Active flux schemes on moving meshes with applications to geometric optics

Citation for published version (APA):

Document license:
CC BY-NC-ND

DOI:
10.1016/j.jcpx.2019.100030

Document status and date:
Published: 04/04/2019

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 16. Sep. 2019
Active flux schemes on moving meshes with applications to geometric optics

Bart S. van Lith a, *, Jan H.M. ten Thije Boonkkamp a, Wilbert L. IJzerman a, b

a Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, the Netherlands
b Signify - High Tech Campus 7, 5656 AE, Eindhoven, the Netherlands

1. Introduction

Illumination optics is a branch of nonimaging optics that is concerned with lighting applications [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 space distributions presents a possibility of such a new approach, as has been noted by Rausch and Herkommer [3,4].

Phase space, the collection of all ray positions and angles, provides a complete description of geometric optics. Knowing how the initial phase space is transformed therefore provides a complete knowledge of an optical system [5]. Point-wise information of the transformation can be found by ray tracing, the current industry standard. Rays propagate in straight lines when the medium has a constant refractive index. Snell's law and the law of specular reflection govern the behaviour of a ray near optical interfaces. Each ray carries a certain amount of energy, which is constant in the absence of attenuation. It is then possible to define energy or power distributions on phase space, and their movement is governed by Liouville's equation.

In a previous paper we derived a first-order upwind finite difference scheme for Liouville’s equation [6]. Although valid and useful, higher-order methods tend to be more efficient with computational resources. A method is needed that is high-order accurate, conservative and has a compact stencil. High-order finite difference methods have large stencils, leading to complications when optical interfaces are present. Active flux schemes, and the closely related multi-moment finite volume schemes, satisfy all these criteria. Both schemes will typically use point values and volume-averaged values, see for...
instance Xiao et al. [7–9], whereas multi-moment schemes may also use other moments, such as surface-averaged values, e.g., see [10]. They have been successfully applied to problems such as the shallow water equations [11], advection-diffusion equations [12], the linearised Euler equations [13], the full Euler equations [14], and to the incompressible Navier-Stokes equations [15]. The active flux scheme can be interpreted as incorporating a semi-Lagrangian method into a finite volume scheme to compute the fluxes.

The essential property of active flux schemes is that fluxes are computed more or less independently from the average values. This is facilitated by separately keeping track of the point values on element boundaries, which is achieved by means of a semi-Lagrangian step. The point values consequently completely determine the fluxes. Applied to geometric optics, this means the point values are found by local ray tracing [16].

Our contribution to active flux schemes is to formulate them as a semi-discrete method on a moving mesh. This to accommodate curved optical interfaces, which manifest themselves as moving boundaries in phase space. Ding et al. have developed a fully discrete active-flux scheme [17], which would also be able to deal with curved interfaces by means of curved space-time meshes. However, three-dimensional geometric optics problems results in a four-dimensional space, which gives rise to a five-dimensional space after inclusion of the evolution coordinate. The approach of Ding et al. would then require a five-dimensional curved mesh to solve three-dimensional optics problems. Simply having one less dimension to mesh seems like an immediate advantage, particularly when computer memory is an issue. Although we restrict ourselves to two-dimensional optics problems in this work, our desire to solve three-dimensional optics problems is our main motivation for looking at semi-discrete active flux schemes on moving meshes.

We therefore 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. This can, in some applications, be an advantage. We believe that illumination optics is precisely such an application for reasons mentioned above.

The contents of this paper are organised as follows: in Section 2 we discuss Liouville’s equation and some basic concepts from illumination optics, in Section 3 we discuss active flux schemes and our novel extensions to moving meshes, in Section 4 we briefly compare our scheme to ray tracing in terms of theoretical performance, in Section 5 we present some numerical experiments where we focus particularly on the comparison with ray tracing, finally in Section 6 we present our conclusions.

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 [5]. 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 $\mathbb{R}^3$, $\vec{p} = (p, p_z)$, is restricted to Descartes’ sphere, $|\vec{p}| = n(z, q)$ with $n$ the refractive index field [5]. At a given position $z$, one can visualise the phase space coordinates as a light ray intercepting 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

$$\frac{dq}{dz} = \frac{\partial h}{\partial p},$$
$$\frac{dp}{dz} = -\frac{\partial h}{\partial q}.$$  \hfill (1a)

\hfill (1b)

where $h$ is the Hamiltonian. In the illumination setting, we use

$$h(z, q, p) = -\sigma \sqrt{n(z, q)^2 - \|p\|^2}.$$  \hfill (2)

Physical solutions require that $|p| \leq n$, a region in momentum-space known as Descartes’ disc. Here, $\sigma \in \{-1, 0, 1\}$ indicates the 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

$$S = \left(\begin{array}{cc} 0 & I \\ -I & 0 \end{array}\right), \quad \nabla = \left(\begin{array}{c} \frac{\partial q}{\partial y} \\ \frac{\partial p}{\partial y} \end{array}\right).$$  \hfill (3)

---

\footnote{1 When first submitting this article, the authors believed this approach to be wholly original. However, it seems that Jin, Deng and Xiao used a similar formulation of their multi-moment scheme in [18]. An anonymous reviewer informed us of this work.}
where $\mathbf{O}$ is a matrix of zeros and $\mathbf{I}$ is the identity matrix, both in $\mathbb{R}^{d \times d}$. Hamilton’s equations (1) can then be more compactly written as

\[
\frac{d\mathbf{y}}{dz} = \mathbf{S}\nabla h.
\]

(4)

