Mass conservative limiting and applications to the approximation of the steady-state radiation transport equations

Jean-Luc Guermond, Zuodong Wang

▸ To cite this version:

Jean-Luc Guermond, Zuodong Wang. Mass conservative limiting and applications to the approximation of the steady-state radiation transport equations. 2024. hal-04770516

HAL Id: hal-04770516
https://hal.science/hal-04770516v1

Preprint submitted on 6 Nov 2024

MASS CONSERVATIVE LIMITING AND APPLICATIONS TO THE APPROXIMATION OF THE STEADY-STATE RADIATION TRANSPORT EQUATIONS

JEAN-LUC GUERMOND[*][‡] AND ZUODONG WANG[†]

Abstract. A limiting technique for scalar transport equations is presented. The originality of the method is that it does not requires solving nonlinear optimization problems nor does it rely on the construction of a low-order approximation. The method has minimal complexity and is numerically demonstrated to maintain high-order accuracy. The performance of the method is illustrated on the radiation transport equation.

Key words. limiting, advection equation, radiation transport equation, stiff sources, conservation equations, asymptotic preserving, invariant domains.

AMS subject classifications. 35L65, 65M12, 65M60, 76V05

1. Introduction.

The objective of the paper is to present two simple limiting techniques for scalar-valued partial differential equations with a structure like the radiation transport equation. The first limiting method is iterative, locally mass conservative, and does not involve solving any nonlinear optimization problem. At convergence, this method enforces the local minimum/maximum principle (assuming that local bounds are known). Two iterations of the method are in general sufficient, but although the method is observed to converge quickly, there is no guarantee that the bounds are enforced everywhere after a number of iterations that is independent of the meshsize. We then propose a second limiting method that is globally mass conservative and that also does not involve solving any nonlinear optimization problem. The second limiting is applied after the first one. The purpose of this second post-processing is simply to certify that some global bound like positivity is exactly and unconditionally enforced while preserving the total mass of the solution. The combination of these two limiting techniques is illustrated using continuous finite elements stabilized with the continuous interior penalty (CIP) technique (a.k.a. edge stabilization) from Douglas and Dupont [7] and Burman and Hansbo [5]. The method presented in the paper is not restricted to continuous elements and CIP though. It can be used with other spatial discretization and other types of stabilization as well. For instance, one can use discontinuous elements (of degree $p \geq 1$) stabilized with the upwind numerical flux. One can also use continuous elements stabilized with other methods like Galerkin Least-Squares, Local Projection Stabilization, Orthogonal Subscale Stabilization, and Subgrid Viscosity.

The original motivation for the work presented here is the solution to the radiation transport equation. Based on our experience on nonlinear hyperbolic systems, we have tried for many years to use upwinding, artificial viscosity, and nonlinear variations thereof to enforce positivity of the angular intensity. But it is well established in the literature that one needs to be careful with this type of method in the context of the radiation transport equations. For instance, it is shown in Adams [1] (see also Larsen [14], Larsen et al. [15], and [9, Rmk. 5.2], [11, §3.2]), that upwinding

---

[*] Department of Mathematics, Texas A&M University 3368 TAMU, College Station, TX 77843, USA.

[†] CERMICS, Ecole des Ponts, 77455 Marne-la-Vallee Cedex 2, France and INRIA Paris, 75647 Paris, France

[‡] This material is based upon work supported in part by the National Science Foundation grant DMS2110868, the Air Force Office of Scientific Research, USAF, under grant/contract number FA9550-18-1-0397, the Army Research Office, under grant number W911NF-19-1-0431, and the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contracts B640889. The support of INRIA through the International Chair program is acknowledged.

and artificial viscosity may lock in the diffusion limit if the artificial viscosity is strong enough so as to guarantee that the local minimum/maximum principle holds. We have long tried to modify the artificial viscosity techniques to overcome this difficulty. For instance in [11] we have adopted a technique inspired by Gosse and Toscani [8] consisting of scaling the transport term, the scattering term, and the artificial diffusion by $\frac{1}{1+\sigma^s h}$ where h is the mesh size and σ^s is the scattering cross section. This gives a method that is indeed positive, convergent, and asymptotic preserving, but it is not locally conservative on nonuniform meshes. Moreover, it does not behave properly with grazing incidences in the diffusion regime unless the grazing boundary data is replaced by its angular average weighted with Chandrasekhar's H-function, thereby seriously diminishing the usefulness of the method as estimating the H-function is nontrivial. In the paper we propose instead to rely on the Galerkin approximation augmented with a traditional linear stabilization technique and to enforce the local minimum/maximum principle by using a mass conservative post-processing technique.

An important aspect of the limiting technique we propose is the estimation of local bounds. The approach we use for this purpose consists of using the method of characteristics as in Yee et al. [28, Eq. (16)]. Note that here we do not propose to use the method of characteristics as a solution method but just as a means to estimate guaranteed local upper and lower bounds. We refer to Lathrop [16], Sanchez and McCormick [25, §II.C.2] for early reviews on solution methods based on the method of characteristics. Once local bounds are found, the next problem consists of enforcing these bounds. This can be done as in Maginot et al. [19] where the authors develop a nonlinear characteristics-based methods that is positivity-preserving. Another idea is to invoke strategies like the so-called flux corrected transport method of Zalesak combining in a nonlinear fashion a low-order and a high-order approximation [29]. The problem with this approach is that it relies on a low-order approximation which itself relies in one way or another on artificial viscosity, which as explained above leads to locking. Another possibility more aligned with what we propose in the paper is to enforce local bounds by using nonlinear optimization techniques. The problem is then to find a maximum-principle-satisfying object that minimizes some distance to the maximum-principle-violating solution while maintaining constant the total mass (either locally or globally). This type of method is developed in e.g., Bochev et al. [3, 4], Yee et al. [28] Peterson et al. [24], and references therein. A fast convergent algorithm for finding such a minimizer is proposed in Liu et al. [18], Ancellin et al. [2]. Here we propose a slightly different approach consisting of just computing a quasi-minimizer with a set of two methods whose algorithmic complexity is significantly lower than that of computing a genuine minimizer. We finally mention that if positivity is the only property one is interested in, one can use "fix-up" techniques like that presented in Lewis and Miller [17, Chap. 4]. This iterative method sets negative fluxes to zero and continues the iterations with the other degree of freedom to restore balance. This idea is generalized to the discontinuous Galerkin setting in Hamilton et al. [12]. Other improvements of the fix-up technique are presented in Maginot et al. [20].

The paper is organized as follows. We introduce in §2 the two limiting techniques mentioned above. We choose to present them without invoking the radiation transport equation as these two methods are quite general and can be used for other purposes. We recall in §3 how local bounds can be extracted for the scalar-valued linear transport equations using the method of characteristics. We also introduce in this section a second-order relaxation technique that is essential to achieve accuracy beyond second-order. We present in §4 the Galerkin approximation of the radiation transport equation stabilized with the continuous interior penalty technique. The full solution method is explained in Algorithms A.1 and A.2. The proposed method is illustrated on the scalar linear transport equation in §5 and on the radiation transport equation in §6.

2. Iterative limiting algorithm.

We present in this section two simple conservative iterative limiting algorithms. The first one preserves mass locally. The second one only preserves mass globally, and is only used as theoretical safeguard. These two algorithms are quite general. They are not specific to the radiation transport equations and can be used in many different contexts.

2.1. The setting.

Let $\{x_i\}_{i\in\mathcal{V}}$ be degrees of freedom that have to be limited, $x_i \in \mathbb{R}$ for all i in the index set $\mathcal{V}$. Let $\mathcal{M} := \sum_{i\in\mathcal{V}} m_i x_i$ be the total mass, where the coefficients m_i are nonnegative and are called mass at i. For all $i \in \mathcal{V}$, let $u_i^{\min}$ and $u_i^{\max}$ be the local minimum and maximum we want to enforce on the i-th degree of freedom. It is henceforth assumed that these local bounds $u_i^{\min}$ and $u_i^{\max}$ satisfy the following reasonable estimate:

$$\sum_{i\in\mathcal{V}} m_i u_i^{\min} \le \mathcal{M} \le \sum_{i\in\mathcal{V}} m_i u_i^{\max}. \tag{2.1}$$

Our objective is to post-process the degrees of freedom $\{x_i\}_{i\in\mathcal{V}}$ to enforce the local bounds while maintaining mass conservation, either locally or globally.

We assume that the lumped mass is non-negative, i.e., $m_i \ge 0$ for all $i \in \mathcal{V}$. This assumption holds, for example, when using finite elements with Bernstein basis of any polynomial degree, and for Lagrange bases of degree 1, 2 and 3 with equally distributed interpolation nodes. We introduce a concept of locality by introducing a notion of stencil. For all $i \in \mathcal{V}$ we assume that there exists a collection of indices $\mathcal{I}(i) \subsetneq \mathcal{V}$ that can exchange mass with i. We call this index set stencil. For instance, for finite elements we have $j \in \mathcal{I}(i)$ if $\varphi_j\varphi_i \not\equiv 0$, where φ_i and φ_j are the global shape function associated with the degrees of freedom i and j. We also define $\mathcal{I}(i)^* := \mathcal{I}(i)\backslash\{i\}$. For all the finite element tests reported in §5 and §6, the set $\mathcal{I}(i)$ is the standard stencil, i.e., $\mathcal{I}(i)$ collects the indices of all the global shape functions φ_j that are such that $\varphi_j\varphi_i \not\equiv 0$.

2.2. Local conservative limiting.

We describe here an iterative algorithm that is locally conservative. The algorithm consists of looping over all the indices i in $\mathcal{V}$ and proceeds as follows. Let $i \in \mathcal{V}$. If $m_i = 0$, then the i-th degree of freedom does not contribute to the global mass; we simply set $y_i = \min(u_i^{\max}, \max(u_i^{\min}, x_i))$ and continue to the next degree of freedom in the list $\mathcal{V}$. This possible modification of the value at i does not change the mass, either locally or globally. Let us assume now that $m_i > 0$. Then either $x_i \le u_i^{\max}$ (i.e., the maximum principle is satisfied) or $u_i^{\max} < x_i$ (i.e., the maximum principle is violated). If $x_i \le u_i^{\max}$, we do nothing and continue to the next index in the loop. If $u_i^{\max} < x_i$, then we compute

$$a_i^+ := \sum_{j\in\mathcal{I}(i)^*} m_j \max(0, u_j^{\max} - x_j), \qquad b_i^+ := \max(x_i - \frac{a_i^+}{m_i}, u_i^{\max}), \tag{2.2a}$$

$$\ell_i^+ := \begin{cases} 0 & \text{if } 0 = a_i^+ \\ m_i \frac{x_i - b_i^+}{a_i^+} & \text{otherwise,} \end{cases} \tag{2.2b}$$

and the actual limiting with respect to $u_i^{\max}$ is done by setting

$$y_j := x_j + \ell_i^+ \max(0, u_j^{\max} - x_j), \quad \forall j \in \mathcal{I}(i)^* \tag{2.3a}$$

$$y_i := b_i^+. \tag{2.3b}$$

Lemma 2.1. *Let $i \in \mathcal{V}$. If $u_i^{\max} < x_i$, then the following holds for all $\{y_j\}_{j\in\mathcal{I}(i)}$ given by (2.3):*
(i) *There is local mass conservation, i.e., $m_i(y_i - x_i) + \sum_{j\in\mathcal{I}(i)^*} m_j(y_j - x_j) = 0$.*

(ii) $x_j \le y_j \le \max(x_j, u_j^{\max})$ *for all* $j \in \mathcal{I}(i)^*$.

(iii) $u_i^{\max} \le y_i \le x_i$. *Moreover,* $y_i < x_i$ *if* $0 < a_i^+$.

Proof. (i) Assume first that $a_i^+ \neq 0$. The equation (2.3a) implies that the mass transferred to the node $j \in \mathcal{I}(i)^*$ is $m_j \ell_i^+ \max(0, u_j^{\max} - x_j)$, while the equation (2.3b) implies that the mass transferred to the node i is $m_i(b_i^+ - x_i)$. The total mass exchange is then

$$\begin{aligned}\Delta_i &:= m_i(y_i - x_i) + \sum_{j\in\mathcal{I}(i)^*} m_j(y_j - x_j)\\ &= m_i(b_i^+ - x_i) + \ell_i^+ \sum_{j\in\mathcal{I}(i)^*} m_j \max(0, u_j^{\max} - x_j) = m_i(b_i^+ - x_i) + \ell_i^+ a_i^+.\end{aligned}$$

But since $a_i^+ \neq 0$, (2.2b) gives $\ell_i^+ = m_i \frac{x_i - b_i^+}{a_i^+}$; hence

$$\Delta_i = m_i(b_i^+ - x_i) + m_i \frac{x_i - b_i^+}{a_i^+} a_i^+ = 0.$$

Otherwise, if $a_i^+ = 0$, then $b_i^+ = \max(x_i, u_i^{\max}) = x_i$ and $\ell_i^+ := 0$. Then again $\Delta_i = 0$.
(ii) The assertion is trivial when $a_i^+ = 0$. Let assume now that $a_i^+ > 0$. As $x_i - \frac{a_i^+}{m_i} \le b_i^+$, we have $\ell_i^+ := m_i \frac{x_i - b_i^+}{a_i^+} \le 1$. Moreover, the assumptions $u_i^{\max} < x_i$ and $b_i^+ := \max(x_i - \frac{a_i^+}{m_i}, u_i^{\max})$, imply that $b_i^+ \le \max(x_i, u_i^{\max}) < x_i$; hence, $0 \le \ell_i^+$. As $0 \le \ell_i^+ \le 1$, we infer that $x_j \le x_j + \ell_i^+ \max(0, u_j^{\max} - x_j) := y_j \le x_j + \max(0, u_j^{\max} - x_j) = \max(x_j, u_j^{\max})$.
(iii) By (2.2a), we have $u_i^{\max} \le b_i^+$ and $b_i^+ = \max(x_i - \frac{a_i^+}{m_i}, u_i^{\max}) \le \max(x_i, u_i^{\max}) = x_i$; hence, $u_i^{\max} \le y_i := b_i^+ \le x_i$. Let us now assume that $a_i^+ \neq 0$. Notice that if $b_i^+ = u_i^{\max}$ then $y_i = b_i^+ < x_i$ (whether a_i^+ is equal to zero or not does not matter here). On the other hand, if $b_i^+ = x_i - \frac{a_i^+}{m_i}$, then the assumption $a_i^+ \neq 0$ implies that $y_i = b_i^+ x_i$. ∎

