ell, \dots, \ell; \ell = 0, \dots, \infty$ and the correction on the in-group scattering is zero. We expect when the regions covering the domain are getting smaller and energy groups are becoming thinner, the error caused by this approximation tends to be zero. The accuracy of the multigroup approximation depends on how the full range of energy is cut and number of regions are considered and also depends on how well the spectrum can be obtained.

Comparing with the exact projection,

- the scattering cross section is now only ℓ -dependent;
- the averaged neutron velocity is angular independent;
- all group constants have no time dependency.

G Black Body Emission

Black body emission is a major physics of thermal radiation transport. It is isotropic; depends on temperature and wavelength; has the maximum energy emission. A black body absorbs all incident radiation, regardless of wavelength and direction. Black body emission follows the *Planck distribution* bounded by a transparent medium with refractive index n :

$$B(\lambda, T) = \frac{2\hbar c^2}{\lambda^5 \left(e^{\frac{\hbar c}{\lambda k T}} - 1 \right)} = \frac{C_1}{\pi n^2 \lambda^5 \left(e^{\frac{C_2}{n \lambda T}} - 1 \right)}, \quad (575)$$

where $C_1 = 2\pi\hbar c^2$ and $C_2 = \frac{\hbar c_0}{k}$. \hbar is the Planck constant; k is the Boltzmann constant; c is the speed of light in the surrounding medium and λ is the wavelength. It is noted that n is the refractive index of the surrounding medium and

$$c = \frac{c_0}{n}. \quad (576)$$

Or

$$B(\nu, T) = \frac{2}{\hbar^2 c^2} \frac{(\hbar \nu)^3}{e^{\frac{\hbar \nu}{k T}} - 1}, \quad (577)$$

where ν is the frequency. The unit is $\frac{J}{m^2 \cdot sr}$. with the assumption of constant refractive index.

We will define

$$\mathcal{B} \equiv \frac{1}{c} \int_{4\pi} B(\nu, T) d\Omega = \frac{4\pi}{c} B \quad (578)$$

for later use which has the unit of $\frac{J \cdot s}{m^3}$.

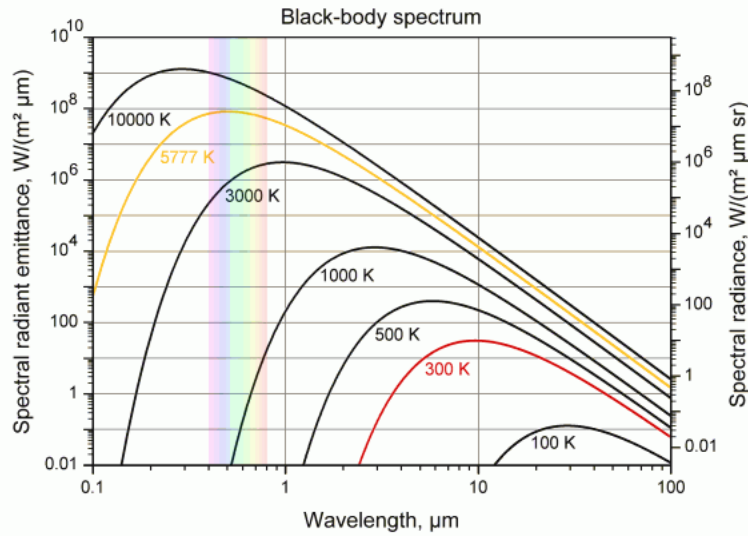


Figure 14 Blackbody spectrum.

Certain properties can be observed:

- *Wien's Displacement Law* If we let the derivative of the Planckian distribution with respect to λ equal to zero (the transcendental equation),

$$\left(1 - \frac{C_2}{5n\lambda T} \right) e^{\frac{C_2}{n\lambda T}} = 1, \quad (579)$$

We obtain the solution

$$n\lambda_{max}T = C_3, \quad (580)$$

where C_3 is known as the third radiation constant, equal to $2897.6 \mu m \cdot K$.

- We also often care the following integral,

$$\int_0^\infty B(\nu, T) d\nu = \frac{1}{4\pi} c_0 a n^2 T^4, \quad (581)$$

where a is the radiation constant $8\pi^5 k^4 / (15\hbar^3 c_0^3)$.

- The *Stefan-Boltzmann Law* The total emissive power in the hemi-sphere

$$E_b = \int_0^\infty \int_{\vec{\Omega} \cdot \vec{n} > 0} |\vec{\Omega} \cdot \vec{n}| B(\nu, T) d\Omega d\nu = \pi \int_0^\infty B(\nu, T) d\nu = n^2 \sigma T^4, \quad (582)$$

where the Stefan-Boltzmann constant, which depends on C_1 and C_2 , is equal to $\frac{\pi^4 C_1}{15 C_2^4} = 5.670 \times 10^{-8} \frac{W}{m^2 \cdot K^4}$. It is noted that the total emissive power is proportional to n^2 . It is seen that

$$\sigma = \frac{1}{4} c_0 a \quad (583)$$

- It is convenient to table the band emission with the following definition:

$$\begin{aligned} \frac{\int_0^\lambda B(\lambda, T) d\lambda}{\int_0^\infty B(\lambda, T) d\lambda} &= \frac{\int_0^\lambda B(\lambda, T) d\lambda}{\frac{1}{4\pi} c_0 a n^2 T^4} = \int_0^{n\lambda T} \frac{4\pi B(\lambda, T)}{c_0 a n^3 T^5} d(n\lambda T) = \int_0^{n\lambda T} \frac{4C_1}{c_0 a (n\lambda T)^5 \left(e^{\frac{C_2}{n\lambda T}} - 1 \right)} d(n\lambda T) \\ &= \int_0^{x=n\lambda T} \frac{15 \left(\frac{C_2}{\pi} \right)^4}{x^5 \left(e^{\frac{C_2}{x}} - 1 \right)} dx \\ &\equiv F(n\lambda T) \end{aligned} \quad (584)$$

It is noted that if we use the frequency for the tabulation, the dependency on the refractive index is gone.

- The asymptotic behavior of Planck's law for large wavelengths: The exponent may be approximated by a two-term Taylor series, leading to

$$B(\lambda, T) \approx \frac{2\pi c_0 k T}{n\lambda^4}, \quad \text{for } \frac{C_2}{n\lambda T} \ll 1, \quad (585)$$

which can be obtained by letting $\hbar \rightarrow 0$, i.e., by allowing photons of arbitrarily small energy content to be emitted, as postulated by classical statistics. The above equation is the Rayleigh-Jeans distribution. It is noted that the equation is accurate only for very large values of $n\lambda T$, so it is of little significance of engineering purpose.

The asymptotic behavior of Planck's law for small wavelengths (neglecting the -1 in the denominator:

$$B(\lambda, T) \approx \frac{C_1}{n^2 \lambda^5} e^{-\frac{C_2}{n\lambda T}}, \quad \text{for } \frac{C_2}{n\lambda T} \gg 1, \quad (586)$$

which is named after Wien. This equation is very accurate over most of the spectrum, thus it is frequently utilized in theoretical analysis in order to facilitate integration.

