Active flux schemes on moving meshes
with applications to geometric optics

by

B.S. van Lith, J.H.M. ten Thije Boonkkamp, W.L. IJzerman
Active flux schemes on moving meshes with applications to geometric optics

Bart S. van Lith\textsuperscript{a,1}, Jan H.M. ten Thije Boonkkamp\textsuperscript{a}, Wilbert L. IJzerman\textsuperscript{a,b}

\textsuperscript{a}Department of Mathematics and Computer Science, Eindhoven University of Technology - P. O. Box 513, NL-5600 MB Eindhoven, The Netherlands.
\textsuperscript{b}Philips Lighting - High Tech Campus 44, 5656 AE, Eindhoven, The Netherlands.

Abstract

Active flux schemes are finite volume schemes that keep track of both point values and averages. The point values are updated using a semi-Lagrangian step, making active flux schemes highly suitable for geometric optics problems on phase space, i.e., to solve Liouville’s equation. However, curved optics lead to moving boundaries in phase space, necessitating moving meshes. To this end, we also introduce a novel way of defining active flux schemes on moving meshes that keeps with the method of lines approach. We show, using scaling arguments, that our scheme outperforms the current industry standard, ray tracing. It has higher accuracy as well as a more favourable time scaling. In numerical experiments, we demonstrate that the active flux scheme is orders of magnitude more accurate and faster than ray tracing.

Keywords: Essentially non-oscillatory, WENO, high-resolution scheme, hyperbolic conservation laws, nonlinear interpolation, spectral analysis.

1. Introduction

Illumination optics is a branch of nonimaging optics that is concerned with lighting applications \cite{1, 2}. The optics involved are very different from imaging optics, in fact, imaging effects in illumination optics are highly undesirable. As such, a radically different approach to optics is needed. Phase

\textsuperscript{1}Corresponding author

Email address: b.s.v.lith@tue.nl (Bart S. van Lith)
space distributions presents the possibility of such a new approach, as has
been noted by Rausch and Herkommer [3, 4].

In a previous paper we derived a first-order upwind finite difference scheme
for Liouville’s equation [5]. Although valid and useful, higher-order meth-
ods also tend to be more efficient with computational resources. A method
is needed that is high-order accurate, conservative and somehow locally de-
defined. High-order finite difference methods have expansive stencils, leading
to complications when optical interfaces are present. Active flux schemes,
otherwise known as multi-moment finite volume schemes, satisfy all these
criteria [6, 7, 8].

The essential characteristic of active flux schemes is that fluxes are com-
puted independently from the average values. This is facilitated by separately
keeping track of the point values on element boundaries by means of a semi-
Lagrangian step. The point values completely determine the fluxes. Applied
to geometric optics, this means the point values are found by local ray tracing
[9].

2. Liouville’s equation and geometric optics

Geometric optics can be cast into a Hamiltonian system with canonical
coordinates \( q \in \mathbb{R}^d \), denoting position, and \( p \in \mathbb{R}^d \) \((d = 1, 2)\) the momentum
coordinates [10]. Here \( d = 1 \) for two-dimensional optics and \( d = 2 \) for three-
dimensional optics. These quantities evolve as a function of the length down
the optical axis, denoted \( z \). The momentum in \( \mathbb{R}^3 \), \( \vec{p} = (p_z, p) \), is restricted
to Descartes’ sphere, \(|\vec{p}| = n(z, q)\) with \( n \) the refractive index field [10]. At
a given position \( z \), one can visualise the phase space coordinates as a light
ray intersecting a screen perpendicular to the optical axis. In this picture, \( q \)
is the position on the screen and \( p \) is related to the angle the ray makes with
the screen. The ray’s evolution as the screen moves along the optical axis is
then given by

\[
\begin{align}
\frac{dq}{dz} &= \frac{\partial h}{\partial p}, \\
\frac{dp}{dz} &= -\frac{\partial h}{\partial q},
\end{align}
\]

where \( h \) is the Hamiltonian. In this setting, \( h(z, q, p) = -\sigma \sqrt{n(z, q)^2 - |p|^2} \),
where \( \sigma \in \{-1, 0, 1\} \) indicates the forward direction of the ray, with \( \sigma = 1 \).
travelling in the positive $z$-direction, $\sigma = -1$ travelling in the negative $z$-direction and $\sigma = 0$ being marginal rays that travel perpendicular to the optical axis. For simplicity of the presentation, we assume that all rays travel in the forward direction, i.e., $\sigma = 1$.

The collection of all positions and momenta is known as phase space $\mathcal{P}$. A set of phase space coordinates will be abbreviated as $y = (q, p) \in \mathcal{P}$. Let us introduce the symplectic matrix $S \in \mathbb{R}^{2d \times 2d}$ and the gradient in phase space given by

$$S = \begin{pmatrix} O & I \\ -I & O \end{pmatrix}, \quad \nabla = \left( \frac{\partial}{\partial q}, \frac{\partial}{\partial p} \right), \quad (2)$$

where $O$ is a matrix of zeros and $I$ is the identity matrix, both in $\mathbb{R}^{d \times d}$. Hamilton’s equations (1) can then be written as

$$\frac{dy}{dz} = S \nabla h. \quad (3)$$

The above discussion is valid for a single ray and we may generalise this to phase space distributions, denoted by $\rho$, by noting that energy is transported along rays [11]. As the problem is formulated independently of time, the light source is stationary and we may therefore also interpret $\rho$ as being a (scaled) power per unit area per solid angle. Since energy, or power in this case, is transported along rays, in the absence of attenuation or diffusion the distribution satisfies

$$\rho(z, q(z), p(z)) = \rho_0(q(0), p(0)), \quad (4)$$

where $(q(z), p(z))$ is a solution to (1). Here, $\rho_0$ is the initial distribution. Note that (4) provides the physically relevant solution even when rays are refracted or reflected. Supposing sufficient smoothness, however, we may differentiate (4) with respect to $z$ and use (1) to obtain Liouville’s equation

$$\frac{\partial \rho}{\partial z} + \left( \frac{\partial h}{\partial p} \cdot \frac{\partial \rho}{\partial q} - \frac{\partial h}{\partial q} \cdot \frac{\partial \rho}{\partial p} \right) = 0. \quad (5)$$

Using the shorthand introduced earlier, (3) allows us to write Liouville’s equation succinctly as

$$\frac{\partial \rho}{\partial z} + (S \nabla h) \cdot \nabla \rho = 0. \quad (6)$$
Assuming that $h$ is twice differentiable, we may rewrite the advection part of Liouville’s equation as

\[(S \nabla h) \cdot \nabla \rho = \nabla \cdot (\rho S \nabla h) - \rho \nabla \cdot (S \nabla h).\] (7)

The divergence of the velocity field is identically zero for sufficiently smooth Hamiltonians, which is more easily seen when written out component-wise, i.e.,

\[\nabla \cdot (S \nabla h) = \sum_i \frac{\partial}{\partial q_i} \left( \frac{\partial h}{\partial p_i} \right) - \frac{\partial}{\partial p_i} \left( \frac{\partial h}{\partial q_i} \right) = 0.\] (8)

Let us write the velocity field as $u = S \nabla h$, then Liouville’s equation may be expressed as

\[\frac{\partial \rho}{\partial z} + \nabla \cdot (\rho u) = 0,\] (9)

a simple continuity equation. This is the conservative form of Liouville’s equation, which is equivalent to what we may call the advective form (5) whenever $h$ is sufficiently smooth. We shall assume $h$ to be piecewise smooth so that in smooth regions we may use either form. Wherever $h$ is discontinuous neither form is applicable and we must use (4).