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 [19]. As the problem is formulated independently of time, the light source is stationary and we may therefore also interpret $\rho$ as power. It is customary in illumination optics to scale the power per unit area per solid angle to human eye sensitivity, after which it is called the luminance or brightness. Moreover, the flow generated by Hamilton’s equations (4) preserves volume [20]. Since power is transported along rays, in the absence of attenuation or diffusion we have that $\rho \frac{d\mathbf{y}}{dz}$ is constant, where $d\mathbf{y} = dq_1 dq_2 dp_1 dp_2$ is the volume element associated to a ray. Therefore, we find that

\[
\rho(\mathbf{z}, \mathbf{q}(\mathbf{z}), \mathbf{p}(\mathbf{z})) = \rho_0(\mathbf{q}(0), \mathbf{p}(0)),
\]

(5)

where $(\mathbf{q}(\mathbf{z}), \mathbf{p}(\mathbf{z}))$ is a solution to (1). Here, $\rho_0$ is the initial distribution, the distribution at $z = 0$. Note that (5) provides the physically relevant solution even when rays are refracted or reflected. Supposing sufficient smoothness, however, we may differentiate (5) with respect to $z$ and use (1) to obtain Liouville’s equation

\[
\frac{\partial \rho}{\partial z} + \frac{\partial h}{\partial \mathbf{p}} \cdot \frac{\partial \rho}{\partial \mathbf{q}} - \frac{\partial h}{\partial \mathbf{q}} \cdot \frac{\partial \rho}{\partial \mathbf{p}} = 0.
\]

(6)

This equation determines the evolution of the luminance in phase space, when moving the screen along the optical axis $z$. Examples of this will be given in Section 5. Using the shorthand introduced earlier, (3) allows us to write Liouville's equation succinctly as

\[
\frac{\partial \rho}{\partial z} + (\mathbf{S}\nabla h) \cdot \nabla \rho = 0.
\]

(7)

Assuming that $h$ is twice differentiable, we may rewrite the advection part of Liouville’s equation as

\[
(\mathbf{S}\nabla h) \cdot \nabla \rho = \nabla \cdot (\rho \mathbf{S}\nabla h) - \rho \nabla \cdot (\mathbf{S}\nabla h).
\]

(8)

The divergence of the velocity field $\mathbf{u} = \mathbf{S}\nabla h$ is identically zero for sufficiently smooth Hamiltonians, which is more easily seen when written out component-wise, i.e.,

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

(9)

Liouville’s equation may therefore be expressed in the conservative form

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

(10)

i.e., the continuity equation. This is the conservative form of Liouville’s equation, which is equivalent to what we may call the advective form (6) 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 (5) together with Snell’s law and the law of specular reflection.

3. Active flux on a moving mesh

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

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

(11)

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

Active flux schemes operate by exploiting the notion that the boundary values completely determine the flux in (11). 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-Lagrangian step [21], otherwise known as ray tracing in an optics context. Like this, the right-hand side of (11) 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 (11) on a moving mesh, as curved optics manifest themselves as moving boundaries in phase space. We consider in particular boundaries that are given by \( q = Q(z) \), e.g., a curved set of mirrors in Section 5.3. Recall that \( z \) acts as a time coordinate, even though it is in fact a spatial coordinate. As such, curved boundaries in real space manifest themselves as moving boundaries in phase space. Moreover, such moving boundaries will always be perfect straight lines perpendicular to the \( q \)-axis at any one fixed \( z \).

Along the optical interfaces, the laws of optics determine what happens to \( \rho \). One very inefficient way of dealing with a moving boundary is to compute a completely new mesh at every time step and interpolate from the old to the new mesh. Assuming such a complete remeshing can be avoided, the mesh has to be deformed slightly to stay aligned to the interface. If the deformation is sufficiently small and smooth, it can be interpreted as a moving mesh.

The mesh may move in any number of ways, e.g., the nodes may be subject to some force law or the positions may be given exactly. Whenever the node positions and velocities are not exactly known, they will need to be found with sufficient accuracy by means of numerical integration or differentiation. Here, we assume for simplicity that we know the node positions exactly and the velocity can be found exactly by differentiation. This in fact covers the majority of the cases occurring in geometric optics. Moving the mesh is needed only to keep the mesh aligned with an optical interface, as that is where Snell’s law and the law of specular reflection come into play. The shape of the optical interfaces, in practice, is either given by piecewise smooth functions or piecewise spline polynomials. We will see an example of this in Section 5.3.

Consider Reynolds’ transport theorem on the, now moving, test domain \( \Omega(z) \), i.e.,

\[
\frac{d}{dz} \int_{\Omega(z)} \rho \, d\mathbf{y} = \int_{\Omega(z)} \frac{\partial \rho}{\partial z} \, d\mathbf{y} + \nabla \cdot (\rho \mathbf{v}) \, d\mathbf{y},
\]

(12)

where \( \mathbf{v} \) is the velocity field of the moving element. Let us fix the shape of \( \Omega \) 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 \( \chi \), whose vertices are given by \((0,0)\), \((1,0)\) and \((0,1)\), see Fig. 1. The coordinates on the reference triangle are denoted \((\xi, \eta)\). The affine transformation from the reference triangle to \( \Omega \) is given by

\[
\begin{pmatrix} q(z) \\ p(z) \end{pmatrix} = \mathbf{A}(z) \begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} q_1(z) \\ p_1(z) \end{pmatrix},
\]

(13)

where \( \mathbf{A} \) is given by

\[
\mathbf{A}(z) = \begin{pmatrix} q_2(z) - q_1(z) & q_3(z) - q_1(z) \\ p_2(z) - p_1(z) & p_3(z) - p_1(z) \end{pmatrix},
\]

(14)

Let us denote the determinant of \( \mathbf{A} \) as \( \mathcal{J} \), then \(|\mathcal{J}(z)| = |\Omega(z)|/|\chi| = 2|\Omega(z)|\). 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. In practice, the smallest angle in the mesh is often a measure of quality, larger being better. We could avoid collinearity by monitoring the minimum angle and remesh whenever it falls below a specified threshold. For typical optical systems, however, this safety mechanism is not needed.