The same idea as above can be used to enforce the minimum principle. Let $i \in \mathcal{V}$. Either $u_i^{\min} \le x_i$ (i.e., the local minimum principle is satisfied) or $x_i < u_i^{\min}$ (i.e., the local minimum principle is violated). If $u_i^{\min} \le x_i$, do nothing and continue to the next index in the list $\mathcal{V}$. If $x < u_i^{\min}$, then compute

$$a_i^- := \sum_{j\in\mathcal{I}(i)^*} m_j \max(0, x_j - u_j^{\min}), \qquad b_i^- := \min(x_i + \frac{a_i^-}{m_i}, u_i^{\min}), \tag{2.4a}$$

$$\ell_i^- := \begin{cases} 0 & \text{if } 0 = a_i^- \\ m_i \frac{x_i - b_i^-}{a_i^-} & \text{otherwise,} \end{cases} \tag{2.4b}$$

and set

$$y_j := x_j + \ell_i^- \max(0, x_j - u_j^{\min}), \quad \forall j \in \mathcal{I}(i) \setminus \{i\} \tag{2.5a}$$

$$y_i := b_i^-. \tag{2.5b}$$

LEMMA 2.2. *Let* $i \in \mathcal{V}$. *If* $x_i < u^{\min}$, *then the following holds for all* $\{y_j\}_{j\in\mathcal{I}(i)}$ *given by* (2.5)*:*

(i) $m_i(y_i - x_i) + \sum_{j\in\mathcal{I}(i)^*} m_j(y_j - x_j) = 0$, *i.e.*, (2.5) *is locally mass conservative*

(ii) $\min(x_j, u_j^{\min}) \le y_j \le x_j$ *for all* $j \in \mathcal{I}(i)^*$.

(iii) $x_i \le y_i \le u_i^{\min}$. *Moreover,* $x_i < y_i$ *if* $0 < a_i^-$.

The algorithm (2.2)–(2.5) guarantees that the local maximum decreases and the local minimum increases until reaching the prescribed values (see Items (iii) and (iii) in Lemmas 2.1 and 2.2). The statements made in Items (ii) and (ii) in Lemmas 2.1 and 2.2 guarantee that by correcting $x_i \to y_i$, the neighboring values that are already in bounds stay in bounds after limiting. The algorithm is iterative, but there is no guarantee that the prescribed bounds are reached in a finite number of iterations. Extensive numerical tests show though that convergence is quick when one starts from a reasonable solution (e.g., a linearly stabilized Galerkin approximation). In all the tests reported in the paper the above algorithm is only applied two times in a row. The limiting algorithm is summarized in Algorithm 2.1. We note that the order of dofs does not influence the behavior of the algorithm.

Algorithm 2.1 Local conservative limiting (2.2)–(2.5)

```
Require: Bounds $\{u_i^{\max}, u_i^{\min}\}_{i\in\mathcal{V}}$, $\{x_i\}_{i\in\mathcal{V}}$.
for $i \in \mathcal{V}$ do                                  ▷ Loop over dofs
    if $m_i = 0$ then
        $y_i = \min(u_i^{\max}, \max(u_i^{\min}, x_i))$.
    else if $u_i^{\max} < x_i$ then                     ▷ Local maximum principle violated
        Compute $a_i^+$, $b_i^+$, $\ell_i^+$ using (2.2).
        $y_i = b_i^+$
        for $j \in \mathcal{I}(i)^*$ do
            $y_j = x_j + \ell_i^+ \max(0, u_j^{\max} - x_j)$
        end for
    else if $x_i < u_i^{\min}$ then                     ▷ Local minimum principle violated
        Compute $a_i^-$, $b_i^-$, $\ell_i^-$ using (2.4).
        $y_i = b_i^-$
        for $j \in \mathcal{I}(i)^*$ do
            $y_j = x_j + \ell_i^- \max(0, x_j - u_j^{\min})$
        end for
    else                                               ▷ Do nothing
        $y_i = x_i$
    end if
end for
return $\{y_i\}_{i\in\mathcal{V}}$
```

2.3. Global conservative limiting. As the iterative algorithm (2.2)–(2.5) does not guarantee that the prescribed bounds are achieved in a finite number of iterations, we now propose a final conservative post-processing that can be used to make sure that global bounds that are essential to the physics are strictly enforced.

Consider the set of degrees of freedom $\{x_i\}_{i\in\mathcal{V}}$, the associated mass $\mathcal{M} := \sum_{i\in\mathcal{V}} m_i x_i$, and the local bounds $\{u_i^{\min}, u_i^{\max}\}_{i\in\mathcal{V}}$, which we recall are assumed to satisfy (2.1). We compute a new set of limited values $\{z_i\}_{i\in\mathcal{V}}$ as follows:

$$\text{For all } i \in \mathcal{V}\text{, compute,} \qquad y_i := \min((\max(x_i, u_i^{\min}), u_i^{\max}). \tag{2.6a}$$

$$\text{Then compute} \qquad \alpha^+ := \max\left(0, \frac{\mathcal{M} - \sum_{j\in\mathcal{V}} m_j y_j}{\sum_{j\in\mathcal{V}} m_j(u_j^{\max} - y_j)}\right), \tag{2.6b}$$

$$\text{and} \qquad \alpha^- := \max\left(0, \frac{\mathcal{M} - \sum_{j\in\mathcal{V}} m_j y_j}{\sum_{j\in\mathcal{V}} m_j(u_j^{\min} - y_j)}\right). \tag{2.6c}$$

$$\text{For all } i \in \mathcal{V}\text{, set,} \qquad z_i := y_i + \alpha^+(u_i^{\max} - y_i) + \alpha^-(u_i^{\min} - y_i). \tag{2.6d}$$

Lemma 2.3. *The following holds for all $i \in \mathcal{V}$:*

$$u_i^{\min} \le z_i \le u_i^{\max}, \tag{2.7}$$

$$\sum_{i\in\mathcal{V}} m_i z_i = \sum_{i\in\mathcal{V}} m_i x_i. \tag{2.8}$$

Proof. (1) By definition $0 \le \alpha^+$. If $\mathcal{M} - \sum_{i\in\mathcal{V}} m_i y_i \le 0$ then $\alpha^+ = 0$. If $\mathcal{M} - \sum_{i\in\mathcal{V}} m_i y_i \ge 0$, then using the assumption (2.1) together with $\sum_{i\in\mathcal{V}} m_i(u_i^{\max} - y_i) \ge 0$ (which holds owing to (2.6a)), we infer that

$$\alpha^+ \le \frac{\mathcal{M} - \sum_{i\in\mathcal{V}} m_i y_i}{\sum_{i\in\mathcal{V}} m_i(u_i^{\max} - y_i)} \le \frac{\sum_{i\in\mathcal{V}} m_i u_i^{\max} - \sum_{i\in\mathcal{V}} m_i y_i}{\sum_{i\in\mathcal{V}} m_i(u_i^{\max} - y_i)} = 1.$$

Hence $0 \le \alpha^+ \le 1$. We proceed similarly to show that $0 \le \alpha^- \le 1$.

(2) Observing that $u_i^{\min} - y_i \le 0$, $0 \le u_i^{\max} - y_i$ (which holds owing to (2.6a)), $0 \le \alpha^+ \le 1$, and $0 \le \alpha^-$, we obtain

$$\begin{aligned} z_i := y_i + \alpha^+(u_i^{\max} - y_i) + \alpha^-(u_i^{\min} - y_i) &\le y_i + \alpha^+(u_i^{\max} - y_i) \\ &\le y_i + (u_i^{\max} - y_i) = u_i^{\max}. \end{aligned}$$

We proceed similarly to prove that $u_i^{\min} \le z_i$.

(3) Since $0 \le \sum_{i\in\mathcal{V}} m_i(u_i^{\max} - y_i)$ and $\sum_{i\in\mathcal{V}} m_i(u_i^{\min} - y_i) \le 0$, we have

$$\begin{aligned} \sum_{i\in\mathcal{V}} m_i z_i &= \sum_{i\in\mathcal{V}} m_i y_i + \alpha^+ \sum_{i\in\mathcal{V}} m_i(u_i^{\max} - y_i) + \alpha^- \sum_{i\in\mathcal{V}} m_i(u_i^{\min} - y_i) \\ &= \sum_{i\in\mathcal{V}} m_i y_i + \max(0, \mathcal{M} - \sum_{i\in\mathcal{V}} m_i y_i) + \min(0, \mathcal{M} - \sum_{i\in\mathcal{V}} m_i y_i) = \mathcal{M}. \end{aligned}$$

This completes the proof. ∎

Remark 2.4 (Local vs. global bounds). Although the bounds $u_i^{\min}$ and $u_i^{\max}$ invoked in the algorithm can in principle be local, we insist that the algorithm (2.6) should only be used to make sure that global bounds are enforced because the mass conservation mechanism is not local. Notice that the local and global limiting algorithms can be used to limit the solution of time-dependent nonlinear conservation equations as well. Preliminary tests (not shown here for brevity) reveal that these two algorithms perform very well when combined as described above. Preliminary tests, (not shown here for brevity) also show that the global limiting algorithm should not be used alone. When used alone, the algorithm may make the approximation to converge to a solution that does not satisfy the Rankin-Hugoniot condition. The purpose of the global limiting is purely theoretical. Its purpose is simply to certify that some global bound (like positivity) is exactly and unconditionally enforced while preserving the total mass of the solution, but the bulk of the job is done by the local limiting algorithm ☐

3. Bounds for the transport equation.

A critical aspect of the algorithm (2.2)–(2.5) is the estimation of the local bounds $\{u_i^{\min}, u_i^{\max}\}_{i\in\mathcal{V}}$. We explain in this section how this can be done for scalar transport equations. To simplify the presentation, we assume that the degrees of freedoms enumerated with the list $\mathcal{V}$ are Lagrange finite elements associated with the Lagrange nodes $\{\boldsymbol{x}_i\}_{i\in\mathcal{V}}$ from a mesh $\mathcal{T}_h$ based on a domain $D \subset \mathbb{R}^d$, $d \in \{1,2,3\}$. In the rest of this section, the symbol x is a position vector in one dimension and $\boldsymbol{x}$ is a position vector in two and higher dimensions.

3.1. Computations of the bounds.

We start by explaining the method we use to compute bounds in one space dimension. We then generalize the method to high space dimensions later. No originality is claimed here as most of what is said in this section can be found in the radiation transport literature, see e.g., Lathrop [16].

3.1.1. One-dimensional case.

Let us consider the one-dimensional transport equation

$$\Omega(x)\partial_x u(x) + \sigma(x)u(x) = q(x). \tag{3.1}$$

We assume to simplify the presentation that both Ω and σ are piecewise contant over the mesh cells. We also assume that Ω is not equal to 0; otherwise, (3.1) is trivial and there is nothing to limit. For each $i \in \mathcal{V}$, let x_i be one of the Lagrangian node in the mesh where one wants to estimate an upper and a lower bound. Let K be the (unique) cell in the mesh $\mathcal{T}_h$ such that the segment $\{x_i - s\Omega(s) \mid s \in \mathbb{R} > 0\} \cap K$ is not empty. This cell exists if x_i is not on the inflow boundary (if x_i is on the inflow boundary one can set $u_i^{\min} = u_i^{\max} = \alpha^\partial(x_i)$ where α^∂ is the inflow boundary data of the problem). Then one defines $x_i^{\text{up}} := \partial K \cap \{x_i - s\Omega(s) \mid s \in \mathbb{R} > 0\}$. The point x_i^{up} is the farthest away from x_i in K along the direction $-\Omega$. This point is at the upwind boundary of K (see the left panel in Figure 3.1 for example). Then, setting $\Omega := \Omega|_K$ and $\sigma := \sigma|_K$, the exact solution to (3.1) is such that

$$u(x_i) = u(x_i^{\text{up}})e^{\frac{\sigma}{\Omega}(x_i^{\text{up}} - x_i)} + \int_{x_i^{\text{up}}}^{x_i} \frac{q(x)}{\Omega} e^{\frac{\sigma}{\Omega}s}\,\mathrm{d}s. \tag{3.2}$$

Setting $q^{\min} := \min_{x\in K} q(x)$, $q^{\max} := \max_{x\in K} q(x)$, this gives $u_i^{\min} \le u(x_i) \le u_i^{\max}$ where

$$u_i^{\min} := u(x_i^{\text{up}})e^{\frac{\sigma}{\Omega}(x_i^{\text{up}} - x_i)} + \frac{q^{\min}}{\sigma}(1 - e^{\frac{\sigma}{\Omega}(x_i^{\text{up}} - x_i)}), \tag{3.3a}$$

$$u_i^{\max} := u(x_i^{\text{up}})e^{\frac{\sigma}{\Omega}(x_i^{\text{up}} - x_i)} + \frac{q^{\max}}{\sigma}(1 - e^{\frac{\sigma}{\Omega}(x_i^{\text{up}} - x_i)}). \tag{3.3b}$$

As these expressions are not robust with respect to σ (in particular they do not make sense for $\sigma = 0$), we instead define $\Delta x := \left|\frac{x_i^{\text{up}} - x_i}{\Omega}\right|$, and use the following bounds motivated by Taylor expansion, when $\frac{\sigma}{\Omega}(x_i - x_i^{\text{up}}) \leq 0.005$:

$$u_i^{\min} := u(x_i^{\text{up}}) e^{\frac{\sigma}{\Omega}(x^0 - x_i)} + q^{\min}\Big(\Delta x - \sigma\frac{\Delta x^2}{2} + \sigma^2\frac{\Delta x^3}{6} - \sigma^3\frac{\Delta x^4}{24}\Big), \tag{3.4a}$$

$$u_i^{\max} := u(x_i^{\text{up}}) e^{\frac{\sigma}{\Omega}(x_i^{\text{up}} - x_i)} + q^{\max}\Big(\Delta x - \sigma\frac{\Delta x^2}{2} + \sigma^2\frac{\Delta x^3}{6}\Big). \tag{3.4b}$$

So assuming that $u(x_i^{\text{up}})$ is known, or one has access to some good approximation thereof by some iteration process, the bounds (3.3) (or (3.4) if $\sigma\Delta x$ is small) gives an estimate of the local bounds of the solution at x_i.

