Applying Variational Integrators to the Geometric Optics Simulation of Continuously Varying Refractive Indices

Special Thanks to Zac Manchester and Advanced Robot Dynamics and Simulation

Grey Golla
Electrical and Computer Enginering
Carnegie Mellon University
Pittsburgh, PA
ggolla@andrew.cmu.edu

***Abstract*—I apply variational integrators to the problem of optical ray tracing in the presence of continuously varying refractive index materials (commonly known as GRadient INdex, or GRIN). Optical ray tracing follows the principle of least time, which has deep parallels with the classical mechanical principle of least action. The variational integrator allows the exact preservation of discretized least time, preventing accumulated drift present in other numerical integration schemes. Additionally, I apply an event-driven model to handle discrete jumps in material. With this, I present a simple and unified model for simulating geometric optics through arbitrary physical systems.**

I. Background

Geometric ray tracing is the standard method for simulating optical systems in a variety of industries including lens design and computational photography. This is a standard system for approximating the complex wave interactions between light and the media it propagates through. Basically, light in the scene is represented by drawing a number of light paths, representing paths through space where infinitesimal beams of light can travel. Typically, these paths originate at light sources or other points of interest, then are allowed to propagate through the system. This construction seems totally unrelated to thinking about light as a wave, but through the Huygens-Fresnel principle we can show a deep connection between geometric optics and wave optics [1].

While a detailed derivation of this link is beyond the scope of this introduction, a qualitative understanding can be gained by looking at the interaction of a planar wavefront with a sharp transition in the index of refraction, which is exactly proportional to a change in the local speed of wave propagation (speed of light). A schematic of this interaction is shown in Fig. 1. Each point on the planar wave can be considered the originating point of a circular wavefront. In free space, these circular wavefronts interfere to yield another parallel planar wavefront. However, at the interface between two different speeds of wave propagation, the circular wavefronts will not interfere to form the same wavefront, but instead another planar wavefront at a different angle. Simple geometry can use this description to derive Snell's law $n_1 sin(\theta_1) = n_2 sin(\theta_2)$, the high school physics law of refraction. The description of circular wavefront propagation at all points can further be used to justify Fermat's principle, or the principle of least time [1].

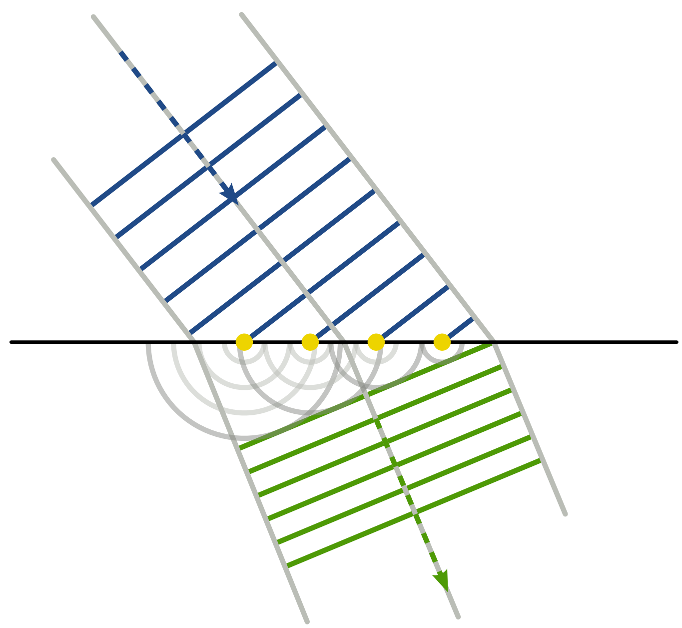

Fig. 1. Planar wave refracting through a change in index. From [2].

The principle of least time states that between any two points, light will travel the path that takes the least time to traverse. In a uniform medium, this is predictably a straight line, but when travelling through changes in refractive index curves and sharp changes in angle are possible. Formally, the time taken for light to traverse a given path is $T = \int n_r(q)ds$. While the principle is called "least" time, in reality we are looking for stationary points, meaning that the variation of path time is 0, $\delta T = 0$. This system is strikingly similar to the principle of stationary action in classical mechanics, where $S = \int L(q(t), \dot{q}, t)dt$ and $\delta S = 0$. This means we can apply variational techniques for simulating these systems to the problem of optical ray tracing.

