KiT-RT: An Extendable Framework for Radiative Transfer and Therapy

JONAS KUSCH, Norwegian University of Life Sciences, Norway
STEFFEN SCHOTTHÖFER, Oak Ridge National Laboratory, USA
PIA STAMMER, Karlsruhe Institute of Technology, Germany, German Cancer Research Center - DKFZ, Germany, and HIDSS4Health-Helmholtz Information and Data Science School for Health, Germany
JANNICK WOLTERS, Karlsruhe Institute of Technology, Germany
TIANBAI XIAO, State Key Laboratory of High Temperature Gas Dynamics, Institute of Mechanics, Chinese Academy of Sciences, China

In this article, we present Kinetic Transport Solver for Radiation Therapy (KiT-RT), an open source C++-based framework for solving kinetic equations in therapy applications available at https://github.com/CSMMLab/KiT-RT. This software framework aims to provide a collection of classical deterministic solvers for unstructured meshes that allow for easy extendability. Therefore, KiT-RT is a convenient base to test new numerical methods in various applications and compare them against conventional solvers. The implementation includes spherical harmonics, minimal entropy, neural minimal entropy, and discrete ordinates methods. Solution characteristics and efficiency are presented through several test cases ranging from radiation transport to electron radiation therapy. Due to the variety of included numerical methods and easy extendability, the presented open source code is attractive for both developers, who want a basis to build their numerical solvers,

Note that all authors contributed equally to this work and are listed alphabetically.
Jonas Kusch has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 491976834. Pia Stammer is supported by the Helmholtz Association under the joint research school HIDSS4Health – Helmholtz Information and Data Science School for Health. The work of Steffen Schotthöfer is funded by the Priority Programme "Theoretical Foundations of Deep Learning (SPP2298)" by the Deutsche Forschungsgemeinschaft. The work of Steffen Schotthöfer is sponsored by the Office of Advanced Scientific Computing Research, U.S. Department of Energy, and performed at the Oak Ridge National Laboratory, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). Tianbai Xiao acknowledges the support by National Science Foundation of China (12302381) and the computing resources provided by Hefei Advanced Computing Center.
Authors' addresses: J. Kusch, Norwegian University of Life Sciences, Drøbakvn 31, Ås, Norway; e-mail: jonas.kusch1@gmail.com; S. Schotthöfer, Oak Ridge National Laboratory, 1 Bethel Valley Road, Oak Ridge, TN 37831, USA; e-mail: schotthofers@ornl.gov; P. Stammer, Karlsruhe Institute of Technology, Englerstraße 2, Karlsruhe, Germany and German Cancer Research Center -DKFZ, Heidelberg, Germany and HIDSS4Health-Helmholtz Information and Data Science School for Health, Karlsruhe/Heidelberg, Im Neuenheimer Feld 280, Germany; e-mail: pia.stammer@kit.edu; J. Wolters, Karlsruhe Institute of Technology, Englerstraße 2, Karlsruhe, Germany; e-mail: jannick.wolters@rwth-aachen.de; T. Xiao, State Key Laboratory of High Temperature Gas Dynamics, Institute of Mechanics, Chinese Academy of Sciences, North Fourth Ring West Road 15, Beijing, China; e-mail: txiao@imech.ac.cn.

0098-3500/2023/12-ART38 $15.00
https://doi.org/10.1145/3630001

and users or application engineers, who want to gain experimental insights without directly interfering with the codebase.

CCS Concepts: • **Mathematics of computing → Solvers**; • **Applied computing → Physics;**

Additional Key Words and Phrases: Kinetic theory, radiation transport, radiation therapy, finite volume methods, machine learning

ACM Reference format:
Jonas Kusch, Steffen Schotthöfer, Pia Stammer, Jannick Wolters, and Tianbai Xiao. 2023. KiT-RT: An Extendable Framework for Radiative Transfer and Therapy. *ACM Trans. Math. Softw.* 49, 4, Article 38 (December 2023), 24 pages.
https://doi.org/10.1145/3630001

1 INTRODUCTION

Personalized medicine in radiation oncology has been an important research topic over the past few decades. To allow for accurate, reliable, and efficient treatment planning tailored toward individual patient needs, there is a growing desire to undertake direct numerical simulation for radiation therapy. High-fidelity numerical solutions enable quantitative estimation of the dose received by the tumor as well as the surrounding tissue while allowing for an automated generation of optimal treatment plans. Besides the aim to ensure sufficient accuracy, such simulations are required to run on limited computational resources such as workstation PCs.

Traditional methods to predict dose distributions in radiation oncology largely rely on simplified models, such as pencil beam models based on the Fermi-Eyges theory [21]. While such models are computationally efficient, they often lack the required accuracy, especially in cases including air cavities or other inhomogeneities [31, 37]. However, **Monte Carlo (MC)** algorithms, which simulate individual interacting particles, achieve a satisfactory accuracy [6]. However, despite ongoing research to accelerate MC methods, their high computational costs currently render them impractical for clinical usage [23, 34]. To obtain a computationally feasible model with comparable accuracy, radiation particles are described on a mesoscopic level through the deterministic linear Boltzmann equation [10, 65–67]. An efficient and accurate numerical approximation to the linear Boltzmann equation can be achieved through the construction of grid-based macroscopic approximations [19, 27, 68, 69]. Variants of grid-based methods for radiation therapy can be found in, for example, References [7, 30, 32, 39, 41, 57].

The available grid-based methods employ various macroscopic approximations of the linear Boltzmann equation that all exhibit certain advantages and shortcomings.

In Reference [19], the modal entropy method called M_N is used as a macroscopic model. This method preserves the positivity of particle distributions while yielding accurate results with little spurious oscillations. However, the need to solve a possibly ill-posed optimization problem in every spatial cell and timestep results in increased computational costs. While analytic solutions to the optimization problems are available at small truncation orders, they cannot capture all physical effects accurately. Further approaches aim at mitigating the challenge of the problem's ill-condition by regularization [4] or reducing the associated computational costs, e.g., using neural networks [61].

In Reference [57] the use of computationally cheap nodal discretizations, known as the S_N method has been proposed. In this case, the solution remains positive, and it is shown that the expensive scattering terms can be approximated efficiently with a Fokker-Planck approximation. Furthermore, the solution can exhibit non-physical artifacts, known as ray-effects [43, 52, 55], which reduce the approximation quality. While methods to mitigate ray-effects exist, see, e.g., References [2, 13, 24, 44, 64], they commonly require picking problem-dependent tuning parameters.

In Reference [39], the modal P_N method has been employed to derive a macroscopic model for radiation treatment planning. While it does not preserve the positivity of the solution and can potentially yield oscillatory approximations, it allows for an efficient numerical treatment of scattering terms. In Reference [41] a combination of S_N and P_N methods that reduces computational costs through a dynamical low-rank approximation [35] has been proposed. The efficiency of this method relies on the ability to describe the solution through a low-rank function.

The variety of different methods allows for individual method choices tailored to different settings. The comparability of different methods in a uniform framework is not only interesting for clinical usage but also for future research in computational radiation therapy. The aim of our open source C++ **Kinetic Transport for radiation therapy (KiT-RT)** framework is therefore to provide a collection of available deterministic methods. Special focus is put on extendability by the use of polymorphism to simplify the implementation of novel solution methods. The methods provided by our framework are optimized for an application on workstation PCs. This meets the typical requirements in radiation therapy applications: For clinical usage, the computational resources are often limited, and the time between recording the CT image and the actual treatment must not exceed a certain time period. Hence, radiation therapy codes that are applicable for clinical use should run efficiently on workstation PCs. Moreover, conventional codes often require structured grids [26, 38, 60, 62, 71], leading to inaccurate representations of structures on CT images. While accuracy in practice is also limited by the CT density values that are given on a structured grid, these are often downsampled to a lower resolution for dose computations. Further, a recomputation of the CT values for unstructured grids is feasible if it improves the quality of dose computations. Therefore, our framework provides functionalities for both unstructured meshes that preserve organ outlines on CT images, as well as standard rectangular grids.

Note that in clinical practice, CT data are sometimes augmented with MRI or PET images, and research on the use of MRIs for delineation as well as delivery has become increasingly popular in recent years [17, 20, 58]. Here, we however focus solely on CT images, as it is the most common imaging modality in practice [17] and provides attenuation as well as electron density information required for the dose calculations [17, 20].

This article aims at presenting the developed framework and its functionality while providing the necessary mathematical and physical background on the principles the software is based on. In Section 2, we provide the underlying physical model as well as its reformulation as a time-dependent partial differential equation. Section 3 discusses different macroscopic models as well as the advantages and disadvantages of the individual underlying directional discretizations. Sections 4 and 5 focus on the used discretizations and software architecture, respectively. A study on parallel efficiency and scaling is done in Section 6. Last, we validate our implementations and analyze their performance for different test cases in Section 7. Extended details of the methods and documentation of the code can be found on https://kit-rt.readthedocs.io.

2 PHYSICAL MODEL

Let us recap the main physical model that is used for computational radiotherapy treatment planning. In the following we will make use of the so-called *continuous slowing down approximation*, which assumes a continuous, deterministic energy loss of particles proportional to the expected energy loss. Note that this model is only valid when considering charged particles, i.e., electrons or protons, whose interactions are dominated by Coulomb's law.

The main goal is to compute the radiation dose distribution,

$$D(\mathbf{x}) = \frac{1}{\rho(\mathbf{x})} \int_0^\infty \int_{\mathbb{S}^2} S(E, \mathbf{x}) \psi(E, \mathbf{x}, \Omega) \, \mathrm{d}\Omega \, \mathrm{d}E, \tag{1}$$

that results from a given treatment plan. Here $E \in \mathbb{R}_+$ is the energy, $\mathbf{x} \in \mathbf{X} \subset \mathbb{R}^3$ denotes the spatial domain, and $\Omega \in \mathbb{S}^2$ is the flight direction of particles. Moreover, $\psi : \mathbb{R}_+ \times \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}$ denotes the radiation flux density and $\rho : \mathbb{R}^3 \to \mathbb{R}$ is the patient tissue density. The stopping power $S : \mathbb{R}_+ \times \mathbb{R}^3 \to \mathbb{R}$ models a continuous energy loss of particles due to scattering with tissue and is defined as

$$S(E, \mathbf{x}) = \int_0^\infty E' \int_{-1}^1 \Sigma(E, E', \mathbf{x}, \mu) \, \mathrm{d}\mu \, \mathrm{d}E', \tag{2}$$

with the scattering cross section $\Sigma : \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}^3 \times [-1, 1] \to \mathbb{R}$. The radiation flux density, which describes the probability of finding a particle at a certain region in phase space, can be computed from the **continuous slowing down (CSD)** approximation [42] of the energy-dependent linear Boltzmann equation,

$$\begin{aligned} &- \partial_E \left(S(E, \mathbf{x}) \psi(E, \mathbf{x}, \Omega) \right) + \Omega \cdot \nabla_x \psi(E, \mathbf{x}, \Omega) + \Sigma_t(E, \mathbf{x}) \psi(E, \mathbf{x}, \Omega) \\ &= \int_{\mathbb{S}^2} \Sigma_s(E, \mathbf{x}, \Omega \cdot \Omega') \psi(E, \mathbf{x}, \Omega') \, \mathrm{d}\Omega'. \end{aligned} \tag{3}$$

This model assumes a continuous energy loss of particles traveling through a background material, which is modeled using the stopping power S. The scattering cross section $\Sigma_s(E, \mathbf{x}, \Omega \cdot \Omega')$ denotes the probability of particles at position $\mathbf{x}$ with energy E changing their flight direction from Ω' to Ω due to a collision with the patient tissue. The total cross section Σ_t is given by

$$\Sigma_t(E, \mathbf{x}) = \Sigma_{s,0}(E, \mathbf{x}) = 2\pi \int_{-1}^1 \Sigma_s(E, \mathbf{x}, \mu) \, \mathrm{d}\mu. \tag{4}$$