The derivatives in the reference coordinates may be found by considering

\[
\frac{\partial}{\partial y_1} = \sum_{j=1}^{2} \frac{\partial \xi_j}{\partial y_1} \frac{\partial}{\partial \xi_j},
\]

(15)

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 \( \frac{\partial \xi_j}{\partial y} = \mathbf{A}^{-1} \), hence
\[ \nabla = \nabla_y = A^{-T}(z)\nabla_{\xi}, \]  
(16)

where \( \nabla_{\xi} = (\partial_{\xi}, \partial_{\eta})^T \) is simply the nabla operator on the standard domain. The velocity field due to the movement of the vertices can be found by differentiating (13), i.e.,

\[ \mathbf{v}(z) = A'(z) \left( \frac{\xi}{\eta} \right) + y'_1(z), \]  
(17)

where the prime denotes differentiation with respect to \( z \).

Next, we apply Liouville’s equation (10) to (12), so that we find

\[ \frac{d}{dz} \int_{\Omega(z)} \rho \, dy = \int_{\Omega(z)} \nabla_y \cdot (\rho(\mathbf{v} - \mathbf{u})) \, dy. \]  
(18)

The volume integral of \( \rho \) is by definition equal to the average value times the volume size. Furthermore, transforming the integral in the right-hand side of (18) to the standard domain yields

\[ \frac{d}{dz} (\tilde{\rho} |\Omega|) = \mathcal{J} \int_{\chi} (A^{-T} \nabla_{\xi}) \cdot (\rho(\mathbf{v} - \mathbf{u})) \, d\xi. \]  
(19)

Since \( A \) is constant with respect to the spatial coordinates, the velocity field on the standard triangle \( \chi \) can be defined as

\[ \hat{\mathbf{u}} = A^{-1} (\mathbf{v} - \mathbf{u}). \]  
(20)

Applying the divergence theorem on the standard triangle, we obtain

\[ \frac{d}{dz} \left( \frac{1}{2} \mathcal{J} \right) = -\mathcal{J} \int_{\partial\chi} \rho (\hat{\mathbf{u}} \cdot \mathbf{n}) \, d\sigma, \]  
(21)

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

\[ \frac{d\mathcal{J}}{dt} = 2\mathcal{J} \int_{\partial\chi} \hat{\mathbf{v}} \cdot \mathbf{n} \, d\sigma, \]  
(22)

where \( \hat{\mathbf{v}} = A^{-1} \mathbf{v} \). This ODE for \( \mathcal{J} \) in an element is sometimes referred to as the geometric conservation law [22,23]. Alternatively, one can set \( \rho = 1 \) in (21) to derive (22). If the numerical solution is to be free-stream preserving, we need (22) and (21) 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 (21). 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.,

\[ f_k(\rho) = 2 \int_{\ell_k} \rho \hat{\mathbf{u}} \cdot \mathbf{n}_k \, d\sigma, \]  
(23)

where \( \ell_k \), \( k = 1, 2, 3 \) are the edges of the standard triangle and \( \mathbf{n}_k \) 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 Fig. 2.

Simpson’s rule is the three-point case of the Gauß-Lobatto rules [24]. 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, \( (p + 1)/2 \) nodes 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 $s(p + 1)/2$.

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 Fig. 2 and define the unit direction vectors $\mathbf{e}_1 = (1, 0)^T$ and $\mathbf{e}_2 = (0, 1)^T$, so that we may approximate the flux over each edge as follows,

$$
\begin{align}
 f_1(\rho) &\approx -\frac{1}{2} (\rho_1 \mathbf{u}_1 + 4\rho_2 \mathbf{u}_2 + \rho_3 \mathbf{u}_3) \cdot \mathbf{e}_2, \\
 f_2(\rho) &\approx \frac{1}{2} (\rho_3 \mathbf{u}_3 + 4\rho_4 \mathbf{u}_4 + \rho_5 \mathbf{u}_5) \cdot (\mathbf{e}_1 + \mathbf{e}_2), \\
 f_3(\rho) &\approx -\frac{1}{2} (\rho_5 \mathbf{u}_5 + 4\rho_6 \mathbf{u}_6 + \rho_3 \mathbf{u}_3) \cdot \mathbf{e}_1,
\end{align}
$$

where $\rho = (\rho_1, \rho_2, \rho_3, \rho_4, \rho_5, \rho_6)^T$ is the vector of point values, while $\mathbf{u}_i$ is the transformed velocity field at the same position as $\rho_i$. Note that the normalisation of $\mathbf{n}_2$ cancels against the length of the hypotenuse. By definition, the total averaged flux over the boundary is given by

$$
F(\rho) = \sum_{k=1}^{3} f_k(\rho). \tag{25}
$$

The boundary integral occurring in (22) 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

$$
\begin{align}
 \frac{d}{dz}(\rho \mathcal{J}) &= -\mathcal{J}(z) F(z), \tag{26a} \\
 \frac{d\mathcal{J}}{dz} &= \mathcal{J}(z) G(z), \tag{26b}
\end{align}
$$

where $F(z) = F(\rho(z))$ approximates the total volume-averaged flux across the boundary.

3.2. $z$-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 $Z$. We define $\rho^t$ to be the approximation for $\rho(z^t)$, etc. In order to find an approximation to the average values at the next $z$-level, a definite integral must be evaluated. Note that if $\rho$ is a constant over the control volume $\Omega$, then also $F(z) = -\rho G(z)$. 

