Adjoint Nonlinear Ray Tracing Supplemental Material

ARJUN TEH, MATTHEW O'TOOLE AND IOANNIS GKIOULEKAS, Carnegie Mellon University, USA

ACM Reference Format:
Arjun Teh, Matthew O'Toole and Ioannis Gkioulekas. 2022. Adjoint Nonlinear Ray Tracing Supplemental Material. *ACM Trans. Graph.* 41, 4, Article 126 (July 2022), 4 pages. https://doi.org/10.1145/3528223.3530077

A DERIVATION OF THE ADJOINT EQUATIONS

To derive the adjoint equations, we start by expanding the optimization objective. We separate the refractive index field η from the state variables and relate them via Lagrange multipliers:

$$\mathcal{L} = C\left(\mathbf{x}\left(\sigma_f\right), \mathbf{v}\left(\sigma_f\right)\right) - \int_0^{\sigma_f} \boldsymbol{\lambda}^\top (\dot{\mathbf{x}} - \mathbf{v})\ \mathrm{d}\sigma - \int_0^{\sigma_f} \boldsymbol{\mu}^\top (\dot{\mathbf{v}} - \eta \nabla \eta)\ \mathrm{d}\sigma - \boldsymbol{\rho} \mathcal{G}\left(\mathbf{x}\left(\sigma_f\right)\right). \tag{1}$$

We omit the inputs to $\mathbf{x}$, $\mathbf{v}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ for brevity. We will also make C and $\mathcal{G}$ implicit in the inputs $\mathbf{x}$ and $\mathbf{v}$.

In Equation (1), we use Lagrange multipliers for both the constraints in the original formulation and an additional constraint omitted in the main text. This is the $\mathcal{G}$ boundary term that represents the exit condition of the ray. In our experiments, $\mathcal{G}$ is the signed distance of a point from a plane. In general, $\mathcal{G}$ could be the signed distance function of an arbitrary mesh. This constraint determines the term σ_f in the Lagrangian, as it dictates the end of the ray trajectory.

In order for the Lagrangian to be equivalent to our original objective, we require $\frac{\partial \mathcal{L}}{\partial \sigma} = 0$, $\frac{\partial \mathcal{L}}{\partial \mathbf{v}} = 0$, and $\frac{\partial \mathcal{L}}{\partial \mathbf{x}} = 0$. These correspond to critical points in time, velocity, and position, respectively. We also require $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\mu}} = 0$, $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}}$, and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\rho}} = 0$. The solutions to these equations are exactly the original constraints of our problem, which are the equations of the dynamics along with the constraint that the ray ends at a desired point (hits a plane, exits an object, etc.).

A.1 Critical point in time

The first equation we examine is the critical point in time, $\frac{\partial \mathcal{L}}{\partial \sigma} = 0$, which describes how the boundary condition will affect the initial conditions of the adjoint equations. Taking the partial derivative of Equation (1) with respect to the (end) time variable σ gives

$$\frac{\partial \mathcal{L}}{\partial \sigma} = \frac{\partial C}{\partial \mathbf{x}}^\top \frac{\partial \mathbf{x}}{\partial \sigma} - \boldsymbol{\lambda}^\top (\dot{\mathbf{x}} - \mathbf{v}) - \boldsymbol{\mu}^\top (\dot{\mathbf{v}} - \eta \nabla \eta) - \boldsymbol{\rho} \frac{\partial \mathcal{G}}{\partial \mathbf{x}}^\top \frac{\partial \mathbf{x}}{\partial \sigma}. \tag{2}$$

We observe that both the $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ terms will be zero when the constraints are satisfied, and the derivative of $\mathbf{x}$ with respect to σ is simply $\mathbf{v}$. So, we can simplify to

$$\frac{\partial \mathcal{L}}{\partial \sigma} = \frac{\partial C}{\partial \mathbf{x}}^\top \mathbf{v} - \boldsymbol{\rho} \frac{\partial \mathcal{G}}{\partial \mathbf{x}}^\top \mathbf{v}. \tag{3}$$