3. Active flux on a moving mesh

The conservative form of Liouville’s equation invites us to integrate over a volume $\Omega \subset \mathcal{P}$ and apply the divergence theorem, i.e.,

\[\frac{d\bar{\rho}}{dz} = -\frac{1}{|\Omega|} \int_{\Omega} \nabla \cdot (\rho u) \, dy = -\frac{1}{|\Omega|} \int_{\partial \Omega} \rho u \cdot n \, dS,\] (10)

where we have divided by $|\Omega|$, the volume of $\Omega$. Here, $\bar{\rho}$ is the average over the volume while $n$ is the outward unit normal. We shall restrict ourselves from here on to a two-dimensional setting, that is $q$ and $p$ reduce to scalars.

Active flux schemes operate by exploiting the notion that the boundary values completely determine the flux in (10). The boundary integral is approximated by quadrature rules that depend on a finite number of boundary values. These values are determined separately from the averages by means of a semi-Lagrange step [16], otherwise known as ray tracing in an optics context. Like this, the right-hand side of (10) can be approximated at any
z, so that the z-integration becomes a definite integral. For the moment, we will assume that we can find the point values to sufficient accuracy and focus on the evolution of the average values. We will, however, have to consider (10) on a moving mesh, as curved optics manifest themselves as moving boundaries in phase space.

Along the optical interfaces, the laws of optics determine what happens to ρ. Assuming a complete remeshing can be avoided, several or all nodes have to be moved in order to stay aligned to the interface. Ding et al. have developed an active flux scheme on a moving mesh by discretising space-time as a whole [24]. We take a different, and so far as we are aware novel, approach and leave time continuous. This fits in with the method of lines paradigm and allows us to discretise time separately and apply any numerical integrator.

The mesh may move in any number of ways, for instance by being specified as a spring-dashpot system. Typically, this involves using a numerical integrator to find the mesh node positions and some approximation to compute the velocities. For simplicity, however, we take node positions and velocities as given. Consider Reynolds’ Transport Theorem on the, now moving, test domain Ω(z), i.e.,

\[
\frac{d}{dz} \int_{Ω(z)} ρ dy = \int_{Ω(z)} \frac{∂ρ}{∂z} + \nabla_y · (ρv) dy,
\]

where \(v\) is the velocity field of the moving element. Let us fix the shape of Ω to be a triangle with vertices \((q_1, p_1)\), \((q_2, p_2)\) and \((q_3, p_3)\). The vertices are allowed to vary with \(z\). Any triangle may be mapped to a reference triangle \(χ\), whose vertices are given by the vertices \((0,0)\), \((1,0)\) and \((0,1)\), see Figure 1. The coordinates on the reference triangle are denoted \((ξ, η)\). The affine transformation from the reference triangle to Ω is given by

\[
\begin{pmatrix} q \\ p \end{pmatrix} = A(z) \begin{pmatrix} ξ \\ η \end{pmatrix} + \begin{pmatrix} q_1 \\ p_1 \end{pmatrix},
\]

where \(A\) is given by

\[
A = \begin{pmatrix} q_2 - q_1 & q_3 - q_1 \\ p_2 - p_1 & p_3 - p_1 \end{pmatrix}.
\]

Let us denote the determinant of \(A(z)\) as \(J(z)\), then \(|J(z)| = |Ω(z)|/|χ| = 2|Ω(z)|\). We choose the labels of the element vertices in physical space in
such a way that $\mathcal{J} > 0$, meaning counter-clockwise. Furthermore, $\mathcal{J} = 0$ occurs only when the vertices are collinear, clearly a situation that is to be avoided.

\begin{equation}
\frac{\partial}{\partial y_i} = \sum_{j=1}^{2} \frac{\partial \xi_j}{\partial y_i} \frac{\partial}{\partial \xi_j},
\end{equation}

where $y_1 = q$, $y_2 = p$, $\xi_1 = \xi$ and $\xi_2 = \eta$. This component-wise expression can be converted to a matrix-vector form by realising that $\frac{\partial \xi}{\partial y} = A^{-1}(z)$, hence

\begin{equation}
\nabla_y = A^{-T}(z) \nabla \xi.
\end{equation}

The velocity field due to the movement of the vertices can be found by differentiating (12), i.e.,

\begin{equation}
v(z) = A'(z) \begin{pmatrix} \xi \\ \eta \end{pmatrix} + y'(z),
\end{equation}

where the prime denotes differentiation with respect to $z$.

Next, we apply Liouville’s equation (9) to (11), so that we find

\begin{equation}
\frac{d}{dz} \int_{\Omega(z)} \rho \, dy = \int_{\Omega(z)} \nabla_y \cdot \left( \rho(v - u) \right) \, dy.
\end{equation}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Mapping from the standard to a general triangle.}
\end{figure}
The volume integral of $\rho$ is by definition equal to the average value times the volume size. Furthermore, transforming the integral to the standard domain yields
\[ \frac{d}{dz} (\bar{\rho} | \Omega(z)) = J(z) \int_{\chi} (A^{-T}(z) \nabla \xi \cdot (\rho(v - u))) \, d\xi. \quad (18) \]

Since $A$ is constant with respect to the spatial coordinates, the velocity field on the standard triangle $\chi$ can be defined as
\[ \tilde{u} = A^{-1}(z) (u - v). \quad (19) \]

Applying the divergence theorem on the standard triangle, we obtain
\[ \frac{d}{dz} \left( \frac{1}{2} \bar{\rho} J \right) = -J \oint_{\partial \chi} \rho(\tilde{u} \cdot n) \, ds, \quad (20) \]
where we have replaced the volume on the left-hand-side by the determinant.

In some situations, there may be some ambient free-stream flux, corresponding to $\rho = \text{const}$. since then the flux is constant too. An example from aerospace would be the ambient free stream far from an airfoil. The scheme can be made free-stream preserving, i.e., preserving a constant state exactly, by considering the evolution of the volume $|\Omega|$, obtained from Reynolds’s transport theorem, leading to
\[ \frac{dJ}{dt} = 2 \oint_{\partial \chi} \tilde{v} \cdot n \, d\sigma, \quad (21) \]
where $\tilde{v} = A^{-1}(z)v$. This ODE for $J$ in an element is sometimes referred to as the geometric conservation law [25, 26]. Alternatively, one can set $\rho = 1$ in (20) to derive (21). If the numerical solution is to be free-stream preserving, we need (21) and (20) to be satisfied simultaneously in a discrete sense, as we will prove in Subsection 3.2.