Fig. 2. Gauß-Lobatto points around the standard triangle $\chi$ with values labelled that should be tracked. The edges are also indicated.
Similar to the results of Acosta Minoli et al. for spectral element methods on moving meshes [22], we find that (26a) and (26b) 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 (26b) rather than the exact value of the Jacobian, even if it is available.

**Theorem 1.** Consider the system (26), where we define \( \tilde{\rho} = \tilde{\rho} \mathcal{J} \), so that

\[
\frac{d\tilde{\rho}}{dz} = -\mathcal{J}(z)F(z),
\]

\[
\frac{d\tilde{\mathcal{J}}}{dz} = \mathcal{J}(z)G(z).
\]

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

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

**Proof.** Let’s write one time step of the integrator as \( \psi \), i.e., \( \tilde{\rho}^{t+1} = \tilde{\rho}^t + \psi(-\mathcal{J}^t F^t) \) where \( \mathcal{J}^t \) is the approximation to \( \mathcal{J}(z^t) \) etc. Any general linear method satisfies \( \psi(c) = c \psi(x) \) for all \( c \in \mathbb{R} \) and linear right-hand side. Next, assume that \( \rho \) is constant, so that \( \tilde{\rho}^t = \rho \mathcal{J}^t \) holds at the current z-step. Furthermore, we have \( F(z) = -\rho G(z) \), so that the right-hand side in (27a) is indeed linear in \( \rho \). The next z-step is then given by

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

The average value is recovered as \( \tilde{\rho}^{t+1} = \frac{\tilde{\rho}^{t+1}}{\tilde{\mathcal{J}}^{t+1}} = \rho \), so that the constant state is preserved. \( \square \)

**Remark 1.** General linear methods are a broad class of numerical integrators [25,26]. 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, we can observe that if we were to use the exact value of the Jacobian \( \mathcal{J}(z^{t+1}) \), we would introduce a relative error in \( \tilde{\rho}^{t+1} \) of size \( \frac{\Delta z}{\mathcal{J}^{t+1}} \), i.e., the ratio of the numerical value of the Jacobian obtained from integrating the geometric conservation law [22] and the exact value of the Jacobian. However, for one-step methods, the exact value of the Jacobian can be used at the beginning of each z-step.

We elect to use the standard fourth-order RK4 method, as this reduces to Simpson’s rule whenever \( \mathcal{J} \) is constant. Note that the system (26) is a one-way coupled system, where \( \tilde{\rho} \) depends on \( \mathcal{J} \), but \( \mathcal{J} \) can be computed independently under the assumption that the mesh motion does not depend on \( \rho \). We denote the RK4 stages of (26b) as

\[
\mathcal{J}^{(1)} = \mathcal{J}(z^t),
\]

\[
\mathcal{J}^{(2)} = \mathcal{J}(z^t) + \frac{1}{2} \Delta z \mathcal{J}^{(1)} G^t,
\]

\[
\mathcal{J}^{(3)} = \mathcal{J}(z^t) + \frac{1}{2} \Delta z \mathcal{J}^{(2)} G^{t+\frac{1}{2}},
\]

\[
\mathcal{J}^{(4)} = \mathcal{J}(z^t) + \Delta z \mathcal{J}^{(3)} G^{t+\frac{1}{2}}.
\]

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

\[
\mathcal{J}^{t+1} = \mathcal{J}(z^t) + \frac{\Delta z}{6} \left[ \mathcal{J}^{(1)} G^t + 2 \left( \mathcal{J}^{(2)} + \mathcal{J}^{(3)} \right) G^{t+\frac{1}{2}} + \mathcal{J}^{(4)} G^{t+1} \right].
\]

\[
\tilde{\rho}^{t+1} = \frac{\mathcal{J}(z^t)}{\mathcal{J}^{t+1}} \tilde{\rho}^t - \frac{\Delta z}{6 \mathcal{J}^{t+1}} \left[ \mathcal{J}^{(1)} F^t + 2 \left( \mathcal{J}^{(2)} + \mathcal{J}^{(3)} \right) F^{t+\frac{1}{2}} + \mathcal{J}^{(4)} F^{t+1} \right].
\]

From (28)–(29), 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 \( \tilde{\rho}^{t+1} = \rho \), 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 [27–29], the active flux method keeps track of point values \( \mathbf{\rho} \) on the boundary separately.\(^2\)

The point values needed to complete the scheme are obtained using a semi-Lagrangian step [21], i.e., by integrating Hamilton’s equations (1) and applying one \( z \)-step of (5), which in the 2D case reduces to

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

This is otherwise known as the method of characteristics (MOC). The characteristics of Liouville’s equation (6) are simply light rays. In the context of optics the MOC is therefore better known as ray tracing [16,1]. Recall that \( \mathbf{u} = S \nabla h \), so that Hamilton’s equations can be written as

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

As the system is Hamiltonian in nature, it is prudent to solve it using a symplectic, or geometric, integrator [25,30]. 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 \( y' \) only has to be determined to second-order accuracy in \( z \), i.e., \( y' = y(z^t) + O(\Delta z^2) \), 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

\[
y' = y^{t+1} - \Delta \mathbf{u} \left( z^t + \frac{1}{2} \Delta z, \frac{y^{t+1} + y^{t+1}}{2} \right),
\]

which is an implicit equation for \( y' \), which 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 (29). To see this, we Taylor expand \( y(z) \) around \( z^{t+\frac{1}{2}} \) to find

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

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

Indeed, taking the average of the two expansions leads to

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

Once the point \( y' \) has been approximated, we perform a local search over the neighbouring elements. By construction the two points \( y' \) 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