The term $\frac{\partial \mathcal{G}}{\partial \mathbf{x}}$ is the normal of the region of interest in the case of a boundary condition based on a signed distance function. Then, the last term can be interpreted as a foreshortening term that attenuates $\boldsymbol{\rho}$ based on how close $\mathbf{v}$ is to being perpendicular to the exit point normal. Setting the expression to 0 and solving for $\boldsymbol{\rho}$ gives us

$$\boldsymbol{\rho} = \frac{\frac{\partial C}{\partial \mathbf{x}}^\top \mathbf{v}}{\frac{\partial \mathcal{G}}{\partial \mathbf{x}}^\top \mathbf{v}}. \tag{4}$$

Therefore, we have that $\boldsymbol{\rho}$ equals the error in position weighted by the ray's exit angle. As the denominator is only zero when the ray is moving parallel to the normal of the exit point, it is not possible to divide by zero when tracing through the exit point.

A.2 Critical point in velocity

We present the derivation of the critical point in velocity in terms of variational calculus. The notation used in this case is δ which represents an infinitesimal variation in the variable or function. A subscript is used when the variation is with respect to one of the parameters of the function. The variation will have the same dimensionality as its variable. For more information on the calculus of variations, we refer to Gelfand and Fomin [1].

We begin by taking the variation of $\mathcal{L}$ with respect to $\mathbf{v}$:

$$\delta_{\mathbf{v}} \mathcal{L} = \delta_{\mathbf{v}} C + \int_0^{\sigma_f} \boldsymbol{\lambda}^\top \delta \mathbf{v}\, \mathrm{d}\sigma - \int_0^{\sigma_f} \boldsymbol{\mu}^\top \delta \dot{\mathbf{v}}\, \mathrm{d}\sigma. \tag{5}$$

Currently, we are taking the variation of the time derivative. We can separate the time derivative and the variation by applying integration by parts to the second integral:

$$\delta_{\mathbf{v}} \mathcal{L} = \delta_{\mathbf{v}} C - \boldsymbol{\mu}^\top (\delta \mathbf{v})\Big|_0^{\sigma_f} + \int_0^{\sigma_f} \boldsymbol{\lambda}^\top \delta \mathbf{v}\, \mathrm{d}\sigma + \int_0^{\sigma_f} \dot{\boldsymbol{\mu}}^\top \delta \mathbf{v}\, \mathrm{d}\sigma. \tag{6}$$

The first two terms on the right-hand side are defined at the beginning and end of the ray, whereas the last two terms are defined over the whole trajectory. This equation needs to be zero for all perturbations of $\delta \mathbf{v}$. We note that $\delta \mathbf{v}$ at the beginning of the ray is always zero, as the initial condition is fixed and cannot be changed. This means that

$$\dot{\boldsymbol{\mu}} = -\boldsymbol{\lambda}, \tag{7}$$

$$\boldsymbol{\mu}(\sigma_f) = \frac{\partial C}{\partial \mathbf{v}}. \tag{8}$$

Author's address: Arjun Teh, Matthew O'Toole and Ioannis Gkioulekas, Carnegie Mellon University, Pittsburgh, USA.

0730-0301/2022/7-ART126
https://doi.org/10.1145/3528223.3530077

A.3 Critical point in position

To find the critical point in position, we follow the same procedure as in the velocity case. We take the variation of $\mathcal{L}$ with respect to $\mathbf{x}$:

$$\delta_{\mathbf{x}}\mathcal{L} = \delta_{\mathbf{x}}C - \int_0^{\sigma_f} \boldsymbol{\lambda}^\top (\delta\dot{\mathbf{x}})\, \mathrm{d}\sigma + \int_0^{\sigma_f} \boldsymbol{\mu}^\top (\nabla\eta\nabla\eta^\top + \eta\nabla\nabla\eta)\delta\mathbf{x}\, \mathrm{d}\sigma - \boldsymbol{\rho}\delta_{\mathbf{x}}\mathcal{G}. \tag{9}$$

We apply integration by parts on $\delta\dot{\mathbf{x}}$ to get the following expression:

$$\delta_{\mathbf{x}}\mathcal{L} = \delta_{\mathbf{x}}C - \boldsymbol{\lambda}^\top (\delta\mathbf{x})\Big|_0^{\sigma_f} + \int_0^{\sigma_f} \left[\dot{\boldsymbol{\lambda}}^\top \delta\mathbf{x} + \boldsymbol{\mu}^\top (\nabla\eta\nabla\eta^\top + \eta\nabla\nabla\eta)\delta\mathbf{x}\right] \mathrm{d}\sigma - \boldsymbol{\rho}\delta_{\mathbf{x}}\mathcal{G}. \tag{10}$$

