Complex Ray Tracing and Cross-Beam Energy Transfer for Laser-Plasma Simulations

Alan Tu

Pittsford Sutherland High School

Pittsford, New York

Advisor: Dr. Adam Sefkow

Laboratory for Laser Energetics

University of Rochester

Rochester, New York

July 2019

ABSTRACT

A ray-tracing code was developed that propagates laser beams by representing them as bundles of rays and then evolving the rays in time according to a set of differential equations. These equations account for the dispersion relation of the electromagnetic waves and the density profile of the background plasma. The energy deposited by the beams, as well as the beam intensities and electric fields, can be calculated and plotted onto a grid. Cross-beam energy transfer (CBET) occurs when laser beams overlap in a plasma and was implemented into the program for two interacting beams. This new program performs the CBET calculation faster than current programs in use. Furthermore, an alternative ray-tracing method was investigated, namely, complex ray tracing, which represents a laser beam with only five rays. The results are identical to those of the standard ray tracing but are achieved faster and can model additional effects such as diffraction and interference. In the future, this work will be implemented into the 3-D hybrid fluid-kinetic code *TriForce*.

I. INTRODUCTION

Current experiments being done on OMEGA involve shooting up to 60 laser beams at a target in an attempt to achieve nuclear fusion. This method is called direct-drive inertial confinement fusion (ICF).[1] The target is an ~l-mm-diam cryogenic capsule that contains a mixture of deuterium and tritium fuel. The energy from the laser beam ablates the outer shell of the capsule, causing the inner fuel to implode by Newton's third law; because of its extremely high temperature and pressure, the ablating material becomes a very hot ionized gas known as plasma.[1] Light propagation in plasma is quite complex since electromagnetic waves can interact with the freely floating electrons of a plasma. Accordingly, many phenomena related to laser–plasma interactions are not well understood.

Ray-tracing algorithms are important in designing and interpreting ICF experiments because they can predict the quantity and location of energy deposited by the laser beam. Based on these simulations, the optimal pointing of laser beams, as well as their respective intensities, can be determined and tested. Ray-tracing models are currently used at the Laboratory for Laser Energetics (LLE) in the hydrodynamic codes *LILAC* and *DRACO*. *LILAC* is a 1-D code that assumes spherical symmetry and consequently cannot take nonuniformities into account. *DRACO* is a 2-D code that assumes azimuthal symmetry and can model some types of perturbations. Both *LILAC* and *DRACO* are fluid-based models, which means that particle velocity distributions are assumed to follow a Maxwell–Boltzmann distribution at each point in the plasma and can be described by a single temperature. Therefore, the fluid model does not follow individual particles to determine the general behavior of the plasma.

TriForce is a 3-D hybrid fluid-kinetic code for ICF and high-energy-density physics research. A pure kinetic model does follow individual particles and does not assume a form for

the particle velocity distribution; therefore, it can more accurately model a plasma that is not in equilibrium (e.g., when a laser beam creates hot electrons and waves). In exchange for greater accuracy, however, kinetic models use more computation time. *TriForce* is a hybrid code in the sense that it uses kinetic algorithms where necessary and fluid algorithms everywhere else.

In this work, preliminary algorithms were developed for use in *TriForce*. These algorithms include a standard ray-tracing code, in which rays simply propagate through plasma; a cross-beam energy transfer code, which models the interaction of two laser beams in resonance with the plasma; and a complex ray-tracing code, which may be a powerful alternative to standard ray tracing because it includes extra physics while requiring fewer rays to model a laser beam.