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

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

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

\[
\text{and}
\]

\[^2\text{In fact, the name “active flux” originates from this distinction. Other finite volume schemes that do not actively keep track of boundary values may be classified as passive flux schemes.}\]
\[
\mathbf{r} = \begin{pmatrix}
-3\rho_1 + 4\rho_2 - \rho_3 \\
-3\rho_1 + 4\rho_6 - \rho_5
\end{pmatrix},
\]
where one can recognise undivided differences over the \(\xi\)- and \(\eta\)-directions, with the off-diagonal elements of \(\mathbf{R}\) 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
\[
2 \int_{\chi} \bar{\rho}(\xi, \eta) \, d\xi \, d\eta = \frac{1}{2} (\rho_2 + \rho_4 + \rho_6),
\]
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. \(\int_{\chi} \varphi \, d\xi \, d\eta = |\chi| = \frac{1}{2}\). As a consequence, the complete local reconstruction \(\rho_{\text{int}}\) on the standard domain is given by
\[
\rho_{\text{int}}(\xi, \eta) = \bar{\rho}(\xi, \eta) + \left(\bar{\rho} - \frac{1}{2}(\rho_2 + \rho_4 + \rho_6)\right) \varphi(\xi, \eta).
\]

3.5 Summary of the algorithm

The algorithm resulting from the synthesis of the discussion above 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 \(z^0 = 0\). A single time step from \(z^t\) to \(z^{t+1}\) is done as follows:

1. For each node in the mesh, perform the following actions:
   (a) Compute the initial position \(\mathbf{y}'\) of the ray coming through the node at \(z^{t+1}\) by (32) and the ray midpoint \(\mathbf{y}^i + \frac{1}{2}\) by (34).
   (b) Perform a local search over the neighbouring elements to find from which element the characteristics originate.
   (c) Transform \(\mathbf{y}'\) and \(\mathbf{y}^i + \frac{1}{2}\) to the standard coordinates, i.e., the inverse transformation to (13).
   (d) Find and store the point values using the method of characteristics (30) and the reconstruction, i.e., (35), (37) and (38).

2. For each element in the mesh, perform the following actions:
   (a) Compute the local transformed velocity field according to (20).
   (b) Compute the flux function at \(z^t\) and \(z^{t+\frac{1}{2}}\) using (24) and (25).
   (c) Update the Jacobians and average values using (28) and (29), respectively.

Naturally, this procedure is simply repeated until \(z^T = \mathcal{Z}\) is reached. 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 [31]. Finally, the algorithm is repeated \(M\) 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 [32,33], 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, midnode nodes always have a search set of two elements, while vertex nodes can have at most a search set of \(\lfloor \frac{\pi}{\alpha_{\text{min}}} \rfloor\), where \(\alpha_{\text{min}}\) 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 \( \Delta z \) must be chosen such that for all nodes, the bounding circle lies entirely within the search set, see Fig. 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 \( a \), we see 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 [34].

Most mesh generators have a minimum angle that can be set, here we used the Triangle software package created by Shewchuck [35], where a minimum angle of 32° usually works well. The minimal distance is estimated as

\[
\Delta y = \frac{1}{2} \min_i (a_i \sin \alpha_i) \geq \frac{1}{2} \sin \alpha_{\min} \min_i a_i, \tag{40}
\]

where \( a_i \) and \( \alpha_i \) 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 of spatial and temporal grid sizes is straightforward: \( \Delta z = O(\Delta y) \).

Earlier, we asserted that the scheme only needs to resolve the point values to second order accuracy in \( \Delta z \). The point values computed from the semi-Lagrange step carry a second-order z-error \( O(\Delta z^2) \), so that integrating the flux around the boundary results in a mixed error \( O(\Delta y \Delta z^2) \). Finally, the CFL condition implies that the total error in the fluxes behaves as \( O(\Delta y^3) \).

### 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. In phase space, this means that \( p \) changes discontinuously as the ray is affected by an optical interface. This translates to nonlocal boundary conditions for Liouville’s equation in the sense that one part of the boundary passes its information on to a completely different part of the boundary, see for instance the “bucket of water” problem in Section 5. Clearly, this behaviour must be incorporated into the solver to find physically relevant solutions.

In our previous work [6], we derived an explicit version of Snell’s law on phase space, relating the incoming momentum \( p \) and the outgoing momentum \( p' \). The law involves the refractive indices \( n_1 \) and \( n_2 \) of the media, together with the \( q \)-component of local surface normal \( v \). We denote Snell’s law as the statement that \( p' = S(p; n_1, n_2, v) \), where

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

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

with \( \psi \leq 0 \) due to angle conventions in Snell’s law. If \( n_2 < n_1 \), there are two cases to be specified, transmission and total internal reflection. Transmission corresponds to \( \delta \geq 0 \), so that the ray continues in the medium with index \( n_2 \). Total internal reflection occurs when \( \delta < 0 \) and the ray is reflected back into the medium with index \( n_1 \). The sign of \( n_2 \), i.e., \( \text{sgn}(n_2) \), in the first case of (41a) allows us to define a mirror embedded in a medium of index \( n_1 \) as a transition from \( n_1 \geq 1 \) to \( n_2 = -n_1 \). One can check that both cases then read exactly the same and the ray will be reflected back into the medium with positive index \( n_1 \).
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, $\hat{y}_{i}, \hat{y}_{i+\frac{1}{2}}^+,$ and $\hat{y}_{i+1}^-$, 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 on phase space 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. We stipulate that ray tracing is usually not performed to find phase space distributions, but rather to find the lower-dimensional distributions of intensity and irradiance. However, for our purposes we assume that it is in fact the space phase distribution that is required. This is not without precedent, as we have noticed in the introduction of this article that phase space methods are rapidly gaining popularity and stature [3,4]. The lower-dimensional distributions can be found by integrating the phase space distribution, e.g., the irradiance results from integrating the phase space distribution over all momenta. For a more extensive discussion on the topic, see [36].

