Software Implementation of the Eikonal Equation

Dmitry S. Kulyabov[*†], **Migran N. Gevorkyan**[*], **Anna V. Korolkova**[*]

[*] *Department of Applied Probability and Informatics*
Peoples' Friendship University of Russia (RUDN University)
6 Miklukho-Maklaya str., Moscow, 117198, Russian Federation
[†] *Laboratory of Information Technologies*
Joint Institute for Nuclear Research
6 Joliot-Curie, Dubna, Moscow region, 141980, Russian Federation

Email: kulyabov_ds@rudn.university, gevorkyan_mn@rudn.university, korolkova_av@rudn.university

The Maxwell equations have a fairly simple form. However, finding solutions to Maxwell's equations is an extremely difficult task. Therefore, various simplifying approaches are often used in optics. One such simplifying approach is to use the approximation of geometric optics. The approximation of geometric optics is constructed with the assumption that the wavelengths are small (short-wavelength approximation). The basis of geometric optics is the eikonal equation. The eikonal equation can be obtained from the wave equation (Helmholtz equation). Thus, the eikonal equation relates the wave and geometric optics. In fact, the eikonal equation is a quasi-classical approximation (the Wentzel–Kramers–Brillouin method) of wave optics. This paper shows the application of geometric methods of electrodynamics to the calculation of optical devices, such as lenses Maxwell and Luneburg. The eikonal equation, which was transformed to the ODE system by the method of characteristics, is considered. The resulting system is written for the case of Maxwell and Luneburg lenses and solved by standard numerical methods. Describes the implementation details and images of the trajectories of rays and fronts of the waves.

Key words and phrases: eikonal equation; Luneburg lens; Maxwell lens; characteristics method; Julia.

CEUR-WS.org/Vol-2177/paper-04-e001.pdf

Copyright © 2018 for the individual papers by the papers' authors. Copying permitted for private and academic purposes. This volume is published and copyrighted by its editors.
In: K. E. Samouylov, L. A. Sevastianov, D. S. Kulyabov (eds.): Selected Papers of the VIII Conference "Information and Telecommunication Technologies and Mathematical Modeling of High-Tech Systems", Moscow, Russia, 20-Apr-2018, published at http://ceur-ws.org

1. Introduction

In this article, we consider the approach to transform the eikonal equations to the ODE system. In the first part we briefly formulate the problem and describe Luneburg and Maxwell lens. The second part describes a software implementation that allows to visualize the trajectory of rays and wave fronts.

2. Application of the characteristic method to the eikonal equation

2.1. The eikonal equation

The eikonal equation can be obtained from Maxwell's equations written for the regions free of currents and charges and under the condition of a time-changing harmonic electromagnetic field in a nonconducting isotropic medium [1–4]. In general, the eikonal equation is written as a partial differential equation of the first order:

$$\begin{cases} |\nabla u(\mathbf{r})|^2 = n^2(\mathbf{r}), & \mathbf{r} \in \mathbb{R}^3, \\ u(\mathbf{r}) = \varphi(\mathbf{r}), & \mathbf{x} \in \Gamma \subset \mathbb{R}^3. \end{cases}$$

where $\mathbf{r} = (x, y, z)^T$ is radius-vector, $\varphi(\mathbf{r})$ is boundary condition, $n(\mathbf{r})$ is refractive index of the medium. The function $u(\mathbf{r})$ is real scalar function with a physical meaning of time. It is also often called as the *eikonal* function [5, 6].

For visualization of lens modeling results, we will consider their projection on the Oxy plane. In this case, the eikonal equation is reduced to a two-dimensional form:

$$\begin{cases} \left(\dfrac{\partial u(x,y)}{\partial x}\right)^2 + \left(\dfrac{\partial u(x,y)}{\partial y}\right)^2 = n^2(x,y), & (x,y) \in \mathbb{R}^2, \\ u(x,y) = \varphi(x,y), & (x,y) \in \Gamma \subset \mathbb{R}^2. \end{cases}$$

Using the characteristic method, the eikonal equation can be transformed into an ODE system that can be solved by standard numerical methods.