One important practical difference between mechanics problems and geometric optics problems is that while geometric problems are generally highly continuous except for contact dynamics, optical problems often have discontinuous changes in the index of refraction because the material a ray is propagating through can discretely change. This is different from contact mechanics because we don't want a constraint keeping light on one side of the material, instead we need to handle a sharp change in propagation direction at the boundary.

II. Implementation

A. Integration in GRIN regions

I chose to implement a simple 2D system, as opposed to a full 3 dimensional ray simulation for a few reasons. First, it was much simpler while still exhibiting all of the interesting behaviors we wish to capture. Second, many real-world optical systems we care about are radially symmetrical, meaning that simulating a planar slice of light's behavior is enough to fully describe the system. Because of this, many industry standard optical design and simulation tools operate in 2D.

The approach I took for designing the core variational integrator is identical to the approach taken for solving the catenary problem, where we solve for y values as a function of uniform discrete x values. The light path is made up of segments with a constant Δx and an arbitrary Δy based on the endpoints of the segment (the y values we solve for).

One interesting note here is that in the catenary problem, these linear segments had a physical explanation: the rope was made of small rigid segments of constant density, meaning that the potential energy (and thus the Lagrangian) of each link was exactly computed by taking the potential energy of a point mass at the center of the link with the appropriate mass: $U(y_1, y_2) = mg(y_1 + y_2)/2$. In ray tracing, there is no such physical argument for a linear segment of light trying to propagate through GRIN materials. I chose to approximate light's behavior by assuming that the index of refraction is constant along the line segment and equal to the true value at the midpoint of the segment, i.e.

$$T(x_1, y_1, y_2) = S(y_1, y_2) * n_r(x_1 + \Delta x/2, (y_1 + y_2)/2) \quad (1)$$

This is a reasonable approximation so long as n_r changes slowly compared to the length of one segment. Our choice of x as the integration variable makes this statement dependent on the slope of the light, meaning as light travels increasingly vertically the approximation will become less accurate.

With the discretized travel time from (1), we can derive the full variational integrator by considering 3 points in the solution and optimizing y_3 to be stationary for $T(x_1, y_1, y_2) + T(x_2, y_2, y_3)$. The integrator uses newton's method to solve (2) for $y_k + 1$ at every discrete step k.

$$D_2T(y_{k-1}, y_k) + D_1T(y_k, y_k + 1) = 0 \quad (2)$$

B. Handling discrete jumps in index

While this system can be arbitrarily accurate for smoothly varying index of refraction, it cannot handle discrete jumps in index well. As mentioned before, sudden changes in material are very common in physical systems, requiring robust handling for our simulation to be useful. I chose to represent a change in material as an implicit surface ($f(x, y) = 0$ for arbitrary f) and define an index of refraction $n_1(x, y)$ for "left" of the material and $n_2(x, y)$ for "right" of the material. This is possible because I am using a constant Δx step, meaning rays will always encounter the surface moving from left to right.

The choice of implicit surface representation is important because many optical systems rely on arbitrarily smooth but mathematically simple shapes such as circles and other polynomial surfaces. Finding the location of the surface, as well as intersecting it with light rays, is simply a root finding problem or solving a system of nonlinear equations. In my case, I intersect the surface with line segments so I use Newton's method to solve for $f(x, y(x))$, where $y(x)$ is the parameterization of the line segment with respect to x. In order to do the refraction, we need the normal of the surface at the point of intersection, which is simply the gradient of the surface at that point.

Unfortunately, the choice of a constant Δx makes applying the refraction difficult, because the surface can intersect the light ray at any point, not just at one of the discretized Δx points. To solve this, we invoke a common approximation within the geometric optics field, the "thin lens approximation". Essentially, the thin lens approximation ignores the propagation distance between nearby refractive surfaces and assumes they happen at the same point. This is generally accurate because the ray path is highly sensitive to changes in angle, but not sensitive to slight position changes [3].

After solving for y_{k+1}, I check for intersections between a surface and the line segment $(x_k, y_k), (x_k + \Delta x, y_{k+1})$. If one occurs, I use Snell's law to compute the refracted direction of the ray at the intersection point, then compute a new y_{k+1} so the line segment now goes in the refracted direction. This roughly ignores the distance between y_k and the intersection point and grows the distance between the intersection point and y_{k+1}.