H The Reflection and Refraction on a Medium Interface

We always assume the unit norm vector of the interface \vec{n} is pointing from medium 1 who has larger refractive index n_1 , to the medium 2 whose refractive index is denoted with n_2 . θ_i is the angle formed by the incident direction $\vec{\Omega}$ and the interface norm \vec{n} . θ_t is the angle formed by the refraction direction $\vec{\Omega}_t$ and the interface norm \vec{n} . θ_r is the angle formed by the reflection direction $\vec{\Omega}_r$ and the negative of the interface norm $-\vec{n}$.

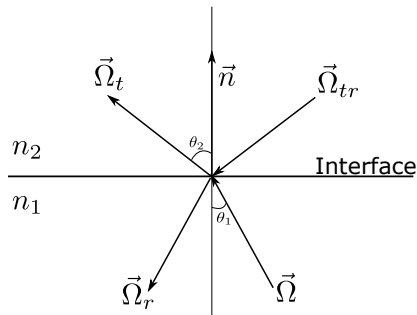


Figure 15 Reflection and refraction.

Specular reflection direction is

$$\vec{\Omega}_r = \vec{\Omega} - 2(\vec{\Omega} \cdot \vec{n})\vec{n}. \quad (587)$$

Based on the Snell's law

$$n_1 \sin \theta_1 = n_2 \sin \theta_2 \quad (588)$$

or

$$n_1 \sqrt{1 - \mu_1^2} = n_2 \sqrt{1 - \mu_2^2}, \quad (589)$$

where $\mu_1 = \vec{\Omega} \cdot \vec{n}$ and $\mu_2 = \vec{\Omega}_2 \cdot \vec{n}$. There is a map from μ_1 to μ_2 , defined on $[\mu_c, 1] \Rightarrow [0, 1]$, where

$$\mu_c = \frac{1}{n_1} \sqrt{n_1^2 - n_2^2} \quad (590)$$

is the cosine of the critical angle. Directly from the Snell's law, we have

$$n_1 \mu_1 d\theta_1 = n_2 \mu_2 d\theta_2. \quad (591)$$

The refraction direction is

$$\vec{\Omega}_2 = \sqrt{1 - |\vec{\Omega}_\perp|^2} \vec{n} + \vec{\Omega}_\perp, \quad (592)$$

where

$$\vec{\Omega}_\perp = \frac{n_1}{n_2} \left[\vec{\Omega} - (\vec{\Omega} \cdot \vec{n}) \vec{n} \right]. \quad (593)$$

The reflectance between two perfect dielectrics can be evaluated with the Fresnel's relation. In such a case, the absorptive index k needs to be introduced.

$$n^2 - k^2 = \frac{\epsilon}{\epsilon_0} = \epsilon_r, \quad (594)$$

where ϵ is the electrical permittivity and ϵ_0 is the absolute dielectric permittivity of classical vacuum ($8.854187817 \times 10^{-12}$ farads per meter).

$$n^2 = \epsilon_r \mu_r, \quad (595)$$

where μ_r is the relative permeability. For most naturally occurring materials, μ_r is very close to one at optical frequencies, therefore the absorptive index k is close to zero.

The reflectance for s-polarized light is

$$R_s^{1 \rightarrow 2}(\mu_1) = \left| \frac{n_1 \mu_1 - n_2 \mu_2}{n_1 \mu_1 + n_2 \mu_2} \right|^2. \quad (596)$$

The transmittivity

$$T_s^{1 \rightarrow 2}(\mu_1) = 1 - R_s^{1 \rightarrow 2}(\mu_1) = \frac{4n_1 \mu_1 n_2 \mu_2}{(n_1 \mu_1 + n_2 \mu_2)^2}. \quad (597)$$

For p-polarized light the reflectance is

$$R_p^{1 \rightarrow 2}(\mu_1) = \left| \frac{n_1 \mu_2 - n_2 \mu_1}{n_1 \mu_2 + n_2 \mu_1} \right|^2. \quad (598)$$

The transmittivity

$$T_p^{1 \rightarrow 2}(\mu_1) = 1 - R_p^{1 \rightarrow 2}(\mu_1) = \frac{4n_1 \mu_2 n_2 \mu_1}{(n_1 \mu_2 + n_2 \mu_1)^2}. \quad (599)$$

The reflectance for unpolarized light is

$$R^{1 \rightarrow 2}(\mu_1) = \frac{R_s^{1 \rightarrow 2}(\mu_1) + R_p^{1 \rightarrow 2}(\mu_1)}{2}. \quad (600)$$

When μ_2 is zero, i.e. the incident direction is aligned with the critical angle $\mu_1 = \mu_c$, the reflectance is one.

For all the directions whose $\mu_1 < \mu_c$ or $\theta_1 > \theta_c$, total reflection will happen. When $\mu_1 = \mu_2 = 1$, the reflectance is $\frac{n_1 - n_2}{n_1 + n_2}$. It is noted that the reflectance evaluated with the opposite direction follows

$$R^{2 \rightarrow 1}(\mu_2) = R^{1 \rightarrow 2}(\mu_1), \quad (601)$$

$$R_s^{2 \rightarrow 1}(\mu_2) = R_s^{1 \rightarrow 2}(\mu_1), \quad (602)$$

$$R_p^{2 \rightarrow 1}(\mu_2) = R_p^{1 \rightarrow 2}(\mu_1). \quad (603)$$

Similar rules apply to the transmittivity equations.

When the equations apply to the specific intensity, we need to take the solid angle change into account.

$$d\Omega_1 = \sin \theta_1 d\theta_1 d\varphi \quad (604)$$

$$d\Omega_2 = \sin \theta_2 d\theta_2 d\varphi = \frac{n_1}{n_2} \sin \theta_1 d\theta_2 d\varphi = \frac{n_1}{n_2} \sin \theta_1 \frac{n_1 \mu_1}{n_2 \mu_2} d\theta_1 d\varphi = \frac{n_1^2}{n_2^2} \frac{\mu_1}{\mu_2} d\Omega_1 \quad (605)$$

We can see

$$\mu_2 d\Omega_2 = \sin \theta_2 d\theta_2 d\varphi = \frac{n_1}{n_2} \sin \theta_1 d\theta_2 d\varphi = \frac{n_1}{n_2} \sin \theta_1 \frac{n_1 \mu_1}{n_2 \mu_2} d\theta_1 d\varphi = \frac{n_1^2}{n_2^2} \mu_1 d\Omega_1 \quad (606)$$

The energy conservation would be

$$I_1(\mu_1) \mu_1 d\Omega_1 dA = R^{1 \rightarrow 2}(\mu_1) I_1(\mu_1) \mu_1 d\Omega_1 dA + I_2(\mu_2) \mu_2 d\Omega_2 dA,$$

which means the transmitted intensity should be

$$I_2(\mu_2) = \frac{n_2^2}{n_1^2}(1 - R^{1 \rightarrow 2}(\mu_1))I_1(\mu_1). \quad (607)$$