3.1. Discretising the boundary flux

To proceed, we need to approximate the right-hand side of (20). First, we can split up the boundary integral into the separate line integrals over each edge. We define $f_k$ as the flux over edge $\ell_k$, i.e.,
\[ f_k(\rho) = 2 \int_{\ell_k} \rho \tilde{u} \cdot n_k \, ds, \quad (22) \]
where $\ell_k$, $k = 1, 2, 3$ are the edges of the standard triangle and $n_k$ are their outward unit normals. Each line integral can now be approximated by a quadrature rule. Here, we choose Simpson’s rule to approximate the integrals, resulting in the addition of the midpoints of each side to the collection of nodes, see Figure 2.

Figure 2: Gauß-Lobatto points around the standard triangle $\chi$ with values labelled that should be tracked. The edges are also indicated.

Simpson’s rule is the three-point case of the Gauß-Lobatto rules [12]. As an alternative, one could consider the Gauß rules, which have a higher order of accuracy for a fixed number of nodes. However, it turns out both give exactly the same order when integrating around the boundary of a polygon.

**Lemma 1.** Given a polygon, Gauß and Gauß-Lobatto rules need exactly the same number of nodes to provide exact integration over the boundary for a polynomial of degree $p$. Specifically, $s(p + 1)/2$ points in total are needed to provide exact integration on a polygon of $s$ sides for all polynomials of degree $p$.

**Proof.** 1. The degree of exactness for Gauß quadrature is $p = 2n_G - 1$, where $n_G$ is the number of nodes per side. Note that for the Gauß nodes, there are no nodes on the vertices of the polygon. The total number of nodes needed to achieve degree of exactness $p$ is therefore $s(p + 1)/2$. 

8
2. For Gauß-Lobatto nodes, the degree is \( p = 2n_{GL} - 3 \), with \( n_{GL} \) the number of nodes per side including the boundary points. The number of nodes per side needed to achieve degree of exactness \( p \) is therefore \( (p + 3)/2 \). However, the nodes on the vertices can be used twice, once for either edge connecting to it. Hence, \( s \) nodes in total can be used twice, so that the total number of nodes needed becomes \( s(p + 3)/2 - s = s(p + 1)/2 \).

From the lemma, we can see that the Gauß-Lobatto rules have a slight advantage over the Gauß rules since each vertex, which is already needed to determine the mesh, can be used twice. We label the points as in Figure 2 and define the unit direction vectors \( e_1 = (1, 0)^T \) and \( e_2 = (0, 1)^T \), so that we may approximate the flux over each edge as follows,

\[
\begin{align*}
    f_1(\rho) &\approx -\frac{1}{3} (\rho_1 \tilde{u}_1 + 4\rho_2 \tilde{u}_2 + \rho_3 \tilde{u}_3) \cdot e_2, \\
    f_2(\rho) &\approx \frac{1}{3} (\rho_3 \tilde{u}_3 + 4\rho_4 \tilde{u}_4 + \rho_5 \tilde{u}_5) \cdot (e_1 + e_2), \\
    f_3(\rho) &\approx -\frac{1}{3} (\rho_5 \tilde{u}_5 + 4\rho_6 \tilde{u}_6 + \rho_1 \tilde{u}_1) \cdot e_1,
\end{align*}
\]

where \( \rho = (\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6)^T \) is the vector of point values, while \( \tilde{u}_i \) is the transformed velocity field at the same position as \( \rho_i \). Note that the normalisation of \( n_2 \) cancels against the length of the hypotenuse. By definition, the total averaged flux over the boundary is given by

\[
F(\rho) = \sum_{k=1}^{3} f_k(\rho). \tag{24}
\]

The boundary integral occurring in (21) is approximated in the same way, and we call it \( G(z) \). Note that \( G \) does not depend on \( \rho \) and can be computed if the mesh velocities are known, or can be approximated up to sufficient order. Together, they form the coupled set of differential equations given by

\[
\begin{align*}
    \frac{d}{dz} (\bar{\rho}J) &= -J(z) F(z), \tag{25a} \\
    \frac{dJ}{dz} &= J(z) G(z), \tag{25b}
\end{align*}
\]

where \( F(z) = F(\rho(z)) \) approximates the total volume-averaged flux across the boundary.
3.2. “Time” integration

To conclude, we introduce the $z$-discretisation $z^t = t \Delta z$ with $t = 0, 1, \ldots, M$ with $M$ the number of steps and $\Delta z = \frac{Z}{M}$ for a given integration length $Z$. In order to find the average values at the next $z$-level, a definite integral must be evaluated, or rather approximated. Note that if $\rho$ is a constant, then also $F(z) = -\rho G(z)$. This follows from the fact that the boundary flux integration is exact if the flux is any cubic polynomial. The only contribution is therefore from the movement of the boundary, leading to a linear total flux even if the flux function itself is nonlinear.

Similar to the results of Acosta Minoli et al. for spectral element methods on moving meshes [25], we find that (25a) and (25b) must be integrated simultaneously using the same time integrator. Perhaps counter-intuitively, a free-stream preserving scheme needs to use the values obtained from numerical integration of (25b) rather than the exact value of the Jacobian, even if it is available.

**Theorem 1.** Consider the system (25), where we define $\tilde{\rho} = \bar{\rho} J$, so that

\[
\frac{d\tilde{\rho}}{dz} = -J(z)F(z), \quad \frac{dJ}{dz} = J(z)G(z).
\]

Then the $z$-integration is free-stream preserving if this system is integrated with any general linear method and the average value is defined as

\[
\bar{\rho}^t := \frac{\tilde{\rho}^t}{J^t}.
\]

**Proof.** Let’s write one time step of the integrator as $\psi$, i.e., $\tilde{\rho}^{t+1} = \tilde{\rho}^t + \psi(-J^t F^t)$ where $J^t$ is the approximation to $J(z^t)$ etc. Any general linear method satisfies $\psi(cx) = c\psi(x)$ for any $c \in \mathbb{R}$ and linear right-hand side. Next, assume that $\rho$ is constant, so that $\tilde{\rho}^t = \rho J^t$ holds at the current $z$-step. Furthermore, we have $F(z) = -\rho G(z)$, so that the right-hand side is indeed linear in $\rho$. The next $z$-step is then given by

\[
\tilde{\rho}^{t+1} = \tilde{\rho}^t + \psi(-J^t F^t) = \tilde{\rho}^t + \psi(\rho J^t G^t) = \tilde{\rho}^t + \rho \psi(J^t G^t) = \rho (\bar{J}^t + \Psi(J^t G^t)) = \rho \bar{J}^{t+1}.
\]

The average value is recovered as $\bar{\rho}^{t+1} = \frac{\tilde{\rho}^{t+1}}{\bar{J}^{t+1}} = \rho$, so that the constant state is preserved. \qed
Remark 1. General linear methods are a broad class of numerical integrators [17, 27]. The class contains all Runge–Kutta methods, linear multistep methods and all predictor-corrector methods composed thereof.