3.1.2. Two-dimensional case. The high-dimensional case is similar to the one-dimensional one. We consider $\mathcal{T}(i)$ the collection of the cells containing $\boldsymbol{x}_i$ and we set $\boldsymbol{x}_i^{\text{up}} := \bigcap_{K\in\mathcal{T}(i)} \partial K \cap \{\boldsymbol{x}_i - s\boldsymbol{\Omega}(s) \mid s \in \mathbb{R} > 0\}$. The point $\boldsymbol{x}_i^{\text{up}}$ (green one) is the farthest away from $\boldsymbol{x}_i$ (red node) in $\mathcal{T}(i)$ along the direction $-\Omega$ (see the right panel in Figure 3.1 for example). One key difference with the one-dimensional case is that $\boldsymbol{x}_i^{\text{up}}$ is not necessarily a Lagrangian node. Then if one only knows u at the vertices of the triangulation, then one needs to reconstruct u at $\boldsymbol{x}_i^{\text{up}}$. For instance, in two space dimensions one can proceed as follows:

1. Compute $\boldsymbol{x}_i^{\text{up}} := \bigcap_{K\in\mathcal{T}(i)} \partial K \cap \{\boldsymbol{x}_i - s\boldsymbol{\Omega}(s) \mid s \in \mathbb{R} > 0\}$.
2. Define $u(\boldsymbol{x}_i^{\text{up}}) := u(\boldsymbol{x}_{i_1}) + \theta(u(\boldsymbol{x}_{i_2}) - u(\boldsymbol{x}_{i_1}))$, where $\theta = \frac{\|(\boldsymbol{x}_i - \boldsymbol{x}_{i_1})\times\boldsymbol{\Omega}\|_{\ell^2}}{\|(\boldsymbol{x}_{i_2} - \boldsymbol{x}_{i_1})\times\boldsymbol{\Omega}\|_{\ell^2}}$.

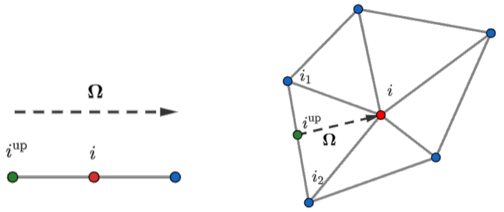


Figure 3.1: Examples for inflow nodes: left panel: 1D case; right panel: 2D case.

3.2. Relaxation. As for any local limiting technique that we know of, second-order relaxation of the bounds must be applied to avoid order reduction. We refer for instance to [10, §4.7] where this question is discussed at length. This issue is addressed by another way in the finite volume literature consists of relaxing the slope reconstructions; see Harten and Osher [13], Schmidtmann et al. [26, §2.1]. Here we adopt the methodology proposed in [10, §4.7]. Let us assume that we have (approximate) knowledge of u at the Lagrange node $\{\boldsymbol{x}_i\}_{i\in\mathcal{V}}$, say $u(\boldsymbol{x}) \approx \sum_{i\in\mathcal{V}} u_i\varphi_i(\boldsymbol{x})$ where $\{\varphi_i\}_{i\in\mathcal{V}}$ are the global Lagrange shape functions. We estimate the local curvature of u by

$$\Delta_i^2 := \frac{\sum_{j\in\mathcal{I}(i)^*} \beta_{ij}(u_i - u_j)}{\sum_{j\in\mathcal{I}(i)^*} \beta_{ij}}, \qquad \text{and set} \qquad \overline{\Delta_i^2} := \text{minmod}\{\Delta_j^2\}_{j\in\mathcal{I}(i)}, \tag{3.5}$$

where $\beta_{ij} = \int_D \nabla\varphi_j\cdot\nabla\varphi_i \,\mathrm{d}x$ are the stiffness coefficients of the Laplace operator. Observe that $\sum_{j\in\mathcal{I}(i)^*} \beta_{ij} = -\beta_{ii} = -\int_D (\nabla\varphi_i)^2 \,\mathrm{d}\boldsymbol{x} \neq 0$. The relaxed local bounds are then defined as follows:

$$\overline{u_i^{\min}} := \max(u_i^{\min} - \overline{\Delta_i^2}, u^{\min}), \qquad \overline{u_i^{\max}} := \min(u_i^{\max} + \overline{\Delta_i^2}, u^{\max}), \tag{3.6}$$

where $u^{\min}$ and $u^{\max}$ are global bounds, if known (see Algorithm A.3). Relaxation is essential to recovers optimal convergence rates, see numerical tests in §5.

4. Approximation of the radiation transport equations.

We are going to illustrate the local mass conserving limiting algorithm (2.2)–(2.5) and the global mass conserving limiting algorithm (2.6) on the radiation transport equation. We introduce here the radiation transport equation and the associated finite element approximation.

4.1. The model problem.

The computational domain D is assumed to be an open, bounded, connected polyhedron in $\mathbb{R}^3$. The boundary of D is denote by ∂D. The symbol $\boldsymbol{n}$ denotes the outer unit normal on ∂D. The unit sphere in $\mathbb{R}^3$ is denoted by $\mathcal{S}$. The surface of the unit sphere is denoted $|\mathcal{S}|$; recall that $|\mathcal{S}| = 4\pi$. We set $\mathcal{O} := \partial D{\times}\mathcal{S}$, and to define the boundary conditions we introduce the inflow boundary $\mathcal{O}_- := \{(\boldsymbol{x},\boldsymbol{\Omega}) \in \mathcal{O} \mid \boldsymbol{\Omega}{\cdot}\boldsymbol{n}(\boldsymbol{x}) < 0\}$.

Given a non-negative source term $q : D{\times}\mathcal{S} \to \mathbb{R}_+$, and a non-negative boundary data $\alpha^\partial : \mathcal{O}_- \to \mathbb{R}_+$, we look for $\Psi : D{\times}\mathcal{S} \to \mathbb{R}_+$ so that

$$\boldsymbol{\Omega}{\cdot}\nabla\Psi(\boldsymbol{x},\boldsymbol{\Omega}) + \sigma^t(\boldsymbol{x})\Psi(\boldsymbol{x},\boldsymbol{\Omega}) = \sigma^s(\boldsymbol{x})\overline{\Psi}(\boldsymbol{x}) + q(\boldsymbol{x},\boldsymbol{\Omega}), \qquad \text{in } D{\times}\mathcal{S} \tag{4.1a}$$

$$\Psi(\boldsymbol{x},\boldsymbol{\Omega}) = \alpha^\partial(\boldsymbol{x},\boldsymbol{\Omega}), \qquad \text{in } \mathcal{O}_- \tag{4.1b}$$

$$\overline{\Psi}(\boldsymbol{x}) := \frac{1}{|\mathcal{S}|}\int_{\mathcal{S}} \Psi(\boldsymbol{x},\boldsymbol{\Omega})\,\mathrm{d}\boldsymbol{\Omega}, \qquad \text{in } D, \tag{4.1c}$$

The dependent variable $\Psi(\boldsymbol{x},\boldsymbol{\Omega})$ is called angular intensity or angular flux, and $\overline{\Psi}(\boldsymbol{x})$ is called scalar intensity or scalar flux. The coefficient $\sigma^s : D \to \mathbb{R}_+$ is the scattering cross section and $\sigma^t : D \to \mathbb{R}_+$ is the total cross section with $\sigma^t \geq \sigma^s$. At some occasions we are also going to use the absorption cross section $\sigma^a := \sigma^t - \sigma^s$.

Our goal is to construct an approximation of (4.1) that is positivity preserving and asymptotic preserving in the diffusion limit. We also want to make sure that the above properties hold with grazing incidences and inhomogeneous materials.

4.2. Angular discretization.

To simplify the presentation of the method we use the discrete ordinate technique to do the discretization with respect to the angles. The resulting angular quadrature is denoted $\{\mu_l, \boldsymbol{\Omega}_l\}_{l\in\mathcal{L}}$ and is assumed to satisfy

$$\sum_{l\in\mathcal{L}} \mu_l = |\mathcal{S}|, \quad \sum_{l\in\mathcal{L}} \mu_l\boldsymbol{\Omega}_l = \boldsymbol{0}, \quad \sum_{l\in\mathcal{L}} \boldsymbol{\Omega}_l|\boldsymbol{c}{\cdot}\boldsymbol{\Omega}_l| = \boldsymbol{0}, \quad \sum_{l\in\mathcal{L}} \mu_l\boldsymbol{\Omega}_l{\otimes}\boldsymbol{\Omega}_l = \frac{|\mathcal{S}|}{3}\mathbb{I}, \tag{4.2}$$

for all $\boldsymbol{c} \in \mathbb{R}^3$, where $\mathbb{I}$ is the $3{\times}3$ identity matrix. We define $L := \mathrm{card}(\mathcal{L})$. All the simulations reported in the paper are done with the S_N technique (Gauss-Legendre quadrature along the polar axis and equi-distributed angles along the azimuth with $\frac{1}{8}N(N+2)$ angles per octant).

4.3. Space discretization.

We are going to use continuous finite elements stabilized with the continuous interior penalty (CIP) technique (a.k.a. edge stabilization) from Douglas and Dupont [7] and Burman and Hansbo [5]. The method presented in the paper is not restricted to continuous elements and CIP though. It can be used with other types of stabilization. For instance, one can use discontinuous elements (of degree $p \geq 1$) stabilized with upwind numerical flux. One can also use continuous elements stabilized with methods like Galerkin Least-Squares (GaLS) (or its SUPG variation), Local Projection Stabilization (LPS), Orthogonal Subscale Stabilization (OSS), and Subgrid Viscosity (SGV).

Let $(\mathcal{T}_h)_{h\in\mathcal{H}}$ be a shape-regular family of unstructured matching meshes exactly covering D. For simplicity we assume that all the elements are generated from a reference element denoted $\widehat{K}$. The geometric transformation mapping $\widehat{K}$ to an arbitrary element $K \in \mathcal{T}_h$ is denoted $T_K : \widehat{K} \longrightarrow K$. We now introduce a reference finite element $(\widehat{K}, \widehat{P}, \widehat{\Sigma})$, which we assume, for simplicity, to be a Lagrange element. We define the following scalar-valued finite element space:

$$V_h = \{v \in \mathcal{C}^0(D;\mathbb{R}) \mid v_{|K}\circ T_K \in \widehat{P},\ \forall K \in \mathcal{T}_h\}. \tag{4.3}$$

The global shape functions are denoted by $\{\varphi_i\}_{i\in\mathcal{V}}$. Recall that $V_h = \operatorname{span}\{\varphi_i\}_{i\in\mathcal{V}}$.

Given any mesh cell K in $\mathcal{T}_h$, we denote by h_K the diameter of K and $\boldsymbol{n}_K$ the outward unit normal at the boundary of K. We set $h := \max_{K\in\mathcal{T}_h} h_K$. The collection of the mesh faces is denoted $\mathcal{F}_h$. The set of interfaces is denoted by $\mathcal{F}_h^\circ$. The set of boundary faces is denoted by $\mathcal{F}_h^\partial$. Each interface F in $\mathcal{F}_h^\circ$ is oriented using a unit normal vector $\boldsymbol{n}_F$. Letting K_l, K_r be the two cells sharing an interface $F \in \mathcal{F}_h^\circ$, we adopt the convention that $\boldsymbol{n}_F$ points from K_l to K_r. Every boundary face $F \in \mathcal{F}_h^\partial$ is oriented by the unit normal $\boldsymbol{n}_F := \boldsymbol{n}_D$. For all $F \in \mathcal{F}_h$ with $F = \partial K_l \cap \partial K_r$, and all function w smooth enough to have traces on F, we define the jump of $\boldsymbol{w}$ across F as

$$[\![w]\!]_F := \lim_{K_l\ni\boldsymbol{y}\to\boldsymbol{x}} w(\boldsymbol{x}) - \lim_{K_r\ni\boldsymbol{y}\to\boldsymbol{x}} w(\boldsymbol{x}). \tag{4.4}$$

4.4. Finite element approximation. First, we define the sesquilinear form associated with the continuous interior penalty. We start by defining $\sigma_\epsilon^t = \sigma^t + \epsilon\sigma_\epsilon$, $\sigma_\epsilon^s = \sigma^s + \epsilon\sigma_\epsilon$, where $\sigma_\epsilon = \epsilon\sup_{\boldsymbol{x}\in D}\sigma^t(\boldsymbol{x})$ and $\epsilon > 0$. The tests reported in the paper are done with $\epsilon = 10^{-10}$. Then, for all $\psi_h, \phi_h \in V_h$, we set

$$s_h(\psi_h,\phi_h) := \varpi \sum_{F\in\mathcal{F}_h^\circ} h_F^2([\![\nabla\psi_h]\!]_{\theta,F}, [\![\nabla\phi_h]\!]_{\theta,F})_{\boldsymbol{L}^2(F)}, \tag{4.5a}$$

$$[\![\nabla\phi_h]\!]_{\theta,F} := (\theta_r\nabla\phi_{h|K_l} - \theta_l\nabla\phi_{h|K_l})\cdot\boldsymbol{n}_F, \tag{4.5b}$$

$$\theta_r = \frac{\sigma_{\epsilon,l}^t}{\sigma_{\epsilon,l}^t + \sigma_{\epsilon,r}^t}, \qquad \theta_r = \frac{\sigma_{\epsilon,r}^t}{\sigma_{\epsilon,l}^t + \sigma_{\epsilon,r}^t}, \tag{4.5c}$$

In all the simulations reported in the paper the parameters, ϖ, and h_F are defined by

$$\varpi := \frac{d^2}{(1+p)^4}, \qquad h_F = \frac{\frac{1}{2}(|K_l| + |K_r|)}{|F|}, \tag{4.6}$$

where d is the space dimension and p is the polynomial degree of the approximation.