An extra coefficient $\frac{n_2^2}{n_1^2}$ needs to be multiplied with the transmitted intensity. This is also known as the *radiance conservation theorem*. Based on this theorem, the interface condition for specific intensity will be

$$I_1(\vec{\Omega}) = R^{1 \rightarrow 2}(|\vec{\Omega} \cdot \vec{n}|)I_1(\vec{\Omega}_r) + (1 - R^{1 \rightarrow 2}(|\vec{\Omega} \cdot \vec{n}|))\frac{n_1^2}{n_2^2}I_2(\vec{\Omega}_t), \vec{\Omega} \cdot \vec{n} < 0 \quad (608a)$$

$$I_2(\vec{\Omega}_t) = R^{1 \rightarrow 2}(|\vec{\Omega} \cdot \vec{n}|)I_2(\vec{\Omega}_{tr}) + (1 - R^{1 \rightarrow 2}(|\vec{\Omega} \cdot \vec{n}|))\frac{n_2^2}{n_1^2}I_1(\vec{\Omega}), \vec{\Omega}_t \cdot \vec{n} > 0 \quad (608b)$$

$\vec{\Omega}_{tr}$ is the reflecting direction of $\vec{\Omega}_t$ with respect to the surface with normal \vec{n} . When $n_1 = n_2$, these conditions reduce to the normal continuity condition in neutron transport.

We can perform the integral

$$\frac{1}{\pi} \int_{\vec{\Omega}_2 \cdot \vec{n} > 0} T^{1 \rightarrow 2}(\vec{\Omega}_1 \cdot \vec{n}) |\vec{\Omega}_2 \cdot \vec{n}| d\Omega_2 = 2 \int_0^1 T^{1 \rightarrow 2}(\mu_1) \mu_2 d\mu_2 = 2 \int_{\mu_c}^1 T^{1 \rightarrow 2}(\mu_1) \frac{n_1^2}{n_2^2} \mu_1 d\mu_1 \quad (609)$$

to obtain the spectral hemispherical transmissivity. The results are

$$\epsilon_s = \frac{4(2n+1)}{3(n+1)^2}, \quad (610)$$

$$\epsilon_p = \frac{4n^3(n^2+2n-1)}{(n^2+1)(n^4-1)} + \frac{2n^2(n^2-1)^2}{(n^2+1)^3} \ln \left(\frac{n+1}{n-1} \right) - \frac{16n^4(n^4+1) \ln n}{(n^2+1)(n^4-1)^2}, \quad (611)$$

$$\epsilon = \frac{\epsilon_s + \epsilon_p}{2} \quad (612)$$

These equations can be used to predict the surface emissivity, which by Kirchhoff's law the emissivity is equal to the transmissivity.

I Rattlesnake Publications

This appendix collects all publications related with Rattlesnake.

Journal articles

- [1] Vincent Laboure, Ryan G. McClarren, Yaqi Wang, “Globally Conservative, Hybrid Self-Adjoint Angular Flux and Least-Squares Method Compatible with Void.” In: *Nuclear Science and Engineering* (2017).
- [2] Sebastian Schunert, Yaqi Wang, Frederick Gleicher, Javier Ortensi, Benjamin Baker, Vincent Laboure, Congjian Wang, Mark DeHart, Richard Martineau, “A flexible nonlinear diffusion acceleration method for the {SN} transport equations discretized with discontinuous finite elements”. In: *Journal of Computational Physics* 338 (2017), pp. 107–136. ISSN: 0021-9991. DOI: [10.1016/j.jcp.2017.01.070](https://doi.org/10.1016/j.jcp.2017.01.070). URL: <http://www.sciencedirect.com/science/article/pii/S0021999117301286>.
- [3] Yaqi Wang, Sebastian Schunert, Mark DeHart, Richard Martineau, Weixiong Zheng, “Hybrid PN - SN with Lagrange multiplier and upwinding for the multiscale transport capability in Rattlesnake”. In: *Progress in Nuclear Energy* (Apr. 2017). DOI: [10.1016/j.pnucene.2017.03.020](https://doi.org/10.1016/j.pnucene.2017.03.020).
- [4] Derek R. Gaston, Cody J. Permann, John W. Peterson, Andrew E. Slaughter, David Andrs, Yaqi Wang, Michael P. Short, Danielle M. Perez, Michael R. Tonks, Javier Ortensi, Ling Zou, Richard C. Martineau, “Physics-based multiscale coupling for full core nuclear reactor simulation”. In: *Annals of Nuclear Energy* 84 (Oct. 2015), pp. 45–54. DOI: [10.1016/j.anucene.2014.09.060](https://doi.org/10.1016/j.anucene.2014.09.060).
- [5] Sebastian Schunert, Yaqi Wang, Richard Martineau, Mark D. DeHart, “A new mathematical adjoint for the modified SAAF-SN equations”. In: *Annals of Nuclear Energy* 75 (2015), pp. 340–352.
- [6] Jon Hansen, Jacob Peterson, Jim Morel, Jean Ragusa, Yaqi Wang, “A Least-Squares Transport Equation Compatible with Voids”. In: *Journal of Computational and Theoretical Transport Special Issue: Papers from the 23rd International Conference on Transport Theory* 43.1–7 (2014).
- [7] Yaqi Wang, Hongbin Zhang, Richard Martineau, “Diffusion Acceleration Schemes for the Self-Adjoint Angular Flux Formulation with a Void Treatment”. In: *Nuclear Science and Engineering* 176 (2014), pp. 201–225.

ANS summaries

- [8] Vincent Laboure, Yaqi Wang, Sebastian Schunert, “Direct S_N treatment of the Scattering Source with Phase Functions”. In: *Transactions of American Nuclear Society* (2017).
- [9] Sebastian Schunert, Hans Hammer, Jijie Lou, Yaqi Wang, Javier Ortensi, Frederick Gleicher, Benjamin Baker, Mark DeHart, Richard Martineau, “Using Directional Diffusion Coefficients for Nonlinear Diffusion Acceleration of the First Order SN Equations in Near-Void Regions”. In: *Transactions of the Am. Nucl. Soc. ANS. Las Vegas, NV*, 2016.
- [10] Sebastian Schunert, Yaqi Wang, Javier Ortensi, Frederick N. Gleicher, Benjamin A. Baker, Mark D. DeHart, Richard C. Martineau, “A Flexible Nonlinear Diffusion Acceleration Method for the First Order SN Equations”. In: *PHYSOR 2016: Unifying Theory and Experiments in the 21st Century*. Sun Valley, ID, USA: American Nuclear Society, May 1, 2016.
- [11] Sebastian Schunert, Yaqi Wang, Javier Ortensi, Frederick N. Gleicher, Benjamin A. Baker, Mark D. DeHart, Richard C. Martineau, “On the Degradation of the Effectiveness of Nonlinear Diffusion Acceleration with Parallel Block Jacobi Splitting”. In: *Nuclear Power: Leading the Supply of Clean, Carbon Free Energy, 2016 ANS Annual Meeting*. New Orleans, LA, USA: American Nuclear Society, 2016.
- [12] Yaqi Wang, Sebastian Schunert, Javier Ortensi, Frederick Gleicher, Benjamin Baker, Gilles Youinou, Mark DeHart, Richard Martineau, “Derivation of the Point-Kinetics Parameters for the Coupled Reactors”. In: *Transactions of American Nuclear Society* (2016).