The theorem ensures that any Runge–Kutta method will provide a free-stream preserving scheme in combination with the spatial discretisation as introduced earlier. Moreover, for one-step methods, the exact value of the Jacobian $J$ can be inserted at the beginning of $z$-step. We elect to use the standard fourth-order RK4 method, as this reduces to Simpson’s rule whenever $J$ is constant. Note that the system (25) is a one-way coupled system, where $\bar{\rho}$ depends on $J$, but $J$ can be computed independently under the assumption that the mesh motion does not depend on $\rho$. We denote the RK4 stages of (25b) as

\begin{align}
J^{(1)} &= J(z^t), \\
J^{(2)} &= J(z^t) + \frac{1}{2} \Delta z J^{(1)} G^t, \\
J^{(3)} &= J(z^t) + \frac{1}{2} \Delta z J^{(2)} G^{t+1/2}, \\
J^{(4)} &= J(z^t) + \Delta z J^{(3)} G^{t+1/2}.
\end{align}

Here, we used the exact value of the Jacobian $J(z^t)$ and the shorthand notation $G^t = G(z^t)$ etc. The resulting $z$-integration scheme is given by

\begin{align}
J^{t+1} &= J(z^t) + \frac{\Delta z}{6} \left[ J^{(1)} G^t + 2 \left( J^{(2)} + J^{(3)} \right) G^{t+1/2} + J^{(4)} G^{t+1} \right], \\
\bar{\rho}^{t+1} &= \frac{J(z^t)}{J^{t+1}} \bar{\rho}^t - \frac{\Delta z}{6 J^{t+1}} \left[ J^{(1)} F^t + 2 \left( J^{(2)} + J^{(3)} \right) F^{t+1/2} + J^{(4)} F^{t+1} \right].
\end{align}

From (27) - (28), one can clearly see that if $G = 0$, the Jacobian is constant and the scheme reduces to Simpson’s rule in $z$, and the whole solver reduces to a more standard active flux scheme. Moreover, whenever $\rho$ is a constant, $F^t = -\rho G^t$, from which one can see that $\bar{\rho}^{t+1} = \rho$, i.e., this numerical scheme will exactly preserve a constant solution.

3.3. Finding point values

Now that we have discussed how to update the averages when the point values are given, we discuss how to find the point values themselves. Where other finite volume schemes typically employ some reconstruction from the
average values, e.g. a WENO scheme [13, 14, 15], the active flux method keeps track of point values \( \rho \) on the boundary \textit{separately}\footnote{In fact, the name “active flux” originates from this distinction. Other finite volume schemes that do not actively keep track of boundary values may be classified as passive flux schemes.}.

The point values needed to complete the scheme are obtained using a semi-Lagrangian step [16], i.e., by integrating Hamilton’s equations (1) and applying (4), which in the 2D case reduces to

\[
\rho(z + \Delta z, q(z + \Delta z), p(z + \Delta z)) = \rho(z, q(z), p(z)).
\]

(29)

This is otherwise known as the method of characteristics (MOC). The characteristics of Liouville’s equation (5) are simply light rays. In the context of optics the MOC is therefore better known as ray tracing [9, 1]. Recall that \( u = S \nabla h \), so that Hamilton’s equations can be written as

\[
\frac{dy}{dz} = u(z, y).
\]

(30)

As the system is Hamiltonian in nature, it is prudent to solve it using a symplectic, or geometric, integrator [17, 18]. The semi-Lagrangian step uses the terminal condition \( y_{t+1} = (q_{t+1}, p_{t+1})^T \), that is to say the ray should end on a node.

As it turns out, the position \( y^t \) only has to be determined to second-order accuracy in \( z \), i.e. \( y^t = y(z^t) + \mathcal{O}(\Delta z^2) \), to achieve global third-order accuracy. More details on why are given in Subsection 3.6. Taking this fact as given for the moment, the implicit midpoint provides an excellent integrator for our purposes. It is a second-order accurate symplectic integrator that is stable under time reversal. It is given by

\[
y^t = y^{t+1} - \Delta z u \left( z^t + \frac{1}{2} \Delta z, \frac{y^t + y^{t+1}}{2} \right),
\]

(31)

which is an implicit equation for \( y^t \) that can be solved using, for instance, Newton’s method.

As an added benefit, the midpoint method comes with a built-in second-order accurate approximation to \( y(z^{t+\frac{1}{2}}) \), which is also required by the scheme...
according to (28). To see this, we Taylor expand to find

\[
y(z^{t+\frac{1}{2}}) = y\left(z^{t+\frac{1}{2}}\right) + \frac{1}{2} \Delta z u \left(z^{t+\frac{1}{2}}, y\left(z^{t+\frac{1}{2}}\right)\right) + O(\Delta z^2), \tag{32a}
\]

\[
y(z^t) = y\left(z^{t+\frac{1}{2}}\right) - \frac{1}{2} \Delta z u \left(z^{t+\frac{1}{2}}, y\left(z^{t+\frac{1}{2}}\right)\right) + O(\Delta z^2). \tag{32b}
\]

Indeed, taking the average of the two expansions lead to

\[
\frac{y(z^t) + y\left(z^{t+\frac{1}{2}}\right)}{2} = y\left(z^{t+\frac{1}{2}}\right) + O(\Delta z^2). \tag{33}
\]

Once the point \(y^t\) has been approximated, we perform a local search over the neighbouring elements. By construction the two points \(y^t\) and \(y^{t+\frac{1}{2}}\) will always be in the same element. After identifying the correct element, both points are transformed to the reference coordinates in \(\chi\). Therefore, a reconstruction of \(\rho\) is required on the standard element.