Next, for all $k \in \mathcal{L}$, we define the bilinear form associated with the operator $\psi \mapsto \boldsymbol{\Omega}_k\cdot\nabla\psi + \sigma^t\psi$ and the bilinear form we use to weakly enforce boundary conditions. For all $k \in \mathcal{L}$, we set

$$t_k(\psi,\phi) = \int_D \big(\boldsymbol{\Omega}_k\cdot\nabla\psi_{h,k}(\boldsymbol{x}) + \sigma^t(\boldsymbol{x})\psi_{h,k}(\boldsymbol{x})\big)\varphi_i(\boldsymbol{x})\,\mathrm{d}x, \tag{4.7}$$

$$b_k(\psi,\phi) := \sum_{F\in\mathcal{F}_h^\partial} \int_F \frac{1}{2}(|\boldsymbol{\Omega}_k\cdot\boldsymbol{n}| - \boldsymbol{\Omega}_k\cdot\boldsymbol{n})\psi(\boldsymbol{x})\phi(\boldsymbol{x})\,\mathrm{d}s. \tag{4.8}$$

The approximation of the angular flux $\boldsymbol{\psi}_h$ is done in $(V_h)^L$. We set $\boldsymbol{\psi}_h := (\psi_{h,1},\ldots,\psi_{h,L}) \in V_h\times\ldots\times V_h$, with $\psi_{h,k} := \sum_{j\in\mathcal{V}}\Psi_{ik}\varphi_j \in V_h$ for all the angular direction in the quadrature $k \in \mathcal{L}$.

Let α_k^∂ be the value of the boundary incidence along the quadrature angle $\boldsymbol{\Omega}_k$. The discrete ordinate Galerkin approximation of (4.1) consists of seeking $\boldsymbol{\Psi}_h \in (V_h)^L$ so that the following holds for all $k \in \mathcal{L}$ and all $i \in \mathcal{V}$:

$$\begin{aligned}(4.9)\qquad & t_k(\psi_{h,k},\varphi_i) + s_h(\psi_{h,k},\varphi_i) + b_k(\psi_{h,k},\varphi_i) \\ &= \int_D \sigma^s(\boldsymbol{x})\overline{\psi}_h(\boldsymbol{x})\varphi_i(\boldsymbol{x})\,\mathrm{d}x + \int_D q(\boldsymbol{x})\varphi_i(\boldsymbol{x})\,\mathrm{d}x + b_k(\alpha_k^\partial,\varphi_i).\end{aligned}$$

We are also going to make use of the diffusion approximation with weakly enforced Dirichlet boundary condition. For this purpose, for all $\phi_h, r_h \in V_h$, we set

(4.10)

$$a(\phi_h,r_h) := \int_D \frac{1}{3\sigma_\epsilon^s(\boldsymbol{x})}\nabla\phi_h(\boldsymbol{x})\cdot\nabla r_h(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \int_D \sigma^a(\boldsymbol{x})\phi_h(\boldsymbol{x})r_h(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} + \frac{1}{4}\int_{\partial D}\phi_h(\boldsymbol{x})r_h(\boldsymbol{x})\,\mathrm{d}s,$$

where we recall that $\frac{1}{2|\mathcal{S}|}\int_{\boldsymbol{\Omega}\in\mathcal{S}}(|\Omega\cdot\boldsymbol{n}| + \Omega\cdot\boldsymbol{n})\,\mathrm{d}s = \frac{1}{|\mathcal{S}|}\int_{\boldsymbol{\Omega}\cdot\boldsymbol{n}<0}|\Omega\cdot\boldsymbol{n}|\,\mathrm{d}s = \frac{1}{4}$.

4.5. Solution method. There are many solution methods to solve (4.9). The method that is used does not really matter for the purpose of the paper which we recall is about conservative limiting not involving computing a low-order solution based on artificial viscosity. The key idea is that limiting is done after (4.9) is solved. We explain in Appendix A the method that we use for all the simulations reported in the paper. As the purpose of the paper is just to discuss limiting, we have adopted a simple source iteration technique preconditioned with a diffusion approximation and using a minimum residual technique.

5. Numerical illustrations, scalar transport equation.

The objective of this section is to illustrate the limiting technique proposed in the paper. We start by testing the method on the scalar advection equation. Examples involving the radiation transport equation are reported in §6.

5.1. Numerical details. The tests are done with continuous finite elements in one and two space dimensions. Unless specified otherwise, the simulations realized in one dimension are done on uniform meshes and those realized in two dimensions are done on unstructured Delaunay meshes. In all the tests reported below the quadratures are exact for the mass matrix. The index I stands for the number of degrees of freedom (or gridpoints) of the approximation. Given a nonzero function $u \in L^1(D)$ and u_h its finite element approximation, we call relative error in the L^1-norm the quantity $\|u - u_h\|_{L^1(D)}/\|u\|_{L^1(D)}$.

5.2. 1D smooth solution. We start by solving a transport problem in one space dimension with a smooth solution. We let $D = (0, 8)$ and solve

$$(5.1)\qquad \Omega\partial_x u + \sigma u = q, \quad \text{a.e. } x \in D, \qquad u(0) = 0.$$

with $\Omega = 1$, $\sigma(x) = 1$, $q(x) = \Omega\pi\sin(\pi x) + \sigma(x)(1 - \cos(\pi x))$. The solution is $u(x) = 1 - \cos(\pi x)$.

We test the method with $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ continuous finite elements on a series of meshes. We compute the relative error in the L^1-norm. The results are shown in Table 5.1. We observe the expected convergence rates: 2, 3, and 4 for $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ finite elements, respectively.

5.3. 1D non-smooth solution. We continue with one-dimensional transport problem with a non-smooth solution. We consider $D = (0, 1)$ and solve

$$(5.2)\qquad \Omega\partial_x u + \sigma u = q, \quad \text{a.e. } x \in D, \qquad u(0) = 0.$$

Table 5.1: Problem (5.1). $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ continuous finite elements. Relative error in the L^1-norm.

$\mathbb{P}_1$			$\mathbb{P}_2$			$\mathbb{P}_3$		
I	L^1-Err	rate	I	L^1-Err	rate	I	L^1-Err	rate
101	2.28E-03	–	101	2.02E-04	–	100	1.07E-04	–
201	4.61E-04	2.32	201	2.22E-05	3.21	202	6.80E-06	3.91
401	1.02E-04	2.18	401	2.67E-06	3.07	400	4.46E-07	3.99
801	2.42E-05	2.08	801	3.30E-07	3.02	799	2.81E-08	4.00
1601	5.92E-06	2.04	1601	4.12E-08	3.01	1600	1.74E-09	4.00

with $\Omega = 1$, and the scalar fields $\sigma(x)$ and $q(x)$ are piecewise constants and given by

$$\sigma(x) = \begin{cases} s_1 & x_0 \le x \le x_1 \\ s_2 & x_1 < x \le x_2 \\ s_3 & x_2 < x \le 1 \end{cases} \qquad q(x) = \begin{cases} q_1 & x_0 \le x \le x_1 \\ q_2 & x_1 < x \le x_2 \\ q_3 & x_2 < x \le 1. \end{cases} \tag{5.3}$$

The exact solution is given by

$$u(x) = \begin{cases} \frac{q_1}{s_1}(1 - \exp(s_1(x_0 - x))) & x_0 \le x \le x_1 \\ u_1 \exp(s_2(x_1 - x)) & x_1 < x \le x_2 \\ u_2 \exp(s_3(x_2 - x)) + \frac{q_3}{s_3}(1 - \exp(s_3(x_2 - x))) & x_2 < x \le 1, \end{cases} \tag{5.4}$$

where $u_1 = \frac{q_1}{s_1}(1 - \exp(s_1(x_0 - x_1)))$ and $u_2 = u_1 \exp(s_2(x_1 - x_2))$. In the simulations reported below we use $x_0 = 0$, $x_1 = 0.3$, $x_2 = 0.6$, $s_1 = 1$, $s_2 = 10^3$, $s_3 = 2$, $q_1 = 1$, $q_2 = 0$, $q_3 = 1$.

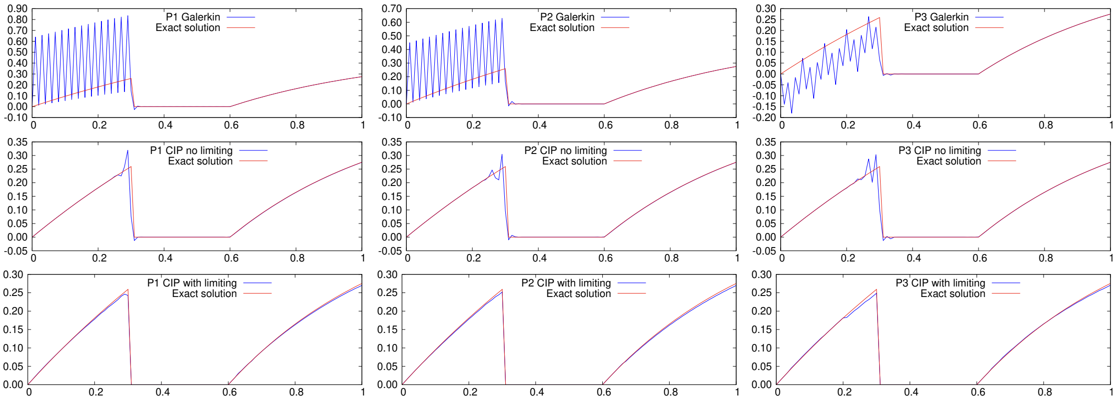


Figure 5.1: Nonsmooth solution (5.2). Left column: $\mathbb{P}_1$, 101 dofs. Center column: $\mathbb{P}_2$, 101 dofs. Right column: $\mathbb{P}_3$, 91 dofs. Top row: Galerkin. Center row: CIP without limiting. Bottom row: CIP with limiting.

We test the method with $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ continuous finite elements. We show in Figure 5.1 the graph of the $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ approximation on a mesh composed of 101, 101 and 91 grid points, respectively. The panels in the top row of the figure show the unstabilized Galerkin solution. oscillations are clearly visible in the interval $(0, x_1)$. The panels in the second row of the figure show the solution stabilized with the CIP method. Most of the oscillations are gone at the exception of overshoots localized at the interface located at x_1. We finally show in the panels of the third row the results obtained with CIP stabilization and the limiting technique proposed in the paper. The maximum principle is satisfied.

Table 5.2: Non-smooth solution (5.2). $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ continuous finite elements. Relative error in the L^1-norm.

$\mathbb{P}_1$			$\mathbb{P}_2$			$\mathbb{P}_3$		
I	L^1-Err	rate	I	L^1-Err	rate	I	L^1-Err	rate
101	1.49E-02	–	101	1.91E-02	–	91	1.68E-02	–
201	5.15E-03	1.54	201	2.82E-03	2.78	181	5.88E-03	1.53
401	1.56E-03	1.72	401	9.59E-04	1.56	361	1.49E-03	1.99
801	3.34E-04	2.23	801	4.12E-04	1.22	721	3.35E-04	2.16
1601	1.01E-04	1.73	1601	1.49E-04	1.47	1441	7.11E-05	2.24

The results of the convergence tests are shown in Table 5.2. We observe the expected convergence rate close to 2 for $\mathbb{P}_3$ elements and between 1.5 and 2 for $\mathbb{P}_1$ and $\mathbb{P}_2$ elements. Note that the convergence rate 2 is optimal since the gradient of the solution is only in BV.

5.4. 2D slip line. We now consider the two-dimensional unit square $D = (0,1)^2$ and solve the problem

$$\Omega\partial_x u = 0, \quad \text{a.e. } x \in D, \qquad u_{\Gamma_1} = 0, \qquad u_{\Gamma_2} = 1, \tag{5.5}$$

with $\Omega = (0,1)^2$, and $\Gamma_1 = \{(0,y) \mid 0 < y \le 1\}$, $\Gamma_2 = \{(x,0) \mid 0 \le x \le 1\}$. The exact solution is discontinuous; it exhibits a slip line align the axis $\{x = y, x > 0\}$. The solution is given by

$$u(x,y) = \begin{cases} 0 & x < y \\ 1 & y \le x. \end{cases} \tag{5.6}$$

We show in the top three panels of Figure 5.2 the graph of the limited CIP solution obtained with $\mathbb{P}_1$, $\mathbb{P}_2$ and $\mathbb{P}_3$ elements on meshes composed of 9535, 9529, and 9541 grid points, respectively. We show in the bottom three panels of the figure the meshes and 11 isolines $\{0.05, 0.1, \ldots, 0.9, 0.95\}$ (4 triangles are shown for each $\mathbb{P}_2$ cell, and 9 triangles are shown for each $\mathbb{P}_3$ cell)

We report in Table 5.3 the convergence rates for $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ continuous finite elements. We observe a rate close to 0.73 for linear elements. The convergence rate for $\mathbb{P}_2$ and $\mathbb{P}_3$ elements is equal to 1 which is optimal as the solution is only in BV. This test confirms the near-optimality of the method for $\mathbb{P}_2$ and $\mathbb{P}_3$ elements.

5.5. 2D non smooth problem. Let $D = (0,1)^2$. We solve the two-dimensional version of the problem (5.2),

$$\boldsymbol{\Omega}\cdot\nabla u + \sigma u = q, \quad \text{a.e. } x \in D, \qquad u(0,y) = 0,\ 0 \le y \le 1. \tag{5.7}$$

Figure 5.2: Slip line problem (5.5). Non-uniform Delaunay meshes. Left: $\mathbb{P}_1$, 9535 grid points; Center: $\mathbb{P}_2$, 9529 grid points; $\mathbb{P}_3$, 9541 grid points.