Requiring that the expression be 0 for all $\delta\mathbf{x}$, we have that

$$\dot{\boldsymbol{\lambda}} = -\left(\nabla\eta\nabla\eta^\top + \eta\nabla\nabla\eta\right)\boldsymbol{\mu}, \tag{11}$$

$$\boldsymbol{\lambda}(\sigma_f) = \delta_{\mathbf{x}}C - \boldsymbol{\rho}\delta_{\mathbf{x}}\mathcal{G}. \tag{12}$$

A.4 Gradient with respect to refractive index

By taking the variation of $\mathcal{L}$ with respect to η, we have

$$\delta_\eta\mathcal{L} = \int_0^{\sigma_f} \left[\boldsymbol{\mu}^\top (\eta\nabla\delta\eta + \delta\eta\nabla\eta)\right] \mathrm{d}\sigma. \tag{13}$$

We select a $\delta\eta$ that ensures that $\delta_\eta\mathcal{L}$ is positive:

$$\delta\eta = \boldsymbol{\mu}^\top\nabla\eta, \tag{14}$$

$$\nabla\delta\eta = \eta\boldsymbol{\mu}^\top. \tag{15}$$

The equation requires solving for $\boldsymbol{\mu}$, which we can do using the equations derived above. Additionally, we need to be able to define $\delta\eta$ and $\nabla\delta\eta$ for our underlying data structure when calculating the gradient with respect to η.

B IMPLEMENTATION OF A TRILINEAR VOLUME

We use a voxel grid to represent the refractive index field, and interpolate using trilinear interpolation. Using trilinear interpolation, we can obtain both the scalar value and the spatial gradient of the refractive index field at any point in the volume. The interpolation weights are trilinear in the sampling position, but linear in the data points. We can express trilinear interpolation as

$$\eta(\mathbf{x};\theta) = \mathbf{w}\cdot\theta = \begin{bmatrix} w_{00}(\mathbf{x}) \\ w_{01}(\mathbf{x}) \\ w_{10}(\mathbf{x}) \\ w_{11}(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \theta_{00} \\ \theta_{01} \\ \theta_{10} \\ \theta_{11} \end{bmatrix}, \tag{16}$$

where θ contains the data values of the volume, and w_{ij} are the weights associated with each data point. These weights are based on the distance between the query and data points. In two dimensions, there are four points that are closest to the query point, giving us four nonzero weights. To obtain the spatial gradient of the field, we can differentiate this equation with respect to $\mathbf{x}$:

$$\frac{\mathrm{d}\eta(\mathbf{x};\theta)}{\mathrm{d}\mathbf{x}} = \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}\mathbf{x}}\cdot\theta = \begin{bmatrix} \frac{\mathrm{d}w_{00}}{\mathrm{d}\mathbf{x}} \\ \frac{\mathrm{d}w_{01}}{\mathrm{d}\mathbf{x}} \\ \frac{\mathrm{d}w_{10}}{\mathrm{d}\mathbf{x}} \\ \frac{\mathrm{d}w_{11}}{\mathrm{d}\mathbf{x}} \end{bmatrix} \cdot \begin{bmatrix} \theta_{00} \\ \theta_{01} \\ \theta_{10} \\ \theta_{11} \end{bmatrix}. \tag{17}$$

The derivative with respect to $\mathbf{x}$ is a column vector for each of the weights, resulting in a 2×4 matrix.