II. STANDARD RAY TRACING

Light acts as both a particle and a wave. The dispersion relation describes the relation between a wave's angular frequency and wavevector while traveling through a given medium. In this case, the medium is plasma, and the dispersion relation for high-frequency electromagnetic waves can be expressed as follows:[2]

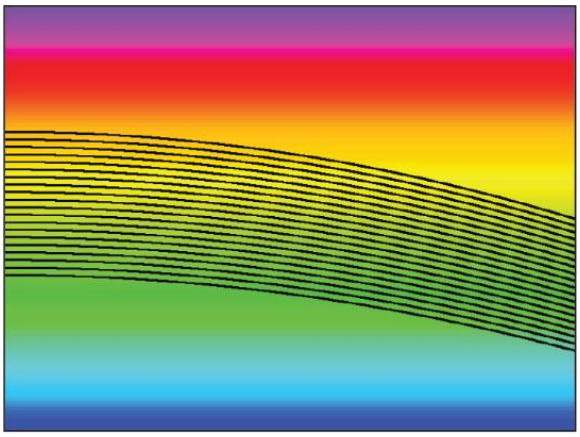
$$\omega^2 = \omega_{\mathrm{p}}^2 + c^2 k^2, \tag{1}$$

where ω is the wave frequency, c is the speed of light, k is the wave number, and ω_{p} is the plasma frequency defined by

$$\omega_{\mathrm{p}} = \sqrt{\frac{n_{\mathrm{e}} e_{\mathrm{c}}^2}{m_{\mathrm{e}} \varepsilon_0}}, \tag{2}$$

where n_{e} is the electron density, e_{c} is the elementary charge, m_{e} is the mass of an electron, and ε_0 is the permittivity of free space. The plasma frequency describes the rate of electron oscillations in the plasma.

An electron density gradient in the plasma will cause a ray to refract toward a region of lower electron density. The critical density [see Eq. (6)] is the density at which the laser light frequency is equal to the plasma frequency; when this happens, the k value in the dispersion equation must equal zero. Therefore, a ray cannot freely propagate in a region with a density above the critical density.



TC14826J1

FIG. 1. A bundle of rays propagating through a plasma. Note that the rays refract toward a region of lower electron density (shown in blue). Fewer rays have been traced for clarity; for best results, hundreds or thousands of rays must be used.

In standard ray tracing, a laser beam is statistically represented by a finite bundle of rays, as shown in Fig. 1. Each ray is traced individually and propagates according to the following differential equations:[3]

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{v}_{\mathrm{g}}, \tag{3}$$

$$\frac{\mathrm{d}\mathbf{v}_{\mathrm{g}}}{\mathrm{d}t} = -\frac{c^2 \nabla n_{\mathrm{e}}}{2 n_{\mathrm{c}}}, \tag{4}$$

where **x** is the position vector, t is time, and $\mathbf{v}_g$ is the group velocity vector, defined by

$$\mathbf{v}_g = \frac{\partial \omega}{\partial \mathbf{k}} = \frac{c^2 \mathbf{k}}{\omega}, \tag{5}$$

where ω is the angular frequency, **k** is the wave number vector, and n_c is the critical density, defined by

$$n_c = \frac{\omega^2 m_e \varepsilon_0}{e_c^2}. \tag{6}$$

Note that, by Eq. (1), the phase velocity ω/k is greater than the speed of light. By Eq. (5), however, the group velocity, which is the rate at which energy propagates, is less than c.

Because these equations have no closed analytic solution for arbitrary electron density, one must numerically evolve each ray in time using small discrete steps, while updating the position and velocity vectors at each time step. An example of tracing ten rays is shown in Fig. 2. The smaller the time step, the more accurate the solution.

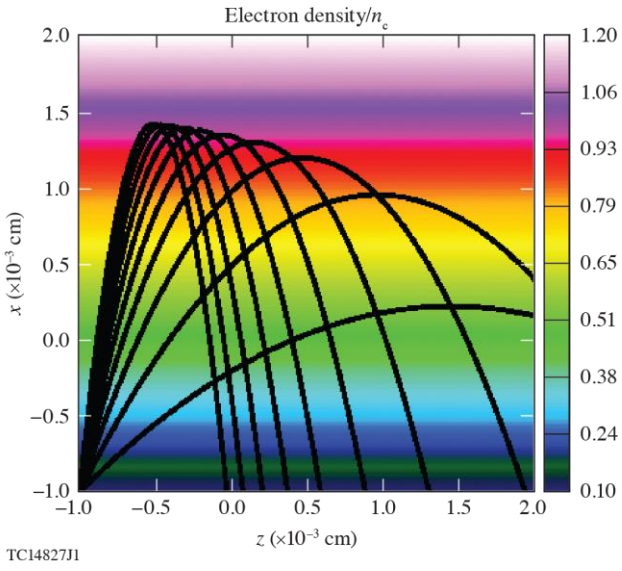


FIG. 2. The numerical trajectories of ten rays traced in a plasma with a linear density gradient. Note that the rays turn at the critical density and follow parabolic paths, as expected by theory.