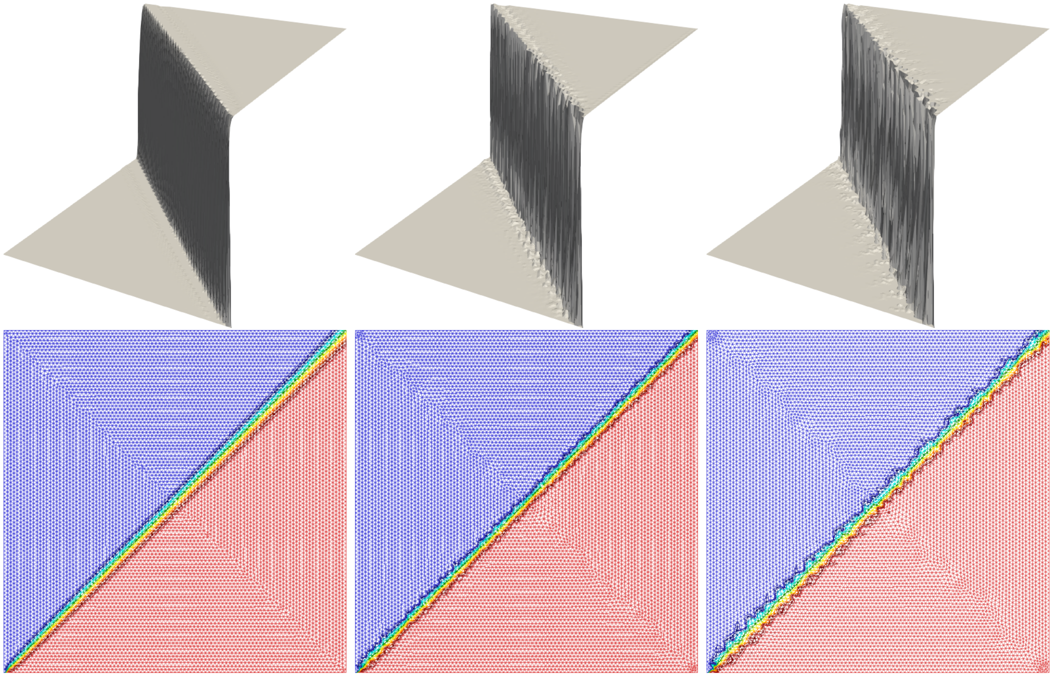


Table 5.3: Problem (5.5). $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ continuous finite elements. Relative error in the L^1-norm.

$\mathbb{P}_1$			$\mathbb{P}_2$			$\mathbb{P}_3$		
I	L^1-Err	rate	I	L^1-Err	rate	I	L^1-Err	rate
961	5.47E-02	–	1681	4.14E-02	–	961	7.09E-02	–
3721	3.45E-02	0.68	6561	2.16E-02	0.96	3721	3.71E-02	0.96
14641	2.12E-02	0.71	25921	1.10E-02	0.98	14641	1.89E-02	0.98
58081	1.28E-02	0.73	103041	5.57E-03	0.99	58081	9.57E-03	0.99
231361	7.75E-03	0.73	410881	2.80E-03	0.99	231361	4.81E-03	1.00

with $\boldsymbol{\Omega} = (1,0)$, and, with the notation $\boldsymbol{x} := (x,y)$, the scalar fields $\sigma(\boldsymbol{x})$ and $q(\boldsymbol{x})$ given by

$$\sigma(x,y) = \begin{cases} s_1 & x_0 \le x \le x_1 \\ s_2 & x_1 < x \le x_2 \\ s_3 & x_2 < x \le 1 \end{cases} \qquad q(x,y) = \begin{cases} q_1 & x_0 \le x \le x_1 \\ q_2 & x_1 < x \le x_2 \\ q_3 & x_2 < x \le 1. \end{cases} \tag{5.8}$$

The exact solution is given by

$$u(x,y) = \begin{cases} \frac{q_1}{s_1}(1-\exp(s_1(x_0-x))) & x_0 \le x \le x_1 \\ u_1 \exp(s_2(x_1-x)) & x_1 < x \le x_2 \\ u_2 \exp(s_3(x_2-x)) + \frac{q_3}{s_3}(1-\exp(s_3(x_2-x))) & x_2 < x \le 1. \end{cases} \tag{5.9}$$

We use the same set of the coefficients x_0, x_1, x_2, s_1, s_2, s_3, and q_1, q_2, q_3 as in §5.3 (i.e., $x_0 = 0$, $x_1 = 0.3$, $x_2 = 0.6$, $s_1 = 1$, $s_2 = 10^3$, $s_3 = 2$, $q_1 = 1$, $q_2 = 0$, $q_3 = 1$).

The results of the convergence tests done on uniform meshes are shown in Table 5.4. The convergence rate varies between 1.1 and 2. Recall that the convergence rate 2 is the optimal since the gradient of the solution is only in BV.

Table 5.4: Problem (5.7). $\mathbb{P}_1$, $\mathbb{P}_2$, and $\mathbb{P}_3$ continuous finite elements. Relative error in the L^1-norm.

$\mathbb{P}_1$			$\mathbb{P}_2$			$\mathbb{P}_3$		
I	L^1-Err	rate	I	L^1-Err	rate	I	L^1-Err	rate
961	5.08E-02	–	1681	2.32E-02	–	961	4.12E-02	–
3721	2.34E-02	1.15	6561	1.04E-02	1.18	3721	1.96E-02	1.10
14641	9.62E-03	1.30	25921	3.74E-03	1.49	14641	7.08E-03	1.49
58081	3.18E-03	1.61	103041	1.09E-03	1.79	58081	1.71E-03	2.06
231361	8.28E-04	1.95	410881	4.64E-04	1.23	231361	4.18E-04	2.04

6. Radiation transport.

In this section, we report the tests done on the radiation transport equation using the algorithm described in the paper.

6.1. Numerical details.

The positive- and asymptotic-preserving algorithm defined in Algorithms A.1 and A.2 is implemented with continuous finite elements of degree $p \in \{1,2,3\}$ on simplices. The meshes in one dimension are uniform. The meshes in two space dimension are non-uniform and composed of triangles. The angular discretization is done with the Gauss-Chebyshev S_N quadrature. In one dimension, the x_1-component of the angles are the N quadrature points of the Gaussian-Legendre quadrature over $[-1,1]$ and the weights are the weights of the Gaussian-Legendre quadrature. In two-dimensions we use the standard triangular S_N quadrature. As the x_3-direction is ignored, there are four quadrants and the total number of angular directions is $\frac{1}{2}N(N+2)$. Unless specified otherwise, the units are cm for lengths, nb.part./cm^2·s·sr for ψ, nb.part./cm^2·s for $\overline{\psi}$, nb.part./cm^3·s for q, and cm^{-1} for the cross sections.

6.2. One-dimensional benchmark tests.

We start by illustrating the performance of the method in one space dimension. We compare the results given by the proposed the method with those given by the unlimited dG1 approximation stabilized by using the upwind flux. We use the angular quadrature S_8 (8 discrete directions in 1D) for all the cases. The angles, characterized by their x_1-component, are enumerated in increasing order from 1 to 8. The data for the four cases considered here are reported in Table 6.1. In each case we give the length of the domain and the number of zones composing the domain. For each zone we give the values of σ^t, σ^s, and q (constants), and we also give the number of cells composing the zone. The boundary condition for cases 1, 3, and 4 are $\boldsymbol{\psi}_{h|\partial D_-} = 0$ (this is the so-called vacuum boundary condition). We enforce a grazing incidence for case 2; we set $\psi_{h,k|\partial D_-} = 0$ for $k \neq 5$, $1 \le k \le 8$, and $\psi_{h,5}(0) = 1.0$.

Table 6.1: Data for the one-dimensional test cases.

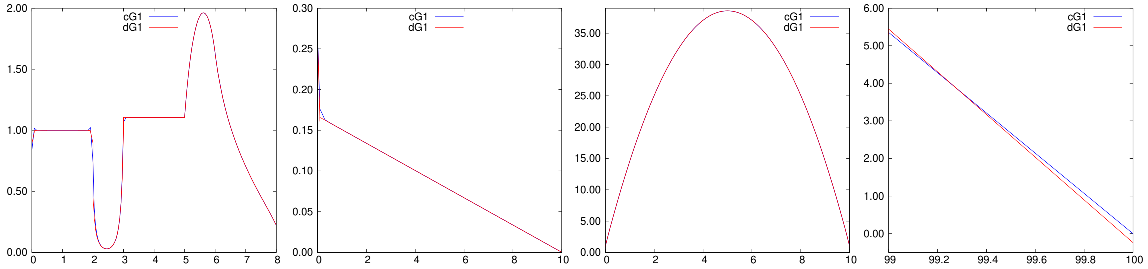
	#zones	5				
Case 1	Length	2.0	1.0	2.0	1.0	2.0
	σ_s	0.0	0.0	0.0	0.9	0.9
	σ_t	50.0	5.0	0.0	1.0	1.0
	q	25.	0.0	0.0	.5	0.0
	#cells	25	25	25	25	25
	B.C.	Vac.				

	#zones	1
Case 2	Length	10.0
	σ_s	100.0
	σ_t	100.0
	q	0.0
	#cells	100
	B.C.	$\psi_5(0) = 1$

	#zones	1
Case 3	Length	10.0
	σ_s	10.0
	σ_t	10.0
	q	0.05
	#cells	100
	B.C.	Vac.

	#zones	1
Case 4	Length	100.0
	σ_s	0.09999
	σ_t	0.1
	q	0.5
	#cells	100
	B.C.	Vac.

Figure 6.1: Comparisons between the present method using cG1 and the upwind dG1 method.



(a) Case 1, $2\overline{\psi}_h$ (b) Case 2, $2\overline{\psi}_h$ (c) Case 3, $2\overline{\psi}_h$ (d) Case 4, $\psi_{h,1}$, zoom

The results of the simulations are reported in Figure 6.1. We show in Panels (6.1a)-(6.1c) the total scalar flux, $2\overline{\psi}_h$, obtained with the dG1 approximation (labeled dG1) and with the proposed method (labeled cG1). We observe a fair agreement between the two methods given the number of grid points (recall that dG1 has two times has many grid points as cG1). Panel (6.1d) shows the angular flux $\psi_{h,1}$ for case 4 in the last cell close to the right boundary. For this case the dG1 approximation gives negative values at $x = 100$ on the angular fluxes 1, 2, and 3 (the values are -0.24, -0.22, -0.066, respectively (approximated to 2 digits)). In all the cases the proposed method is always nonnegative.

6.3. Grazing incidences. We now focus on the second test case discussed in §6.2. A grazing incidence is enforced on the left boundary of the domain (only $\psi_{h,5}(0)$ is nonzero). We also have $\sigma^t h = \sigma^s h = 10$; that is, the diffusive regime is dominant. The conjunction of these two conditions produces a boundary layer as shown in Chandrasekhar [6], Malvagi and Pomraning [21]. Moreover, as established in Theorem 5.4 in [9], convergence of the dG approximation only occurs in a weak norm; more precisely, convergence on $\overline{\psi}_h$ occurs in the Sobolev space $H^s(D)$ only for smoothness indices s strictly less that $\frac{1}{2}$. That is to say, the values of $\overline{\psi}_h$ at the boundary does not converge (recall that boundary traces do not exit when $s < \frac{1}{2}$). Let us for a moment denote by $\alpha^\partial(\boldsymbol{\Omega}, \boldsymbol{x})$ the boundary data of the problem. Then as observed in Adams [1, Eq. 66] (see also Prop. 3.6 in [9]) the leading term of the diffusion expansion of the dG approximation at the boundary is

$$\overline{\psi}_h(\boldsymbol{x})_{|\partial D} \approx \frac{1}{2\pi}\int_{\boldsymbol{\Omega}\cdot\boldsymbol{n}\le 0} (|\boldsymbol{\Omega}\cdot\boldsymbol{n}| + \tfrac{3}{2}|\boldsymbol{\Omega}\cdot\boldsymbol{n}|^2)\alpha^\partial(\boldsymbol{\Omega}, \boldsymbol{x})\,\mathrm{d}\boldsymbol{\Omega}, \tag{6.1}$$

which is almost equal to $\frac{1}{2\pi}\int_{\boldsymbol{\Omega}\cdot\boldsymbol{n}\le 0}\frac{\sqrt{3}}{2}|\boldsymbol{\Omega}\cdot\boldsymbol{n}|\mathrm{H}(|\boldsymbol{\Omega}\cdot\boldsymbol{n}|)\alpha^{\partial}(\boldsymbol{\Omega},x)\,\mathrm{d}\boldsymbol{\Omega}$, (where H is Chandrasekar's function), because $\frac{\sqrt{3}}{2}\mu\mathrm{H}(\mu)\approx 0.91\mu+1.635\mu^2\approx\mu+\frac{3}{2}\mu^2$ up to a "a few percents" over the interval $\mu\in[0,1]$. Hence, although, strictly speaking, the dG approximation is not asymptotic preserving when there are grazing incidences (unless the mesh or the shape functions are designed to resolve boundary layers), due to the above observation, it is a common practice in the radiation transport literature to say that the dG approximation with the upwind numerical flux is asymptotic preserving regardless of whether the boundary data is isotropic or not.