To simplify the evaluation of material properties, we follow the common assumption that all materials are water-equivalent and differ only in density (e.g., References [39, 57, 72]), i.e.,

$$\begin{aligned} S(E, \mathbf{x}) &= S^{H_2O}(E) \rho(\mathbf{x}), \\ \Sigma_t(E, \mathbf{x}) &= \Sigma_t^{H_2O}(E) \rho(\mathbf{x}), \\ \Sigma_s(E, \mathbf{x}, \Omega \cdot \Omega') &= \Sigma_s^{H_2O}(E, \Omega \cdot \Omega') \rho(\mathbf{x}), \end{aligned} \tag{5}$$

where we leave out the superscript H_2O in the following. Cross sections for water are taken from the ICRU database [28]. Having defined the prerequisites of our physical model, we can focus on bringing it into a form that allows for computing numerical approximations efficiently. It turns out that the energy variable in Equation (3) can be treated as a pseudo-time, which facilitates solving the CSD equation. For a given maximal energy $E_{\max}$ let us define the transformed energy as

$$\widetilde{E}(E) := \int_0^{E_{\max}} \frac{1}{S(E')} \, \mathrm{d}E' - \int_0^E \frac{1}{S(E')} \, \mathrm{d}E' \tag{6}$$

and the transformed particle density as

$$\widetilde{\psi}(\widetilde{E}, \mathbf{x}, \Omega) := S(E) \rho(\mathbf{x}) \psi(E(\widetilde{E}), \mathbf{x}, \Omega). \tag{7}$$

Then, multiplying Equation (3) with $S(E)$ and plugging in the defined transformation gives

$$\partial_{\widetilde{E}} \widetilde{\psi}(\widetilde{E}, \mathbf{x}, \Omega) + \Omega \cdot \nabla_x \frac{\widetilde{\psi}(\widetilde{E}, \mathbf{x}, \Omega)}{\rho(\mathbf{x})} + \widetilde{\Sigma}_t(\widetilde{E}) \widetilde{\psi}(\widetilde{E}, \mathbf{x}, \Omega) = \int_{\mathbb{S}^2} \widetilde{\Sigma}_s(\widetilde{E}, \Omega \cdot \Omega') \widetilde{\psi}(\widetilde{E}, \mathbf{x}, \Omega') \, \mathrm{d}\Omega', \tag{8}$$

where we define $\widetilde{\Sigma}_t(\widetilde{E}) := \Sigma_t(E(\widetilde{E}))$ and $\widetilde{\Sigma}_s(\widetilde{E}, \Omega \cdot \Omega') := \Sigma_s(E(\widetilde{E}), \Omega \cdot \Omega')$. Dropping the tilde notation and treating $\widetilde{E}$ as a pseudo-time t gives a slightly modified version of the classical linear

Boltzmann equation,

$$\begin{aligned} \partial_t \psi(t,\mathbf{x},\Omega) + \Omega \cdot \nabla_x \frac{\psi(t,\mathbf{x},\Omega)}{\rho(\mathbf{x})} + \Sigma_t(t)\psi(t,\mathbf{x},\Omega) &= \int_{\mathbb{S}^2} \Sigma_s(t,\Omega\cdot\Omega')\psi(t,\mathbf{x},\Omega')\,\mathrm{d}\Omega' \\ \psi(t=0,\mathbf{x},\Omega) &= S(E_{\max})\rho(\mathbf{x})\psi(E_{\max},\mathbf{x},\Omega). \end{aligned} \tag{9}$$

Hence, the CSD equation can be treated numerically with classical closure methods and space-time discretizations. Note that a drawback of this formulation is the inverse tissue density arising in the flux term. This term can require significantly reduced timestep sizes when small densities arise in the CT scan. Let us first discuss methods to discretize the directional domain, yielding macroscopic evolution equations.

3 MACROSCOPIC MODELS

This section discusses macroscopic models to Equation (9). These models are derived from nodal and modal discretizations of the directional domain. Let us derive an evolution equation to describe the moments of the radiation flux with respect to the real-valued spherical harmonics basis functions, which we denote by

$$\mathbf{m} = \left(m_0^0, m_1^{-1}, m_1^0, m_1^1, \cdots, m_N^N\right)^T \in \mathbb{R}^{(N+1)^2}.$$

We wish to determine the expansion coefficients with respect to the spherical harmonics also called moments

$$u_\ell^k(t,\mathbf{x}) := \int_{\mathbb{S}^2} \psi(t,\mathbf{x},\Omega) m_\ell^k(\Omega)\,\mathrm{d}\Omega.$$

Moreover, there exists a diagonal matrix $\Sigma_\ell(t)$ with entries $\Sigma_{\ell,kk} = \Sigma_\ell^k := 2\pi \int_{[-1,1]} P_\ell(\mu)\Sigma_s(t,\mu)\,\mathrm{d}\mu$ where P_ℓ are the Legendre polynomials such that

$$\int_{\mathbb{S}^2}\int_{\mathbb{S}^2} m_\ell^k(\Omega)\Sigma_s(t,\Omega\cdot\Omega')\psi(t,\mathbf{x},\Omega')\,\mathrm{d}\Omega' d\Omega = \Sigma_\ell^k(t)u_\ell^k(t,\mathbf{x}).$$

Deriving evolution equations for a finite number of moments by testing the original transport equation with the spherical harmonics bases up to some order N leads to equations that depend on the unknown moments of order $N+1$, i.e., we need to find a closure. The most commonly used closure sets the moments of order $N+1$ to zero. The resulting method is called the spherical harmonics (P_N) method [15]

$$\partial_t \mathbf{u}(t,\mathbf{x}) = -\mathbf{A}\cdot\nabla_{\mathbf{x}}\frac{\mathbf{u}(t,\mathbf{x})}{\rho(\mathbf{x})} - \Sigma_t(t)\mathbf{u}(t,\mathbf{x}) + \Sigma\mathbf{u}(t,\mathbf{x}),$$

where $\mathbf{A}\cdot\nabla_{\mathbf{x}} := \mathbf{A}_1\partial_x + \mathbf{A}_2\partial_y + \mathbf{A}_3\partial_z$ with $\mathbf{A}_i := \int_{\mathbb{S}^2} \mathbf{m}\mathbf{m}^T\Omega_i\,\mathrm{d}\Omega$ and $\Sigma = \mathrm{diag}(\Sigma_0^0, \Sigma_1^{-1}, \Sigma_1^0, \Sigma_1^1, \ldots, \Sigma_N^N)$. While P_N is a computationally efficient method (especially for scattering terms), it does not preserve the positivity of the radiation flux approximation and can lead to spurious oscillations [53]. A closure that mitigates oscillations and preserves positivity at significantly increased computational costs is the M_N closure.

The M_N closure [46, 47] employs the principle of minimal mathematical, i.e., maximal physical entropy to close the moment system. To this end, we define the twice differentiable, strictly convex kinetic entropy density $\eta : \mathbb{R}_+ \to \mathbb{R}$. Different, problem-specific entropy densities can be defined, e.g., the Maxwell–Boltzmann entropy $\eta(g) = g\log(g) - g$. Thus, one can close the system by choosing the reconstructed radiation flux density $\psi_{\mathbf{u}}$ that fulfills

$$\psi_{\mathbf{u}} = \arg\min_g \int_{\mathbb{S}^2} \eta(g)\,\mathrm{d}\Omega \quad \text{s.t. } \mathbf{u} = \int_{\mathbb{S}^2} \mathbf{m}g\,\mathrm{d}\Omega. \tag{10}$$

Using the minimizer $\psi_{\mathbf{u}}$ to close the moment system, we obtain

$$\partial_t \mathbf{u}_\ell(t,\mathbf{x}) + \nabla_x \cdot \int_{\mathbb{S}^2} \Omega \otimes \mathbf{m}_\ell(\Omega) \frac{\psi_{\mathbf{u}}(t,\mathbf{x},\Omega)}{\rho(\mathbf{x})} \, \mathrm{d}\Omega + \Sigma_t(t)\mathbf{u}_\ell(t,\mathbf{x}) = \Sigma_\ell \mathbf{u}_\ell(t,\mathbf{x}). \tag{11}$$

The minimal entropy method preserves important properties of the underlying equation [4, 47], i.e., the positivity of the solution, hyperbolicity of the moment system, dissipation of mathematical entropy, and the H-Theorem. The minimal entropy closed moment system is invariant under Galilean transforms. Last, if collision invariants of the Boltzmann equations are used as modal basis functions, then the moment system yields a local conservation law.

In practice, the dual formulation of the optimization problem (10) is solved with a Newton method with line-search. We also implement more efficient neural network–based solvers, as presented in Reference [61]. There, a neural network $\mathrm{N}_\theta(\mathbf{u})$ is used to approximate the objective function of (10) at its optimal point $\psi_{\mathbf{u}}$, i.e.,

$$\mathrm{N}_\theta(\mathbf{u}) \approx \int_{\mathbb{S}^2} \eta(\psi_{\mathbf{u}}) \, \mathrm{d}\Omega, \tag{12}$$

and recover $\psi_{\mathbf{u}}$ via the dual formulation and first-order optimality conditions. However, the entropy closure becomes ill conditioned [4] and thus, a numerical optimizer requires a large number of iterations to compute a solution. To mitigate this issue, a regularized version of the entropy closure problem has been proposed by Reference [4], which adds a regularization term to the optimization problem (10). Then Equation (10) is reformulated as

$$\min_g \int_{\mathbb{S}^2} \eta(g) \, \mathrm{d}\Omega + \frac{1}{\gamma} \left\| \mathbf{u} - \int_{\mathbb{S}^2} \mathbf{m} g \, \mathrm{d}\Omega \right\|_2^2, \tag{13}$$

where $\gamma > 0$ is a regularization parameter.

The S_N method [48] employs a nodal discretization for the directional domain. To facilitate the computation of integral terms that arise due to scattering, the nodal point sets are commonly chosen according to a quadrature rule with points Ω_q and weights w_q with $q = 1, \cdots, Q$. Possible quadratures include the product rule, spherical Monte Carlo, Levelsymmetric [50], Lebedev [51], and LDFESA [33]. A comparison of different quadrature sets and their approximation behavior for S_N methods can be found in Reference [14].

The evolution equations for $\psi_q(t,\mathbf{x}) := \psi(t,\mathbf{x},\Omega_q)$ are then given by

$$\partial_t \psi_q(t,\mathbf{x}) + \Omega_q \cdot \nabla_x \frac{\psi_q(t,\mathbf{x})}{\rho(\mathbf{x})} + \Sigma_t(t)\psi_q(t,\mathbf{x}) = \sum_{p=1}^{Q} w_p \Sigma_s(t, \Omega_q \cdot \Omega_p)\psi_p(t,\mathbf{x}). \tag{14}$$

A main disadvantage of S_N methods are so-called ray-effects [43, 52, 55], which are spurious artifacts that stem from the limited number of directions in which particles can travel. Moreover, radiation therapy applications exhibit forward-peaked scattering, which cannot be well captured by classical quadrature rules. To allow for moderate computational costs when computing scattering terms and to efficiently treat forward-peaked scattering, one can transform the nodal solution to a modal description and apply the more efficient P_N methodology for scattering terms.

4 DISCRETIZATION METHODS

4.1 Spatial Discretization

The KiT-RT framework is based on unstructured, cell-centered grids as spatial discretization. In the following, we restrict ourselves to a two-dimensional spatial grid; however, the notations can be straightforwardly extended to three or one spatial dimension. An unstructured grid $\widetilde{\mathbf{X}} = \{\mathbf{X}_i\}_{i \in I}$ is a partition of a bounded spatial domain $\mathbf{X} \subset \mathbb{R}^d$. A grid cell $\mathbf{X}_i$ holds information about the

coordinates of its centroid $\mathbf{x}_i$, its measure A_i, the indices of its boundary vertices, indices of its neighbor cells $N(i)$ and cell faces. The information of the cell faces is encoded in the unit-normal vector of the face dividing cell i and its neighbor $j \in N(i)$, multiplied with the measure of the face and is denoted by $\mathbf{n}_{i,j}$. The grids used in this work are either triangular or quadrilateral, unstructured grids in two spatial dimensions.