As the rays travel through the plasma, they deposit energy in the plasma. The amount of energy deposited by a ray in an interval dt is given by the following relation:[4]

$$E_{\text{dep}} = \frac{v_{\text{ei}} n_{\text{e}}}{n_{\text{c}}} E_{\text{inc}} \text{d}t, \tag{7}$$

where E_{inc} is the energy of the ray incident on the interval, dt is the time step, and v_{ei} is the electron–ion collision rate given by

$$v_{\text{ei}} = \frac{n_{\text{e}} e^2}{m_{\text{e}}} \eta, \tag{8}$$

where η is the plasma resistivity.

We deposit energy using a first-order linear interpolation method, which is shown in Fig. 3. For each time step, we determine the location of the ray and find its nearest node. We calculate the distances dl and dm as shown in the figure, therefore dividing the cell into four quadrants. We split the energy deposited into the four surrounding cells based on the fractional area of each quadrant. This is more effective than zeroth-order interpolation, the so-called nearest neighbor method, because a ray that barely passes through a cell will not deposit all of its energy in that cell.

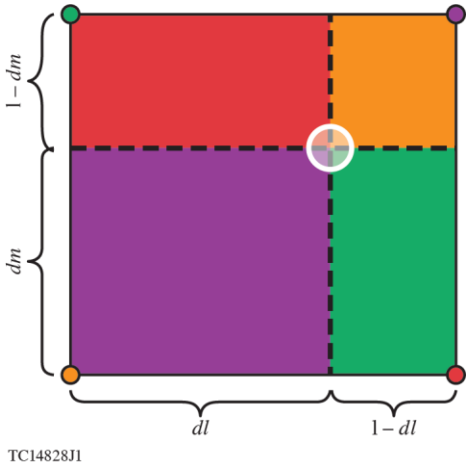


FIG. 3. First-order interpolation method for one grid cell.

Figure 4 shows results from this ray-tracing code when applied to an OMEGA capsule electron density distribution, as calculated by *LILAC*. The plot in Fig. 4(a) is from the ray-tracing code developed in this project; the plot in Fig. 4(b) is from a different ray-tracing code in use at LLE that has already been validated. The density profiles are the same but represented with different color maps. The ray trajectories match very closely.