- [13] Sebastian Schunert, Yaqi Wang, Javier Ortensi, Frederick Gleicher, Mark DeHart, Richard Martineau, “A High-Order Nonlinear Diffusion Acceleration for the SN Equations Discretized with the Discontinuous FEM II: Fourier Analysis”. In: *Transactions of the American Nuclear Society* 113 (2015). Accepted.
- [14] Sebastian Schunert, Yaqi Wang, Javier Ortensi, Frederick Gleicher, Mark DeHart, Richard Martineau, “A High-Order Nonlinear Diffusion Acceleration for the SN Equations Discretized with the Discontinuous FEM I: Theory and Numerical Results”. In: *Transactions of the American Nuclear Society* 113 (2015). Accepted.
- [15] Yaqi Wang, “A Remez Algorithm on Finding Polar Angle Quadrature Sets for the Method of Characteristics”. In: *Transactions of the American Nuclear Society* 112 (2015), pp. 413–415.
- [16] Hongbin Zhang, Yaqi Wang, Ling Zou, David Andrs, Haihua Zhao, Richard C. Martineau, “Coupling of RELAP-7 with the Three-Dimensional Kinetics Code RattleSnake”. In: *Transactions of American Nuclear Society* 108 (1) (2013), pp. 863–865.
- [17] Frederick N. Gleicher, Yaqi Wang, Derek Gaston, Richard Martineau, “The Method of Manufactured Solutions for RattleSnake, A S_N Radiation Transport Solver Inside the MOOSE Framework”. In: *Transactions of American Nuclear Society* 106:1 (2012), pp. 372–374.

M&C papers

- [18] Hans Hammer, Jim E. Morel, Yaqi Wang, “Nonlinear Diffusion Acceleration in Voids for the Weighted Least-Square Transport Equation”. In: *M&C 2017, International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*. ANS. Jeju, Korea, 2017.
- [19] Vincent Laboure, Ryan G. McClarren, “Energy-Dependent Implicit Filtered P_N Method for Thermal Radiation Transport”. In: *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2017)*. American Nuclear Society (ANS). Jeju, Korea, Apr. 2017.
- [20] Zachary M. Prince, Jean C. Ragusa, “Multiphysics Core Dynamics Simulation Using the Improved Quasi-Static Method”. In: *M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*. Jeju, Korea, Apr. 2017.
- [21] Jacob R. Peterson, Hans R. Hammer, Jim E. Morel, Jean C. Ragusa, Yaqi Wang, “Conservative Nonlinear Diffusion Acceleration Applied to the Unweighted Least-Squares Transport Equation in MOOSE”. In: *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*. ANS. Nashville, Tennessee, 2015.
- [22] Yaqi Wang, Mark D. DeHart, Derek R. Gaston, Frederick N. Gleicher, Richard C. Martineau, Javier Ortensi, John W. Peterson, Sebastian Schunert, “Convergence study of Rattlesnake solutions for the two-dimensional C5G7 MOX benchmark”. In: *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*. Nashville, Tennessee, Apr. 2015.
- [23] Yaqi Wang, “Nonlinear Diffusion Acceleration for the Multigroup Transport Equation Discretized with S_N and Continuous FEM with Rattlesnake”. In: *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. Sun Valley, Idaho, May 2013, pp. 2648–2665.

PHYSOR papers

- [24] Vincent Laboure, Yaqi Wang, Mark D. DeHart, “Least-Squares P_N Formulation of the Transport Equation Using Self-Adjoint-Angular-Flux Consistent Boundary Conditions.” In: *PHYSOR 2016 - Unifying Theory and Experiments in the 21st Century*. ANS. Sun Valley, ID, May 2016.
- [25] Yaqi Wang, Sebastian Schunert, Mark D. DeHart, Richard C. Martineau, “Hybrid PN-SN Calculations with SAAF for the Multiscale Transport Capability in Rattlesnake”. In: *PHYSOR 2016 - Unifying Theory and Experiments in the 21st Century*. Sun Valley, Idaho, USA, May 2016.
- [26] Yaqi Wang, Frederick N. Gleicher, “Revisit Boundary Conditions for the Self-Adjoint Angular Flux Formulation”. In: *PHYSOR 2014 - The Role of Reactor Physics toward a Sustainable Future*. The Westin Miyako, Kyoto, Japan, Sept. 2014.

Reports

- [27] Yaqi Wang, Javier Ortensi, *YAKXS - The XML Multigroup Cross Section Library*. INL. 2016.