4.2 Finite-volume Methods

The nodal and modal methods are different approaches to discretizing the velocity space of the Boltzmann equation. However, all of them result in a system of transport equations that can be solved using a finite-volume scheme. Thus, we first describe a method agnostic finite-volume scheme and afterwards point out the differences of the S_N, P_N, and M_N-based implementations. We denote the temporal variable by t; however, the results hold for energy interpreted as pseudo-time as well. Let $\mathbf{g}(t, \mathbf{x}) \in \mathbb{R}^m$ be the vector of conserved variables of a system of transport equations,

$$\partial_t \mathbf{g}(t, \mathbf{x}) + \nabla_{\mathbf{x}} \cdot \mathbf{F}(\mathbf{g}(t, \mathbf{x})) = \mathbf{R}(t, \mathbf{x}, \mathbf{g}(t, \mathbf{x})), \qquad \mathbf{x} \in \mathbf{X},\ t \in [0, t_f), \tag{15}$$

where $\mathbf{F}$ is the general flux function describing the solution transport and $\mathbf{R}$ is a general right-hand side, containing velocity discretizations of collision terms, sources, and absorption terms. The main discretization strategy is to divide the spatial domain into an unstructured grid with N_x cells and the time domain into N_t discrete values $0 = t_0 < \cdots < t_{N_t - 1}$. We consider the solution as an average over one space-time cell,

$$\mathbf{g}_i^n = \frac{1}{A_i} \int_{\mathbf{X}_i} \mathbf{g} \, \mathrm{d}\mathbf{x}, \tag{16}$$

and average Equation (15) over one space-time cell,

$$\begin{aligned} &\frac{1}{\Delta t A_i} \int_{\mathbf{X}_i} \int_{t_n}^{t_{n+1}} \partial_t \mathbf{g}(t, \mathbf{x}) \, \mathrm{d}t \, \mathrm{d}\mathbf{x} + \\ &\frac{1}{\Delta t A_i} \int_{\mathbf{X}_i} \int_{t_n}^{t_{n+1}} \nabla_{\mathbf{x}} \cdot F(\mathbf{g}(t, \mathbf{x})) \, \mathrm{d}t \, \mathrm{d}\mathbf{x} = \frac{1}{\Delta t A_i} \int_{\mathbf{X}_i} \int_{t_n}^{t_{n+1}} R(t, \mathbf{x}, \mathbf{g}(t, \mathbf{x})) \, \mathrm{d}t \, \mathrm{d}\mathbf{x}, \end{aligned} \tag{17}$$

where $\Delta t = t_{n+1} - t_n$. Solving the integrals using the Gauss theorem for the advection term and an explicit Euler scheme for the time derivative yields

$$\begin{aligned} &\frac{1}{\Delta t} \left(\mathbf{g}_i^{n+1} - \mathbf{g}_i^n \right) + \\ &\frac{1}{\Delta t A_i} \int_{t_n}^{t_{n+1}} \sum_{j \in N(i)} F(\mathbf{g}(t, x_{i,j})) \cdot \mathbf{n}_{i,j} \, \mathrm{d}t = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} R(t, \mathbf{x}, \mathbf{g}(t, x_i)) \, \mathrm{d}t, \end{aligned} \tag{18}$$

where $g(t, x_i)$ is the conserved variable evaluated at cell i and $g(t, x_{i,j})$ is the conserved variable evaluated at the interface between cell i and its neighbor j. To compute the actual value of $\mathbf{g}_j^{n+1}$, one needs to find approximations for the flux integral. A common ansatz is of the form

$$F(\mathbf{g}_j^n, \mathbf{g}_i^n) \approx \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} F(\mathbf{g}(t, x_{i,j})) \cdot \mathbf{n}_{i,j} \, \mathrm{d}t, \tag{19}$$

where the numerical flux $F(\mathbf{g}_j^n, \mathbf{g}_i^n)$ at face (i, j) is approximated using the cell averaged conserved variable at cell i and j. For transport equations, a well-known numerical flux is given by the Upwind scheme [45]

$$F(\mathbf{g}_j^n, \mathbf{g}_i^n)_{up} = F(\mathbf{g}_i^n) \cdot \mathbf{n}_{i,j} H(\mathbf{n}_{i,j} \cdot \mathbf{v}) + F(\mathbf{g}_j^n) \cdot \mathbf{n}_{i,j} \left(1 - H(\mathbf{n}_{i,j} \cdot \mathbf{v}) \right), \tag{20}$$

where H is the heaviside step function and $\mathbf{v}$ is the transport velocity vector. Finally, we approximate the source, absorption and collision terms using the current cell average. Thus the explicit solution iteration of a first-order scheme reads

$$\mathbf{g}_i^{n+1} = \mathbf{g}_i^n - \frac{\Delta t}{A_i} \sum_{j \in N(i)} F(\mathbf{g}_j^n, \mathbf{g}_i^n)_{up} + \Delta t R(t, \mathbf{x}_i, \mathbf{g}_i^n). \tag{21}$$

Since the scattering term R is commonly stiff, **implicit-explicit (IMEX)** schemes can be used to remove influences of scattering from timestep restrictions. If we assume a linear scattering term, that is with a given matrix $\mathbf{R}_i^{n+1}$, then we have $R(t^{n+1}, \mathbf{x}_i, \mathbf{g}_i^n) = \mathbf{R}_i^{n+1}\mathbf{g}_i^n$, and the IMEX scheme reads

$$(\mathbf{I} - \Delta t \mathbf{R}_i^{n+1})\mathbf{g}_i^{n+1} = \mathbf{g}_i^n - \frac{\Delta t}{A_i} \sum_{j \in N(i)} F(\mathbf{g}_j^n, \mathbf{g}_i^n)_{up}. \tag{22}$$

4.3 Second-order Finite-volume Schemes

The KiT-RT solver provides the option to evaluate the space and time discretizations using second-order accurate schemes. To this end, we use a Heun scheme for the temporal discretization and second-order upwind flux for the numerical flux [8]. Whereas first-order spatial fluxes assume a constant solution value $\mathbf{g}_i^n$ in a cell i, a second-order upwind scheme is based on a linear reconstruction of the conserved variable. Therefore the inputs $\mathbf{g}_i^n$ and $\mathbf{g}_j^n$ to the numerical flux of Equation (20) are replaced by

$$\tilde{\mathbf{g}}_i^n = \mathbf{g}_i^n + \Psi_i \nabla_{\mathbf{x}} \mathbf{g}_i^n \cdot \mathbf{r}_{i,j}, \tag{23}$$

$$\tilde{\mathbf{g}}_j^n = \mathbf{g}_j^n + \Psi_j \nabla_{\mathbf{x}} \mathbf{g}_j^n \cdot \mathbf{r}_{j,i}, \tag{24}$$

where $\mathbf{r}_{i,j}$ is the vector pointing from cell centroid $\mathbf{x}_i$ of cell i to the interface midpoint between cells i and j and Ψ_i is the flux limiter for cell i. This reconstruction is formally second-order accurate on regular grids [3] assuming exact evaluation of the gradient $\nabla_{\mathbf{x}} \mathbf{g}_i^n$. The gradient of the conserved variable $\mathbf{g}_i^n$ is evaluated using the Green–Gauss theorem with interpolated solution values at the cell interfaces,

$$\nabla_{\mathbf{x}} \mathbf{g}_i^n \approx \frac{1}{A_i} \sum_{j \in N(i)} \frac{1}{2} \left(\mathbf{g}_i^n + \mathbf{g}_j^n \right) \cdot \mathbf{n}_{\mathbf{i,j}}. \tag{25}$$

Second- or higher-order upwind spatial discretizations require the use of flux limiters to prevent the generation of oscillations in shock regions and to achieve a monotonicity-preserving scheme. In the KiT-RT package, the Barth and Jespersen [8] limiter as well as the Venkatakrishnan limiter [70] are implemented. As an example, we show the computation of the Barth and Jespersen limiter at cell i,

$$\Psi_i = \min_{j \in N(i)} \begin{cases} \min(1, \frac{\mathbf{g}_{\max} - \mathbf{g}_i}{\Delta_2}), & \text{if } \Delta_2 > 0 \\ \min(1, \frac{\mathbf{g}_{\min} - \mathbf{g}_i}{\Delta_2}), & \text{if } \Delta_2 < 0 \\ 1, & \text{else} \end{cases}, \tag{26}$$

where we have

$$\Delta_2 = \frac{1}{2} \nabla_{\mathbf{x}} \mathbf{g}_i^n \cdot \mathbf{r}_{i,j}, \tag{27}$$

$$\mathbf{g}_{\max} = \max(\mathbf{g}_i, \mathbf{g}_j), \tag{28}$$

$$\mathbf{g}_{\min} = \min(\mathbf{g}_i, \mathbf{g}_j). \tag{29}$$

The second-order Heun scheme for temporal discretization is a two-step Runge–Kutta scheme with the iteration formula

$$\begin{aligned} \mathbf{g}_i^* &= \mathbf{g}_i^n - \frac{\Delta t}{A_i} \sum_{j \in N(i)} F(\mathbf{g}_j^n, \mathbf{g}_i^n)_{up} + \frac{\Delta t}{A_i} R(\mathbf{g}_i^n), \\ \mathbf{g}_i^{**} &= \mathbf{g}_i^* - \frac{\Delta t}{A_i} \sum_{j \in N(i)} F(\mathbf{g}_j^*, \mathbf{g}_i^*)_{up} + \frac{\Delta t}{A_i} R(\mathbf{g}_i^*), \\ \mathbf{g}_i^{n+1} &= \frac{1}{2} \left(\mathbf{g}_i^n + \mathbf{g}_i^{**} \right), \end{aligned} \tag{30}$$

which is based on the implicit trapezoidal integration method.

4.4 Numerical Fluxes

In the following, we adapt the introduced numerical methods to the method-specific notation and present the detailed implementation. The space- and time-averaged conservative variables $\mathbf{g}_i^n$ for the nodal discretization at cell i and timestep n are given by the vector of the radiation flux $\boldsymbol{\psi}_i^n = [\psi_1, \ldots, \psi_{N_q}]^T \in \mathbb{R}^{N_q}$ evaluated at the quadrature points. The different methods are distinguishable by their numerical flux function. The corresponding numerical flux for the S_N method is given by

$$F(\boldsymbol{\psi}_\mathbf{i}^\mathbf{n}) = \Omega \otimes \frac{\boldsymbol{\psi}_i^n}{\rho(\mathbf{x_i})} \tag{31}$$

and the corresponding upwind flux reads

$$F(\boldsymbol{\psi}_j^n, \boldsymbol{\psi}_i^n)_{up} = \Omega \cdot \mathbf{n}_{i,j} \frac{\boldsymbol{\psi}_i^n}{\rho(\mathbf{x_i})} H(\mathbf{n}_{i,j} \cdot \Omega) + \Omega \cdot \mathbf{n}_{i,j} \frac{\boldsymbol{\psi}_j^n}{\rho(\mathbf{x_j})} \left(1 - H(\mathbf{n}_{i,j} \cdot \Omega) \right). \tag{32}$$