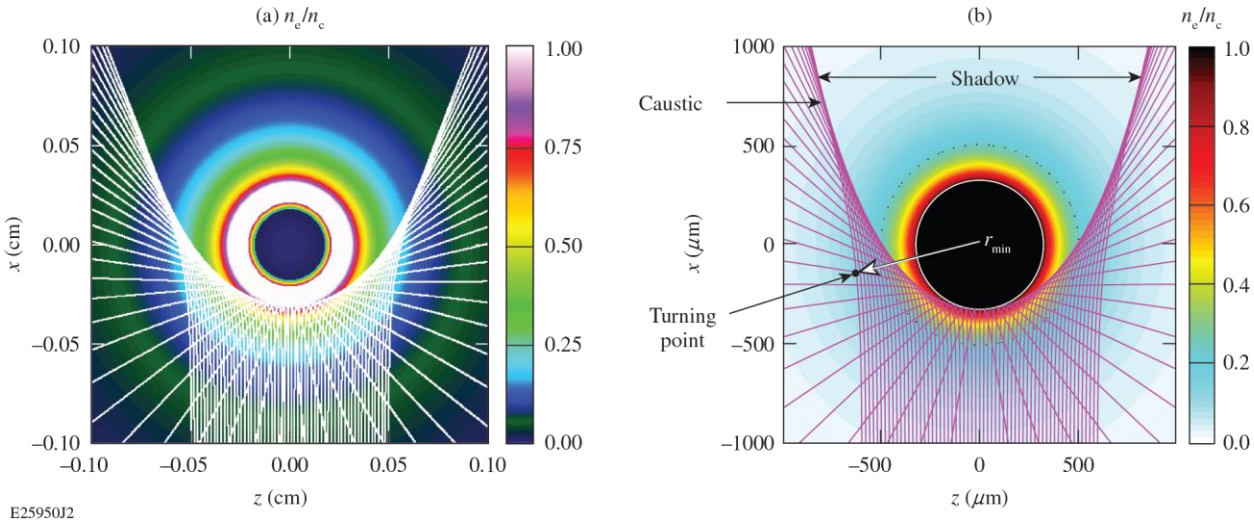


FIG. 4. (a) Results from the ray-tracing code developed in this work and (b) another ray-tracing code in use at LLE, both using an OMEGA capsule electron density profile. Notice the close similarities.

III. CROSS-BEAM ENERGY TRANSFER

Figure 5 shows two laser beams overlapping without any mutual interaction or energy deposition in the plasma. The electric-field contours are expressed in terms of a_0, a dimensionless quantity defined by

$$a_0 = e_c E / (m_e c \omega), \tag{9}$$

where E is the electric field. In this figure, the chosen width of the laser beam is arbitrary. To obtain the electric field on the grid, we let each ray carry electric field information, which is then "deposited" (using the first-order linear interpolation method shown in Fig. 3) in each cell along its trajectory.

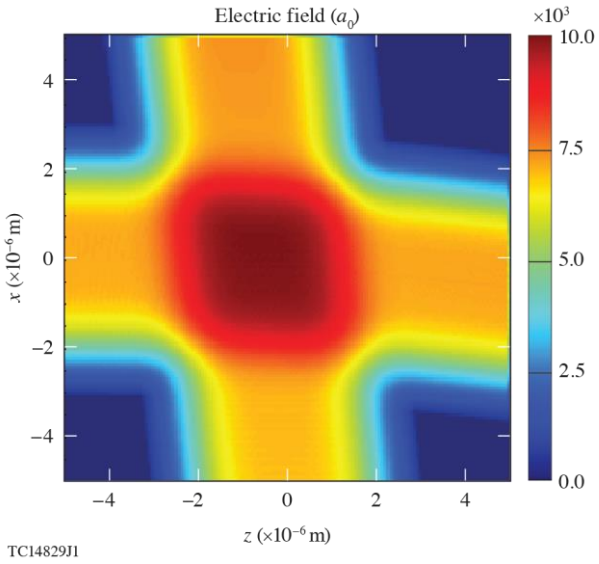


FIG. 5. Electric field of two intersecting laser beams that do not transfer energy between each other. One beam travels from bottom to top, and the other beam travels from left to right.

In reality, however, many complex effects occur when laser beams overlap with each other within a plasma. One of these effects is cross-beam energy transfer (CBET), which is driven by ion-acoustic waves, one of many collective modes that a plasma can exhibit. The plasma flow velocity causes a Doppler shift; therefore, two interacting beams will have slightly different frequencies relative to each other. Beat waves will result, and the acoustic waves will act as a grating to divert energy from one beam to another. Generally speaking, the beam traveling more "against" the bulk plasma flow will lose energy. In OMEGA implosions, the plasma flows radially outward and the rays are shot inward, so incoming rays will lose energy to outgoing rays, significantly decreasing energy deposition in the desired areas around the capsule. CBET may be responsible for up to a 50% decrease in hydrodynamic efficiency in OMEGA implosions.[5]

A simulation program was developed that models CBET for the simple case of two intersecting beams. To properly model CBET, a five-step process was used: (1) obtain electric fields at each point on the grid; (2) map the ray trajectories to the grid; (3) find all the

interactions (i.e., where the rays intersect); (4) calculate the gain coefficients for the interactions; and (5) solve by iteration.