To apply the adjoint equations to these values, we need to know how the variation of the data points affects the variation of the calculated η value. For this, we take the variation of η with respect to the data values θ:

$$\frac{\mathrm{d}\eta(\mathbf{x};\theta)}{\mathrm{d}\theta} = \mathbf{w}\cdot\,\mathrm{d}\theta = \begin{bmatrix} w_{00}(\mathbf{x}) \\ w_{01}(\mathbf{x}) \\ w_{10}(\mathbf{x}) \\ w_{11}(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} \mathrm{d}\theta_{00} \\ \mathrm{d}\theta_{01} \\ \mathrm{d}\theta_{10} \\ \mathrm{d}\theta_{11} \end{bmatrix}. \tag{18}$$

We can then replace $\mathrm{d}\eta$ in Equation (??) with this relation:

$$d_\eta\mathcal{L} = \int_0^{\sigma_f} \left(\eta\nabla\mathbf{w}^\top\,\mathrm{d}\theta + \mathbf{w}^\top\,\mathrm{d}\theta\nabla\eta\right)^\top \boldsymbol{\mu}. \tag{19}$$

We rearrange Equation (19) so that $\mathrm{d}\theta$ is factored out:

$$d_\eta\mathcal{L} = \int_0^{\sigma_f} \boldsymbol{\mu}^\top \left(\eta\nabla\mathbf{w}^\top + \nabla\eta\mathbf{w}^\top\right) \mathrm{d}\theta. \tag{20}$$

This gives us a rule for updating the values of the θ in order to satisfy Equations (14) and (15).

C USING ARC-LENGTH PARAMETERIZATION

We consider the arc-length parameterized version of Hamilton's Equations (??)-(??) [2],

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}s} = \frac{\mathbf{v}}{\eta}, \tag{21}$$

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}s} = \nabla\eta. \tag{22}$$

By discretizing these equations using the symplectic Euler integrator, we have:

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \frac{\mathbf{v}_i}{\eta\left(\mathbf{x}_{i-1}\right)}\Delta s, \tag{23}$$

$$\mathbf{v}_i = \mathbf{v}_{i-1} + \nabla\eta\left(\mathbf{x}_{i-1}\right)\Delta s. \tag{24}$$

We can then reverse these equations to obtain:

$$\mathbf{x}_{i-1} = \mathbf{x}_i - \frac{\mathbf{v}_i}{\eta\left(\mathbf{x}_{i-1}\right)}\Delta s, \tag{25}$$

$$\mathbf{v}_{i-1} = \mathbf{v}_i - \nabla\eta(\mathbf{x}_{i-1})\Delta s. \tag{26}$$

We note that, in this case, the backward Equation (25) for $\mathbf{x}_{i-1}$ is implicit and cannot be solved explicitly. This is because, in the forward direction, Equations (23)-(24) use $\mathbf{x}_{i-1}$. To solve Equation (25), we would need to use an implicit solver such as Newton's Method. This would greatly increase the computational cost of the gradient calculation. Alternatively, we would need to store the trajectory of the ray during forward tracing, and use it during backward tracing. This would increase the memory requirements of our algorithm, making them similar to those of automatic differentiation techniques that record the entire trajectory through the computation graph.

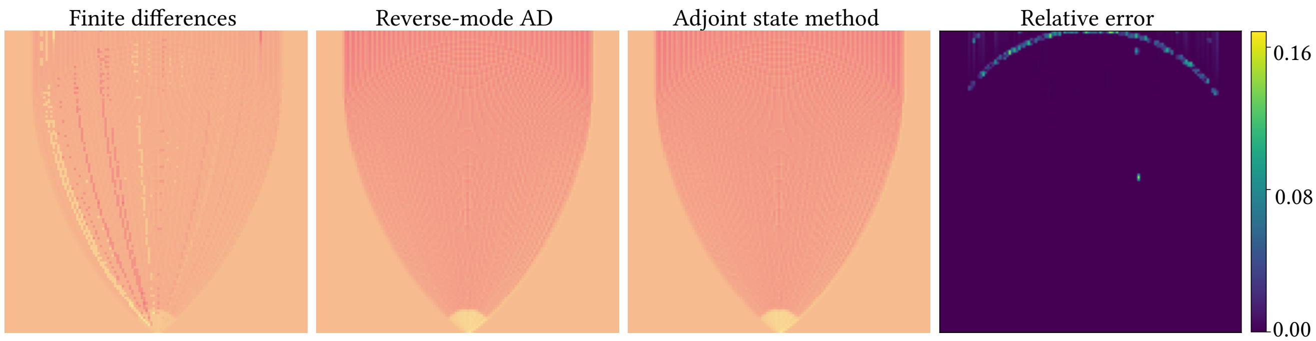

Fig. 1. Comparison of gradients generated using finite differences, reverse-mode AD and the adjoint method. We trace rays through the Luneburg lens, then calculate the loss $\mathbf{1}^\top \mathbf{x}(\sigma_f)$. The adjoint and reverse-mode AD methods produce gradients that closely match, as seen in the relative error plot. The result of finite differences deviates significantly, due to its instability when tracing.