For modal discretizations, $\mathbf{g}_i^n$ is given by the moment vector $\mathbf{u}_i^n \in \mathbb{R}^{(N+1)^2}$ at cell i and timestep n. The numerical flux for the P_N method then reads

$$F(\mathbf{u}_\mathbf{i}^\mathbf{n}) = \left[\mathbf{A}_1 \mathbf{u}_i^n, \mathbf{A}_2 \mathbf{u}_i^n, \mathbf{A}_3 \mathbf{u}_i^n \right]^T, \tag{33}$$

where A_i are the flux Jacobians emerging from the spherical harmonics recursion scheme. To evaluate the numerical flux with an upwind scheme, we decompose the flux Jacobians in their positive and negative definite parts,

$$A_l = A_l^+ + A_l^-, \qquad l = 1, \ldots, d. \tag{34}$$

Then the numerical flux is given by

$$F(\mathbf{u}_i^n, \mathbf{u}_j^n)_{up} = \sum_{l=1}^{d} \left(A_l^+ \frac{\mathbf{u}_i^n}{\rho(\mathbf{x}_i)} + A_l^- \frac{\mathbf{u}_j^n}{\rho(\mathbf{x}_j)} \right) n_l H(n_l) + \left(A_l^- \frac{\mathbf{u}_i^n}{\rho(\mathbf{x}_i)} + A_l^+ \frac{\mathbf{u}_j^n}{\rho(\mathbf{x}_j)} \right) n_l \left(1 - H(n_l) \right). \tag{35}$$

In contrast to the P_N method, the flux function of the M_N method cannot be expressed as a matrix multiplication but reads

$$F(\mathbf{u}_\mathbf{i}^\mathbf{n}) = \int_{\mathbb{S}^2} \Omega \otimes \mathbf{m}(\Omega) \psi_{\mathbf{u}_n^i}(\Omega) \, \mathrm{d}\Omega, \tag{36}$$

where $\psi_{\mathbf{u}_n^i}$ is the reconstructed radiation flux density of the minimal entropy closure at the cell averaged moment $\mathbf{u}_n^i$. Using a quadrature rule for the velocity integral discretization and a numerical

flux for every quadrature point, we arrive at the kinetic numerical flux,

$$F(u_j^n, u_i^n)_{up} = \sum_{q=1}^{Q} w_q \mathbf{m}_q \Omega_q \cdot \mathbf{n}_{i,j} \left[\frac{\psi_{\mathbf{u}_i^n, q}}{\rho(\mathbf{x}_i)} H(\Omega_q \cdot \mathbf{n}_{i,j}) + \frac{\psi_{\mathbf{u}_j^n, q}}{\rho(\mathbf{x}_j)} \left(1 - H(\Omega_q \cdot \mathbf{n}_{i,j})\right) \right]. \tag{37}$$

Note that the updated solution of the M_N method must still be a feasible moment for the minimal entropy closure of Equation (10). To ensure this, one must employ a flux-limiter [38], construct a realizability reconstruction [40], or employ the regularized entropy closure formulation.

The numerical framework supports the usual Neumann and Dirichlet boundary conditions.

5 SOFTWARE ARCHITECTURE

The design principle of the KiT-RT software package is focused on efficient implementation, high re-usability of its components, and ease of extension. It contains a set of efficient numerical solvers for radiation transport, which are constructed of basic, reusable building blocks. These building blocks can be freely arranged to implement new solvers or tools for completely different applications. However, KiT-RT is equipped with an easy-to-use command line interface based on readable configuration files, which allows easy manipulation of the solvers. Thus the software is attractive for developers, who want to experiment with the framework and build their numerical solvers as well as users and application engineers, who want to gain experimental insights without directly interfering with the codebase.

KiT-RT is implemented in modern C++ and mainly uses polymorphism for its construction. In the following, we present the class structures used to build the numerical solvers and explain the used building blocks, which are displayed in Figure 1. Most building blocks consist of a virtual base class, which contains a static factory method to build an instance of the concrete derived class, defined by the given configuration details. Furthermore, the virtual base class defines the interface of this building block with other parts of the KiT-Framework.

5.1 Solver Class

The virtual SolverBase class is the basic blueprint for all time- or energy-dependent finite-volume solvers of the KiT-RT framework. It holds an instance of the Config, NumericalFlux, ProblemBase, QuadratureBase, and Mesh class.

It controls the screen, log, and volume output of the solver. The screen output provides instantaneous feedback of the solver state via the command line and gives information on the current iteration, the total mass of the system, the residual of the radiation flux as well as the flow field, and whether logs and volume outputs have been written to file. The file log carries the same information as the screen output in a tabular format. Last, the volume output consists of vtk files with solver and problem-specific solution data. The output data can be specified in the solver configuration.

The method `Solve()` of the SolverBase class drives the execution of all derived solvers by iterating over the time discretization of the numerical methods described in Section 4. This main time iteration is displayed in Algorithm 1. Each command is specified in the derived solver classes, such as the P_N solver, and does not induce any additional communication overhead for the parallelization architecture. The class PNSolver inherits from SolverBase. It does not own additional instances of other custom building blocks and overwrites the sub-routines of Algorithm 1 for the P_N equation-specific numerical method, which allows for runtime solver assembly. Its child class is the CSDPNSolver, which is the implementation of the P_N-based continuous slowing-down solver, that overwrites the solver-preprocessing routines for the continuous slowing down

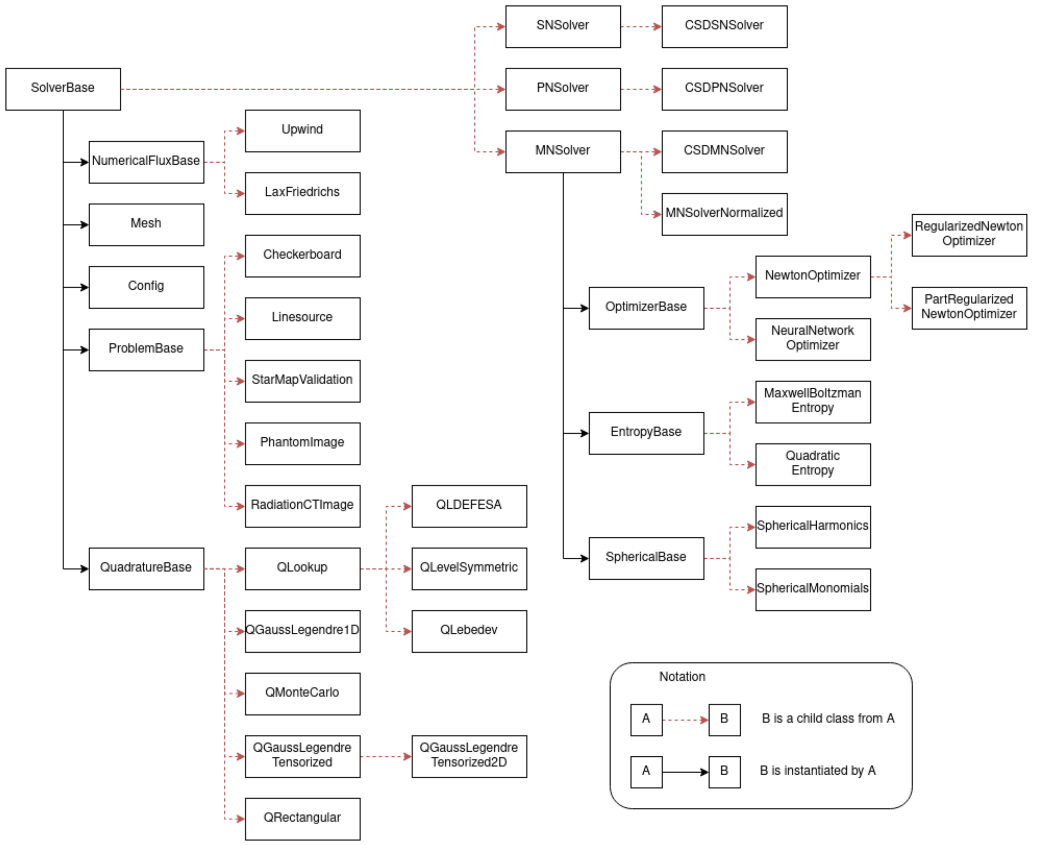


Fig. 1. Class and inheritance structure of the virtual SolverBase Class. Each instantiated Solver has class members and routines specific to its numerical structure.

specific energy transformation. The P_N-based solvers produce radiation flux and moments as output.

The class SNSolver adapts the sub-routines of Algorithm 1 for the ordinate-based numerical methods and is the parent class of the CSDSNSolver. S_N-based solvers produce the radiation flux as output.

Last, the MNSolver class contains the implementation of the M_N numerical method. It further holds the module SphericalBase, which controls the choice of basis functions $\mathbf{m}(\mathbf{v})$ of the velocity space, the module EntropyBase, which controls the choice of the entropy functional for the entropy closure, and, last, the module OptimizerBase, which controls the choice of numerical optimizer used to compute the entropy closure. The class CSDMNSolver inherits from the MNSolver class and analogously overwrites the sub-routines of Algorithm 1 for the continuous-slowing down equations. M_N-based solvers produce the radiation flux, moments, and dual variable of the entropy closure as output.

5.2 Mesh Class

The mesh class handles the computational meshes of the spatial discretization of the underlying differential equation. It can handle one-dimensional (1D) meshes and 2D unstructured triangular

ALGORITHM 1: Solution algorithm for `SolverBase.Solve()`

Input: $\mathbf{u}_i^0$ or $\boldsymbol{\psi}_i^0$: Initial condition of the test case
$\widetilde{\mathbf{X}}$: Mesh
$\mathbf{R}_i$: Boundary and source terms

Result: $\mathbf{u}_i^{n_{tf}}$ or $\boldsymbol{\psi}_i^{n_{tf}}$: Solution values at final time n_{tf} (and intermediate results, if specified)

if *modal* **then**
  $\mathbf{g}_i^0 \leftarrow \mathbf{u}_i^0 \quad \forall i \in I$ /* Modal initial Condition */
end
if *nodal* **then**
  $\mathbf{g}_i^0 \leftarrow \boldsymbol{\psi}_i^0 \quad \forall i \in I$ /* Nodal initial Condition */
end
for $n = 0, \dots, n_{tf}$ **do**
  $\mathbf{g}_i^*, \mathbf{g}_i^{k=0} \leftarrow \mathbf{g}_i^n \qquad \forall i \in I$ /* Runge Kutta initialization */
  for $k = 0, \dots, k_{rk} - 1$ **do**
    if M_N *method* **then**
      $\boldsymbol{\psi}_{\mathbf{u},i}^k \leftarrow \texttt{Closure}(\mathbf{u}_i^n) \quad \forall i \in I$ /* Entropy closure */
    end
    if 2^{nd} *order* **then**
      $\Psi_i^k \leftarrow \texttt{SlopeLimiter}(i,j) \quad \forall i \in I$ /* Compute and limit spatial slopes */
    end
    $\mathbf{F}(\mathbf{g}_i^k) \leftarrow \sum_{j \in N(i)} \mathbf{F}_{\text{up,i,j}}^k \qquad \forall i \in I$ /* Flux computation */
    $\mathbf{g}_i^{k+1} \leftarrow \mathbf{g}_i^k - \frac{\Delta t}{A_i}\left(\mathbf{F}(\mathbf{g}_i^k) + \mathbf{R}(\mathbf{g}_i^k)\right) \quad \forall i \in I$ /* Pseudo time integration */
  end
  $\mathbf{g}_i^{n+1} \leftarrow \frac{1}{2}\left(\mathbf{g}_i^* + \mathbf{g}_i^{k_{rk}}\right) \qquad \forall i \in I$ /* Time integration */
end

and quadrilateral meshes in the SU2 [59] mesh format. The mesh class keeps a record of all geometry and adjacency information required for the finite-volume methods with first and second-order fluxes.

5.3 Computational Problem Class

The problem class handles the setup of computational problems and test cases. It sets the initial conditions for the solution of the numerical solver and manages space-, time-, or energy-dependent material properties for the solver. The abstract class ProblemBase holds pointers to the Mesh and Config classes and creates instances of specific problems depending on the chosen configurations. Each implemented problem has two child classes, one for the ordinate-based sovers and one for moment-based solvers. The moment-based problem classes compute the moments of the initial condition and sources of the corresponding kinetic densities specified in the ordinate-based problem class.