As mentioned, we have already determined the electric field on the grid. Now let us take a closer look at the following steps:

A. Mapping ray trajectories and finding interactions

The determination of ray intersections requires finding and saving the locations where individual rays cross the half-grid, as shown in Fig. 6, which has a reduced grid size of 10 × 10 for clarity.

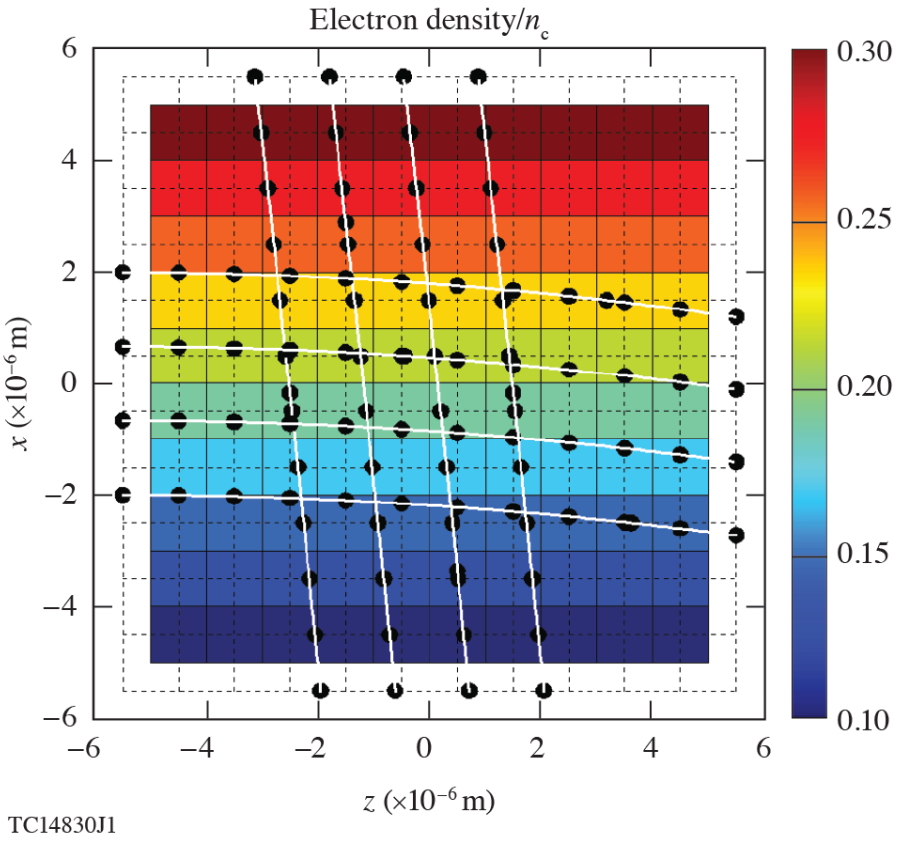


FIG. 6. Keeping track of ray intersections with the grid. The solid lines define the normal grid, and the dashed lines define the half-grid. The half-grid has been used because it allows a node (the place where two normal grid lines intersect; also, where we store electric field and electron density information) to be at the center of each cell. The rays are the white lines and their intersections with the half-grid are the black dots.

For each ray, we store an array of the ray's coordinates along its trajectory. At each time step, we check if the ray has crossed a grid line. For example, if there is a grid line at x = 0.100 cm and the ray's x coordinate changes from 0.098 to 0.101, then the ray would have crossed a grid line. We interpolate to find the coordinates of the crossing and save them as well; these coordinates will be used later in the calculation of energy transfer. A multidimensional array keeps track of the cells passed through by each ray.

Two rays are said to intersect if they both pass through the same grid cell, even if they do not physically intersect. For each intersection between rays from different beams, we calculate the amount of energy transfer, which is explained below.