References

- [1] Derek Gaston, Chris Newman, Glen Hansen, and Damien Lebrun-Grandié. “MOOSE: A Parallel Computational Framework for Coupled Systems of Nonlinear Equations”. In: *Nuclear Engineering and Design* 239.10 (2009), pp. 1768–1778.
- [2] Derek R. Gaston, Cody J. Permann, John W. Peterson, Andrew E. Slaughter, David Andrs, Yaqi Wang, Michael P. Short, Danielle M. Perez, Michael R. Tonks, Javier Ortensi, Ling Zou, and Richard C. Martineau. “Physics-based multiscale coupling for full core nuclear reactor simulation”. In: *Annals of Nuclear Energy* 84 (2015), pp. 45–54. DOI: [10.1016/j.anucene.2014.09.060](https://doi.org/10.1016/j.anucene.2014.09.060).
- [3] E. E. Lewis and W. F. Miller. *Computational Methods of Neutron Transport*. John Wiley & Sons, 1984.
- [4] James J. Duderstadt and Louis J. Hamilton. *Nuclear Reactor Analysis*. New York: John Wiley & Sons, Inc., 1976.
- [5] G. J. Bell and S. Glasstone. *Nuclear Reactor Theory*. New York: Van Nostrand Reinhold Company, 1970.
- [6] Michael F. Modest. *Radiative Heat Transfer*. Elsevier LTD, Oxford, Apr. 11, 2013. ISBN: 0123869447. URL: http://www.ebook.de/de/product/19724908/michael_f_modest_radiative_heat_transfer.html.
- [7] D. S. Anikonov, A. E. Kovtanyuk, and Iu. V. Prokhorov. *Transport Equation and Tomography*. New York: Brill Academic Publishers, 2002. ISBN: 90-6764-354-8.
- [8] K. N. Liou. *An Introduction to Atmospheric Radiation*. 2nd. New York: Academic Press, 2002.
- [9] John I. Castor. *Radiation Hydrodynamics*. CAMBRIDGE UNIV PR, Sept. 11, 2006. 368 pp. ISBN: 0521540623. URL: http://www.ebook.de/de/product/6212488/john_i_castor_radiation_hydrodynamics.html.
- [10] Dimitri Mihalas and Barbara Weibel Mihalas. *Foundations of Radiation Hydrodynamics*. DOVER PUBN INC, July 11, 1999. 752 pp. ISBN: 0486409252. URL: http://www.ebook.de/de/product/3334110/dimitri_mihalas_barbara_weibel_mihalas_foundations_of_radiation_hydrodynamics.html.
- [11] Gerald C. Pomraning. *The Equations of Radiation Hydrodynamics*. Oxford, 1973.
- [12] Ivan Lux and Laszlo Koblinger. *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*. CRC Press, 1990.
- [13] T.L. Becker, A.B. Wollaber, and E.W. Larsen. “A Hybrid Monte Carlo - Deterministic Method for Global Particle Transport Calculations”. In: *Nuclear Science and Engineering* 155 (2007), pp. 155–167.
- [14] R.J. Howerton, R.E. Dye, and S.T. Perkins. *Evaluated Nuclear Data Library*. Tech. rep. UCRL-50400. Lawrence Livermore National Laboratory, 1981.
- [15] M.B. Chadwick, M. Herman, P. Obložinský, M.E. Dunn, Y. Danon, A.C. Kahler, D.L. Smith, B. Pritychenko, G. Arbanas, R. Arcilla, R. Brewer, D.A. Brown, R. Capote, A.D. Carlson, Y.S. Cho, H. Derrien, K. Guber, G.M. Hale, S. Hoblit, S. Holloway, T.D. Johnson, T. Kawano, B.C. Kiedrowski, H. Kim, S. Kunieda, N.M. Larson, L. Leal, J.P. Lestone, R.C. Little, E.A. McCutchan, R.E. MacFarlane, M. MacInnes, C.M. Mattoon, R.D. McKnight, S.F. Mughabghab, G.P.A. Nobre, G. Palmiotti, A. Palumbo, M.T. Pigni, V.G. Pronyaev, R.O. Sayer, A.A. Sonzogni, N.C. Summers, P. Talou, I.J. Thompson, A. Trkov, R.L. Vogt, S.C. van der Marck, A. Wallner, M.C. White, D. Wiarda, and P.G. Young. “ENDF/B-VII.1 Nuclear Data for Science and Technology: Cross Sections, Covariances, Fission Product Yields and Decay Data”. In: *Nuclear Data Sheets* 112.12 (Dec. 2011), pp. 2887–2996. DOI: [10.1016/j.nds.2011.11.002](https://doi.org/10.1016/j.nds.2011.11.002).
- [16] Keiichi Shibata, Osamu IWAMOTO, Tsuneo NAKAGAWA, Nobuyuki IWAMOTO, Akira ICHIHARA, Satoshi KUNIEDA, Satoshi CHIBA, Kazuyoshi FURUTAKA, Naohiko OTUKA, Takaaki OHASAWA, Toru MURATA, Hiroyuki MATSUNOBU, Atsushi ZUKERAN, So KAMADA, and Jun-ichi KATAKURA. “JENDL-4.0: A New Library for Nuclear Science and Engineering”. In: *Journal of Nuclear Science and Technology* 48.1 (2011), pp. 1–30. DOI: [10.1080/18811248.2011.9711675](https://doi.org/10.1080/18811248.2011.9711675). eprint: <http://www.tandfonline.com/doi/pdf/10.1080/18811248.2011.9711675>.
- [17] Arjan Koning, Robin Forrest, Mark Kellett, Robert Mills, Hans Henriksson, and Yolanda Rugama. *The JEFF-3.1 Nuclear Data Library*. Research Report 6190. Paris, France: Nuclear Energy Agency, Organisation for Economic Co-operation and Development, 2006.
- [18] Z.G. Ge, Y.X. Zhuang, T.J. Liu, J.S. Zhang, H.C. Wu, Z.X. Zhao, and H.H. Xia. “The Updated Version of Chinese Evaluated Nuclear Data Library (CENDL-3.1)”. In: *Journal of the Korean Physical Society* 59 (2011), p. 1052.