### 3.4. Reconstruction

The approximation in each element consists of a polynomial reconstruction, made up of a quadratic interpolation of the point values together with a bubble function that compensates for the average value. In the following discussion, we again omit any element markers for brevity. The quadratic interpolation on the standard triangle can be written as

\[
\tilde{\rho}(\xi) = \frac{1}{2} \xi^T R \xi + r^T \xi + \rho_1, \tag{34}
\]

with \(\xi = (\xi, \eta)^T\). The symmetric matrix \(R \in \mathbb{R}^{2 \times 2}\) approximates the Hessian matrix while the vector \(r \in \mathbb{R}^2\) approximates the gradient, both at \(\xi = 0\). To determine them, we simply substitute the node locations and demand that the value of \(\tilde{\rho}\) equals the point value of \(\rho\). This leads to

\[
R = 4 \begin{pmatrix}
\rho_1 - 2\rho_2 + \rho_3 & \rho_1 - \rho_2 + \rho_4 - \rho_6 \\
\rho_1 - \rho_2 + \rho_4 - \rho_6 & \rho_1 - 2\rho_6 + \rho_5
\end{pmatrix} \tag{35a}
\]

and

\[
r = \begin{pmatrix}
-3\rho_1 + 4\rho_2 - \rho_3 \\
-3\rho_1 + 4\rho_6 - \rho_5
\end{pmatrix}, \tag{35b}
\]

where one can recognise undivided differences over the \(\xi\)- and \(\eta\)-directions, with the off-diagonal elements of \(R\) representing the mixed spatial derivative.
The eagle-eyed observer will notice that this interpolation does not necessarily accommodate the average value \( \bar{\rho} \). Indeed, the average value of \( \bar{\rho} \), given by

\[
2 \int \int_{\chi} \bar{\rho}(\xi, \eta) \, d\xi \, d\eta = \frac{1}{3} (\rho_2 + \rho_4 + \rho_6),
\]

may certainly be different from \( \bar{\rho} \). To ensure that the average value of the reconstruction is equal to \( \bar{\rho} \), and to thus ensure that the scheme is conservative, we must add a third-order function that compensates for the deficit. This function must furthermore not interfere with the quadratic interpolation already established, hence it must be zero on the triangle edges. Such a function is called a bubble function, which can be interpreted as the product of all three barycentric coordinates, i.e.,

\[
\varphi(\xi, \eta) = 60 \xi \eta (1 - \xi - \eta).
\]

The bubble function is normalised such that it has unit average value, i.e.,

\[
\int \int_{\chi} \varphi \, d\xi \, d\eta = |\chi| = \frac{1}{2}.
\]

As a consequence, the total local reconstruction of \( \rho \) on the standard domain is given by

\[
\rho_{\text{tot}}(\xi, \eta) = \bar{\rho}(\xi, \eta) + \left( \bar{\rho} - \frac{1}{3} (\rho_2 + \rho_4 + \rho_6) \right) \varphi(\xi, \eta).
\]

### 3.5. Summary of the algorithm

The algorithm resulting from the synthesis of the previous discussion can be summarised as follows. As input the algorithm requires a triangular mesh with midpoints added, together with an initial condition that provides point values and averages at \( z^0 = 0 \).

1. For each node in the mesh, perform the following actions:
   - (a) Compute the position of the ray coming through the node at \( z^{t+1} \) by (31) and the ray midpoint by (33).
   - (b) Perform a local search over the neighbouring elements to find from which element the characteristics originate.
   - (c) Transform \( y^t \) and \( y^{t+\frac{1}{2}} \) to the standard coordinates, i.e., the inverse transformation to (12).
   - (d) Find and store the point values using the reconstruction, i.e., (34), (35), (37) and (38).

2. For each element in the mesh, perform the following actions:
(a) Compute the local transformed velocity field according to (19).
(b) Compute the flux function at $z^t$ and $z^{t+\frac{1}{2}}$ using (23) and (24).
(c) Update the average values using (28).

3. Go to step 1 unless $z^{t+1} = Z$.

Step 1 requires a loop over all the nodes, where each substep requires a constant amount of work. Step 2 requires a loop over all elements, where again each substep requires a constant amount of work. Thus, the amount of work of step 1 and 2 results in an amount of work scaling with $O(N_{\text{nodes}} + N_{\text{elements}})$. The number of nodes scales linearly with the number of elements, which can be seen by realising that the mesh can be interpreted as a planar graph and using Euler’s characteristic [19]. Step 3 results in the algorithm repeating $M$ times. To conclude, the work needed by the algorithm scales as $O(MN_{\text{elements}})$.

3.6. A CFL condition

As with many hyperbolic solvers, the active flux scheme is subject to a Courant-Friedrichs-Lewy (CFL) condition, see for instance [20, 21], i.e.,

$$\Delta z \leq \frac{\Delta y}{u_{\text{max}}},$$

where $\Delta y$ is the smallest spatial size in the mesh and $u_{\text{max}}$ is the maximum local velocity occurring in the problem. Here, the CFL condition arises from the assumption that the semi-Lagrangian step returns a point in a neighbouring element. For each node, we restrict the search set to all elements for which the node in question lies on the boundary. Thus, midpoint nodes always have a search set of two elements, while vertex nodes can have at most a search set of $\frac{2\pi}{\alpha_{\min}}$, where $\alpha_{\min}$ is the minimum angle in the mesh.

For stability, the analytical domain of dependence must lie entirely within the numerical domain of dependence, which is the search set. The analytical domain of dependence can be bounded by a circle of radius $u_{\text{max}}\Delta z$ centred on the node, which we call the bounding circle. The time step $\Delta t$ must be chosen such for all nodes, the bounding circle lies entirely within the search set, see Figure 3.

The shortest distance from each node to the boundary of its numerical domain of dependence can be found by dropping an altitude line. Clearly, the shortest altitude line in the mesh originates from a midpoint. Calling the edge length $a$, basic trigonometry tells us that $\Delta y = \frac{1}{2}a\sin\alpha$, which must
be bigger than the radius of the bounding circle to guarantee stability. The minimum angle often serves as a measure for mesh quality [22].

Most mesh generators have a minimum angle that can be set, here we used the Triangle software package created by Shewchuck [23], where a minimum angle of $32^\circ$ usually works well. The minimal distance is estimated as

$$\Delta y = \frac{1}{2} \min_i (a_i \sin \alpha_i) \geq \frac{1}{2} \sin \alpha_{\text{min}} \min_i a_i,$$

(40)

where $\alpha_i$ and $a_i$ are the collection of angles and edges in the mesh for $i = 1, 2, \ldots$. To find the maximum velocity simply requires a loop over all nodes in the mesh, evaluating the local velocity field.

With the CFL condition, the scaling behaviour between spatial and temporal grid sizes is straightforward: $\Delta z = \mathcal{O}(\Delta y)$. Earlier, we asserted that the scheme only needs to resolve the point values to second order accuracy in $\Delta z$. The flux is integrated around the boundary so that the fluxes carry a mixed error of $\mathcal{O}(\Delta y \Delta z^2)$. Thus, the scaling of the CFL condition then implies that the total error scales in the fluxes scales as $\mathcal{O}(\Delta y^3)$. 

Figure 3: Sketch of the numerical domain of dependence of a node, its neighbouring elements, and a bound on the analytical domain of dependence, the circle. The radius of the circle is $b_{\text{max}} \Delta t$. 