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 (5) 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 $\mathcal{O}(N_i^{-1/2})$, see for instance [37]. 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

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

(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 $N \frac{|\Omega|}{\sum_{h \in H} |\Omega|}$. 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 roughly the same size, we have $N_i = \mathcal{O}\left(\frac{\sqrt{E}}{E}\right)$, with $E$ the number of elements. As a consequence, the error scales as

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

(44)
where the second term comes from the fact that we are using a constant approximation on each element. 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

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

(45)

Furthermore, for each ray, the appropriate element has to be found in order to evaluate the Monte Carlo average. This costs another $O(E)$ time per ray as a search over all elements has to be performed. Note that this may be shortened to $O(\log E)$ provided one uses a binary-tree data structure to organise the elements. However, we did not pursue this strategy. We assume that computing the final state of each ray takes constant time, so that the resulting time scaling for the Monte Carlo ray tracing algorithm becomes the number of rays multiplied by the time per ray, i.e.,

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

(46)

The active flux scheme exhibits an altogether faster time scaling, as the connectivity between elements is exploited. The CFL condition implies that the number of $z$-steps scales as $O(\sqrt{E})$. 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 $O\left(E^{3/2}\right)$. However, near interfaces a linear search over the elements is needed, leading to a search space of size $O(\sqrt{E})$. Therefore, the time scaling for optical problems for the active flux scheme will become

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

(47)

Again, a binary-tree structure may be imposed allowing for binary searches, which can be performed in $O(\log E)$ 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

$$e_{\text{AF}} = O\left(E^{-3/2}\right).$$

(48)

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.

Alternatively, one may formulate the time and error scaling in terms of the degrees of freedom, denoted $D$. For ray tracing, the number of degrees of freedom is simply the number of rays used. We argued above that the number of rays should scale as $E^2$, so that $D = O(\sqrt{E})$.

For the active flux scheme, the number of degrees of freedom can be found by invoking Euler’s characteristic and interpreting the mesh as a planar graph. We have $v - e + f = 2$, where $v$ are the vertices, $e$ the edges and $f$ the faces, where $f$ includes the infinitely large outer face. Every face, except the outer one, is associated with an average value, while every vertex is associated with a point value. Each edge is also associated with a point value, since the midpoint of each edge carries a point value. Thus, we find $D = v + e + f - 1 = 1 + 2e$. Next, we note that each interior element has three edges and one face, but each edge is also shared between two elements, so that $e \approx \frac{2}{3} f \approx \frac{3}{2} E$. This approximation becomes better and better as $E$ increases, since the number of interior elements increases faster than those on the boundary. Therefore, we find that $D \approx 3E$ for the active flux scheme. This allows us to translate Table 1 into Table 2.

In this way, it becomes clear that the active flux scheme actually uses slightly more time per degree of freedom, but yields a much higher convergence rate. To be sure, the cost outweighs the benefits. This can be seen more clearly by looking at how the error scales with the computation time. For ray tracing we find $e_{\text{RT}} = O\left(t^{-1/2}\right)$, while for the active flux scheme we find $e_{\text{AF}} = O\left(t^{-3/4}\right)$. These relations can be regarded as an efficiency measure, as they tell us how effective the methods are with spending computation time. For example, suppose we want to reduce the error by a factor of two, then these
relations will tell us that the running time of the active flux scheme will increase by a factor $2^{1/2} \approx 2.51$, while running time for ray tracing will increase by a factor of 64.

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

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

(49)

where we choose $n_1 = 1.4$ and $n_2 = 1$. Using an initial luminance that has support with $q < 0$ and $p > 0$, the solution features both refraction and total internal reflection. An exact solution was derived in [6], making this an ideal test problem. In the special case that the initial solution has compact support in the quadrant with $\{q < 0, p > 0\}$ it is given by

$$\rho(z, q, p) = \begin{cases} \rho_0 \left( q - \frac{z - p}{\sqrt{n_1^2 - p^2}}, p \right) & \text{if } q < 0, p \geq 0, \\ \rho_0 \left( \frac{z - p}{\sqrt{n_1^2 - p^2}}, -q, -p \right) & \text{if } q < 0, -p_c < p < 0, \\ \rho_0 \left( \Delta z - z, \frac{p'}{\sqrt{n_1^2 - p'^2}}, p' \right) & \text{if } q > 0, p \geq 0, \\ 0 & \text{otherwise}. \end{cases}$$

(50a)

where $p_c = \sqrt{n_1^2 - n_2^2}$, $p'$ is such that $S(p'; n_1, n_2, -1) = p$, with $S$ being Snell’s law from (41). Finally, $\Delta z$ is defined as

$$\Delta z = \frac{q}{p} \sqrt{n_1^2 - p^2}. $$

(50b)

The first case of (50a) is simple propagation without refraction or reflection. The second case is total internal reflection, as $n_1 > n_2$, and $p_c \approx 0.98$ is the critical momentum. The third case is refraction according Snell’s law. It’s important to note that the optical axis is chosen along the surface, so that higher momenta are refracted into lower momenta.