There are a few complications to this plan. First, it is possible that instead of refraction, a decrease in refractive index could cause total internal reflection, where the ray is perfectly reflected along the surface. This is handled in exactly the same way as normal refraction. A related problem is that refraction might cause the direction of the ray to travel backwards, i.e. the vector has a negative x component. My discretization of a constant (positive) Δx does not allow this scenario, meaning that the only option is to terminate the ray trace at that point.

III. Results

This framework is capable of simulating both classical (constant index) and GRIN scenarios, I started by verifying the simulation on some well-understood optical systems, including a spherical thin lens in Fig. 2. The simulation shows that with a lens made of two spherical sections and a higher index of refraction inside than outside, parallel rays intersect at the focal point of the lens.

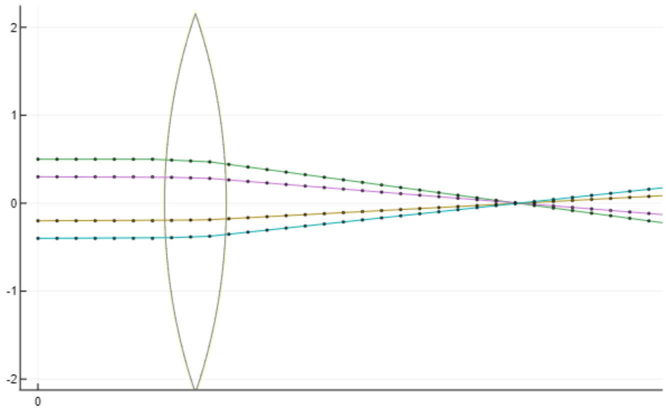

Fig. 2. Simulation of a spherical lens focusing parallel rays to its focal point

A simple GRIN scenario that I thought was neat was the continuously varying optical fiber, which is analogous to a classical optical fiber that uses total internal reflection to guide light from one end to the other by reflecting it off the interior surface of the fiber, simulated in Fig. 3. This result also shows a weakness in my rendering system. I render the surfaces by using Julia's `contour!` plotting command to plot the contour of the function at value 0. Unfortunately, if the evaluation points are misaligned then the contour ends up being shown in slightly the wrong position, which is why the light rays reflect above the rendered location of the bottom edge of the fiber.

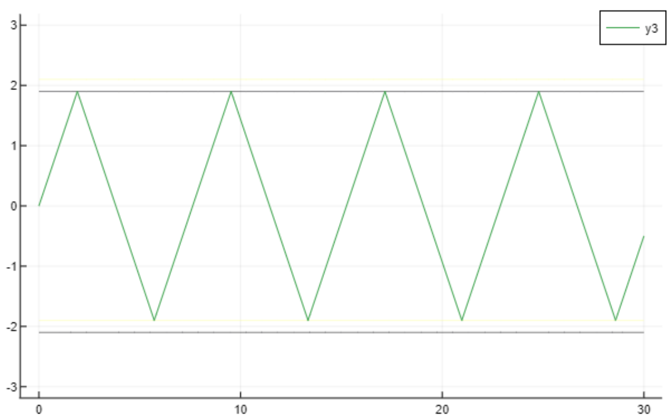

Fig. 3. Simulation of an optical fiber with walls at +/-2

The GRIN analog to this has an index that decreases in both directions away from the center of the fiber. Note that in Fig. 4 we plot inverse refractive index (wave propagation velocity), so it increases away from the center. Observe how, as opposed to the discrete materials, the light continuously curves as it travels through the medium.

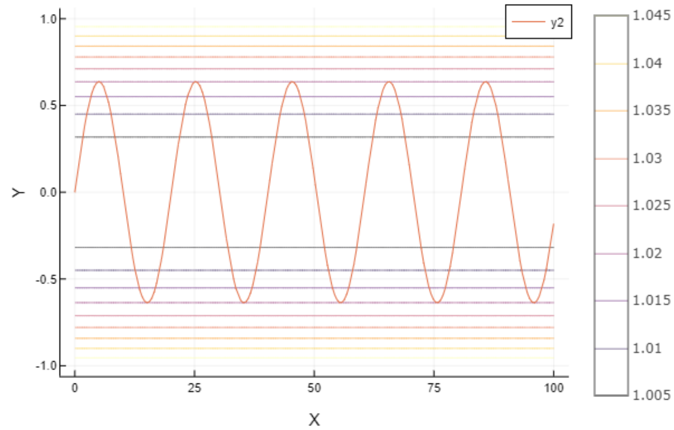

Fig. 4. Simulation of a continuously varying material forming a gradient optical fiber.