3.7. Optical systems

For optical systems the characteristics, light rays, change discontinuously when encountering an optical interface. In particular, a ray that encounters an interface will be refracted according to Snell’s law or reflected according to the law of specular reflection. Clearly, this behaviour must be incorporated into the solver to find physically relevant solutions. In our previous work [5], we derived an explicit version of Snell’s law on phase space, i.e.,

\[ S(p; n_1, n_2, \nu) := \begin{cases} 
  p - \left( \psi + \text{sgn}(n_2) \sqrt{\delta} \right) \nu & \text{if } \delta \geq 0, \\
  p - 2\psi \nu & \text{if } \delta < 0,
\end{cases} \tag{41a} \]

where

\[ \delta := n_2^2 - n_1^2 + \psi^2 \quad \text{and} \quad \psi := p\nu \pm \sqrt{n_1^2 - p^2 \sqrt{1 - \nu^2}}, \tag{41b} \]

where \( \psi \leq 0 \) due to angle conventions in Snell’s law, \( n_1 \) and \( n_2 \) are the refractive indices and \( \nu \) is the \( q \)-component of the surface normal. Since \( \rho \) is transported along rays and rays are discontinuous, the solution itself will be discontinuous, even for smooth initial conditions.

As a consequence, it is necessary to align elements along the interface. It turns out that mesh alignment is a very important principle for geometric optics problems. If the mesh is properly aligned with an optical interface, the active flux scheme works with very little modification. Note that for refractive surfaces, it is also necessary to have two values for all the nodes that are on the interface, one for each one-sided limit. Finally, optical interfaces may have critical momenta at which the behaviour of Snell’s law changes from refractive to reflective. This occurs when \( \delta = 0 \) according to (41a). It is prudent to include nodes at critical momenta to ensure continuity of the flux across each interface edge.

In implementing the active flux scheme, we need to know which elements are on an interface. The point values for these elements have to be treated slightly differently. If a backward ray from these nodes crosses the interface, Snell’s law or the law of specular reflection has to be applied. As the point values on element boundaries uniquely specify the flux, this is sufficient to capture the physical solution. There are two useful remarks to make about elements that connect to the interface:

- Only nodes that lie on the interface will be affected by the interface, due to the CFL condition.
• Since $\rho$ is continuous along rays, we can treat the triplet of point values as lying in the same element.

Both statements can be used to simplify the implementation somewhat. The first remark allows us to safely ignore the interface for all nodes except those that actually lie on it. The second remark allows us to treat interface elements exactly the same as regular interior elements once the correct neighbour has been found.

4. Ray tracing and time scaling

We compare our active flux scheme to forward ray tracing, the industry standard as far as the authors are aware. In particular, we use forward ray tracing with a bin count to reconstruct the numerical solution. It is possible that more sophisticated ways of reconstructing the solution are employed in commercial software packages, however, such procedures would be kept confidential for obvious reasons.

To obtain a good comparison, the bin count is performed on the same elements as used by the active flux scheme. With the Monte Carlo process, the average value of each element is approximated by the empirical average, i.e.,

$$\bar{\rho}_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} \rho \, dy \approx \frac{1}{N_i} \sum_{k=1}^{N_i} \rho_{ik},$$

(42)

where $\bar{\rho}_i$ is the average value over element $\Omega_i$, $N_i$ is the number of rays arriving in the element and $\rho_{ik}$ are the values defined by (4) and the ray origins. The initial position of each ray is drawn from a uniform distribution over phase space. The estimate for the average on each bin $i$ converges with a standard deviation of $O\left(\frac{1}{\sqrt{N_i}}\right)$, see for instance [28]. Although the standard deviation is a statistical error, we use it as a straight-forward error estimate. The error behaviour will therefore be given by

$$\frac{1}{N_i} \sum_{k=1}^{N_i} \rho_{ik} = \bar{\rho}_i + O\left(\frac{1}{\sqrt{N_i}}\right),$$

(43)

as $N_i \to \infty$. As the total number of rays, $N$, is uniformly distributed across the initial phase space, we can compute the expected number of rays in each element by $N \frac{|\Omega_i|}{\sum |\Omega_k|}$. Thus, a uniform distribution on the initial phase space
translates to a uniform distribution, although deformed, in the target phase space.

From this observation, we can see that in order to have a constant local accuracy, we need the elements to have roughly the same size. This precludes the ability to locally adapt the element size, also known as $h$-adaptivity. The active flux scheme, on the other hand, does allow for $h$-adaptivity. If all elements have the roughly same size, the error scales as

$$e_{RT} = \mathcal{O}\left(\sqrt{\frac{E}{N}}\right) + \mathcal{O}\left(\frac{1}{\sqrt{E}}\right). \quad (44)$$

The error is minimal when both terms are equal, from which we find that the optimal number of rays scale as $N \sim E^2$. Using this scaling, the global error will scale as

$$e_{RT} = \mathcal{O}\left(\frac{1}{\sqrt{E}}\right). \quad (45)$$

Furthermore, for each ray, the appropriate element has to be found to evaluate the Monte Carlo average. This takes another $\mathcal{O}(E)$ time as a search over all elements has to be performed. Note that this may be shortened to $\mathcal{O}(\log E)$ provided one uses a binary-tree data structure to organise the elements. However, we did not pursue this strategy. The resulting time scaling for the Monte Carlo ray tracing algorithm therefore becomes

$$t_{RT} = \mathcal{O}\left(E^3\right). \quad (46)$$

The active flux scheme exhibits an altogether faster time scaling, as the connectivity between elements is exploited. The CFL implies that the number of $z$-steps scales as $\mathcal{O}(\sqrt{E})$. Due to the known connectivity, each element can be updated in constant time. Thus, the time scaling behaviour for the active flux scheme without optical interfaces is $\mathcal{O}\left(E^{3/2}\right)$. However, near interfaces a linear search over the elements is needed, leading to a search space of $\mathcal{O}(\sqrt{E})$. Therefore, the time scaling for optical problems for the active flux scheme will become

$$t_{AF} = \mathcal{O}\left(E^2\right). \quad (47)$$

Again, a binary-tree structure may be imposed allowing for binary searches, which can be performed in $\mathcal{O}(\log E)$ time. However, for simplicity of the algorithm and a fair comparison, we did not include this option. Finally, the active flux scheme is third-order accurate, meaning

$$e_{AF} = \mathcal{O}\left(E^{-3/2}\right). \quad (48)$$
Table 1: Summary of the time and error scaling of active flux and ray tracing methods.

<table>
<thead>
<tr>
<th></th>
<th>active flux</th>
<th>ray tracing</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>$O(E^2)$</td>
<td>$O(E^3)$</td>
</tr>
<tr>
<td>error</td>
<td>$O(E^{-3/2})$</td>
<td>$O(E^{-1/2})$</td>
</tr>
</tbody>
</table>

To conclude, based on scaling arguments, the active flux scheme is both faster and more accurate than forward ray tracing with bin count, the reference method. The time and error scaling for both methods are summarised in Table 1.