2.2. Transformations of the eikonal equation to the ODE system

By using the method of characteristics [7–12], we may convert the eikonal equation for the plane case to the form of the differential equation system with functions: $x(t)$, $y(t)$, $p_1(t)$, $p_2(t)$:

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = \dfrac{p_1}{n^2}, \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = \dfrac{p_2}{n^2}, \\ \dfrac{\mathrm{d}p_1}{\mathrm{d}t} = \dfrac{1}{n}\dfrac{\partial n}{\partial x}, \\ \dfrac{\mathrm{d}p_2}{\mathrm{d}t} = \dfrac{1}{n}\dfrac{\partial n}{\partial y}. \end{cases} \quad \begin{array}{l} \text{Initial conditions} \\ x(t)|_{t=0} = x_0, \\ y(t)|_{t=0} = y_0, \\ p_1(t)|_{t=0} = c_1 n(x_0, y_0), \\ p_2(t)|_{t=0} = c_2 n(x_0, y_0), \end{array}$$

where

$$p_1 = \frac{\partial u}{\partial x}, \quad p_2 = \frac{\partial u}{\partial y}.$$

The following relation $c_1^2 + c_2^2 = 1$ is imposed on constants c_1 and c_2. Thus we may take: $c_1 = \cos(\alpha)$ and $c_2 = \sin(\alpha)$. Initial conditions give a mathematical description of the source of the rays. For example, to model a point source, we need to fix the initial

coordinates x_0, y_0 and change the angle α, which will set the angle of the beam exit from the source-point. To simulate the radiating surface, on the contrary, it is necessary to fix the angle α and change the coordinates x_0 and y_0.

The parameter t has a physical meaning of the signal passing time from the point (x_0, y_0) to the point (x, y).

In polar coordinates, the eikonal equation has the following form:

$$\left(\frac{\partial u(r,\varphi)}{\partial r}\right)^2 + \frac{1}{r^2}\left(\frac{\partial u(r,\varphi)}{\partial \varphi}\right)^2 = n^2(r),$$

and corresponding system of ODEs will have the form:

$$\begin{cases} \dfrac{\mathrm{d}r}{\mathrm{d}t} = p_r, \\ \dfrac{\mathrm{d}\varphi}{\mathrm{d}t} = \dfrac{p_\varphi}{r}, \\ \dfrac{\mathrm{d}p_r}{\mathrm{d}t} = n\dfrac{\partial n}{\partial r} + \dfrac{p_\varphi^2}{r}, \\ \dfrac{\mathrm{d}p_\varphi}{\mathrm{d}t} = -\dfrac{p_\varphi p_r}{r}. \end{cases} \qquad \begin{array}{l} \text{Initial conditions} \\ r(t)|_{t=0} = r_0, \\ \varphi(t)|_{t=0} = \varphi_0, \\ p_r(t)|_{t=0} = c_1 n(r_0), \\ p_\varphi(t)|_{t=0} = c_2 n(r_0). \end{array}$$

Let's consider the examples of lenses [13, 14].

2.3. Luneburg lens

The Luneburg lens [15–18] is a spherical lens of radius R with center at point (X_0, Y_0) (consider the projection on the plane Oxy) with a refractive index of the following form

$$n(x,y) = \begin{cases} n_0\sqrt{2 - \left(\dfrac{r}{R}\right)^2}, & r \leqslant R, \\ n_0, & r > R, \end{cases}$$

where $r(x,y) = \sqrt{(x-X_0)^2 + (y-Y_0)^2}$ is the distance from the center of the lens to an arbitrary point in the (x, y) plane. The formula implies that the coefficient n continuously varies from $n_0\sqrt{2}$ to n_0 starting from the center of the lens and ending with its boundary. The refractive index of the medium outside the lens is constant and is equal to n_0. Usually n_0 is equal to 1.

To solve the eikonal equation by the method of characteristics, it is necessary to find partial derivatives of the function $n(x, y)$. For the Luneburg lens case the partial derivatives are:

$$\frac{\partial n(x,y)}{\partial x} = -\frac{n_0^2(x-X_0)}{R^2 n(x,y)}, \quad \frac{\partial n(x,y)}{\partial y} = -\frac{n_0^2(y-Y_0)}{R^2 n(x,y)}, \quad r \leqslant R.$$

Outside lens region the derivatives are equal to 0.