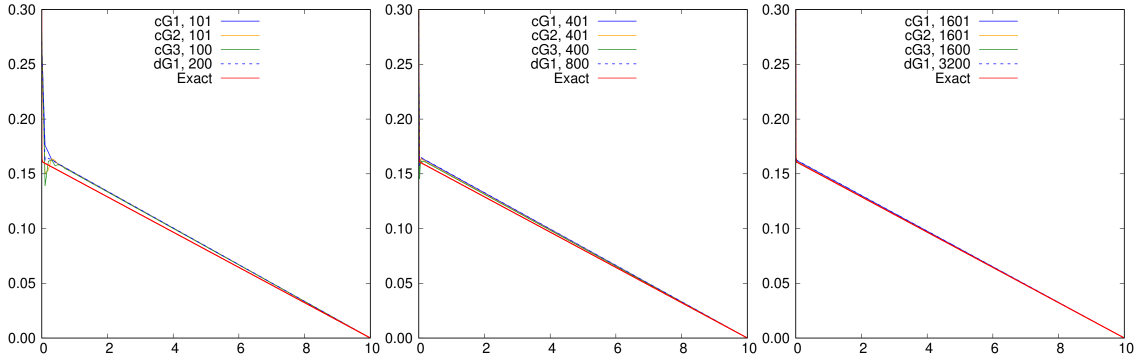


Figure 6.2: Total scalar flux, $2\overline{\psi}_h$, for the grazing problem with cG1, cG2, cG3, and dG1. Left: 100 grid points. Center: 400 grid points. Right: 1600 grid points

Note that the above comments regarding dG are independent of the polynomial degree, i.e., (6.1) does not depend on p. Note also that the numerical results reported in the second panel of Figure 6.1 show that the result holds true as well for the stabilized cG1 approximation. The key here is that in both cases the boundary condition is enforced weakly using the bilinear form b_k defined in (4.8). We show in Figure 6.2 simulations done with the proposed method using cGp, $p\in\{1,2,3\}$, on meshes composed on 100 (left panel), 400 (center panel), and 1600 (right panel) grid points. In each case, the number of cells is adjusted so as to maintain the same number of grid points for all $p\in\{1,2,3\}$. We also report in this figure the results from the dG1 approximation with 200, 800, and 3200 grid points. The red line labeled "exact" is obtained by using cG3 with 300000 grid points. We observe that, for each number of degrees of freedom all the methods give almost exactly the same results (regardless of whether the method is dG or cG). We notice also that with 100 grid points the numerical solution is indeed "a few percents" away from the exact solution, as claimed in [1, Eq. 66]. The reader is invited to zoom on the right panel where we used 1600 gridpoints: the dG1 solution and the three cG solutions all align on one line that is still slightly away from the exact solution. This confirms the theoretical result established in Theorem 5.4 in [9] where it is shown that when grazing incidences are enforced, convergence occurs in weak norms only and the convergence rate is weak (the convergence rate behaves like $\mathcal{O}(h^{\frac{1}{2}})$ in one space dimension).

Remark 6.1 (Comparison with [11]). The positive and asymptotic preserving method presented in [11] behaves properly in the diffusion limit with grazing incidences only if the asymptotic boundary value (6.1) (or Chandrasekar's exact value) is enforced weakly (just enforcing $\alpha^{\partial}(\boldsymbol{\Omega},\boldsymbol{x})$ is not asymptotic preserving). This is not the case here. The present method properly works in every regime by just enforcing the boundary conditions using the bilinear form b_k defined in (4.8). ☐

We finally repeat the above test by adding a vacuum region on the left while still enforcing the same grazing incidence boundary condition as above. This test is meant to assess the behavior of the proposed method in the presence of interfaces with vacuum and grazing incidences. We show the results in Figure 6.3 using 200 grid points in the center panel and 800 grid points in the right panel. We observe that the proposed method properly behaves. The cG and dG results almost coincide in both cases. We observe again, that due to the grazing incidence all the methods are (almost) asymptotic preserving up to "a few percents".

#zones	2	
Length	10.0	10.0
σ_s	0.0	100.0
σ_t	0.0	100.0
q	0.0	0.0
B.C.	$\psi_5(0) = 1$	

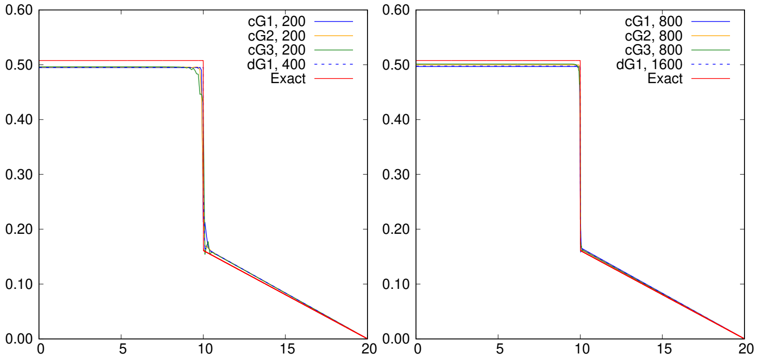


Figure 6.3: Total scalar flux, $2\overline{\psi}_h$, for the grazing problem with a vacuum interface using cG1, cG2, cG3, and dG1. Left: data. Center: 200 grid points. Right: 800 grid points

6.4. Diffusion limit with constant cross sections. To verify that the method does not lock in the diffusion regime, we consider the two-dimensional test reported in [11, §5.2.1]. The computational domain is $D = (0, 1)^2$. The cross sections are constant $\sigma^t = \sigma^s = \frac{1}{\epsilon}$ with $\epsilon > 0$. The source is isotropic and given by $q(\boldsymbol{x}) = \epsilon\frac{2}{3}\pi^2 \sin(\pi x_1) \sin(\pi x_2)$. When $\epsilon \to 0$, the regime becomes dominated by diffusion. The asymptotic limit in this case is $\psi^0(\boldsymbol{x}) = \sin(\pi x_1) \sin(\pi x_2)$.

We solve (4.1) with the method described in the paper using linear elements ($p = 1$). The tests are done using five meshes with 140, 507, 1927, 7545, and 29870 Lagrange nodes, respectively; the corresponding mesh-sizes are approximately $h \approx 0.1$, 0.5, 0.025, 0.125, and 0.00625. We use the S_6 quadrature (24 angular directions).

Table 6.2: Diffusion limit. Convergence test with respect to the mesh-size and ϵ.

ϵ	I	rel($\\|\boldsymbol{e}\\|_{L^2}$)	rate	rel($\\|\nabla\boldsymbol{e}\\|_{L^2}$)	rate
10^{-3}	140	3.48E-02	–	7.57E-02	–
	507	5.19E-03	2.96	1.91E-02	2.14
	1927	2.35E-03	1.18	5.88E-03	1.77
	7545	2.91E-03	-.31	2.32E-03	1.36
	29870	3.05E-03	-.07	1.44E-03	0.69
10^{-4}	140	1.52E-02	–	3.92E-02	–
	507	3.39E-03	2.33	1.20E-02	1.84
	1927	6.35E-04	2.51	4.23E-03	1.56
	7545	1.82E-04	1.83	1.19E-03	1.86
	29870	2.70E-04	-.57	3.14E-04	1.93

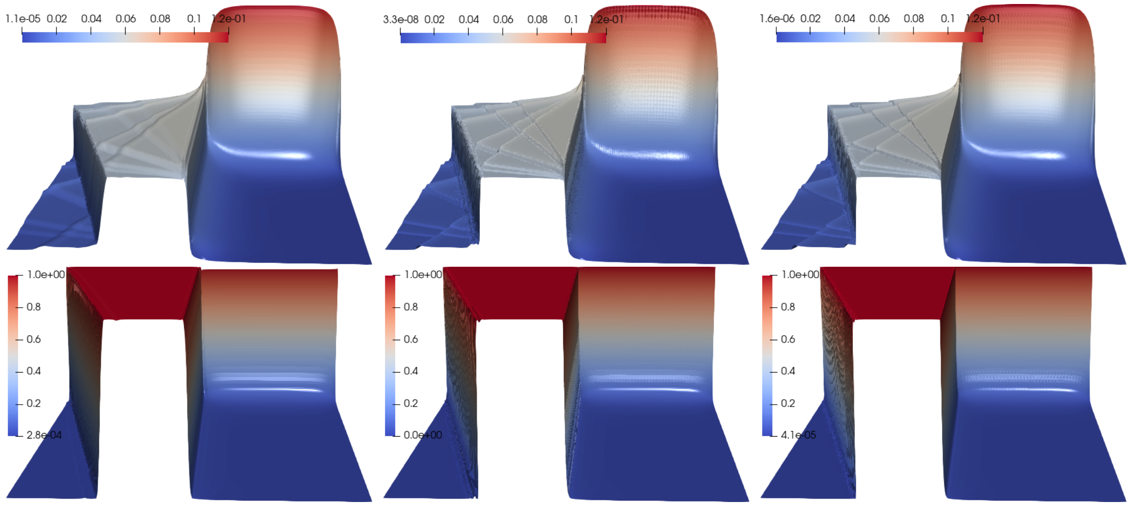
ϵ	I	rel($\\|\boldsymbol{e}\\|_{L^2}$)	rate	rel($\\|\nabla\boldsymbol{e}\\|_{L^2}$)	rate
10^{-5}	140	1.25E-02	–	2.24E-02	–
	507	3.16E-03	2.13	6.98E-03	1.81
	1927	7.66E-04	2.12	2.65E-03	1.45
	7545	1.70E-04	2.21	7.11E-04	1.93
	29870	2.81E-05	2.62	2.38E-04	1.59
10^{-6}	140	1.23E-02	–	1.95E-02	–
	507	3.14E-03	2.12	5.75E-03	1.90
	1927	7.84E-04	2.08	1.98E-03	1.60
	7545	1.92E-04	2.06	7.06E-04	1.51
	29870	4.59E-05	2.08	2.35E-04	1.60

We show in Table 6.2 the results of the test for the following four values of the small parameter $\epsilon \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. We report in this table the relative L^2-norm and the relative H^1-

semi-norm of the difference between $\overline{\psi}_h$ and the Lagrange interpolant of the asymptotic limit ψ^0. We observe that, as proved in [9, Th. 5.3] for the upwind dG1 approximation, the scalar flux $\overline{\psi}_h$ converges optimally to ψ^0 when ϵ is significantly smaller than the mesh-size. The convergence order is $\mathcal{O}(h^2)$ in the L^2-norm. It seems that some super-closeness phenomenon occurs in the H^1-semi-norm since the rate behaves like $\mathcal{O}(h^{1.5})$. Tests with quadratic and cubic elements demonstrate the same behavior. These tests are not reported for brevity.

6.5. Reflection effects. We now reproduce a test case proposed in [11] using exactly the same finite element grids and angular discretization in order to illustrate that the proposed method is indeed more accurate than that from the reference.

Figure 6.4: Grazing and reflection effects: Top: scalar intensity $\overline{\psi}_h$. Bottom: first angular intensity, $\psi_{h,1}$. Left, $\mathbb{P}_1$, 76230 grid points. Center $\mathbb{P}_2$, 303893 grid points. Right: $\mathbb{P}_3$, 682990 grid points.

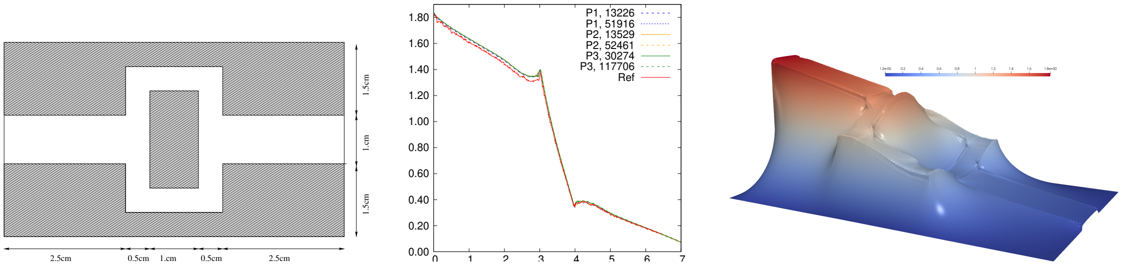


We consider the two-dimensional domain $D = (0,1)^2$ composed of two regions: one that is optically thick and one without any scattering. The cross sections are distributed as follows: $\sigma^t(\boldsymbol{x}) = 100$, $\sigma^s(\boldsymbol{x}) = 99$ if $x_2 \geq 0.5$ (optically thick and diffusive zone), and $\sigma_t(\boldsymbol{x}) = \sigma_s(\boldsymbol{x}) = 0$ if $x_2 \leq 0.5$ (void). The angular approximation is done with the S_6 quadrature (24 angular directions). We enforce a grazing incidence boundary condition on the leftmost boundary: this boundary is illuminated with intensity 1 along the first direction of the quadrature $\boldsymbol{\Omega}_1 := (0.93802334, 0.25134260, 0.23861919)$ (eight digits truncation). All the other incoming angular fluxes are set to zero on this boundary. All the incoming angular fluxes are set to zero on the other three boundaries. The approximation in space for the asymptotic-preserving method is done on a non-uniform grid composed of 151434 triangles. There are 76230 $\mathbb{P}_1$ grid points (i.e., 1 829 520 dofs in total), 303893 $\mathbb{P}_2$ grid points (i.e., 7 239 432 dofs in total), 682990 $\mathbb{P}_3$ grid points (i.e., 16 391 760 dofs in total).

The results are shown in Figure 6.4. Comparing these results with what is shown in the two leftmost panels in Fig. 3 in [11], we observe that the present method is significantly more accurate than that in [11] while being positivity-preserving and asymptotic preserving. We also notice the ray effect in the vacuum region $\{x_2 \leq 0.5\}$, which is an artifact of the S_N method (recall that we are just using 24 angular directions on purpose). That the ray effect is so crisply captured clearly demonstrates that the space approximation is very accurate; one cannot discern any smoothing induced by numerical dissipation.

6.6. Crooked pipe problem. We now solve a problem known in the literature as the crooked pipe problem. We adopt here the setting used in Olivier et al. [22, §7.3]. The geometry of the problem is shown in the left panel of Figure 6.5, $D = (0, 7)\times(-2, 2)$. In the pipe we have $\sigma^s = 0.2$, $\sigma^a = 10^{-3}$, $q = 10^{-7}$. The characteristics of the material composing the walls are $\sigma^s = 200$, $\sigma^a = 10^{-3}$, and $q = 10^{-7}$. The boundary condition is $\Psi(x, y) = 2/|S| = 1/2\pi$ on $\{x = 0, |y| \le 0.5\}$ and $\Psi(x, y) = 0$ on the rest of the boundary.