One thing that is particularly peculiar about geometric optics problems on phase space are the nonlocal boundary conditions, meaning that completely separate parts of phase space are in connection through Snell’s law or the law of specular reflection. For suitable initial conditions, the bucket of water problem showcases both. A typical result is shown in Fig. 4, which was generated with an initial luminance.
The 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.

| $\max |\Omega|$ | $D$ | $e_{MF}$ | $y_{MF}$ | $N_{RT}$ | $e_{RT}$ | $y_{RT}$ |
|------------|------|---------|----------|----------|----------|----------|
| $6.4 \cdot 10^{-3}$ | 3,694 | $1.9 \cdot 10^{-2}$ | 1.3 | 9,472 | $2.9 \cdot 10^{-2}$ | 0.6 |
| $3.2 \cdot 10^{-3}$ | 7,302 | $1.2 \cdot 10^{-2}$ | 2.0 | 37,383 | $2.3 \cdot 10^{-2}$ | 1.2 |
| $1.6 \cdot 10^{-3}$ | 14,400 | $6.2 \cdot 10^{-3}$ | 2.1 | 148,523 | $1.6 \cdot 10^{-2}$ | 1.0 |
| $8.0 \cdot 10^{-4}$ | 28,610 | $3.0 \cdot 10^{-3}$ | 2.2 | 590,070 | $1.1 \cdot 10^{-2}$ | 1.0 |
| $4.0 \cdot 10^{-4}$ | 56,782 | $1.4 \cdot 10^{-3}$ | 2.7 | 2,347,001 | $7.8 \cdot 10^{-3}$ | 1.0 |
| $2.0 \cdot 10^{-4}$ | 114,234 | $5.4 \cdot 10^{-4}$ | 3.1 | 9,534,170 | $5.5 \cdot 10^{-3}$ | 1.0 |
| $1.0 \cdot 10^{-4}$ | 227,754 | $2.2 \cdot 10^{-4}$ | 3.6 | 38,096,352 | $3.9 \cdot 10^{-3}$ | 1.0 |
| $5.0 \cdot 10^{-5}$ | 453,616 | $8.2 \cdot 10^{-5}$ | 2.9 | |

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 luminance that results in a solution that is piecewise smooth to the right and left of the interface. To do so, we use a piecewise polynomial with compact support given by

$$\varphi(x) := \begin{cases} (1 - x^2)^4, & \text{for } |x| < 1, \\ 0, & \text{otherwise}, \end{cases}$$

which is a $C^3$-function on $\mathbb{R}$. The smooth initial luminance is defined as

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

with the constants defined as $q_0 = -\frac{1}{2}$, $\lambda_q = \frac{1}{2}$, $p_0 = \frac{9}{20}$, $\lambda_p = \frac{9}{25}$, $p_1 = \frac{22}{25}$, 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. In both cases, we compute an approximation to the $L^1(\mathcal{P})$-norm of the difference with the exact solution by using a quadrature rule. For convenience, we use the fifth-order quadrature rule of Dunavant [38], since this has only internal nodes, an accuracy higher than third-order, and all positive weights. We require a quadrature rule involving internal nodes only, since the ray tracing solution is discontinuous on the element boundaries. The positive weights are chosen for stability. To compute the value of the active flux solution on the quadrature nodes, we apply the same reconstruction as in Section 3.4.

We assume that the error scaling as a function of the number of elements is assumed to satisfy

$$e = C_e E^{-\frac{2}{5}},$$

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 3 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 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 $\max |\Omega| = 8.0 \cdot 10^{-4}$.

The computation times of the previous test are shown in Table 4. Like the error scaling, the time scaling is determined empirically using a power law assumption, i.e.,

$$t = C t E^\gamma,$$
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 2 days using roughly 150 million rays. We can now compare also the error-versus-time behaviour. For an error around $4 \cdot 10^{-3}$, ray tracing takes more than six hours, while the active flux scheme is done in 3.4 seconds. It seems an understatement to say that the speed-up is significant.

We have compiled the error-versus-time behaviour in Fig. 5. The steepness of the curve measures, in a sense, the efficiency of the method. It shows how much bang we are getting for our buck. In the figure, we can clearly see that the active flux scheme is the method of choice. Furthermore, the theoretical predictions of Section 4 are confirmed.

### 5.2. Convergence on a moving mesh

After confirming the convergence of the standard active flux scheme for geometric optics problems, we also wish to confirm that a moving mesh does not interfere with convergence. Therefore, we repeat the previous convergence test, but now with a moving mesh. The corner nodes are displaced with a periodic motion of amplitude $\Delta_{\text{min}}$, where $\Delta_{\text{min}}$ is the shortest edge in the mesh. The odd-numbered nodes are moved horizontally (q-direction) and the even-numbered nodes are moved vertically (p-direction), see Fig. 6. All nodes are given a unique phase to avoid rigid motion of parts of the mesh. The frequency is chosen as $\Delta_{\text{min}} / \Delta_t$, so that the mesh motion is repeated three times.

We present the convergence data of the active flux scheme on the moving mesh in Table 5. Interestingly, the errors are very close to the errors of the stationary mesh, with roughly the first 13 digits or more agreeing. We can therefore conclude that the moving mesh has almost no influence on the convergence behaviour of the active flux scheme in the case where both node positions and velocities are given exactly. This is by far the most common case in geometric optics, as the mesh movement is only needed to keep the mesh aligned with any optical interfaces. We will see an example of this in the next Section 5.3.

### 5.3. Compound parabolic concentrator

The compound parabolic concentrator (CPC) provides a rather difficult curved optics test case, see Fig. 7. The CPC is an ideal concentrator and it features rays that are reflected any number of times. That is, for any integer $r \geq 0$, there are rays...
Fig. 6. Example of a course mesh at several times: \( \frac{z}{Z} = 0 \) (red dot-dashed, squares), \( \frac{z}{Z} = \frac{1}{12} \) (green dashed, triangles), \( \frac{z}{Z} = \frac{1}{6} \) (blue solid, circles).