Table 1. Performance comparison between the adjoint method and reverse-mode AD for one optimization iteration.

Implementation	Peak memory	Time
AD (Enoki)	5.61 GB	2.15 s
Adjoint (Enoki)	0.29 GB	0.88 s
AD (PyTorch)	16.01 GB	3.29 s
Adjoint (PyTorch)	0.14 GB	4.25 s

D VERIFYING THE ADJOINT METHOD WITH AUTOMATIC DIFFERENTIATION

We verify that the adjoint method produces gradient values equal to those from automatic differentiation. We trace rays through the Luneburg lens and then calculate the gradient using both reverse-mode AD and the adjoint method. We visualize the gradients for a slice of the volume in Figure 1. The difference between the estimates remains low throughout most of the volume.

E COMPARISONS IN PYTORCH AND ENOKI

We implemented the adjoint method in both PyTorch and Enoki and present performance numbers for both versions in Table 1. In both frameworks, the adjoint method performs significantly better than reverse-mode AD in terms of memory usage. In Enoki, the adjoint method performs faster as well, but not in PyTorch. We attribute this to the fact that accessing the data structure implemented in PyTorch takes longer than traversing the computation graph.

F IMPORTANCE OF REVERSIBILITY

We use Figure 2 to highlight the importance of reversibility. In particular, we show that using the non-reversible formulation based on the arc-length parameterization results in biased gradients. Using these biased gradients for optimization results in an optimization trajectory that both converges to a higher loss and is unstable.

REFERENCES

[1] Izrail Moiseevitch Gelfand and Sergei Vasilyevich Fomin. 1963. *Calculus of variations.* Courier Corporation.

[2] Ivo Ihrke, Gernot Ziegler, Art Tevs, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. 2007. Eikonal rendering: Efficient light transport in refractive objects. *ACM TOG* (2007).

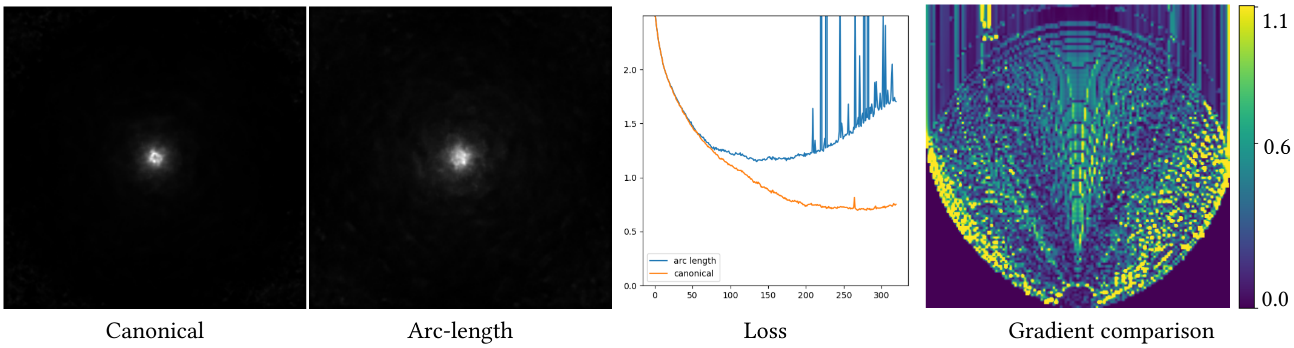

Fig. 2. Comparison of optimization results with the adjoint method using the reversible canonical parameterization and the non-reversible arc-length parameterization. We optimize a refractive index volume to focus single collimated light source to the center of the sensor plane. Both optimization variants achieve a focus, but the optimization using arc-length parameterization achieves a worse loss value and becomes unstable. At the right, we show the relative difference between the gradients computed using the canonical and arc-length parameterizations.