The solver framework comes with several pre-implemented test cases and functionalities. This includes standard 1D and 2D test cases such as line source and checkerboard for the radiative transfer solvers, as well as isotropic and directed sources with different background media that can be loaded from a user-supplied image file, for the continuous slowing down solvers. Custom test cases can be easily added by the user, based on the provided examples and our modular approach.

5.4 Quadrature Class

The virtual QuadratureBase class creates instances of specific numerical quadratures using its static factory method. The quadratures are intended to integrate over the velocity space of the Boltzmann equation; however, they can be applied to other use cases as well. The implemented quadratures are distinguished by the dimension of the integrated velocity space and the integration area. Each quadrature has a specifiable order and manages the integration points in Cartesian and spherical coordinates as well as the corresponding quadrature weights. By default, the quadrature rules integrate over the unit sphere, but the integration region can be scaled.

5.5 IO/Use of Config Files

The KiT-RT solver is a command line interface-based program and takes one argument: the configuration file. This file is parsed and the specified modules of the KiT-RT framework are arranged for a solver instance or another custom tool. The configuration file is a document containing option specifications of the form CONFIG_OPTION=VALUE. A solver configuration contains information about file input and output, where the location of the mesh file, the volume output files, and the log files are specified. Then the computational problem and the problem-specific parameters, e.g., scattering coefficient, final time, spatial dimension, and boundary conditions, are set. Next, the solver-specific options are set. In the example of an M_N solver, the choice of velocity basis, the maximal degree of the basis functions, the CFL number, spatial integration order, entropy functional, optimizer, quadrature, and quadrature order are set. Finally, quantities for the screen, volume, and log output are specified along with their output frequency. Example configuration files for the numerical results of Section 7 can be found in the Github Repository https://github.com/CSMMLab/KiT-RT.

5.6 Practices of Modern Software Development

The entire solver and associated documentation is put under the version control system git [16] to greatly enhance collaborative workflows. Additionally, the web-hosting service GitHub is used to provide global access to the code that is licensed under the open source MIT license.

To further improve collaborations, the service also acts as a central host for progress and issue tracking, deployment, and maintaining code integrity. The latter is obtained for example through automated testing in terms of unit tests, which ensure the validity of smaller code instances such as functions or classes by testing predefined inputs against their expected result. Further, regression tests are used that validate the correctness of the solver as a whole based on small test problems and compare obtained results to reference solutions. These tests are automatically executed every time code changes are submitted to the main development branches or if a merge request is opened.

If any of the automated tests fail for a new submit, then it is rejected for merging, ensuring code integrity and quality on the important development streams at all times. Combining the test information, we can further define metrics such as a test coverage describing the percentage of code lines validated by any form of testing and ultimately helping to build trust in the code framework. For the KiT-RT framework, the test coverage is reported to the coveralls.io service at https://coveralls.io/github/CSMMLab/KiT-RT. The KiT-RT framework features relatively modest software dependencies, but being able to build and run the code correctly can be troublesome on many systems. To circumvent this issue, we provide a pre-configured build environment through the containerization engine Docker [54]. These so-called Docker containers have been developed for consistent software development and deployment and work as isolated instances with a minimal software stack comparable to lightweight virtual machines. The respective specialized docker image is also publicly available at https://hub.docker.com/r/kitrt/test.

As mentioned previously, GitHub can also be used for the deployment of precompiled software packages and the associated documentation. The documentation is automatically generated as

part of a complete software build. It is written in the reStructuredText Markup language and uses the documentation framework Sphinx [11], which compiles the Markup files to a series of linked HTML files or in other words a local website. To make the website itself publicly available it is hosted by ReadTheDocs[1] service under the URL https://kit-rt.readthedocs.io.

With all these tools in mind, the development workflow can be described as follows: Starting on GitHub, each developer can create a new branch based on the development branch or fork the entire KiT-RT framework to obtain a personal workspace. After the developers have added their changes, they can file a merge request that will automatically be tested by the continuous integration processes, and a core developer will perform a code review of all changes. Provided all tests succeed and the core developer is satisfied with the added/changed code quality, it will be merged into the development branch. If enough new features have been added to the development branch, then it will be merged into the master branch and the software will obtain a new version number (major or minor).

6 PARALLEL SCALING

In the following, we investigate the parallel performance of the three base solver implementations S_N, P_N, and M_N, where we follow Reference [56] for the brief review of parallel scalings. The speedup of a parallel algorithm is defined as

$$S(n,p) = \frac{T^*(n)}{T(n,p)}, \tag{38}$$

where $T^*(n)$ is the execution time of the best inherently serial algorithm with input size n and $T(n,p)$ the time for the parallel implementation with p processing workers and input size n. In theory the best possible speedup is linear [22], i.e., $S(n,p) = p$. Further, the measure of parallel efficiency is

$$E(n,p) = \frac{S(n,p)}{p}. \tag{39}$$

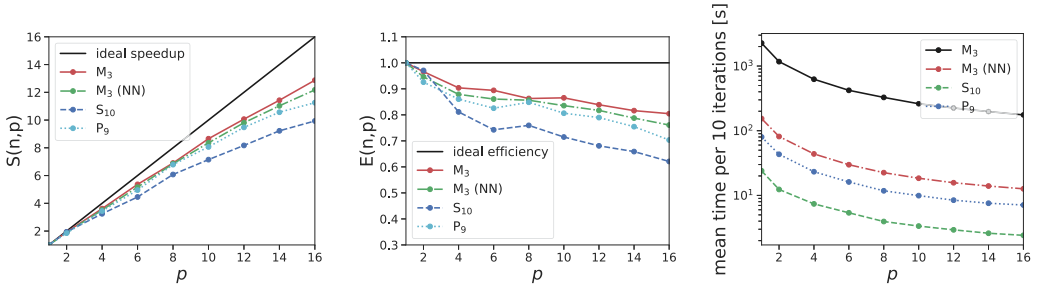
The upper bound for the speedup is given by Amdahl [29]

$$S(n,p) \leq \frac{1}{f + (1-f)/p}, \tag{40}$$

caused by communication overhead of the parallel workers as well as the fraction of inherently serial code f. In the following, we focus on numerical studies for strong scaling of KiT-RT, since the software is designed for shared memory parallelism with OpenMP on workstations with limited memory resources. Thus, we inspect the performance of the S_{10}, P_9, M_3 and neural network–based M_3 method in the checkerboard test case for a fixed grid size of $n = 700,000$ cells. The grid is partitioned and distributed among p cores, and $S(n,p)$ is displayed in Figure 2. The detailed computational setup for the methods and the simulation performance is discussed in Section 7.2 Figure 2 shows a comparison of the solvers' parallel scalings and efficiency. The M_3 methods exhibit the best speedup, caused by a high computational load per core, which reflects in overall higher timings. Additionally, M_N methods require less communication between cores, since the size of the PDE system is generally smaller than for P_N and S_N methods for similar numerical accuracy.

The performance of the continuous slowing down solver implementations follows the corresponding base solver performance, since the same spatial, velocity, and (pseudo) temporal discretizations are used.

[1] https://readthedocs.org/



(a) Strong scaling (b) Parallel efficiency (c) Mean iteration time

Fig. 2. Strong parallel scaling, and parallel efficiency, for the (neural network based) M_3, S_{10}, and P_9 solver in the checkerboard test case with 700,000 grid cells. The mean iteration time is computed per 10 iterations.

7 VALIDATION

For validation and a comparison of the implemented solvers, we consider a selection of the test cases provided within the problem class (5.3). The S_N solver is validated with a comparison with Kinetic.jl [73, 74]. The continuous slowing down solvers are further compared to a reference Monte Carlo solution computed using TOPAS [60] as well as the validated spherical harmonics solver StarMap [62].

7.1 Inhomogeneous Linesource

In the following, we compare the numerical results for the continuous slowing down approximation of our framework to a Monte Carlo solution, computed using TOPAS [60] as well as the staggered-grid spherical harmonics solver StarMap [62]. The problem considered is an inhomogeneous Linesource test case, which extends the classical Linesource benchmark [9, 25, 26] to a steady-state but energy-dependent setting. We consider a spatial domain $\mathbf{X} = [0,1]^2$ and a velocity domain $\mathbf{V} = P_{\mathbb{R}^2}(\mathbb{S}^2)$. As background density, we choose a piecewise constant function

$$\rho(\mathbf{x}) = 1 + 4 \cdot \mathbb{1}_{X_{\text{right}}}(\mathbf{x}), \qquad \mathbf{x} \in \mathbf{X}. \tag{41}$$

In the left part of the spatial domain $\mathbf{X}_{\text{left}} = [0,1] \times [0, 0.56]$ a reduced density is used, compared to the right part of the spatial domain $\mathbf{X}_{\text{right}} = [0,1] \times [0.56, 1]$. At a maximal energy of $E_{\max} = 1$, i.e., initial pseudo time, a particle beam is positioned in the center of the spatial domain $\mathbf{x}_0 = [0.5, 0.5]^T$, which is modeled as

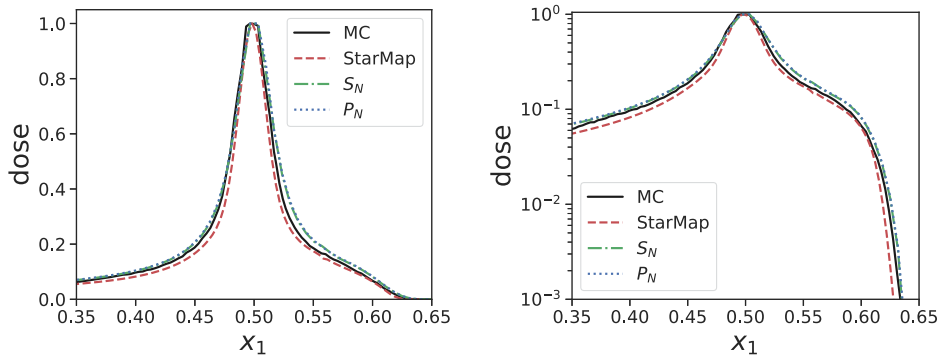
$$f(E, \mathbf{x}, \mathbf{v}) = \frac{1}{2\pi\epsilon^2} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_0\|^2}{2\sigma^2}\right), \quad E = E_{\max}, \tag{42}$$

$$f(E, \mathbf{x}, \mathbf{v}) = 0, \qquad E = E_{\max}, \mathbf{x} \in \partial\mathbf{X}. \tag{43}$$

Here a standard deviation of $\epsilon = 0.01$ is chosen to obtain a sharp particle beam in the center. No source term is used in the simulation and boundary conditions are zero-valued Dirichlet conditions, i.e.,

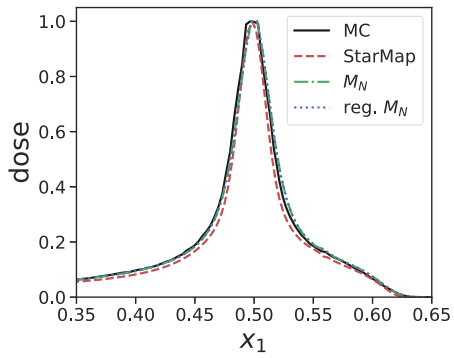
$$f(E, \mathbf{x}, \mathbf{v}) = 0, \qquad \mathbf{x} \in \partial\mathbf{X}. \tag{44}$$

The scattering cross section σ_s of the collision operator of the continuous slowing down approximation (9) is set constant to 1. The spatial grid for all deterministic methods is a structured rectangular grid with 300^2 cells. Due to the functionality of the Monte Carlo software, we use a three-dimensional grid and project the x_3-domain onto the $x_1 - x_2$ plane. To allow for feasible costs, the Monte Carlo method uses a coarser grid resolution of 100 spatial cells per dimension, and 100,000 Monte Carlo runs are computed to reduce statistical noise. The S_N solver uses a product quadrature

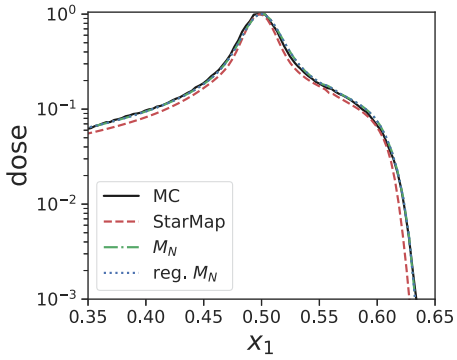


(a) S_{20}, P_8, linear scale

(b) S_{20}, P_8, log scale



(c) M_3, reg M_3, linear scale



(d) M_3, reg M_3, log scale

Fig. 3. Comparison of simulation results of deterministic and stochastic methods. The M_3 and regularized M_3 methods coincide best with the Monte Carlo reference, surpassing StarMap in accuracy, as seen in the logarithmic scale plots.