- [19] A.I. Blokhin, A.V. Ignatyuk, V.N. Manokhin, M.N. Nikolaev, and V.G. Pronyaev. *BROND-2.2, Russian Evaluated Neutron Reaction Data Library*. Tech. rep. IAEA-NDS-90 Rev.8. International Atomic Energy Agency, 1984.
- [20] O. C. Zienkiewicz and Z. Zhu R. L. Taylor. *The Finite Element Method: Its Basis and Fundamentals*. Butterworth-Heinemann, 2005.
- [21] Bernardo Cockburn, George E. Karniadakis, and Chi-Wang Shu. “The Development of Discontinuous Galerkin Methods”. In: *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, 2000, pp. 3–50. DOI: [10.1007/978-3-642-59721-3_1](https://doi.org/10.1007/978-3-642-59721-3_1).
- [22] J.R. Askew. *A characteristics formulation of the neutron transport equation in complicated geometries*. Tech. rep. United Kingdom: AEEW-M-1108, 1972.
- [23] J. R. Askew. “Review of the status of collision probability methods”. In: *Proc. Sem. Numerical Reactor Calculations*. International Atomic Energy Agency, Vienna. 1972, pp. 185–196.
- [24] K Lathrop and B Carlson. “Numerical solution of the Boltzmann Transport Equation”. In: *Journal of Computational Physics* 1.2 (Nov. 1966), pp. 173–197. DOI: [10.1016/0021-9991\(66\)90002-7](https://doi.org/10.1016/0021-9991(66)90002-7).
- [25] Benjamin S. Kirk, John W. Peterson, Roy H. Stogner, and Graham F. Carey. “libMesh : a C++ library for parallel adaptive mesh refinement/coarsening simulations”. In: *Engineering with Computers* 22.3 (Dec. 2006), pp. 237–254.
- [26] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Karpeyev, D. Kaushik, M. Knepley, L. Curfman McInnes, K. Rupp, B. Smith, S. Zampini, and H. Zhang. *PETSc Users Manual*. 3.6. Mathematics and Computer Science Division, Argonne National Laboratory. June 2015.
- [27] *MPI: A Message-Passing Interface Standard Version 3.1*. June 2015.
- [28] Y. Wang, S. Schunert, J. Ortensi, F.N. Gleicher, B.A. Baker, M. DeHart, and Martineau R.C. “Demonstration of MAMMOTH Fully-Coupled Multiphysics Simulation with the Godiva Benchmark Problem”. In: *M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*. Jeju, Korea, 2017.
- [29] J.D. Hales, R.L. Williamson, S.R. Novascone, G. Pastore, B.W. Spencer, D.S. Stafford, K.A. Gamble, D.M. Perez, R.J. Gardner, Liu W., J. Galloway, C. Matthews, C. Unal, and N. Carlson. *BISON Theory Manual The Equations Behind Nuclear Fuel Analysis*. 1.3. Idaho National Laboratory. 2016.
- [30] R.L. Williamson, J.D. Hales, S.R. Novascone, M.R. Tonks, D.R. Gaston, C.J. Permann, D. Andrs, and R.C. Martineau. “Multidimensional multiphysics simulation of nuclear fuel behavior”. In: *J. Nucl. Mater.* 423 (2012).
- [31] J.D. Hales, S.R. Novascone, R.L. Williamson, D.R. Gaston, and M.R. Tonks. “Solving nonlinear solid mechanics problems with the Jacobian-free Newton Krylov method”. In: *Comput. Model. Eng. Sci.* 84 (2012).
- [32] Ray A. Berry, John W. Peterson, Hongbin Zhang, Richard C. Martineau, Haihua Zhao, Ling Zou, and David Andrs. *RELAP-7 Theory Manual*. INL. Dec. 2014.
- [33] Nicholas D. Stehle, Dmitriy Y. Anistratov, and Marvin L. Adams. “A hybrid transport-diffusion method for 2D transport problems with diffusive subdomains”. In: *Journal of Computational Physics* 270 (Aug. 2014), pp. 325–344.
- [34] S. Chandrasekhar. *Radiative Transfer*. Dover Publications Inc., Mar. 28, 2003. 416 pp. ISBN: 0486605906. URL: http://www.ebook.de/de/product/3303044/s_chandrasekhar_radiative_transfer.html.
- [35] J. Morel. *A Non-Local Tensor Diffusion Theory*. Research Report LA-UR-07-5257. Los Alamos, NM: Los Alamos National Laboratory, 2007.
- [36] T.J. Trahan and E.W. Larsen. “3-D ANISOTROPIC NEUTRON DIFFUSION IN OPTICALLY THICK MEDIA WITH OPTICALLY THIN CHANNELS”. In: *Proc. Int. Conf. on Mathematics and Computational Methods Applied in Nuclear Science and Engineering (MC 2011)*. ANS. Rio de Janeiro, Brazil, 2011.
- [37] E.W. Larsen and J.E. Morel. “Nonlocal Diffusion Coefficients for Neutronic Systems Containing Voided Subregions”. In: *M&C 2017, International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*. ANS. Jeju, Korea, 2017.
- [38] John R. Howell, Robert Siegel, and M. Pinar Menguc. *Thermal Radiation Heat Transfer*. 6th. Taylor & Francis Inc, Oct. 5, 2015. 1016 pp. ISBN: 978-1466593268. URL: http://www.ebook.de/de/product/24179465/john_r_howell_robert_siegel_m_pinar_menguc_thermal_radiation_heat_transfer.html.

- [39] Yeram Sarkis Touloukian. *Thermophysical Properties of High Temperature Solid Materials*. New York: Macmillan, 1967.
- [40] W. D. Wood, H. W. Deem, and C. F. Lucks. *Thermal Radiative Properties*. New York: Springer, 1964.
- [41] G. G. Gubareff, J. E. Janssen, and R. H. Torberg. *Thermal radiation properties survey*. Minneapolis: Honeywell Research Center, 1960.
- [42] J. E. Morel and J. M. McGhee. “A Self-Adjoint Angular Flux Equation”. In: *Nuclear Science and Engineering* 132 (1999), pp. 312–325.
- [43] Jennifer Liscum-Powell. *Finite Element Numerical Solution of a Self-Adjoint Transport Equation for Coupled Electron-Photon Problems*. Tech. rep. SAND2000-2017. Sandia National Laboratory, 2000.
- [44] Jennifer L. Liscum-Powell, Anil K. Prinja, Jim E. Morel, and Leonard J. Lorence, Jr. “Finite Element Solution of the Self-Adjoint Angular Flux Equation for Coupled Electron-Photon Transport”. In: *Nuclear Science and Engineering* 142 (2002), pp. 270–291.
- [45] Jim E. Morel, B. Todd Adams, Taewan Noh, John M. McGhee, Thomas M. Evans, and Todd J. Urbatsch. “Spatial Discretizations for Self-Adjoint Forms of the Radiative Transfer Equations”. In: *Journal of Computational Physics* 214 (2006), pp. 12–40.
- [46] Yaqi Wang, Hongbin Zhang, and Richard Martineau. “Diffusion Acceleration Schemes for the Self-Adjoint Angular Flux Formulation with a Void Treatment”. In: *Nuclear Science and Engineering* 176 (2014), pp. 201–225.
- [47] Jon Hansen, Jacob Peterson, Jim Morel, Jean Ragusa, and Yaqi Wang. “A Least-Squares Transport Equation Compatible with Voids”. In: *Journal of Computational and Theoretical Transport Special Issue: Papers from the 23rd International Conference on Transport Theory* 43.1–7 (2014).
- [48] Jacob R. Peterson, Hans R. Hammer, Jim E. Morel, Jean C. Ragusa, and Yaqi Wang. “Conservative Nonlinear Diffusion Acceleration Applied to the Unweighted Least-Squares Transport Equation in MOOSE”. In: *Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*. ANS. Nashville, Tennessee, Apr. 2015.
- [49] H Hammer, J.E. Morel, and Y. Wang. “Nonlinear Diffusion Acceleration in Voids for the Weighted Least-Square Transport Equation”. In: *M&C 2017, International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*. ANS. Jeju, Korea, 2017.
- [50] Yaqi Wang and Frederick N. Gleicher. “Revisit Boundary Conditions for the Self-Adjoint Angular Flux Formulation”. In: *PHYSOR 2014 - The Role of Reactor Physics toward a Sustainable Future*. The Westin Miyako, Kyoto, Japan, Sept. 2014.
- [51] Alexander N. Brooks and Thomas J.R. Hughes. “Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations”. In: *Computer Methods in Applied Mechanics and Engineering* 32.1-3 (Sept. 1982), pp. 199–259. DOI: [10.1016/0045-7825\(82\)90071-8](https://doi.org/10.1016/0045-7825(82)90071-8).
- [52] Sebastian Schunert, Yaqi Wang, Richard Martineau, and Mark D. DeHart. “A new mathematical adjoint for the modified SAAF-SN equations”. In: *Annals of Nuclear Energy* 75 (2015), pp. 340–352.
- [53] Vincent M. Laboure, Ryan G. McClarren, and Yaqi Wang. “Globally Conservative, Hybrid Self-Adjoint Angular Flux and Least-Squares Method Compatible with Void.” In: *Nuclear Science and Engineering* (2017).
- [54] Vincent M. Laboure, Yaqi Wang, and Mark D. DeHart. “Least-Squares P_N Formulation of the Transport Equation Using Self-Adjoint-Angular-Flux Consistent Boundary Conditions.” In: *PHYSOR 2016 - Unifying Theory and Experiments in the 21st Century*. ANS. Sun Valley, ID, May 2016.
- [55] Thomas A. Brunner, James P. Holloway. “Two-dimensional time dependent Riemann solvers for neutron transport”. In: *Journal of Computational Physics* 210 (2005), pp. 386–399.
- [56] Vincent M. Laboure, Ryan G. McClarren, and Cory D. Hauck. “Implicit Filtered P_N for High-Energy Density Thermal Radiation Transport using Discontinuous Galerkin Finite Elements.” In: *Journal of Computational Physics* 321 (2016), pp. 624–643. DOI: [10.1016/j.jcp.2016.05.046](https://doi.org/10.1016/j.jcp.2016.05.046). arXiv: [1601.08242](https://arxiv.org/abs/1601.08242) [physics.comp-ph]. URL: <http://www.sciencedirect.com/science/article/pii/S0021999116301917>.
- [57] Vincent M. Laboure. “Improved Fully-Implicit Spherical Harmonics Methods for First and Second Order Forms of the Transport Equation using Galerkin Finite Element”. PhD thesis. Texas A&M University, 2016.