2.4. Maxwell's fish eye lens

The Maxwell's fish eye lens [19] is also a spherical lens of radius R with center at point (X_0, Y_0) (consider the projection on the plane Oxy) with a refractive index of the following form:

$$n(x,y) = \begin{cases} \dfrac{n_0}{1+\left(\dfrac{r}{R}\right)^2}, & r \leqslant R, \\ n_0, & r > R. \end{cases}$$

To solve the eikonal equation by the method of characteristics, it is necessary to find partial derivatives of the function $n(x, y)$. For Maxwell's lens case these derivatives may be written in the form:

$$\frac{\partial n(x,y)}{\partial x} = -\frac{2n^2(x,y)(x-X_0)}{n_0 R^2}, \ \frac{\partial n(x,y)}{\partial y} = -\frac{2n^2(x,y)(y-Y_0)}{n_0 R^2}, \ r \leqslant R.$$

3. Numerical simulation of Luneburg and Maxwell lens

3.1. Description of the numerical modeling

We carry on numerical modeling for lenses with a radius $R = 1$, the refractive index of the external medium $n_0 = 1$, the center of the lens was placed in the point $(X_0, Y_0) = (2, 0)$, the boundary region was set as the rectangle $x_{\min} = 0$, $x_{\max} = 5$, $y_{\min} = -1.5$ and $y_{\max} = 1.5$. The point source was placed on the lens boundary at $(x_0, y_0) = (0, 0)$. 50 values of the α parameter have bee taken from the interval $[-\pi/2 + \pi/100, \pi/2 - \pi/100]$, which allowed to simulate rays trajectories from a point source within an angle slightly smaller than 180°. The t parameter was changed within the $[0, 5]$ interval.

Each α parameter value sets new initial conditions for the ODE system. The process of numerical simulation consists in multiple solution of this system for different initial conditions. The numerical solution of the ODE system for a particular initial condition gives us a set of points $(x_I, y_I), I = 1, \ldots, N$ approximating the trajectories of a particular beam. After performing calculations for all the selected initial conditions, we obtain a set of rays. To visualize the rays, it is enough to depict each of the obtained numerical solutions. The result of the simulation may be seen in 1 and 2.

To visualize the wave fronts with the resulting numerical data it is necessary to carry out additional manipulations. From each numerical solution, we must select points (x_I, y_I) that correspond to a specific point in time t_I.

The use of a numerical method with a fixed step gives an advantage, since each numerical solution will be obtained for the same uniform grid $t_0 < t_1 < \ldots < t_i < \ldots < t_n$.

3.2. The description of software implementation

For the simulation of rays transmission through the lens we used Julia language [20] and Python with Matplotlib and NumPy for visualization. To solve the ODE system we used the classical Runge–Kutta methods of order 6 with a constant step, implemented by us in Julia as a separate function.

The main program files are located in the `src` directory. Files of Julia source code have `jl` extension. Note that the Julia language encourages usage of Unicode variable so we may use various characters of utf-8 encoding, which are often found in mathematical formulas. So, we used Greek letters and symbols of partial differentiation ∂.

– `eikonal.jl` — the right part of the system of differential equations, for all lenses it has the same form, differing only in functions `n(x,y)` and ∂`n(x,y)`.

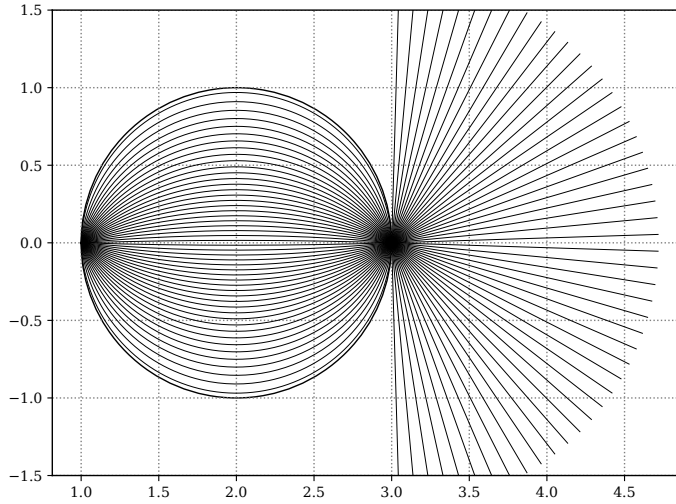

Figure 1. The trajectories of the rays inside of Maxwell's lens for a point source and $n_0 = 1$.

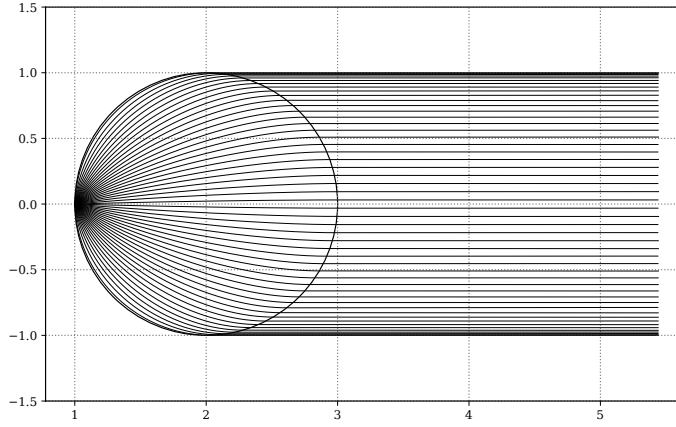

Figure 2. The trajectories of the rays inside of Luneburg lens for a point source and $n_0 = 1$.