5. Results

The simplest nontrivial optics problem is a single flat interface between two media of constant refractive index. This problem was previously dubbed the “bucket of water” problem, as the problem roughly represents a beam of light shone at a water surface. The refractive index is given by

$$n(q) = \begin{cases} 
n_1 & \text{if } q < 0, \\
n_2 & \text{if } q \geq 0,
\end{cases} \quad (49)$$

where we choose $n_1 = 1.4$ and $n_2 = 1$. Using an initial condition that has support with $q < 0$ and $p > 0$, the solution features both refraction and total internal reflection. An exact solution was derived in [5], making this an ideal test problem. A typical result is shown in Figure 4, which was generated with an initial condition

$$\rho_0(q,p) = \begin{cases} 
1 & \text{if } -\frac{1}{5} < q < -\frac{1}{2} \text{ and } 0 < p < \frac{13}{10}, \\
0 & \text{otherwise}.
\end{cases} \quad (50)$$

The figure shows that the effects of transmission and reflection can clearly be seen. The active flux scheme does produce negative values for $\rho$, as some spurious oscillations occur around sharp gradients. Notice, however, that the transition is sharp around the optical interface, which is to be expected as interface nodes have two values, one for each one-sided limit.

5.1. Convergence and computation time

To verify the assertions made earlier about the computation times and convergence rates of both active flux and ray tracing, we perform some numerical experiments. For the convergence test, we need to use an initial
Figure 4: Numerical solution to the bucket of water problem using 15282 elements and 829 time steps.
condition that results in a solution that is piecewise smooth to the right and left of the interface. To do so, we use the bump function

$$\varphi(x) := \begin{cases} e^{-\frac{1}{1-x^2}}, & \text{for } |x| < 1, \\ 0, & \text{otherwise}, \end{cases}$$

(51)

which is a $C_0^\infty$ function. The smooth initial condition is defined as

$$\rho_0(q,p) = \varphi\left(\frac{q - q_0}{\lambda_q}\right) \left[\varphi\left(\frac{p - p_0}{\lambda_p}\right) + \varphi\left(\frac{p - p_1}{\sigma_p}\right)\right],$$

(52)

with the constants defined as $q_0 = -\frac{1}{2}$, $\lambda_q = \frac{1}{4}$, $p_0 = \frac{9}{20}$, $\lambda_p = \frac{9}{20}$, $p_1 = \frac{23}{20}$ and $\sigma_p = \frac{3}{20}$. This initial condition is chosen such that the light that is reflected is completely separate from the light that is transmitted, i.e., the critical momentum is avoided. As such, this ensures that the solution stays piecewise smooth, where the only discontinuity in the solution occurs at the interface.

For the number of rays, we used $N_{RT} = \frac{1}{25} E^2$ with $E$ the number of elements. As outlined earlier, this is the optimal scaling for the number of rays compared to the number of elements. The constant $\frac{1}{25}$ is chosen to give reasonable results. The error is computed using the $L^2$-norm on phase space, hence

$$e = \left( \int_{\mathbb{R}^2} (\rho_{AF}(Z,q,p) - \rho(Z,q,p))^2 \, dy \right)^{\frac{1}{2}},$$

(53)

with $\rho_{AF}$ the numerical solution and $\rho$ the exact solution. The integral is numerically approximated with sufficient accuracy. The exact solution is known for the bucket of water problem [5]. Furthermore, the error scaling as a function of the number of elements is assumed to satisfy

$$e = C_e E^{-\frac{\gamma}{2}},$$

(54)

with $\gamma$ the order of convergence and $C_e > 0$ some constant. Both algorithms, active flux and ray tracing, were implemented in Fortran and run on a single core of a Dell Optiplex 7010 for fair comparison.

Table 2 shows convergence data for the two methods. Note that the table does not show ray tracing data for the smallest element size, as the computation time grew prohibitively large. Both schemes hold up to the
Table 2: Errors and estimated orders versus the maximal element size for the convergence test. Active flux is indicated with subscript AF while ray tracing is indicated with subscript RT.

<table>
<thead>
<tr>
<th>max_i</th>
<th>Ω_i</th>
<th>ε_AF</th>
<th>γ_AF</th>
<th>N_RT</th>
<th>ε_RT</th>
<th>γ_RT</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 · 10^{-2}</td>
<td>2.1 · 10^{-2}</td>
<td>795</td>
<td>3.5 · 10^{-2}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3 · 10^{-2}</td>
<td>9.1 · 10^{-3}</td>
<td>1.2</td>
<td>14,933</td>
<td>1.6 · 10^{-2}</td>
<td>1.2</td>
<td></td>
</tr>
<tr>
<td>3.2 · 10^{-3}</td>
<td>2.9 · 10^{-3}</td>
<td>1.7</td>
<td>235,225</td>
<td>8.0 · 10^{-3}</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>8.0 · 10^{-4}</td>
<td>5.5 · 10^{-4}</td>
<td>2.4</td>
<td>3,785,359</td>
<td>3.9 · 10^{-3}</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>2.0 · 10^{-4}</td>
<td>8.4 · 10^{-5}</td>
<td>2.7</td>
<td>31,330,826</td>
<td>1.9 · 10^{-3}</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>5.0 · 10^{-5}</td>
<td>1.2 · 10^{-5}</td>
<td>2.8</td>
<td>980,917,344</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Theoretical arguments, the predicted error scaling is verified by the data. The active flux scheme is indeed third-order accurate while the ray tracing method provides first-order accuracy. It is interesting to note that the lowest error achieved by ray tracing is roughly comparable to the error from the active flux scheme with \( \max_i |\Omega_i| = 3.2 \cdot 10^{-3} \).

The computation times of the previous test are shown in Table 3. Like the error scaling, the time scaling is determined empirically using a power law assumption, i.e.,

\[
    t = C_t E^s, \tag{55}
\]

with some constant \( C_t > 0 \), which is valid as the number of elements grows large. The time scaling for ray tracing indeed comes out as predicted in Section 4, i.e., cubic in the number of elements. The time scaling of the active flux scheme is close to the predicted quadratic. For ray tracing on the other hand, the limit is indeed reached.

Now that the theoretical time scaling is confirmed for both methods, we can estimate the computation time for the last entry of the table for ray tracing: it would take over 21 days using roughly 980 million rays. We can now compare also the error-versus-time behaviour. For an error around \( 3 \cdot 10^{-3} \), ray tracing takes almost eight hours, while the active flux scheme is done in 4.6 seconds. It seems an understatement to say that the speed-up is significant.

5.2. CPC

The compound parabolic concentrator (CPC) provides a rather difficult test case for curved optics, see Figure 5. The CPC is an ideal concentrator and it features rays that are reflected any number of times. That is, for
Table 3: Number of elements $E$, computation times $t$ and their scaling $s$ of the previous test. Active flux is indicated with subscript AF while ray tracing is indicated with subscript RT.