- [58] Ryan G. McClarren, Cory D. Hauck. “Robust and accurate filtered spherical harmonics expansions for radiative transfer”. In: *Journal of Computational Physics* 229 (2010), pp. 5597–5614.
- [59] David Radice, Ernazar Abdikamalov, Luciano Rezzolla, Christian D. Ott. “A new spherical harmonics scheme for multi-dimensional radiation transport I. Static matter configurations”. In: *Journal of Computational Physics* 242 (2013), pp. 648–669.
- [60] Martin Frank, Cory Hauck, and Kerstin Kuepper. “Convergence of Filtered Spherical Harmonic Equations for Radiation Transport”. In: *Communications in Mathematical Sciences* 14 (2016), pp. 1443–1465.
- [61] Vincent M. Laboure and Ryan G. McClarren. “Energy-Dependent Implicit Filtered P_N Method for Thermal Radiation Transport”. In: *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2017)*. American Nuclear Society (ANS). Jeju, Korea, Apr. 2017.
- [62] Adams, M. L. “Discontinuous finite element transport solutions in thick diffusive problems”. In: *Nuclear Science and Engineering* 137.3 (2001), pp. 298–333.
- [63] Yaqi Wang. “Adaptive Mesh Refinement Solution Techniques for the Multigroup S_N Transport Equation Using a High-Order Discontinuous Finite Element Method”. PhD thesis. Texas A&M University, 2009.
- [64] Beatrice M. Riviere. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*. CAMBRIDGE UNIV PR, Dec. 11, 2008. 210 pp. ISBN: 089871656X. URL: http://www.ebook.de/de/product/23834215/beatrice_m_riviere_discontinuous_galerkin_methods_for_solving_elliptic_and_parabolic_equations_theory_and_implementation.html.
- [65] K.S. Smith. “Nodal Method Storage Reduction by Nonlinear Iteration”. In: *Trans. Am. Nucl. Soc.*, 44, p. 265. ANS. 1983.
- [66] L. LuLu, K. Smith, and B. Forget. “Techniques for Stabilizing Coarse-Mesh Finite Difference (CMFD) in Methods of Characteristics (MOC)”. In: *MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) method*. ANS. Nashville, TN, Apr. 2015.
- [67] Ser Gi Hong, Kang-Seog Kim, and Jae Seung Song. “Fourier Convergence Analysis of the Rebalance Methods for Discrete Ordinates Transport Equations in Eigenvalue Problems”. In: *Nuclear Science and Engineering* 164 (2010), pp. 33–52.
- [68] Sebastian Schunert, Yaqi Wang, Frederick Gleicher, Javier Ortensi, Benjamin Baker, Vincent Laboure, Congjian Wang, Mark DeHart, and Richard Martineau. “A flexible nonlinear diffusion acceleration method for the {SN} transport equations discretized with discontinuous finite elements”. In: *Journal of Computational Physics* 338 (June 2017), pp. 107–136. ISSN: 0021-9991. DOI: [10.1016/j.jcp.2017.01.070](https://doi.org/10.1016/j.jcp.2017.01.070). URL: <http://www.sciencedirect.com/science/article/pii/S0021999117301286>.
- [69] H. Park, D. A. Knoll, and C. K. Newman. “Nonlinear Acceleration of Transport Criticality Problems”. In: *Nuclear Science and Engineering* 172 (2012), pp. 52–65.
- [70] D. A. Knoll, H. Park, and C. Newman. “Acceleration of k-Eigenvalue/Criticality Calculations Using the Jacobian-Free Newton-Krylov Method”. In: *Nuclear Science and Engineering* 167 (2011), pp. 133–140.
- [71] S. Schunert, H. Hammer, J. Lou, Y. Wang, J. Ortensi, F. Gleicher, B. Baker, M. DeHart, and R. Martineau. “Using Directional Diffusion Coefficients for Nonlinear Diffusion Acceleration of the First Order SN Equations in Near-Void Regions”. In: *Transactions of the Am. Nucl. Soc.* ANS. Las Vegas, NV, 2016.
- [72] A. Zhu, M. Jarrett, Y. Xu, B. Kochunas, E. Larsen, and T. Downar. “An optimally diffusive Coarse Mesh Finite Difference method to accelerate neutron transport calculations”. In: *Annals of Nuclear Energy* 95 (2016), pp. 116–124.
- [73] S. Yuk and N.Z. Cho. “Improved Two-Level p-CMFD Acceleration in Neutron Transport Calculations”. In: *M&C 2017, International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*. ANS. Jeju, Korea, 2017.
- [74] M. M. Nanneh and M. M. R. Williams. “A Diffusion-Transport Theory Hybrid Method for Calculating Neutron Flux Distributions in Slab Lattices”. In: *Atomkernenergie Kerntechnik* 47 (1985), pp. 221–224.
- [75] Ricardo C. Barros, Musa Yavuz, Marcos P. de Abreu, Hermes Alves Filho, and José Antônio M. Mello. “Progress in Spectral Nodal Methods Applied to Discrete Ordinates Transport Problems”. In: *Progress in Nuclear Energy* 33.1/2 (1998), pp. 117–154.