Finally, showing off the full behavior of the system, I combined a spherical lens with the continuously varying optical fiber to focus parallel beams into a point, then a continuous optical fiber, this time with a bend in it. The lens makes the parallel beams each have a different angle entering the fiber so that each ray will not overlap and thus can be separated at the far end of the optical fiber.

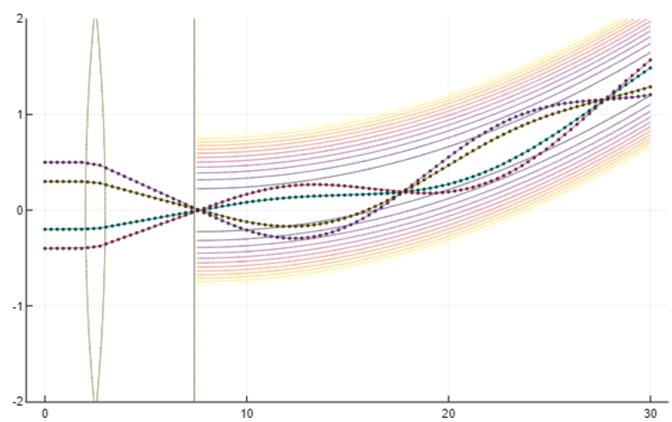

Fig. 5. Combined discrete and continuous materials showing parallel beams focused into a continuous waveguide.

IV. Discussion

A. *Parameterization*

I knew from the beginning that choosing to parameterize with a constant Δx was an interesting choice. Starting with the advantages, the first is simplicity. The obvious advantage is only needing to track 1 coordinate instead of a coordinate pair, but the biggest advantage is the simplicty of solving each step. Fully parameterizing both x and y by t would require constraining the length of each step, e.g. by adding a constraint to the optimization for the length of the $(\Delta x, \Delta y)$ line segment, so I'd have to use something much slower and more complex than Newton's method.

I've already mentioned most of the problems with this approach in earlier sections, but the biggest is that it constrains the problem types you can effectively simulate. Anything where the light reflects or refracts back on the x axis will simply not work, each ray can only pass a given x coordinate once. Additionally, as the rays get steeper, the effective accuracy of the simulation drops because each linear segment of light gets longer, meaning that only "mostly horizontal" optical systems will be effectively traced. Many optical systems, including most lens designs, follow this property, but it is certainly not universal.

The effective accuracy dropping with the slope of the light is a big problem when trying to reason about the required step of the simulation. As stated above, for the integration to be accurate we need the index of refraction to be roughly constant over the scale of the linear segments along the path. With a constant Δx, however, we cannot say anything strong about the length of these linear segments without constraining the angles of light paths encountered, which is difficult to do in general.

B. Handling discrete changes in index

Another effect of my discretization choice is the inability to handle discrete index changes precisely. As mentioned above, while the delta in the index of refraction and the normal of the surface are precisely used (up to the linearization between two Δx points), the location of the refraction is moved to the preceding Δx. This introduces some error in the simulation compared to basic ray tracing systems with no support for GRIN materials. As mentioned above, the thin lens approximation is taken liberally throughout the optics community, but it would be better to avoid this. The effect of this is very clear in Fig. 6, where the refraction for the second surface of the lens clearly happens in the middle of the lens instead of along its actual back surface.

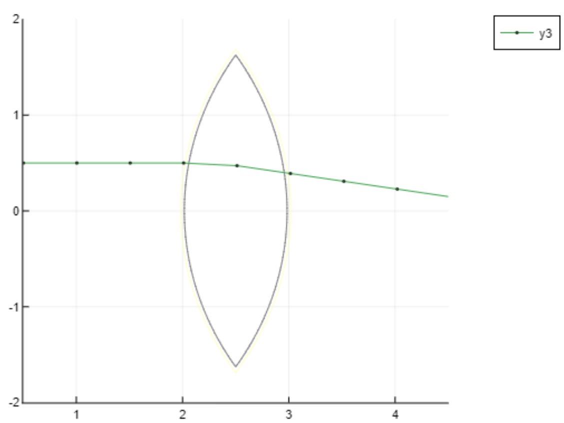

Fig. 6. Simulation of the spherical lens with intentionally large Δx. Note how the second refraction happens in the middle of the lens.