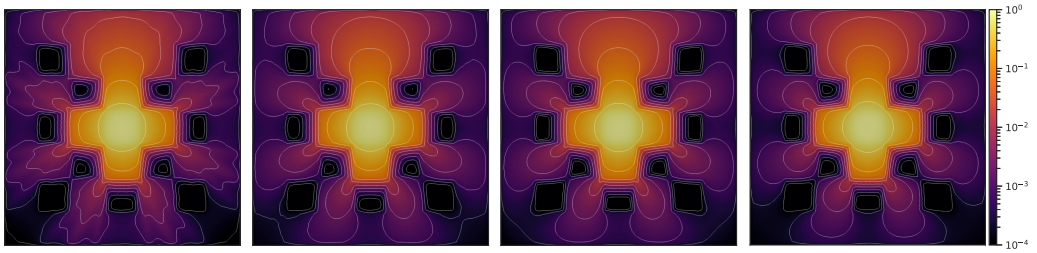
rule of order 20 for the streaming step and spherical moments up to order 8 to compute scattering terms. Similarly, the P_N solver employs spherical moments up to order 8 and the M_N solver as well as its regularized counterpart uses spherical harmonics of order 3. The timestep restriction of all deterministic methods picks a CFL number of 0.7. All methods are second order in space and time using the Barth and Jespersen slope limiter. The resulting dose profile, i.e., the zeroth-order moment of the kinetic density,

$$u_0(\mathbf{x}, t) = \int_{\mathrm{V}} f(t, \mathbf{x}, \mathbf{v}) \, \mathrm{d}\mathbf{v}, \tag{45}$$

is plotted in Figure 3 along the x_1-axis in the interval $x_1 \in [0.3, 0.65]$ at $x_2 = 0.5$. All methods show similar behavior and agree well with the Monte Carlo results. Moreover, it is observed that the regularized M_N method seems to coincide with its non-regularized counterpart and is closer to the Monte Carlo reference than the S_N and P_N methods. The StarMap solution mostly undershoots the Monte Carlo reference, whereas all KiT-RT solutions slightly overshoot it.

7.2 Checkerboard

The checkerboard test case [12] mimics a nuclear reactor block with a strong radiative source in the domain center, which is denoted by $\mathbf{X}_q$, and several highly absorptive regions $\mathbf{X}_a$ placed in a



(a) S_{10} (b) P_5 (c) M_3 (d) regularized M_3

Fig. 4. Simulation results of various KiT-RT solver for the checkerboard test-case in log scale; S_{10}, P_5, M_3, and regularized M_3 solver (left to right).

checkerboard pattern around it, see Figure 4. The corresponding time-dependent linear Boltzmann equation reads

$$\partial_t \psi + \Omega \nabla_{\mathbf{x}} \psi + \Sigma_t = \int_{\mathbb{S}^2} \Sigma_s(t, \mathbf{x}, \Omega, \Omega_*) \psi(\Omega_*) \, \mathrm{d}\Omega_* + q(\mathbf{t}, \mathbf{x}, \Omega) \tag{46}$$

for $\mathbf{x} \in [0,7]^2$, $t \in [0,10)$, and $\Omega \in P_{\mathbb{R}^2}(\mathbb{S}^2)$. This corresponds to Equation (9) with $\rho(\mathbf{x}) = 1$. We equip the equation with Dirichlet boundary conditions and initial condition,

$$\psi(t, \mathbf{x}, \Omega) = 0, \qquad \mathbf{x} \in \partial \mathbf{X}, \tag{47}$$

$$\psi(t, \mathbf{x}, \Omega) = 0, \qquad t = 0, \tag{48}$$

to obtain a well-posed problem. Furthermore, the scattering kernel k and source term q are assumed to be isotropic and constant in time. The scattering cross and absorption cross sections are given by

$$\Sigma_s(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \in X_a \\ 1 & \text{else} \end{cases}, \qquad \Sigma_t(\mathbf{x}) = \begin{cases} 10 & \mathbf{x} \in \mathbf{X}_a \\ 1 & \text{else} \end{cases}, \tag{49}$$

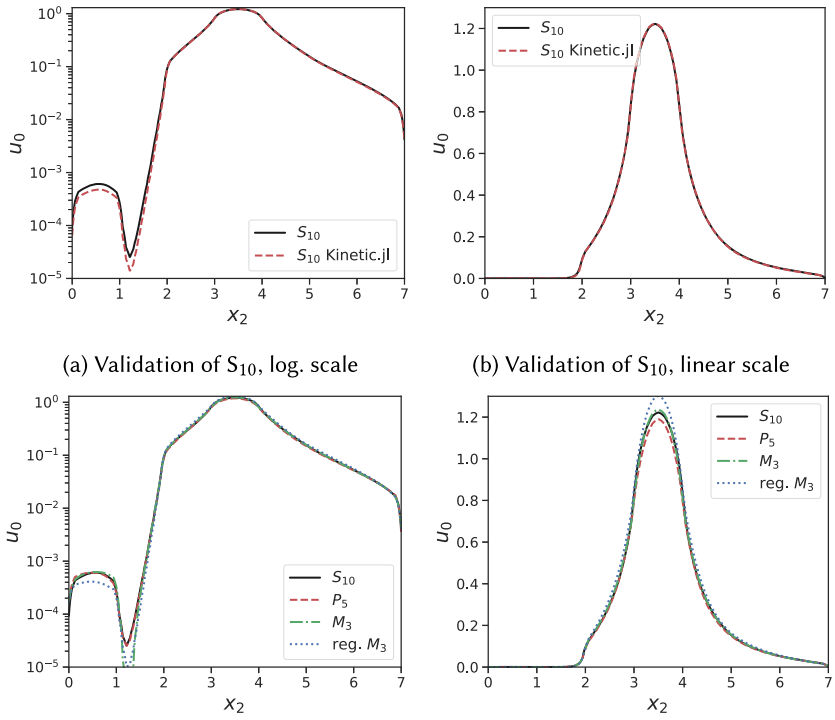
and the isotropic source is given by

$$q(\mathbf{x}, \Omega, t) = \begin{cases} 1 & \mathbf{x} \in \mathbf{X}_q \\ 0 & \text{else} \end{cases}. \tag{50}$$

We create an unstructured triangular mesh with 25,000 cells to discretize the spatial domain with regard to the absorption and source regions, such that the region boundaries coincide with the mesh faces. The simulation is computed until the final time $t_f = 10$ using various solver configurations. All employed solvers use a second-order upwind flux as the spatial discretization and a second-order explicit Runge Kutta scheme for temporal discretization with CFL number equal 0.45, since M_N solvers with non-regularized entropy closure require a CFL number smaller than 0.5 for stability [5, 38, 40]. The solution computed at final time $t_f = 10$ is displayed in Figure 4, where we can see the scalar flux

$$\Psi(\mathbf{x}, t) = \int_{\mathbb{S}^2} \psi \, \mathrm{d}\Omega, \tag{51}$$

in the contour plot in log scale. The radiation flux is highest at the source region $\mathbf{X}_q$ and almost zero in the absorption region $\mathbf{X}_a$ for all solvers. Toward the top of the domain, the radiation travels freely, whereas toward the left, right, and bottom, the radiation expansion is damped by absorption regions. Figure 4 shows the the S_{10} solver with a tensorized Gauss Legendre Quadrature, the P_5 solution with a spherical harmonics basis, the M_3 solution with a spherical harmonics basis and a



(a) Validation of S_{10}, log. scale

(b) Validation of S_{10}, linear scale

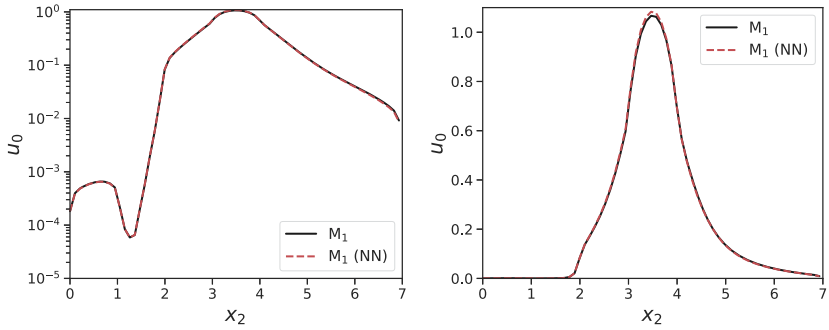
(c) KiT-RT solver comparison, log. scale

(d) KiT-RT solver comparison, linear scale

Fig. 5. Validation of the KiT-RT S_{10} solver against kinetic.jl [74] ((a) and (b)) and comparison of all KiT-RT solvers ((c) and (d)) using cross sections for the checkerboard test-case. Left: Results in log scale, right: results in linear scale.

Newton optimizer with line-search configured to accuracy 1e−7 and the regularized M_3 solution using the same optimizer and basis. As seen in Figure 5, the deviation between implementations of the S_{10} solver and the reference solution given in Reference [74] is below the 1e−3, which is the characteristic length of a grid cell. Furthermore, the cross sections of all KiT-RT-based methods coincide with grid accuracy, except for the non-regularized M_3 method. Note that the regularized M_3 solution is much closer to the other methods due to better condition of the entropy closure problem near absorption regions.

Next, we compare a non-regularized M_1 solver using a monomial basis and a Newton solver–based entropy computation with an M_1 solver using a neural network–based entropy closure [61]. In the KiT-RT package, we have compiled the trained networks in Reference [61] to C++ and have implemented a fast tensorflow [1] back-end for seamless integration into the KiT-RT framework. Both the Newton-based and neural network–based solutions correspond well, as Figure 6 shows. Only at the top of the domain, i.e., at $x \approx 6.5$ in the cross-section plot, do we see a small deviation between the methods. In this region at the wave-front of the radiative transport, the moments of the kinetic equation are close to the boundary of the realizable set, where, on the one hand, the Newton-based solver needs more iterations and thus more wall time to compute the solution to the optimization problem, and, on the other hand, the neural network accuracy declines slightly. We validate the speedup through the neural network entropy closure using a larger mesh of $700,000$ grid cells. The time consumption of the M_1 and neural network–based M_1 solver is illustrated in



(a) Neural entropy solver comparison (b) Neural entropy solver comparison

Fig. 6. Comparison of a Newton-based M_1 and neural network–based M_1 solver for the checkerboard test case in log scale (left) and linear scale (right). The solution of both methods correspond well.

Table 1. Wall Time Comparison for One Iteration of the Neural Network Based M_1 Solver Compared to the Newton-based M_1 Solver for the Checkerboard Test Case

	Mean iteration time [s]			
Comp. cores	Newton	Neural Network	Time reduction (%)	Speedup (×)
4	757.89	80.81	89.33	9.45
12	258.65	33.61	87.01	7.81

Table 1, where one can see that the neural network–based closure accelerates the computational time by 89.33–87.01%.