- [76] Enrico Girardi and Jean-Michel Ruggieri. “Mixed First- and Second-Order Transport Method Using Domain Decomposition Techniques for Reactor Core Calculations”. In: *Proc. International Conference on Supercomputing in Nuclear Application*. Paris, France, 2003.
- [77] Enrico Girardi, Jean-Michel Ruggieri, Patricia Sireta, and Guillaume Ritter. “A New Method for the Treatment of Local Strong Heterogeneities and its Application to the Phebus Experimental Facility”. In: *PHYSOR 2004 - The Physics of Fuel Cycles and Advanced Nuclear Systems : Global Developments*. American Nuclear Society, Lagrange Park, IL. Chicago, Illinois, Apr. 2004.
- [78] Dmitriy Y. Anistratov and Nicholas D. Stehle. “Computational Transport Methodology Based on Decomposition of a Problem Domain into Transport and Diffusive Subdomains”. In: *Journal of Computational Physics* 231 (2012), pp. 8009–8028.
- [79] Sergiy Manolov, Jim E. Morel, and Cristian Rabiti. “Hybrid Sn-Diffusion and Sn-P3 Transport Calculations”. In: *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. Sun Valley, Idaho, May 2013, pp. 2748–2758.
- [80] Yaqi Wang, Sebastian Schunert, Mark D. DeHart, and Richard C. Martineau. “Hybrid PN-SN Calculations with SAAF for the Multiscale Transport Capability in Rattlesnake”. In: *PHYSOR 2016 - Unifying Theory and Experiments in the 21st Century*. Sun Valley, Idaho, USA, May 2016.
- [81] K. S. Smith. “Assembly Homogenization Techniques for Light Water Reactor Analysis”. In: *Progress in Nuclear Energy* 17,3 (July 1986), pp. 303–335.
- [82] Yaqi Wang, Sebastian Schunert, Mark DeHart, Richard Martineau, and Weixiong Zheng. “Hybrid PN - SN with Lagrange multiplier and upwinding for the multiscale transport capability in Rattlesnake”. In: *Progress in Nuclear Energy* (Apr. 2017). DOI: [10.1016/j.pnucene.2017.03.020](https://doi.org/10.1016/j.pnucene.2017.03.020).
- [83] W. H. Reed and T. R. Hill. *Triangular mesh methods for the neutron transport equation*, Los Alamos Scientific Laboratory Report. Tech. rep. LA-UR-73-479. Los Alamos National Laboratory, 1973.
- [84] Yaqi Wang, Sebastian Schunert, Javier Ortensi, Frederick Gleicher, Benjamin Baker, Gilles Youinou, Mark DeHart, and Richard Martineau. “Derivation of the Point-Kinetics Parameters for the Coupled Reactors”. In: *Transactions of American Nuclear Society* (2016).
- [85] R. Avery. *Coupled fast-thermal power breeder reactor*. US Patent No: 2,992,982, Dec. 10, 1957.
- [86] Karl Ott. “Quasistatic Treatment of Spatial Phenomena in Reactor Dynamics”. In: *Nuclear Science and Engineering* 26.4 (1966), pp. 563–565. DOI: [10.13182/NSE66-A18428](https://doi.org/10.13182/NSE66-A18428). eprint: <http://www.tandfonline.com/doi/pdf/10.13182/NSE66-A18428>.
- [87] Sandra Dulla, Ernest H. Mund, and Piero Ravetto. “The quasi-static method revisited”. In: *Progress in Nuclear Energy* 50.8 (Nov. 2008), pp. 908–920. DOI: [10.1016/j.pnucene.2008.04.009](https://doi.org/10.1016/j.pnucene.2008.04.009).
- [88] Zachary M. Prince and Jean C. Ragusa. “Multiphysics Core Dynamics Simulation Using the Improved Quasi-Static Method”. In: *M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*. Jeju, Korea, Apr. 2017.
- [89] Thomas G. Saller, Randal S. Baker, and Jon A. Dahl. “On-The-Fly Multigroup Weighting in PARTISN”. In: *M&C 2017 - International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering*. Jeju, Korea, Apr. 2017.
- [90] Vicente Hernandez, Jose E. Roman, and Vicente Vidal. “SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems”. In: *ACM Transactions on Mathematical Software* 31.3 (Sept. 2005), pp. 351–362. DOI: [10.1145/1089014.1089019](https://doi.org/10.1145/1089014.1089019).
- [91] D. R. Ferguson and K. L. Derstine. “Optimized Iteration Strategies and Data Management Considerations for Fast Reactor Finite Difference Diffusion Theory Codes”. In: *Nuclear Science and Engineering* 64 (1977), pp. 593–604.
- [92] Marvin L. Adams and Edward W. Larsen. “Fast iterative methods for discrete-ordinates particle transport calculations”. In: *Progress in Nuclear Energy* 40.1 (2002), pp. 3–159. ISSN: 0149-1970. DOI: [DOI: 10.1016/S0149-1970\(01\)00023-3](https://doi.org/10.1016/S0149-1970(01)00023-3).
- [93] Edward W. Larsen and Jim E. Morel. “Advances in Discrete-Ordinates Methodology”. In: *Nuclear Computational Science*. Springer Netherlands, Dec. 2009, pp. 1–84. DOI: [10.1007/978-90-481-3411-3_1](https://doi.org/10.1007/978-90-481-3411-3_1).
- [94] Yung-An Chao. “A new and rigorous SP N theory for piecewise homogeneous regions”. In: *Annals of Nuclear Energy* 96 (Oct. 2016), pp. 112–125. DOI: [10.1016/j.anucene.2016.06.010](https://doi.org/10.1016/j.anucene.2016.06.010).

- [95] S. Montgomery-Smith, David A. Jack, and Douglas E. Smith. “A systematic approach to obtaining numerical solutions of Jeffery’s type equations using Spherical Harmonics”. In: *Composites Part A: Applied Science and Manufacturing* 41.7 (July 2010), pp. 827–835. DOI: [10.1016/j.compositesa.2010.02.010](https://doi.org/10.1016/j.compositesa.2010.02.010).
- [96] Yaqi Wang. *INSTANT Theory Manual - Part I. PN-hybrid-FEM for the Multigroup Transport Equation*. 1. Idaho National Laboratory. Idaho Falls, ID, USA, Feb. 2012.
- [97] Vincent M. Laboure, Yaqi Wang, and Sebastian Schubert. “Direct S_N treatment of the Scattering Source with Phase Functions”. In: *Transactions of American Nuclear Society* (2017).
- [98] K. D. Lathrop and B. G. Carlson. *Discrete ordinates angular quadrature of the neutron transport equation*. Tech. rep. LA-3186. Los Alamos National Laboratory, 1964.
- [99] Alain Hébert. *Advanced Reactor Physics*. Presses internatlonals Polytechnique, 2009.
- [100] W. F. Walters. *Use of the Chebyshev-Legendre Quadrature Set in Discrete-Ordinate Codes*. Tech. rep. LA-UR-87-3621. Los Alamos Scientific Laboratory, 1987.
- [101] Akio Yamamoto, Masato Tabuchi, Naoki Sugimura, Tadashi Ushio, and Masaaki Mori. “Derivation of Optimum Polar Angle Quadrature Set for the Method of Characteristics Based on Approximation Error for the Bickley Function”. In: *Journal of Nuclear Science and Technology (Tokyo, Japan)* 44.2 (2007), pp. 129–136.
- [102] Yaqi Wang. “A Remez Algorithm on Finding Polar Angle Quadrature Sets for the Method of Characteristics”. In: *Transactions of the American Nuclear Society* 112 (2015), pp. 413–415.