B. Calculating gain coefficients for interactions and updating energies

For each interaction, we calculate the gain coefficient from the following formula:[6]

$$\left(L_{\mathrm{s}}^{ijkl}\right)^{-1} = \frac{e^2 \left|E_{k0}\right|^2}{4 m_{\mathrm{e}} c \omega_{ij} k_{\mathrm{B}} T_{\mathrm{e}} \left(1 + 3T_{\mathrm{i}}/ZT_{\mathrm{e}}\right)} \frac{n_{\mathrm{e}}}{n_{\mathrm{c}}} \frac{\omega_{\mathrm{s}}}{v_{\mathrm{ia}}} P\left(\eta_{ijkl}\right), \tag{10}$$

$$P(\eta) = \frac{\left(v_{\mathrm{ia}}/\omega_{\mathrm{s}}\right)^2 \eta}{\left(\eta^2 - 1\right)^2 + \left(v_{\mathrm{ia}}/\omega_{\mathrm{s}}\right)^2 \eta^2}, \tag{11}$$

$$\eta_{ijkl} = \frac{\omega_{kl} - \omega_{ij} - \left(\mathbf{k}_{kl} - \mathbf{k}_{ij}\right) \cdot \mathbf{u}}{\omega_{\mathrm{s}}}, \tag{12}$$

where L_{s}^{ijkl} is the laser absorption length (*ij* refers to the i^{th} ray at the j^{th} location on its path, and likewise for *kl*), *e* is the elementary charge, E_{k0} is the initial electric field of the pump ray, m_{e} is the electron mass, *c* is the speed of light, ω_{ij} and ω_{kl} are the frequencies of the seed beam and the pump beam, respectively, k_{B} is the Boltzmann constant, T_{e} is the electron temperature, T_{i} is the ion temperature, *Z* is the ionization state, n_{e} is the electron density, n_{c} is the critical density, v_{ia} is the ion-acoustic wave energy damping rate, $\mathbf{k}_{ij}$ and $\mathbf{k}_{kl}$ are the seed and pump ray vectors, respectively, **u** is the plasma flow velocity, and ω_{s} is the acoustic frequency given by

$$\omega_{\mathrm{s}} = k_{\mathrm{iaw}}\, c_{\mathrm{s}}, \tag{13}$$

where c_{s} is the ion-acoustic wave speed given by

$$c_{\mathrm{s}} = \sqrt{\frac{k_{\mathrm{B}}\left(ZT_{\mathrm{e}} + 3T_{\mathrm{i}}\right)}{m_{\mathrm{i}}}}, \tag{14}$$

where m_{i} is the ion mass, and

$$k_{\mathrm{iaw}} = \left|\mathbf{k}_{kl} - \mathbf{k}_{ij}\right|. \tag{15}$$

We have access to the electric field value in the current cell because we saved it to the grid at an earlier step. The vectors $\mathbf{k}_{ij}$ and $\mathbf{k}_{kl}$ can be deduced simply from the coordinates of each ray's intersection with the two grid lines of the cell, as shown in Fig. 7.

Finally, the energy transfer is determined by the following formula:[6]

$$W_{i,\,j+1} = W_{ij}\exp\left(\frac{s_{ij}W_{kl}}{L_{\mathrm{s}}^{ijkl}\sqrt{\varepsilon}}\right), \tag{16}$$

where W_{ij} is the seed ray's energy (normalized to the incident energy) in the current cell, $W_{i,j+1}$ is the seed ray's energy in the next cell, s_{ij} is the distance traveled by the seed ray in the current cell, W_{kl} is the pump ray's energy in the current cell, L_{s}^{ijkl} is the gain coefficient for the interaction in the current cell, and $\varepsilon = 1 - n_{\mathrm{e}}/n_{\mathrm{c}}$.