<table>
<thead>
<tr>
<th>$E$</th>
<th>$t_{AF}$</th>
<th>$s_{AF}$</th>
<th>$t_{RT}$</th>
<th>$s_{RT}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>141</td>
<td>0.6 s</td>
<td></td>
<td>0.051 s</td>
<td></td>
</tr>
<tr>
<td>611</td>
<td>1.4 s</td>
<td>0.58</td>
<td>0.12 s</td>
<td>0.54</td>
</tr>
<tr>
<td>2,425</td>
<td>4.6 s</td>
<td>0.86</td>
<td>6.8 s</td>
<td>2.91</td>
</tr>
<tr>
<td>9,728</td>
<td>24.3 s</td>
<td>1.20</td>
<td>7 min 0 s</td>
<td>2.97</td>
</tr>
<tr>
<td>39,157</td>
<td>4 min 19 s</td>
<td>1.70</td>
<td>7 h 37 min 45 s</td>
<td>3.00</td>
</tr>
<tr>
<td>156,598</td>
<td>55 min 54 s</td>
<td>1.85</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

any integer $r$, there are rays that are reflected $r$ times. There are even so-called whispering modes that move continuously along one wall of the optic, reflecting an infinite number of times. These features make it a challenging test case for both ray tracing and the active flux scheme.

Figure 5: Sketch of the CPC.

The CPC is an example of an ideal concentrator, i.e., it takes all light that falls into the aperture under an acceptance angle $\theta$ and rejects anything else. It concentrates light into a the smallest possible space, an aperture of width $2a$. The length of the optic is given by

$$Z = a \frac{(1 + \sin \theta) \cos \theta}{\sin^2 \theta}. \quad (56)$$

Since the concentrator is ideal in space, it must be an ideal diluter in angular space, which consequently means the largest possible angular width, so that
Figure 6: Numerical solution to the CPC at various $z$ levels.

$-1 \leq p \leq 1$. The advection speed at $p = \pm 1$ is infinite, hence, we have to set some maximum momentum $p_{\text{max}} = 1 - \varepsilon$ for some $\varepsilon > 0$. We choose $\varepsilon = \frac{1}{2}l_{\text{min}}$, where $l_{\text{min}}$ is the minimum edge length in the initial grid. Furthermore, it is computationally easier to use the CPC in the reverse direction, as an ideal diluter. The initial conditions therefore consist of

$$\rho_0(q,p) = \begin{cases} 1 & \text{if } -a \leq q \leq a \text{ and } -p_{\text{max}} \leq p \leq p_{\text{max}}, \\ 0 & \text{otherwise.} \end{cases} \quad (57)$$

We use the parameters $a = \frac{1}{2}$ and $\theta = \frac{\pi}{6}$ for the CPC. We fix the left and right sides of the grid to the position of the mirrors, while the rest of the grid is uniformly stretched. We set a maximum element size of $10^{-2}$, resulting in 3345 triangles and 10201 total degrees of freedom. The CFL condition results in 2765 time steps. The outcomes are shown at several $z$ levels. In Figure 6, the solution is shown at $z = \frac{1}{3}Z$ and at $z = \frac{2}{3}Z$. In Figure 7 the final solution at $z = Z$ is shown.

The analytical solution at $z = Z$ is simply a rectangular area, $-1 \leq q \leq 1$ and $-\frac{1}{2} \leq p \leq \frac{1}{2}$, where $\rho = 1$ and otherwise $\rho = 0$. Due to the maximum momentum, holes have appeared in our solution. This is a restriction due to the fact that we must have a maximum velocity in our scheme. Other than that, the solution is rather good. Due to the constant-state preservation of the scheme, the error in the inner region is within machine precision.
6. Conclusions

We have demonstrated that it is possible to construct an active flux scheme for Liouville’s equation of geometric optics. Active flux schemes are highly suitable to geometric optics problems on phase space for several reasons: first, they are defined on elements rather than a uniform grid; second, they apply local ray tracing to find the boundary values. The fact that the active flux scheme is defined on elements allows us to align the computational mesh with any optical interface present. Local ray tracing means an easy implementation of optical laws, i.e., Snell’s law and the law of specular reflection. It also means that symplectic integrators can be used in the ray tracer.

It was deemed necessary, at least in the illumination optics setting, to consider moving boundaries. This is due to the fact that curved optical interfaces manifest themselves as moving boundaries in phase space. The principle of mesh alignment dictates that the mesh moves along with the boundary. To this end we introduced a novel way of implementing active flux schemes on moving meshes, one where the “time” coordinate is left continuous. This as opposed to earlier methods that discretise space-time as a whole. Our approach falls in with the method of lines paradigm and allows for a great range of numerical integrators to be used.

Leaving time continuous leads to an additional equation for the Jaco-
bian determinant, sometimes known as the geometric conservation law. For a method to be free-stream preserving, both the original PDE and the geometric conservation law must hold simultaneously on a discrete level. We proved that for general linear methods, this simply entails using the same numerical integrator for both semi-discrete equations. This class of numerical integrators includes all Runge-Kutta and linear multistep methods, and any predictor-corrector methods composed thereof.

In comparison to forward ray tracing with bin counting, the current industry standard, the active flux scheme is both faster and more accurate, which was verified by numerical experiments. In particular, we showed that ray tracing leads to an $O(E^3)$ run time, with $E$ the number of elements, while active flux runs in only $O(E^2)$ for illumination problems. In terms of accuracy, active flux achieves third-order while ray tracing leads to a first-order method. In one example, we observed a run time of 4.6 seconds on active flux, while ray tracing would take over 21 days to achieve the same global error.

At the moment, only two-dimensional optics was considered. An obvious next step would be to extend the method to a three-dimensional optics setting. Similar scaling arguments as presented in this work suggest that the gains will not be as great in the three-dimensional setting, yet still significant: $O(E^2)$ for active flux versus $O(E^{5/2})$ for ray tracing [29]. This difference will, of course, be more pronounced for larger computations that require high accuracy.

Acknowledgements

We’d like to thank Phil Roe for his help and suggestions.

References


## PREVIOUS PUBLICATIONS IN THIS SERIES:

<table>
<thead>
<tr>
<th>Number</th>
<th>Author(s)</th>
<th>Title</th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>18-02</td>
<td>T.G.J. Beelen</td>
<td>Determining the essentially different partitions of all Chinese convex tangrams</td>
<td>March ‘18</td>
</tr>
<tr>
<td>18-03</td>
<td>N.K. Yadav, J.H.M. ten Thije, Boonkkamp, W.L. IJzerman</td>
<td>Computation of double freeform optical surfaces using a Monge-Ampère solver: application to beam shaping</td>
<td>August ‘18</td>
</tr>
<tr>
<td>18-04</td>
<td>J. de Graaf</td>
<td>Some Mathematical Comments on Thermostatics</td>
<td>August ‘18</td>
</tr>
<tr>
<td>18-05</td>
<td>B.S. van Lith, J.H.M. ten Thije, Boonkkamp, W.L. IJzerman</td>
<td>Active flux schemes on moving meshes with applications to geometric optics</td>
<td>September ‘18</td>
</tr>
</tbody>
</table>