Figure 6.5: Crooked pipe problem. Left: setting of the problem. Center: profile of the total scalar flux, $|\mathcal{S}|\overline{\psi}_h$ for various meshes and polynomial degrees. The symbol "Ref." stands for the data from [22]. Right: total scalar flux distribution ($\mathbb{P}_3$)



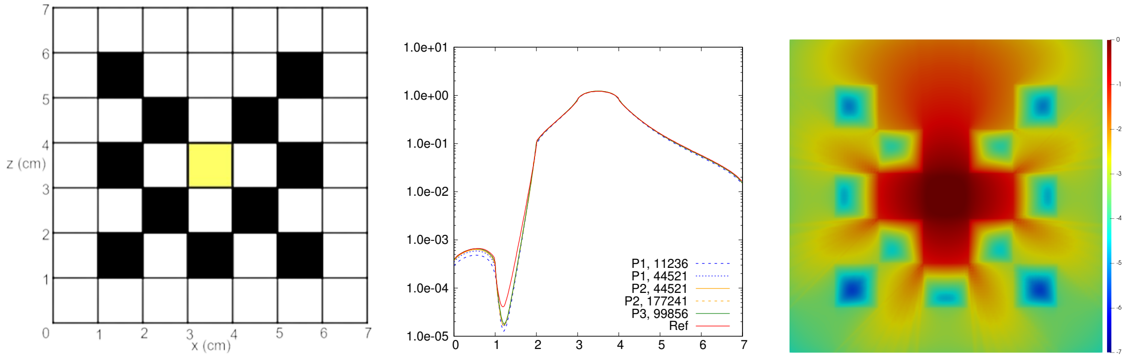
We use unstructured triangular Delaunay meshes. The meshes are generated so that the pipe/wall interface is exactly represented. The computations are done on various meshes and with various polynomial degrees ranging form $\mathbb{P}_1$ to $\mathbb{P}_3$: $\mathbb{P}_1$ with 13226 grid points, $\mathbb{P}_1$ with 51916 grid points, $\mathbb{P}_2$ with 13529 grid points, $\mathbb{P}_2$ with 524621 gridpoints, $\mathbb{P}_3$ with 30274 grid points, and $\mathbb{P}_3$ with 117706 grid points.

We show in the center panel of Figure 6.5, the profile of the total scalar intensity, $|\mathcal{S}|\overline{\psi}_h$, along the segment $\{0 \le x \le 7,\ y = 0\}$. There are six different simulations. The angular discretization is done as in [22, §7.3] using the S_{24} angular quadrature ($\frac{1}{2}N(N + 2) = 312$ angles). We also report in this panel the results given in Olivier et al. [22, Fig. 10] (red line). We observe that the results from the six simulations with the proposed method collapse on a single curve, suggesting that all the spatial features are resolved. There are slight discrepancies of a few per cents with the results from [22], but overall the agreement is satisfactory.

6.7. Lattice problem. We continue with a benchmark test from Peng and McClarren [23, §5.2] called "Lattice problem" therein. The computational domain is $D = (0, 7)^2$. The material is organized in a checkerboard fashion. Each elementary region has size 1×1. The details of the geometry are shown in the left panel of Figure 6.6. There are 11 purely absorbing regions ($\sigma^t = 10$, $\sigma^s = 0$, black boxes in Figure 6.6), there are 37 regions that are purely scattering ($\sigma^t = 1$, $\sigma^s = 1$, white boxes in Figure 6.6), and there is one region with a strong source and scattering material ($q = 1$ $\sigma^t = 1$, $\sigma^s = 1$, yellow box). The homogeneous Dirichlet boundary condition is enforced over the entire boundary of the domain.

The meshes are generated so that the material interfaces are exactly represented. The computations are done on three meshes with polynomial degrees ranging from $\mathbb{P}_1$ to $\mathbb{P}_3$. Five different simulations done: $\mathbb{P}_1$ with 11236 grid points, $\mathbb{P}_1$ with 44521 grid points, $\mathbb{P}_2$ with 44521 grid points, $\mathbb{P}_2$ with 177241 grid points, and $\mathbb{P}_3$ with 99856 grid points. The angular quadrature is done with 312 angular directions (this is the S_{24} quadrature).

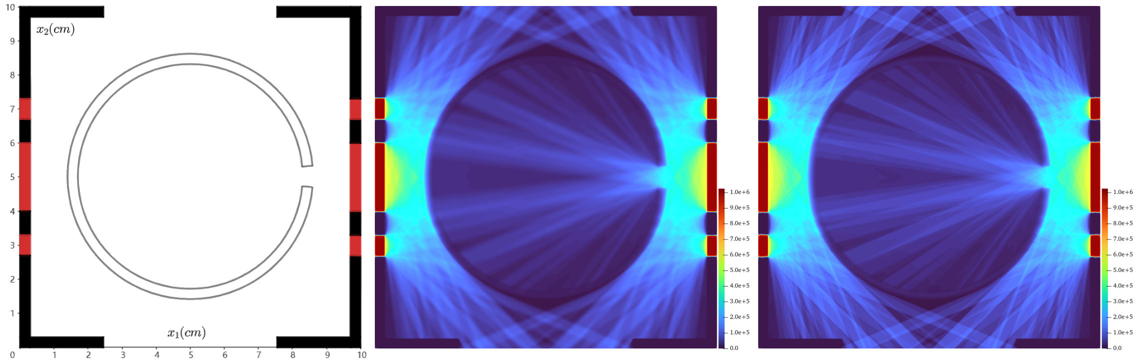
Figure 6.6: Lattice problem. Left: setting of the problem. Center: profile of the total scalar flux, $|\mathcal{S}|\overline{\psi}_h$ for various meshes and polynomial degrees. The symbol "Ref." stands for the data collected from [23]. Right: total scalar flux distribution in logscale ($\mathbb{P}_3$, 99856 grid points)



We show in the center panel of Figure 6.6 the profile of the total scalar intensity, $|\mathcal{S}|\overline{\psi}_h$, along the segment $\{x = 3.5,\ 0 \le y \le 7\}$. We also report in this figure results from Peng and McClarren [23, Fig. 8(e)]. These results are shown in red and identified with the symbol "Ref." The representation is done in log scale along the y-axis. We observe that the agreement between the present method and the reference results is overall quite satisfactory.

6.8. Hohlraum. We finish with a test that is purely qualitative and loosely inspired from the hohlraum problem in Southworth et al. [27, IV.B]. The objective is to give some feeling on how the method behaves when solving a somewhat realistic problem. The problem under consideration is a very simplistic representation of a hohlraum device used in the indirect-drive approach of inertial confinement fusion. The dimensions are not given in the reference, but we use a square domain $D = (0, 10)^2$ (i.e., 10cm×10cm). These values are significantly larger than those of an actual hohlraum, but as the problem is linear, everything can be recalled by a length scale. The zero incidence boundary condition is imposed. The thickness of the walls of the cavity is 0.3. The width of the opening at the top and bottom of the cavity is 5.1. The "spherical" capsule inside the cavity is centered at $(5., 5.)$. Its internal radius is 3.3 and its external radius is 3.6. The opening on the right side of the capsule (simulating a filling hole) is a cone with half angle equal to $5°$ and vertex located at $(5., 5.)$. The sources are meant to simulated simulate the heating of the wall by lasers, hence the sources are only located in the walls of the cavity in the region composed of the points $\boldsymbol{x} = (x_1, x_2)$ where $(0 \le x_1 \le 0.3$ or $9.7 \le x_1 \le 10)$ and $(2.7 \le x_2 \le 3.3$ or $4 \le x_2 \le 6$ or $6.7 \le x_2 \le 7.3)$. The constant value of the source is arbitrarily set to $q = 10^8$. The cross sections are distributed as follows. "Gold" wall of the cavity: $\sigma^t = 10^2$, $\sigma^s = 2.5$. "Helium filling" around the spherical capsule: $\sigma^t = 10^{-4}$ $\sigma^s = 10^{-4}$. "Plastic" capsule wall: $\sigma^t = 10$, $\sigma^s = 6$. "Hydrogen fuel" inside the capsule: $\sigma^t = 10^{-2}$, $\sigma^s = 10^{-2}$. Again, these values are inspired from [27, IV.B] and are not to be taken as actual values. Note that the problem is heterogeneous. It involves streaming and diffusive regions.

Figure 6.7: Hohlraum problem. Left: geometry of the problem. Center, scalar flux, $\overline{\psi}_h$, $\mathbb{P}_1$, 682261 grid points, S_{12} quadrature. Right, scalar flux, $\overline{\psi}_h$, $\mathbb{P}_3$, 864754 grid points, S_{12} quadrature.



We show in the center and right panels of Figure 6.7 the results of two simulations using continuous $\mathbb{P}_1$ and $\mathbb{P}_3$ elements. The meshsize is approximately 0.0125 for the $\mathbb{P}_1$ approximation (682261 grid points) and 0.035 for the $\mathbb{P}3$ approximation (864754 grid points). We use the S_{12} angular quadrature as in [27, IV.B]. This makes 84 directions. The figure shows the scalar flux, $\overline{\psi}_h$, in both cases. The results look visually similar to what is reported in Figure 3(b) in [27] although the color scheme is slightly different.

Appendix A. Preconditioned source iteration and limiting.

We explain here the preconditioned source iteration method that is used in the radiation transport tests reported in the paper. We start with a few definitions to help us make the algorithm more concise. We define the mapping $\Phi : V_h \to V_h$ so that for all $p \in V_h$ (not to be confused with the polynomial degree), $\Phi(p)$ solve the diffusion equation $a(\Phi(p), v) = \int_D pv \,\mathrm{d}x$ for all $v \in V$ (notice that the boundary condition is homogeneous). For all angular quadrature index $k \in \mathcal{L}$, we define the mapping $\Psi^0_k : V_h \to V_h$ so that for all scalar flux $\phi \in V_h$, $\Psi^0_k(\phi)$ solves (4.9) with homogenous boundary condition (i.e., zero incidence) and zero source term (i.e., $q \equiv 0$). Likewise we define $\Psi^*_k \in V_h$ so that Ψ^*_k solves (4.9) with the correct boundary condition and the correct source term q. In summary, $\Phi(p)$, $\Psi^0_k(\phi)$ and Ψ^*_k are defined so that the following holds for all $v \in V_h$:

$$a(\Phi(p), v) = \int_D pv \,\mathrm{d}x, \tag{A.1}$$

$$t(\Psi^0_k(\phi), v) + s_h(\Psi^0_k(\phi), v) + b_h(\Psi^0_k(\phi), v) = \int_D \sigma^s \phi v \,\mathrm{d}x, \tag{A.2}$$

$$t(\Psi^*_k, v) + s_h(\Psi^*_k, v) + b_h(\Psi^*_k, v) = \int_D qv \,\mathrm{d}x + b_k(\alpha^\partial_k, v). \tag{A.3}$$

We set $\Psi^0(\phi) := (\Psi^0_1(\phi), \dots, \Psi^0_L(\phi)) \in V_h^L$ and $\Psi^* := (\Psi^*_1, \dots, \Psi^*_L) \in V_h^L$. Then $\Psi(\phi) := \Psi^0(\phi) + \Psi^*$ solves (4.9) iff $\overline{\Psi(\phi)} = \phi$. Hence, we have to solve the linear system: Find $\phi \in V_h$ so that

$$\phi - \overline{\Psi^0(\phi)} = \overline{\Psi^*}. \tag{A.4}$$

This can be done in a multitude of ways. The simple algorithm we use in the numerical simulations reported in the paper proceeds as follows. We initialize the algorithm by setting $\Phi^0 = \Phi(q)$. Using

Krylov's method to construct the solution, it is natural to define the first search direction to be the residual of (A.4), i.e., $p^0 = \phi^0 - \overline{\Psi^0(\phi^0)} - \overline{\Psi^*}$. Let ϕ^n (here $n \in \mathbb{N}$ is the loop index) be some estimate of ϕ. We add the direction induced by the diffusion limit $r^n := \Phi(\sigma^s p^n)$. Then we search for α^n, $\beta^n \in \mathbb{R}$ so that the new update

$$\phi^{n+1} = \phi^n + \alpha^n p^n + \beta^n r^n, \tag{A.5}$$

minimizes $\|\phi^{n+1} - \overline{\Psi(\phi^{n+1})}\|$ where $\|\cdot\|$ is some norm induced by some inner product $(\cdot,\cdot)$ that can be chosen by the user. As the next search direction is the residual $p^{n+1} = \phi^{n+1} - \overline{\Psi^0(\phi^{n+1})}$, after some algebraic manipulations we obtain

$$p^{n+1} = p^n + \alpha^n(p^n - \overline{\Psi^0(p^n)}) + \beta^n(r^n - \overline{\Psi^0(r^n)}). \tag{A.6}$$

To avoid stalling, which is a standard for steepest descent methods, we enforce p^{n+1} to be orthogonal to p^n. Setting $dp := p^n - \overline{\Psi^0(p^n)}$ and $dr := r^n - \overline{\Psi^0(r^n)}$, the two constraints on α^n and β^n are then

$$\|p^n\|^2 + \alpha^n(p^n, dp^n) + \beta^n(p^n, dr^n) = 0, \tag{A.7}$$

$$\min\left[\|p^n\|^2 + 2\alpha^n(p^n, dp^n) + 2\beta^n(p^n, dr^n) + \|\alpha^n dp^n + \beta^n dr^n\|^2\right]. \tag{A.8}$$

The solution to this quadratique system is

$$\beta^n = \|p^n\|^2 \frac{\|dp^n\|^2(p^n, dr^n) - (dp^n, dr^n)(p^n, dp^n)}{2(dp^n, dr^n)(p^n, dp^n)(p^n, dr^n) - \|dp^n\|^2(p^n, dr^n)^2 - \|dr^n\|^2(p^n, dp^n)^2}, \tag{A.9}$$

$$\alpha^n = \frac{-\|p^n\|^2 - \beta^n\,(p^n, dr^n)}{(p^n, dp^n)}, \tag{A.10}$$

and (A.5) gives the next estimate of the solution. At this point one applies the global mass conserving limiting algorithm (2.6) to $\{\phi^{n+1}\}_{i\in\mathcal{V}}$ with the global mass $\mathcal{M} := \sum_{i\in\mathcal{V}} m_i \phi_i^{n+1}$ and $\phi_i^{\min} = 0$ (and one can also enforce $\phi_i^{\max}$ if it happens that the maximum is a priori known). The algorithms stops when $\|p^{n+1}\|/\|\phi^{n+1}\|$ is smaller than some tolerance.