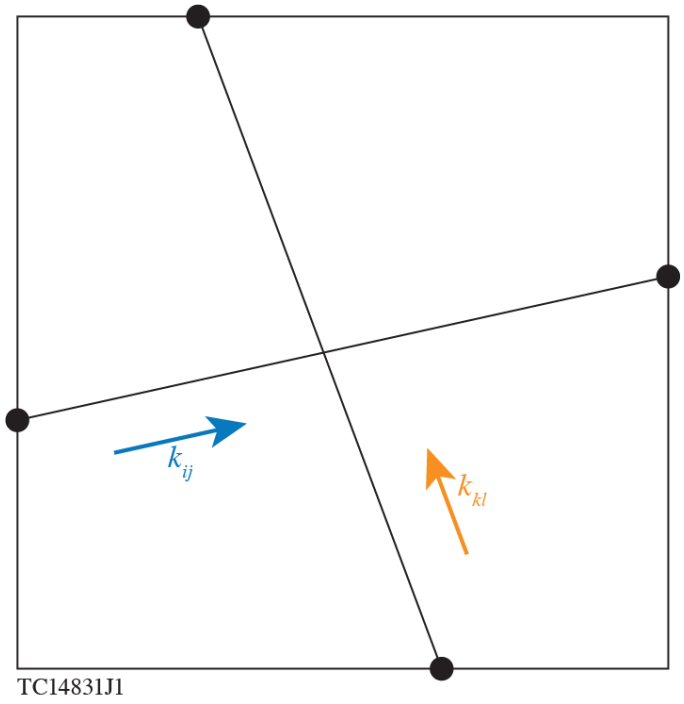


FIG. 7. The $\mathbf{k}_{ij}$ and $\mathbf{k}_{kl}$ vectors. One grid cell is shown.

Once we determine the energy transfer for a single intersection, we must propagate the energy change to all downstream cells. After doing this for all possible ray intersections, we iterate the process if necessary, a step that is discussed in more detail below. In the case where a ray from one beam intersects multiple rays from another beam in a given cell, we calculate the energy transfer using the vector of only one of the crossing rays while using the total intensity of all the crossing rays.

C. Results

After writing this program, we compared our results to those from a CBET program developed by R. K. Follett[6] that uses the same equations [Eqs. (10)–(12)] but different numerical algorithms. The comparison is shown below in Fig. 8.

The ray-tracing code developed matches Follett's CBET results very closely; in addition, the calculations are performed 10× faster. At higher intensities, however, our results differ significantly from Follett's. This is most likely because we have not yet implemented the final step in the CBET process which is solving by iteration. Each energy update affects CBET interactions downstream, so it is critical that we calculate the interactions in the correct order that would occur physically. This is not always possible, however, especially in more-complicated cases, so we perform iteration; that is, we recalculate all the CBET interactions with our newly modified values and repeat this until the change in values becomes very small. Essentially, we get closer and closer to the true answer. Future work will implement iteration into this process.

IV. COMPLEX RAY TRACING

Complex ray tracing, as described by Harvey *et al*.,[7] is a powerful alternative to standard ray tracing. In complex ray tracing, a beam is represented by only five rays (in 2-D): a chief ray (a.k.a., base ray), two waist rays, and two divergence rays, as shown in Fig. 9.

The electric field or intensity at any point can be calculated by finding the distance between the point and three of the rays along a line perpendicular to the chief ray, as shown in Fig. 10.[8] The method therefore assumes a functional relation between the rays themselves, whereas all rays are independent in standard ray tracing. To find the (normalized) electric field at