Table 5

<table>
<thead>
<tr>
<th>( \bar{E} )</th>
<th>( \varepsilon_{AF} )</th>
<th>( \gamma_{AF} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,192</td>
<td>1.9 ( \cdot ) 10(^{-2} )</td>
<td>2.0</td>
</tr>
<tr>
<td>2,368</td>
<td>1.2 ( \cdot ) 10(^{-2} )</td>
<td>3.0</td>
</tr>
<tr>
<td>4,720</td>
<td>6.2 ( \cdot ) 10(^{-3} )</td>
<td>2.1</td>
</tr>
<tr>
<td>9,408</td>
<td>3.0 ( \cdot ) 10(^{-3} )</td>
<td>2.2</td>
</tr>
<tr>
<td>18,763</td>
<td>1.4 ( \cdot ) 10(^{-3} )</td>
<td>2.3</td>
</tr>
<tr>
<td>37,817</td>
<td>5.4 ( \cdot ) 10(^{-4} )</td>
<td>2.7</td>
</tr>
<tr>
<td>75,594</td>
<td>2.2 ( \cdot ) 10(^{-4} )</td>
<td>2.6</td>
</tr>
<tr>
<td>150,698</td>
<td>8.2 ( \cdot ) 10(^{-5} )</td>
<td>2.9</td>
</tr>
</tbody>
</table>

Fig. 7. Sketch of the CPC.

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.

The CPC is an example of an ideal concentrator, i.e., it takes all light that falls into the aperture under a given acceptance angle \( \theta \) and rejects anything else. It concentrates light into the smallest possible space, an aperture of half-width \( a \). These
two parameters, $a$ and $\theta$, completely determine the shape of the CPC. The length of the optic is such that the entrance aperture has maximum width, and is given by
\[
Z = a \frac{(1 + \sin \theta) \cos \theta}{\sin^2 \theta}.
\] (56)

Conservation of volume in phase space implies that if we obtain maximum concentration in position space, there must be maximum dilution in angular space. This means the output distribution has the largest possible angular width, $[-\frac{\pi}{2}, \frac{\pi}{2}]$. This corresponds in our case $-1 \leq p \leq 1$, since we choose $n_1 = 1$. The CPC is completely determined by the half-width of the exit aperture $a$ and the acceptance angle $\theta$. The right wall of the CPC is given by $q = Q_r(z)$, where $Q_r$ is given by
\[
Q_r(z) = \frac{1}{2\cos^2 \theta} \left( c_1 z + b_1 + \sqrt{c_2 z + b_2} \right),
\] (57a)

where the coefficients are given by
\[
c_1 = -\sin(2\theta), \quad b_1 = a (\cos(2\theta) - 4\sin(\theta) - 3), \quad c_2 = 8a (2\cos \theta + \sin(2\theta)), b_2 = 8ab_1.
\] (57b)

The left wall is simply given by $q = -Q_r(z)$. The mesh is stretched uniformly with the motion of the walls, and the exact velocities can be found by differentiating (57a). The nonlocal boundary conditions that implement reflection along the mirrors depend on the normal direction in real space, and therefore on the velocity of the mirrors in phase space.

The advection speed $\frac{p}{\sqrt{1-p^2}}$ blows up at $p = \pm 1$, so that it is necessary to set some maximum momentum $p_{\text{max}} = 1 - \varepsilon$ for some $\varepsilon > 0$. We choose $\varepsilon = \frac{1}{2}a_{\text{min}}$, where $a_{\text{min}}$ is the minimum edge length in the initial grid. Furthermore, it is computationally slightly more convenient to use the CPC in the reverse direction, as an ideal spatial diluter. The initial conditions therefore consist of
\[
\rho_0(q, p) = \begin{cases} 
1 & \text{if } -a \leq q \leq a \text{ and } -p_{\text{max}} \leq p \leq p_{\text{max}}, \\
0 & \text{otherwise}.
\end{cases}
\] (58)

We use the parameters $a = \frac{1}{2}$ and $\theta = \frac{\pi}{4}$ for the CPC. From Fig. 7, one can see that as we move along the $z$-axis, the reflecting walls move outwards. Recall that $z$ acts as a time coordinate, meaning this problem involves two moving boundaries that start at $\pm a$ and move out to $\pm \frac{a}{100} = \pm 1$. To accommodate these moving boundaries, we stretch the grid uniformly in the $q$-direction, making sure that the left and right edges of the computational domain remain perfectly aligned with the reflecting walls.

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 Fig. 8, the solution is shown at $z = \frac{1}{3}Z$ and at $z = \frac{2}{3}Z$. In Fig. 9 the final solution at $z = Z$ is shown. All figures are plotted with square axes, so that the moving boundaries can be distinguished between the three time levels shown.

The analytical solution at $z = Z$ is simply a rectangular area, $-1 \leq q \leq 1$ and $-1 \leq p \leq 1$, 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.

Three-dimensional optical systems give rise to four-dimensional phase space, which imply a five-dimensional problem after inclusion of the evolution coordinate. To keep the problem manageable, we proposed a semi-discrete version of the active flux scheme. Although we restricted ourselves to two-dimensional optics problems in this work, our goal is to solve three-dimensional optics problems in the future. 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. It was therefore also necessary to consider moving meshes. 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 Jacobian 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, it is sufficient to use 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 3.4 seconds for the active flux scheme, while ray tracing would take over 2 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: the running time will scale as $O(E^3)$ for active flux versus $O(E^2)$ for ray tracing [36]. The third-order convergence will still give a significant advantage over the first-order convergence of ray-tracing. This difference will, of course, be more pronounced for larger computations that require high accuracy.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was generously supported by Signify. The authors would especially like to thank Phil Roe for his help and suggestions.
References