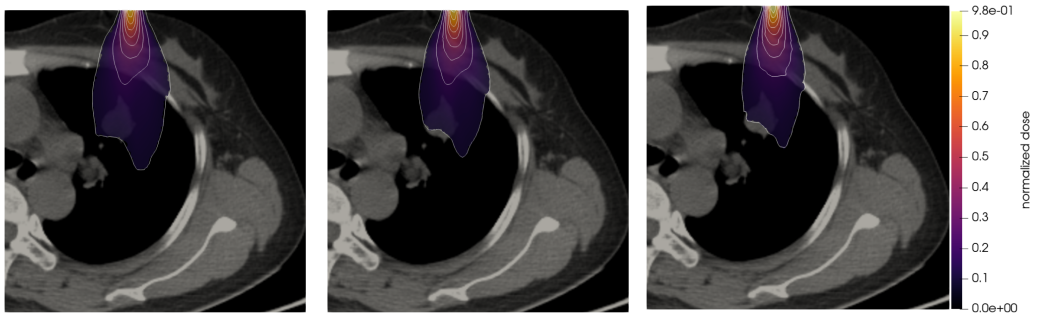
7.3 Beam in 2D Patient CT

Having validated the CSD solvers against StarMAP and a Monte Carlo framework in Section 7.1, we now examine a realistic 2D CT scan of a lung patient as a proof of concept for the application of our framework to radiation therapy computations. The patient data were retrieved from an open source dataset [49] in The Cancer Imaging Archive [18]. The patient is irradiated with an electron beam of $E_{\max} = 20$ MeV. We model this beam as the initial condition

$$\psi(E_{max}, \mathbf{x}, \Omega) = \frac{1}{(2\pi)^{3/2}\sigma_{\Omega_2}\sigma_x\sigma_y} \cdot \exp(-(\mu_{\Omega_2} - \Omega_2)^2/2\sigma_{\Omega_2}) \cdot \exp(-(\mu_{x_1} - x_1)^2/2\sigma_{x_1}) \cdot \exp(-(\mu_{x_2} - x_2)^2/2\sigma_{x_2}),$$

where $(\mu_{x_1}, \mu_{x_2}) = (2.5\text{cm}, 5.8\text{cm})$ is the beam position within the 6cm × 6cm domain and $\mu_{\Omega_2} = \frac{\pi}{2}$rad is the beam direction. The remaining parameters are chosen as $\sigma_{x_1} = \sigma_{x_2} = \sigma_{\Omega_2} = 0.1$. To determine a tissue density ρ for given gray-scale values of the CT image, we set the maximum density, represented by white pixels, to the density of bone $\rho_{\text{bone}} = 1.85\ \text{g/cm}^3$. The remaining tissue is scaled such that the minimum pixel value of zero corresponds to a minimum density of $\rho_{\min} = 0.05\ \text{g/cm}^3$. This corresponds approximately to the lower bound of observed lung densities [36].

Figure 7 compares the normalized dose for a CSD S_{13}, P_{13}, and M_5 solver. While all methods show similar behavior and can capture the effects of heterogeneities in the patient density, some differences, e.g., in the maximum depth of the S_{13} solution compared to P_{13} and M_5 or the shape of the lowest two isolines can be observed. The cross sections in Figure 8 further show that the



(a) S_{13} (b) P_{13} (c) M_5

Fig. 7. Patient CT scan with lung tumor with simulation results for the (a) S_{13}, (b) P_{13}, and (c) M_5 solver with spherical harmonics basis and partially regularized entropy.

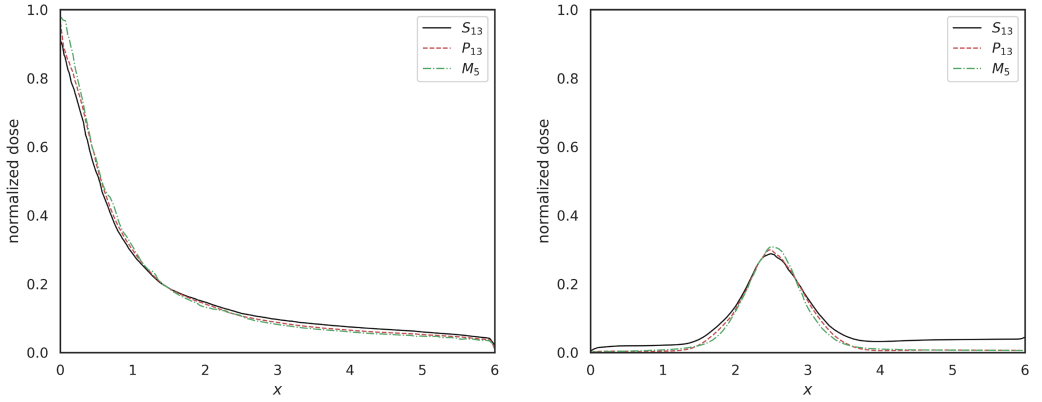


Fig. 8. Vertical (at $x = 2.5$ cm) and horizontal (at $y = 5$ cm) cross section through the normalized dose in the patient CT. Comparison of the S_{13}, P_{13}, and partially regularized M_5 solver.

Table 2. Wall Time Comparison for the S_{13}, P_{13}, and Partially Regularized M_5 Solver on Eight Cores for the 2D Beam in Patient CT Test Case

	S_{13}	P_{13}	M_5
Mean iteration time [s]	3836.82	18633.03	50729.98

S_{13} dose has a lower maximum and higher minimum value than the M_5 and to a lesser extent also P_{13} solutions. In conjunction with results from a homogeneous beam test case (see Reference [63]), this indicates that the modal methods perform slightly better for forward-peaked sources than the standard S_N method.

Table 2 presents runtime comparisons for three KiT-RT solvers. It has to be noted that the performance depends on the choice of several hyperparameters as well as the application case. For a fair comparison, it would be necessary to analyze the performance for fixed error levels and a variety of test cases with reliable reference solutions. The S_N method is the fastest, since computational cost of the numerical flux and the scattering operator is of the order of magnitude of an inner product, followed by P_N method, whos numerical flux cost is dominated by the matrix-vector evaluation of the Roe matrix and the moment vector. The M_N methods cost is mostly driven by the expense

of the entropy closure. The memory footprint of the P_N and M_N method is given by the size of the spherical harmonics basis, that is N^2 for a basis of degree N. Remark, that M_N need far less basis functions to achieve the same accuracy as P_N. For the S_N method the total system size is $2N^2$, given by the number of quadrature points of the tensorized Gauss Legendre quadrature.

The observed runtimes for S_N and P_N are slightly quicker than those of a large, parallelized MC code like TOPAS. In terms of speed, KiT-RT is currently not competitive with simplified models such as pencil beam methods. Since these do not solve the full transport problem, they can, however, not guarantee the same accuracy, especially in the presence of heterogeneities.

8 CONCLUSION

In this work, we have presented a collection of deterministic transport solvers for radiation therapy applications. The methods agree well with results obtained with conventional radiation therapy codes. Due to the use of polymorphism, we can guarantee a straightforward extension to further numerical methods, which facilitates the investigation of novel radiation therapy solvers and their comparison to conventional methods.

ACKNOWLEDGEMENTS

All authors of this work have contributed equally to this project. The authors thank Thomas Camminady for his help with implementing spherical quadrature rules.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Retrieved from https://www.tensorflow.org/

[2] IK Abu-Shumays. 2001. Angular quadratures for improved transport computations. *Transp. Theory Statist. Phys.* 30, 2-3 (2001), 169–204.

[3] Michael Aftosmis, Datta Gaitonde, and T. Sean Tavares. 1995. Behavior of linear reconstruction techniques on unstructured meshes. *AIAA J.* 33, 11 (1995), 2038–2049. https://doi.org/10.2514/3.12945 arXiv:https://doi.org/10.2514/3.12945

[4] Graham W. Alldredge, Martin Frank, and Cory D. Hauck. 2019. A regularized entropy-based moment method for kinetic equations. *SIAM J. Appl. Math.* 79, 5 (2019), 1627–1653.

[5] Graham W. Alldredge, Cory D. Hauck, and André L. Tits. 2012. High-order entropy-based closures for linear transport in slab geometry ii: A computational study of the optimization problem. *SIAM J. Sci. Comput.* 34, 4 (2012), B361–B391. https://doi.org/10.1137/11084772X arXiv:https://doi.org/10.1137/11084772X

[6] P. Andreo. 1991. Monte Carlo techniques in medical radiation physics. *Phys. Med. Biol.* 36, 7 (Jul. 1991), 861–920. https://doi.org/10.1088/0031-9155/36/7/001

[7] Richard Barnard, Martin Frank, and Michael Herty. 2012. Optimal radiotherapy treatment planning using minimum entropy models. *Appl. Math. Comput.* 219, 5 (2012), 2668–2679.

[8] Timothy Barth and Dennis Jespersen. [n. d.]. *The Design and Application of Upwind Schemes on Unstructured Meshes.* https://doi.org/10.2514/6.1989-366

[9] William Bennett and Ryan G. McClarren. 2022. Benchmarks for infinite medium, time dependent transport problems with isotropic scattering. *J. Comput. Theor. Transport* 51, 4 (2022), 205–221. https://doi.org/10.1080/23324309.2022.2103151

[10] Christoph Börgers. 1998. Complexity of monte carlo and deterministic dose-calculation methods. *Phys. Med. Biol.* 43, 3 (1998), 517.

[11] Georg Brandl. 2021. Sphinx Documentation. Retrieved from http://sphinx-doc.org/

[12] Thomas A. Brunner and James Paul Holloway. 2005. Two-dimensional time dependent Riemann solvers for neutron transport. *J. Comput. Phys.* 210, 1 (2005), 386–399.

[13] Thomas Camminady, Martin Frank, Kerstin Küpper, and Jonas Kusch. 2019. Ray effect mitigation for the discrete ordinates method through quadrature rotation. *J. Comput. Phys.* 382 (2019), 105–123.