- `maxwell.jl` — refractive index `n(x,y)` and its derivatives ∂`n (x,y)` for the Maxwell lens.
- `luneburg.jl` — similar to the previous item, but for Luneberg lens.
- `parameters.jl` — common parameters for all lenses: lens radius, refractive index of the external medium, coordinates of the center of the lens, area boundaries for Cartesian and polar coordinates, etc.
- `RK.jl` — a separate function that implements the Runge-Kutta method.

Outside the directory `src` are scripts that trigger calculations. The script `cartesian1.jl` performs calculations for the point source, the script `cartesian2.jl` for parallel rays, and the script `polar.jl` for polar coordinates. Depending on the argument, the files corresponding to the desired lens are loaded. The calculation process for all lenses is the same. The script `plot.py` is written in Python and is used to visualize the trajectories of rays and fronts of the waves.

Consider, for example, the script `cartesian1.jl`, briefly explaining the Julia syntax. At the very beginning of the script the variables and functions are selectively imported from files.

```
include("src/RK.jl")
include("src/parameters.jl")

if length(ARGS) > 0
  ARGS[1] == "maxwell"  && include("src/maxwell.jl")
  ARGS[1] == "luneburg" && include("src/luneburg.jl")
else
  include("src/maxwell.jl")
end

include("src/eikonal.jl")
```

Julia is a JIT-compiled language, so it considers the speed of a compiled language with the flexibility of an interpreted one. In this case, it allows us to include the necessary code during the program execution and we get versions of the `n(x,y)` and ∂`n(x,y)` functions for Maxwell and Luneberg lens depending on which command-line argument is passed when the program is run. The operator `&&` allows you write conditional execution in the short form. So, the code from the maxwell.jl file will be enabled only if the first conditional expression is true (if `maxwell` argument is passed).

Then follows the process of calculation. As described above, we have to solve the ODE system for the set of initial values. Each solution gives the coordinates (x, y) of the corresponding ray. The calculation results for each ray are stored in a separate file.

```
isdir(dir) ? true : mkdir(dir)

# x_0, y_0, p1_0, p2_0
xn_0 = Vector{Float64}(4)
i = 0
# The point source. Rays are radiated at an angle alpha
for alpha in linspace(alpha_min, alpha_max, 50)
  i = i + 1
  file = open("./$(dir)/data$i.txt", "w")
  xn_0[1:2] = [x_0, y_0]
  xn_0[3:4] = [cos(alpha)*n(x_0, y_0), sin(alpha)*n(x_0, y_0)]

  tn, xn = RK.RKp6n1(eikonal, xn_0, h, t_start, t_stop)

  for (t, x, y) in zip(tn, xn[:, 1], xn[:, 2])
    write(file, "$t,$x,$y\n")
  end

  close(file)
end
```

3.3. Visualization of trajectories and fronts

Since the solution of the ODE system gives the trajectory of a single ray, visualization of rays trajectories is a simple task. The solutions for each system are saved as three columns of data: t, x and y. To visualize the ray it is sufficient to read columns x and y and and plot them.

A more complex task is to visualize the wave fronts. Geometrically, the wave front is formed from points of the beam corresponding to the same time. Thus, to get all the points of the front, it is necessary to consider the coordinates x and y corresponding to the same time $t$from each file with ODE solutions. Since we used the Runge–Kutta method with a constant step, the same time in different files corresponds to the same line number. The following code fragment shows how a list containing the points of the wave fronts is formed.

```
# The lines_num variable contains the number of lines in the file
levels = [[] for i in range(lines_num)]
# Sequentially open all files for reading
for fname in fnames:
  with open(fname) as f:
    for li, line in enumerate(f.readlines()):
      levels[li].append([float(s) for s in line.strip('\n').split(',')])
levels = np.array(levels)
```

As a result, we form the nested list, each element containing all the points of the corresponding front.

4. Conclusions

The paper presents the description of the numerical solution of the eikonal equation for the case of Luneburg and Maxwell lenses. The results are visualized as trajectories of rays passing through lenses and as fronts of electromagnetic waves.