More generally, the problem of handling discrete changes in index is difficult because within one linear segment, we want to handle both the bending caused by the change in GRIN material and the corner caused by the boundary in materials. I chose to first bend light based on the GRIN material and then discretely refract, but there may be other choices that yield lower error.

Another problem that caused noticeable artifacts in certain scenes was "double refraction". Because I detect an intersection on the line segment $(x_k, y_k), (x_k + \Delta x, y_{k+1})$ and then adjust y_{k+1} based on this refraction, it is possible that the new line segment no longer intersects with the surface. This allows a second refraction to take place, causing highly inaccurate results in some cases. Fig. 7 shows a schematic drawing of that effect.

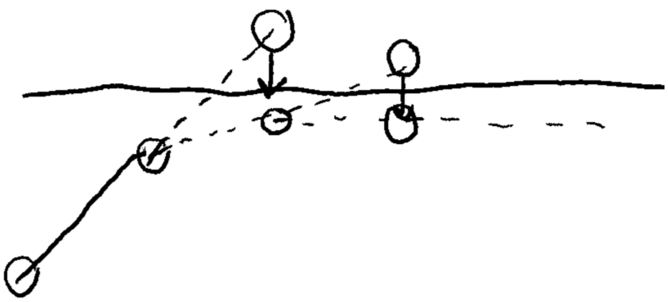

Fig. 7. Drawing of the "double refraction" effect where the light ray approaches a horizontal surface from the bottom left corner. It refracts at the first arrow, pushing the line segment back into the first medium where it then refracts again.

Solving this problem is not easy. One option would be to adjust y_{k+1} to be on the ray that results from doing the refraction at the exact intersection point, but then the new line segment $(x_k, y_k), (x_k + \Delta x, y_{k+1})$ would not have the correct propagation direction. This could be solved by storing the propagation direction separately from the location, but that increases complexity across the entire simulation. In general, any option that does not create a special "intermediate" $x_{k+0.5}$ point for the intersection is going to have this problem, regardless of the parameterization used.

C. Future work

The obvious extension of this framework would be to allow the tracing of 3D paths. While none of the code could be transferred, the framework of a variational integrator combined with line segments intersecting implicit surfaces will transfer mostly unchanged. As mentioned above, using a parameterization other than constant Δx will require adding a constraint to the variational integrator, requiring a more complex solver, and to make a generally useful 3D path tracer that is basically necessary.

Another very simple extension could be adding support for anisotropic materials, which have a different effective index of refraction depending on the direction light travels within them. This would be very simple to add to the current system, and could easily extend a hypothetical future 3D version as well.

Finally, a huge application of optical path tracing is various forms of optimization work, e.g. for optical systems design. Using this ray tracer as the core of an optimizer for optical parameters would allow the generation of optimal GRIN materials to create, for example, exotic forms of lenses with superior properties to conventional single-index lenses. The relative simplicity of this system, with the Δx paramterization, might make optimizing over it easier than more complex alternatives.

V. Conclusion

This system successfully simulates a wide range of real optical systems, combining discrete jumps in refractive index with GRadient INdex materials nearly arbitrarily. The system was validated against a variety of real-world optical systems, including a convex-convex spherical lens focusing parallel rays to its focal point (Fig. 2) and an optical fiber capturing light through total internal reflection (Fig. 3). The simulation was used to show interesting GRIN scenarios serving as optical fiber-like objects. Despite limitations, using a constant Δx as the parameterization has computational and simplicity benefits and works for a variety of real-world problems in optics. The system can be fruitfully extended in a variety of ways including moving to 3D, changing the parameterization to a more generic one, or adding optimization functionality to design new optical systems.

The code for this project can be found at https://github.com/Bobobalink/Variational-GRIN

References

[1] "Fermat's principle," Wikipedia, 25-Oct-2022. [Online]. Available: https://en.wikipedia.org/wiki/Fermat%27s_principle. [Accessed: 19-Dec-2022].

[2] A. Nordmann, "File:refraction - huygens-fresnel principle.svg," Wikimedia Commons, 11-Jul-2020. [Online]. Available: https://commons.wikimedia.org/wiki/File:Refraction_-_Huygens-Fresnel_principle.svg. [Accessed: 19-Dec-2022].

[3] "Thin Lens," Wikipedia, 17-Sep-2021. [Online]. Available: https://en.wikipedia.org/wiki/Thin_lens. [Accessed: 19-Dec-2022].