[14] Thomas Camminady, Martin Frank, and Jonas Kusch. 2021. Highly uniform quadrature sets for the discrete ordinates method. In *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science and Engineering*. 25–29.
[15] Kenneth M. Case and Paul Frederick Zweifel. 1967. Linear transport theory.
[16] Scott Chacon and Ben Straub. 2014. *Pro Git*. Springer Nature.
[17] Hersh Chandarana, Hesheng Wang, R. H. N. Tijssen, and Indra J. Das. 2018. Emerging role of MRI in radiation therapy. *J. Magn. Reson. Imag.* 48, 6 (2018), 1468–1478.
[18] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, L. Tarbox, and F. Prior. 2013. The cancer imaging archive (TCIA): Maintaining and operating a public information repository. *J. Digit. Imag.* 26, 6 (2013), 1045–1057. https://doi.org/10.1007/s10278-013-9622-7
[19] Roland Duclous, Bruno Dubroca, and Martin Frank. 2010. A deterministic partial differential equation model for dose calculation in electron radiotherapy. *Phys. Med. Biol.* 55, 13 (2010), 3843.
[20] Jens M. Edmund and Tufve Nyholm. 2017. A review of substitute CT generation for MRI-only radiation therapy. *Radiat. Oncol.* 12 (2017), 1–15.
[21] Leonard Eyges. 1948. Multiple scattering with energy loss. *Phys. Rev.* 74, 10 (1948), 1534.
[22] V. Faber, Olaf M. Lubeck, and Andrew B. White. 1986. Superlinear speedup of an efficient sequential algorithm is not possible. *Parallel Comput.* 3, 3 (1986), 259–260. https://doi.org/10.1016/0167-8191(86)90024-4
[23] Matthias Fippel and Martin Soukup. 2004. A Monte Carlo dose calculation algorithm for proton therapy. *Med. Phys.* 31, 8 (2004), 2263–2273.
[24] Martin Frank, Jonas Kusch, Thomas Camminady, and Cory D. Hauck. 2020. Ray effect mitigation for the discrete ordinates method using artificial scattering. *Nucl. Sci. Eng.* 194, 11 (2020), 971–988.
[25] Barry D. Ganapol. 2008. *Analytical Benchmarks for Nuclear Engineering Applications*. OECD Publishing. https://doi.org/10.1016/j.anucene. 11. 07. 2008.
[26] C. Kristopher Garrett and Cory D. Hauck. 2013. A comparison of moment closures for linear kinetic transport equations: The line source benchmark. *Transport Theory Statist. Phys.* 42, 6-7 (2013), 203–235.
[27] Kent A. Gifford, John L. Horton, Todd A. Wareing, Gregory Failla, and Firas Mourtada. 2006. Comparison of a finite-element multigroup discrete-ordinates code with Monte Carlo for radiotherapy calculations. *Phys. Med. Biol.* 51, 9 (2006), 2253.
[28] V. Grégoire and T. R. Mackie. 2011. State of the art on dose prescription, reporting and recording in Intensity-Modulated Radiation Therapy (ICRU report No. 83). *Cancer/Radiothér.* 15, 6-7 (2011), 555–559.
[29] John L. Gustafson. 2011. *Amdahl's Law*. Springer, Boston, MA, 53–60. https://doi.org/10.1007/978-0-387-09766-4_77
[30] Hartmut Hensel, Rodrigo Iza-Teran, and Norbert Siedow. 2006. Deterministic model for dose calculation in photon radiotherapy. *Phys. Med. Biol.* 51, 3 (2006), 675.
[31] Kenneth R. Hogstrom, Michael D. Mills, and Peter R. Almond. 1981. Electron beam dose calculations. *Phys. Med. Biol.* 26, 3 (1981), 445.
[32] Mi Huang. 2015. *Application of Deterministic 3D SN Transport Driven Dose Kernel Methods for Out-of-field Dose Assessments in Clinical Megavoltage Radiation Therapy*. Ph. D. Dissertation. Georgia Institute of Technology.
[33] J. J. Jarrel and M. L. Adams. 2011. Discrete-ordinates quadrature sets based on linear discontinuous finite elements. In *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering* (American Nuclear Society, Downers Grove, IL).
[34] Xun Jia, Jan Schümann, Harald Paganetti, and Steve B. Jiang. 2012. GPU-based fast Monte Carlo dose calculation for proton therapy. *Phys. Med. Biol.* 57, 23 (2012), 7783.
[35] O. Koch and C. Lubich. 2007. Dynamical low-rank approximation. *SIAM J. Matrix Anal. Appl.* 29, 2 (2007), 434–454. https://doi.org/10.1137/050639703
[36] Ehiichi Kohda and Naoyuki Shigematsu. 1989. Measurement of lung density by computed tomography: implication for radiotherapy. *Keio J. Med.* 38, 4 (1989), 454–463.
[37] Thomas Krieger and Otto A. Sauer. 2005. Monte Carlo- versus pencil-beam-/collapsed-cone-dose calculation in a heterogeneous multi-layer phantom. *Phys. Med. Biol.* 50, 5 (Feb. 2005), 859–868. https://doi.org/10.1088/0031-9155/50/5/010
[38] C. Kristopher Garrett, Cory Hauck, and Judith Hill. 2015. Optimization and large scale computation of an entropy-based moment closure. *J. Comput. Phys.* 302 (2015), 573 – 590.
[39] Kerstin Kuepper. 2016. *Models, Numerical Methods, and Uncertainty Quantification for Radiation Therapy*. Ph. D. Dissertation. Universitätsbibliothek der RWTH Aachen.
[40] Jonas Kusch, Graham W. Alldredge, and Martin Frank. 2019. Maximum-principle-satisfying second-order intrusive polynomial moment scheme. *SMAI J. Comput. Math.* 5 (2019), 23–51.
[41] Jonas Kusch and Pia Stammer. 2023. A robust collision source method for rank adaptive dynamical low-rank approximation in radiation therapy. *arXiv preprint arXiv:2111.07160* (2021).

[42] Edward W. Larsen, Moyed M. Miften, Benedick A. Fraass, and Iain A. D. Bruinvis. 1997. Electron dose calculations using the method of moments. *Med. Phys.* 24, 1 (1997), 111–125.
[43] Kaye D. Lathrop. 1968. Ray effects in discrete ordinates equations. *Nucl. Sci. Eng.* 32, 3 (1968), 357–369.
[44] Kaye D. Lathrop. 1971. Remedies for ray effects. *Nucl. Sci. Eng.* 45, 3 (1971), 255–268.
[45] Randall J. LeVeque. 1992. *Numerical Methods for Conservation Laws (2nd ed.).* Birkhäuser. 1–214 pages.
[46] C. Levermore. 1996. Moment closure hierarchies for kinetic theories. *J. Statist. Phys.* 83 (1996), 1021–1065.
[47] C. David Levermore. 1997. Entropy-based moment closures for kinetic equations. *Transport Theory Statist. Phys.* 26, 4-5 (1997), 591–606.
[48] Elmer Eugene Lewis and Warren F. Miller. 1984. Computational methods of neutron transport. 17, 24 (1 1984). https://www.osti.gov/biblio/5538794
[49] P. Li, S. Wang, T. Li, J. Lu, Y. Huang Fu, and D. Wang. 2020. A large-scale CT and PET/CT dataset for lung cancer diagnosis. *Cancer Imag. Arch.* (2020). https://doi.org/10.7937/TCIA.2020.NNC2-0461
[50] Gianluca Longoni. 2004. *Advanced Quadrature Sets and Acceleration and Preconditioning Techniques for the Discrete Ordinates Method in Parallel Computing Environments.* Ph. D. Dissertation. University of Florida.
[51] G. Marchuk and V. I. Lebedev. 1986. Numerical methods in the theory of neutron transport. (1 1986). Retrieved from https://www.osti.gov/biblio/7057084
[52] Kirk A. Mathews. 1999. On the propagation of rays in discrete ordinates. *Nucl. Sci. Eng.* 132, 2 (1999), 155–180.
[53] Ryan G. McClarren, James Paul Holloway, and Thomas A. Brunner. 2008. On solutions to the Pn equations for thermal radiative transfer. *J. Comput. Phys.* 227, 5 (2008), 2864–2885. https://doi.org/10.1016/j.jcp.2007.11.027
[54] Dirk Merkel. 2014. Docker: Lightweight linux containers for consistent development and deployment. *Linux J.* 2014, 239 (2014), 2.
[55] J. E. Morel, T. A. Wareing, R. B. Lowrie, and D. K. Parsons. 2003. Analysis of ray-effect mitigation techniques. *Nucl. Sci. Eng.* 144, 1 (2003), 1–22.
[56] Kenneth Moreland and Ron Oldfield. 2015. Formal metrics for large-scale parallel performance. In *High Performance Computing*, Julian M. Kunkel and Thomas Ludwig (Eds.). Springer International Publishing, Cham, 488–496.
[57] Edgar Olbrant and Martin Frank. 2010. Generalized fokker–planck theory for electron and photon transport in biological tissues: Application to radiotherapy. *Comput. Math. Methods Med.* 11, 4 (2010), 313–339.
[58] Ricardo Otazo, Philippe Lambin, Jean-Philippe Pignol, Mark E. Ladd, Heinz-Peter Schlemmer, Michael Baumann, and Hedvig Hricak. 2021. MRI-guided radiation therapy: An emerging paradigm in adaptive radiation oncology. *Radiology* 298, 2 (2021), 248–260. https://doi.org/10.1148/radiol.2020202747 PMID: 33350894.
[59] Francisco Palacios, Juan Alonso, Karthikeyan Duraisamy, Michael Colonno, Jason Hicken, Aniket Aranake, Alejandro Campos, Sean Copeland, Thomas Economon, Amrita Lonkar, Trent Lukaczyk, and Thomas Taylor. [n. d.]. Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design. In *Proceedings of the 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition.* https://doi.org/10.2514/6.2013-287
[60] J. Perl, J. Shin, J. Schumann, B. Faddegon, and H. Paganetti. 2012. TOPAS: An innovative proton Monte Carlo platform for research and clinical applications. *Med. Phys.* 39, 11 (Nov. 2012), 6818–6837. https://doi.org/10.1118/1.4758060
[61] Steffen Schotthöfer, Tianbai Xiao, Martin Frank, and Cory Hauck. 2022. Structure preserving neural networks: A case study in the entropy closure of the boltzmann equation. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 19406–19433. https://proceedings.mlr.press/v162/schotthofer22a.html
[62] Benjamin Seibold and Martin Frank. 2014. StaRMAP—A second order staggered grid method for spherical harmonics moment equations of radiative transfer. *ACM Trans. Math. Softw.* 41, 1, Article 4 (Oct. 2014), 28 pages. https://doi.org/10.1145/2590808
[63] Pia Stammer. 2023. *Uncertainty Quantification and Numerical Methods in Charged Particle Radiation Therapy.* Ph. D. Dissertation. Karlsruher Institut für Technologie. https://doi.org/10.5445/IR/1000158316 46.21.02; LK 01.
[64] John Tencer. 2016. Ray effect mitigation through reference frame rotation. *J. Heat Transf.* 138, 11 (2016).
[65] Jouko Tervo and Pekka Kolmonen. 2002. Inverse radiotherapy treatment planning model applying Boltzmann-transport equation. *Math. Models Methods Appl. Sci.* 12, 01 (2002), 109–141.
[66] J. Tervo, P. Kolmonen, M. Vauhkonen, L. M. Heikkinen, and J. P. Kaipio. 1999. A finite-element model of electron transport in radiation therapy and a related inverse problem. *Inverse Probl.* 15, 5 (1999), 1345.
[67] J. Tervo, M. Vauhkonen, and E. Boman. 2008. Optimal control model for radiation therapy inverse planning applying the Boltzmann transport equation. *Lin. Algebr. Appl.* 428, 5-6 (2008), 1230–1249.
[68] Oleg N. Vassiliev, Todd A. Wareing, Ian M. Davis, John McGhee, Douglas Barnett, John L. Horton, Kent Gifford, Gregory Failla, Uwe Titt, and Firas Mourtada. 2008. Feasibility of a multigroup deterministic solution method for three-dimensional radiotherapy dose calculations. *Int. J. Radiat. Oncol. Biol. Phys.* 72, 1 (2008), 220–227.

[69] Oleg N. Vassiliev, Todd A. Wareing, John McGhee, Gregory Failla, Mohammad R. Salehpour, and Firas Mourtada. 2010. Validation of a new grid-based Boltzmann equation solver for dose calculation in radiotherapy with photon beams. *Phys. Med. Biol.* 55, 3 (Jan. 2010), 581–598. https://doi.org/10.1088/0031-9155/55/3/002

[70] V. Venkatakrishnan. [n. d.]. On the accuracy of limiters and convergence to steady state solutions. In *31st Aerospace Sciences Meeting*. 880. https://doi.org/10.2514/6.1993-880 arXiv:https://arc.aiaa.org/doi/pdf/10.2514/6.1993-880

[71] Hans-Peter Wieser, Eduardo Cisternas, Niklas Wahl, Silke Ulrich, Alexander Stadler, Henning Mescher, Lucas-Raphael Müller, Thomas Klinge, Hubert Gabrys, Lucas Burigo, et al. 2017. Development of the open-source dose calculation and optimization toolkit matRad. *Med. Phys.* 44, 6 (2017), 2556–2568.

[72] M. K. Woo and J. R. Cunningham. 1990. The validity of the density scaling method in primary electron transport for photon and electron beams. *Med. Phys.* 17, 2 (1990), 187–194. https://doi.org/10.1118/1.596497

[73] Tianbai Xiao. 2021. A flux reconstruction kinetic scheme for the Boltzmann equation. *J. Comput. Phys.* 447 (2021), 110689.

[74] Tianbai Xiao. 2021. Kinetic.jl: A portable finite volume toolbox for scientific and neural computing. *J. Open Source Softw.* 6, 62 (2021), 3060. https://doi.org/10.21105/joss.03060

Received 12 May 2022; revised 11 July 2023; accepted 9 October 2023