Acknowledgments

The work is partially supported by Russian Foundation for Basic Research (RFBR) grants No 16-07-00556. Also the publication was prepared with the support of the "RUDN University Program 5-100".

References

1. M. Born, E. Wolf, Principles of Optics: Electromagnetic Theory of Propagation, Interference, and Diffraction of Light, 7th Edition, Cambridge University Press, Cambridge, 1999.
2. J. A. Stratton, Electromagnetic Theory, MGH, 1941.
3. L. D. Landau, E. M. Lifshitz, The Classical Theory of Fields, 4th Edition, Course of Theoretical Physics. Vol. 2, Butterworth-Heinemann, 1975.
4. L. D. Landau, E. M. Lifshitz, L. P. Pitaevskii, Electrodynamics of Continuous Media, 2nd Edition, Course of Theoretical Physics. Vol. 8, Butterworth-Heinemann, 1984.
5. H. Bruns, Das Eikonal, Vol. 35, S. Hirzel, Leipzig, 1895.
6. F. Klein, Über das Brunssche Eikonal, Zeitscrift für Mathematik und Physik 46 (1901) 372–375.
7. W. Jeong, R. Whitaker, A Fast Eikonal Equation Solver for Parallel Systems, SIAM conference on ... 84112 (2007) 1–4.
8. R. Kimmel, J. A. Sethian, Computing Geodesic Paths on Manifolds, Proceedings of the National Academy of Sciences 95 (15) (1998) 8431–8435. arXiv:1011.1669v3, doi:10.1073/pnas.95.15.8431.

9. H. Zhao, A Fast Sweeping Method for Eikonal Equations, Mathematics of Computation 74 (250) (2004) 603–628. doi:10.1090/S0025-5718-04-01678-3.
10. G. Beliakov, Numerical Evaluation of the Luneburg Integral and Ray Tracing, Applied Optics 35 (7) (1996) 1011–1014. doi:10.1364/AO.35.001011.
11. P. A. Gremaud, C. M. Kuster, Computational Study of Fast Methods for the Eikonal Equation, SIAM Journal on Scientific Computing 27 (6) (2006) 1803–1816. doi:10.1137/040605655.
12. S. Bak, J. McLaughlin, D. Renzi, Some Improvements for the Fast Sweeping Method, SIAM Journal on Scientific Computing 32 (5) (2010) 2853–2874. doi:10.1137/090749645.
13. D. S. Kulyabov, A. V. Korolkova, L. A. Sevastianov, M. N. Gevorkyan, A. V. Demidova, Geometrization of Maxwell's Equations in the Construction of Optical Devices, in: V. L. Derbov, D. E. Postnov (Eds.), Proceedings of SPIE. Saratov Fall Meeting 2016: Laser Physics and Photonics XVII and Computational Biophysics and Analysis of Biomedical Data III, Vol. 10337 of Proceedings of SPIE, SPIE, 2017, pp. 103370K1–7. doi:10.1117/12.2267959.
14. D. S. Kulyabov, A. V. Korolkova, L. A. Sevastianov, M. N. Gevorkyan, A. V. Demidova, Algorithm for Lens Calculations in the Geometrized Maxwell Theory, in: V. L. Derbov, D. E. Postnov (Eds.), Saratov Fall Meeting 2017: Laser Physics and Photonics XVIII; and Computational Biophysics and Analysis of Biomedical Data IV, Vol. 10717 of Proceedings of SPIE, SPIE, Saratov, 2018, pp. 107170Y–1–6. doi:10.1117/12.2315066.
15. R. K. Luneburg, Mathematical Theory of Optics, University of California Press, Berkeley & Los Angeles, 1964.
16. S. P. Morgan, General Solution of the Luneberg Lens Problem, Journal of Applied Physics 29 (9) (1958) 1358. doi:10.1063/1.1723441.
17. J. A. Lock, Scattering of an Electromagnetic Plane Wave by a Luneburg Lens I Ray Theory, Journal of the Optical Society of America A 25 (12) (2008) 2971. doi:10.1364/JOSAA.25.002971.
18. J. A. Lock, Scattering of an Electromagnetic Plane Wave by a Luneburg Lens II Wave Theory, Journal of the Optical Society of America A 25 (12) (2008) 2980. doi:10.1364/JOSAA.25.002980.
19. J. C. Maxwell, Solutions of Problems (prob. 3, vol. VIII, p. 188), The Cambridge and Dublin mathematical journal 9 (1854) 9–11.
20. A. Joshi, R. Lakhanpal, Learning Julia, Packt Publishing, 2017.