The final and key part of the algorithm is the conservative limiting (local and global). For each angle $k \in \mathcal{L}$, we proceed as follows: (1) For every $i \in \mathcal{V}$, we use the method of characteristics explained in §3 to compute the lower and upper bounds on the angular intensity, $\{\Psi_{i,k}^{\min}, \Psi_{i,k}^{\max}\}_{i\in\mathcal{V}}$. (2) We then apply the local mass conserving algorithm (2.2)–(2.5) to $\{\Psi_{i,k}\}_{i\in\mathcal{V}}$ using the bounds $\{\Psi_{i,k}^{\min}, \Psi_{i,k}^{\max}\}_{i\in\mathcal{V}}$ computed above. (3) Finally we apply the global mass conserving algorithm (2.6) to $\{\overline{\Psi}_{i,k}\}_{i\in\mathcal{V}}$ with the global mass $\mathcal{M} := \sum_{i\in\mathcal{V}} m_i \overline{\Psi}_{i,k}$, $\overline{\Psi}_{i,k}^{\min} = 0$ (one can also enforce $\overline{\Psi}_{i,k}^{\max}$ if the maximum happens to be a priori known).

Acknowledgments. The first author heartily thanks Dr. Dominic Caron for many fruitful discussions they had at TAMU from 2020 to 2022 over possible extensions of [11]. These investigations unfortunately did not bear fruits but they eventually led the first author to the strategy presented in the paper (stabilized Galerkin + characteristics-based limiting). The first author also thanks the organizers of the 2024 "Mathematics for neutronics" one-day workshop, Feb 24, 2014 at Lab. J.-L. Lions, Paris, for the stimulating discussions that lead to the completion of this exhausting project. Finally, the first author thanks M. Adams, J. Morel (TAMU) and T. Bailey (LLNL) for their indefectible support for this project over the years. The second author sincerely thanks Dr. Ari Rappaport for his kind help, his patience, and meaningful discussions about programming.

References.

Algorithm A.1 Preconditioned source iteration

```
Require: Tolerance $\delta > 0$, upper bounds $\{\Phi_i^{\max}\}_{i\in\mathcal{V}}$ (optional).
$n = 0$, $e = \infty$.    ▷ initialize iteration count and error
$\phi^0 = \Phi(q)$, $p^0 = \phi^0 - \overline{(\Psi^0(\phi^0) + \Psi^*)}$ ← solution of (A.3).    ▷ Initialize scalar flux and residual
while $e > \delta$ do    ▷ iteratively update $\Psi^0(\phi)$
    $dp^n \leftarrow p^n - \overline{\Psi^0(p^n)}$    ▷ set $dp^n$
    $r^n \leftarrow \Phi(\sigma^s p^n)$, solution of (A.1) with source $\sigma^s p^n$.    ▷ set $r^n$
    $dr^n \leftarrow r^n - \overline{\Psi^0(r^n)}$    ▷ set $dr^n$
    $\beta^n \leftarrow \frac{\|p^n\|^2\big(\|dp^n\|^2(p^n,dr^n)-(dp^n,dr^n)(p^n,dp^n)\big)}{2(dp^n,dr^n)(p^n,dp^n)(p^n,dr^n)-\|dp^n\|^2(p^n,dr^n)^2-\|dr^n\|^2(p^n,dp^n)^2}$.    ▷ optimize $\beta^n$
    $\alpha^n \leftarrow \frac{-\|p^n\|^2-\beta^n\,(p^n,dr^n)}{(p^n,dp^n)}$.    ▷ optimize $\alpha^n$
    $p^{n+1} \leftarrow p^n + \alpha^n dp^n + \beta^n dr^n$.    ▷ update $p^n$
    $\phi^{n+1} \leftarrow \phi^n + \alpha^n p^n + \beta^n r^n$.    ▷ update $\phi^n$
    $\mathcal{M} := \sum_{i\in\mathcal{V}} m_i \Phi^{n+1}$    ▷ compute the global mass
    $\Phi_i^{n+1,\min} = 0$ for all $i \in \mathcal{V}$.    ▷ set lower bound
    Optional: $\Phi_i^{n+1,\max} \leftarrow \Phi_i^{\max}$ for all $i \in \mathcal{V}$.    ▷ set upper bound
    $\{\Phi_i^{n+1}\}_{i\in\mathcal{V}} \leftarrow$ (2.6) with above parameters.    ▷ post-processing by global limiter
    $e \leftarrow \frac{\|\phi^{n+1}-\phi^n\|}{\|\phi^{n+1}\|}$.    ▷ estimate error at current step
    $n \leftarrow n + 1$    ▷ go to next iteration
end while
$\{\Psi_{i,k}\}_{i,k\in\mathcal{V}\times\mathcal{L}} \leftarrow \Psi^0(\phi^n) + \Psi^*$.    ▷ Compute final solution
return $(\Psi_{i,k})_{i,k\in\mathcal{V}\times\mathcal{L}}$
```

Algorithm A.2 Conservative limiting

```
Require: Numerical solution $\{\Psi_{i,k}\}_{i,k\in\mathcal{V}\times\mathcal{L}}$, Iteration number $it^{\max}$.
for $it \in \{1 : it^{\max}\}$ do    ▷ iteratively apply local limiter
    for $(i,k) \in \mathcal{V}\times\mathcal{L}$ do    ▷ loop on each DoF
        $\Psi_{i,k}^{\max}, \Psi_{i,k}^{\min} \leftarrow$ (3.3) with scattering source defined with $\overline{\Psi_i}$.    ▷ estimate local bounds
        $\Psi_{i,k}^{\max}, \Psi_{i,k}^{\min} \leftarrow$ Relaxation Algorithm A.3.    ▷ relax local bounds
        $\Psi_{i,k} \leftarrow$ (2.2)–(2.5) with above bounds.    ▷ Local conservative limiting
    end for
end for
$\mathcal{M}_k := \sum_{i\in\mathcal{V}} m_i \Psi_{i,k}$ for all $k \in \mathcal{L}$.    ▷ set global mass for each angular
$\Psi_{i,k}^{\min} = 0$ for all $(i,k) \in \mathcal{V}\times\mathcal{L}$.    ▷ set lower bounds for global limiter
Optional: $\Psi_{i,k}^{\max} \leftarrow$ input data, for all $(i,k) \in \mathcal{V}\times\mathcal{L}$.    ▷ set upper bounds for global limiter
$\{\Psi_{i,k}\}_{i,k\in\mathcal{V}\times\mathcal{L}} \leftarrow$ (2.6), with above bounds.    ▷ apply global conservative limiter
return $\{\Psi_{i,k}\}_{i,k\in\mathcal{V}\times\mathcal{L}}$
```

[1] M. L. Adams. Discontinuous finite element transport solutions in thick diffusive problems. *Nuclear Science and Engineering*, 137(3):298–333, 2001.

[2] M. Ancellin, B. Després, and S. Jaouen. Extension of generic two-component VOF interface advection schemes to an arbitrary number of components. *J. Comput. Phys.*, 473:111721, 20, 2023.

Algorithm A.3 Relaxation

```
Require: Bounds $\{u_i^{\max}, u_i^{\min}\}_{i\in\mathcal{V}}$, $\{u_i\}_{i\in\mathcal{V}}$, FEM basis $\{\varphi_i\}_{i\in\mathcal{V}}$, $u^{\min}$, $u^{\max}$.
$\beta_{ij} := \int_D \nabla\varphi_i \cdot \nabla\varphi_j \,\mathrm{d}\boldsymbol{x}$.                              ▷ set stiffness matrix
$\alpha_i \leftarrow \frac{\sum_{j\in\mathcal{I}(i)^*} d_{ij}\beta_{ij}(u_i - u_j)}{\sum_{j\in\mathcal{I}(i)^*} d_{ij}}$ for each $i \in \mathcal{V}$.     ▷ estimate second-order increment
for $i \in \mathcal{V}$ do                                                  ▷ tighten bounds by minmod
    $\beta_i \leftarrow \alpha_i$.                                           ▷ initialize relaxation parameter
    for $j \in \mathcal{I}(i)^*$ do                                         ▷ minmod process
        if $\beta_i\alpha_j \le 0$ then
            $\beta_i \leftarrow 0$.
        else if $|\beta_i| > |\alpha_j|$ then
            $\beta_i \leftarrow \alpha_j$.
        end if
    end for
end for
$u_i^{\max} \leftarrow \min(u_i^{\max} + |\beta_i|, u^{\max})$ for each $i \in \mathcal{V}$.          ▷ relax upper bound
$u_i^{\min} \leftarrow \max(u_i^{\min} - |\beta_i|, u^{\min})$ for each $i \in \mathcal{V}$.          ▷ relax lower bound
return $\{u_i^{\max}, u_i^{\min}\}_{i\in\mathcal{V}}$
```

[3] P. Bochev, D. Ridzal, G. Scovazzi, and M. Shashkov. Formulation, analysis and numerical study of an optimization-based conservative interpolation (remap) of scalar fields for arbitrary Lagrangian–Eulerian methods. *J. Comput. Phys.*, 230(13):5199–5225, 2011.

[4] P. Bochev, D. Ridzal, and K. Peterson. Optimization-based remap and transport: A divide and conquer strategy for feature-preserving discretizations. *J. Comput. Phys.*, 257:1113–1139, 2014.

[5] E. Burman and P. Hansbo. Edge stabilization for Galerkin approximations of convection-diffusion-reaction problems. *Comput. Methods Appl. Mech. Engrg.*, 193(15-16):1437–1453, 2004.

[6] S. Chandrasekhar. *Radiative Transfer.* Oxford University Press, 1950.

[7] J. Douglas and T. Dupont. Interior penalty procedures for elliptic and parabolic galerkin methods. In R. Glowinski and J. L. Lions, editors, *Computing Methods in Applied Sciences*, pages 207–216, Berlin, Heidelberg, 1976. Springer Berlin Heidelberg.

[8] L. Gosse and G. Toscani. An asymptotic-preserving well-balanced scheme for the hyperbolic heat equations. *C. R. Math. Acad. Sci. Paris*, 334(4):337–342, 2002.

[9] J.-L. Guermond and G. Kanschat. Asymptotic analysis of upwind discontinuous Galerkin approximation of the radiative transport equation in the diffusive limit. *SIAM J. Numer. Anal.*, 48(1):53–78, 2010.

[10] J.-L. Guermond, M. Nazarov, B. Popov, and I. Tomas. Second-order invariant domain preserving approximation of the Euler equations using convex limiting. *SIAM J. Sci. Comput.*, 40(5):A3211–A3239, 2018.

[11] J.-L. Guermond, B. Popov, and J. Ragusa. Positive and asymptotic preserving approximation of the radiation transport equation. *SIAM J. Numer. Anal.*, 58(1):519–540, 2020.

[12] S. Hamilton, M. Benzi, and J. Warsa. Negative flux fixups in discontinuous finite element s_n transport. Saratoga Springs, NY, 2009.

[13] A. Harten and S. Osher. Uniformly high-order accurate nonoscillatory schemes. I. *SIAM J.*

Numer. Anal., 24(2):279–309, 1987.

[14] E. W. Larsen. On numerical solutions of transport problems in the diffusion limit. *Nuclear Science and Engineering*, 83(1):90–99, 1983.

[15] E. W. Larsen, J. E. Morel, and W. F. Miller, Jr. Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes. *J. Comput. Phys.*, 69(2):283–324, 1987.

[16] K. Lathrop. Spatial differencing of the transport equation: Positivity vs. accuracy. *J. Comput. Phys.*, 4(4):475–498, 1969.

[17] E. E. Lewis and W. F. Miller. *Computational methods of neutron transport.* American Nuclear Society, 1993.

[18] C. Liu, B. Riviere, J. Shen, and X. Zhang. A simple and efficient convex optimization based bound-preserving high order accurate limiter for cahn–hilliard–navier–stokes system. *SIAM Journal on Scientific Computing*, 46(3):A1923–A1948, 2024.

[19] P. G. Maginot, J. E. Morel, and J. C. Ragusa. A non-negative moment-preserving spatial discretization scheme for the linearized Boltzmann transport equation in 1-D and 2-D Cartesian geometries. *J. Comput. Phys.*, 231(20):6801–6826, 2012.

[20] P. G. Maginot, J. C. Ragusa, and J. E. Morel. Nonnegative methods for bilinear discontinuous differencing of the sn equations on quadrilaterals. *Nuclear Science and Engineering*, 185(1): 53–69, 2017.

[21] F. Malvagi and G. C. Pomraning. Initial and boundary conditions for diffusive linear transport problems. *J. Math. Phys.*, 32(3):805–820, 1991.

[22] S. Olivier, W. Pazner, T. S. Haut, and B. C. Yee. A family of independent variable Eddington factor methods with efficient preconditioned iterative solvers. *J. Comput. Phys.*, 473:111747, 39, 2023.

[23] Z. Peng and R. G. McClarren. A sweep-based low-rank method for the discrete ordinate transport equation. *J. Comput. Phys.*, 473:111748, 2023.

[24] K. Peterson, P. Bochev, and D. Ridzal. Optimization-based, property-preserving algorithm for passive tracer transport. *Computers & Mathematics with Applications*, 159:267–286, 2024.

[25] R. Sanchez and N. J. McCormick. A review of neutron transport approximations. *Nuclear Science and Engineering*, 80(4):481–535, 1982.

[26] B. Schmidtmann, R. Abgrall, and M. Torrilhon. On third-order limiter functions for finite volume methods. *Bull. Braz. Math. Soc. (N.S.)*, 47(2):753–764, 2016.

[27] B. S. Southworth, M. Holec, and T. S. Haut. Diffusion synthetic acceleration for heterogeneous domains, compatible with voids. *Nuclear Science and Engineering*, 195(2):119–136, 2021.

[28] B. C. Yee, S. S. Olivier, T. S. Haut, M. Holec, V. Z. Tomov, and P. G. Maginot. A quadratic programming flux correction method for high-order dg discretizations of sn transport. *J. Comput. Phys.*, 419:109696, 2020.

[29] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31(3):335–362, 1979.