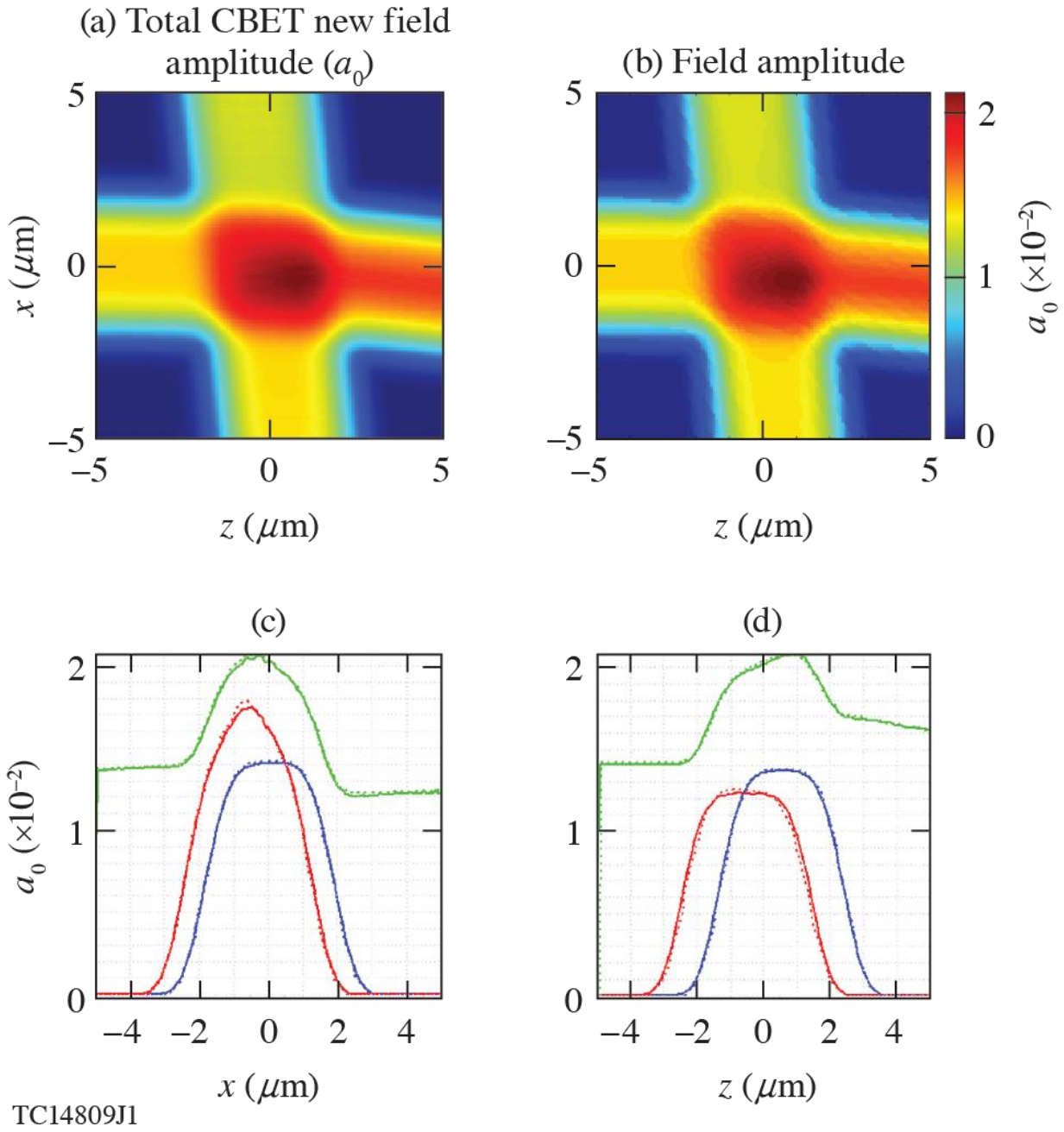


FIG. 8. [(a),(b)] Comparison of Follett's code with the developed code: (a) Our results and (b) Follett's results. Energy from the upward-traveling beam has been transferred to the rightward-traveling beam (cf., Fig. 5). [(c),(d)] Our results are shown as solid lines and Follett's results are shown as dotted lines. The blue, green, and red lines show the electric-field profiles taken at the minimum z (or x) value, midpoint value, and maximum value, respectively.

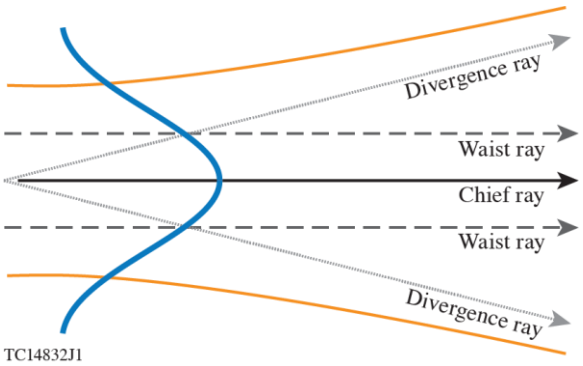


FIG. 9. The rays needed for complex ray tracing. The solid blue line is the transverse profile of the beam intensity.

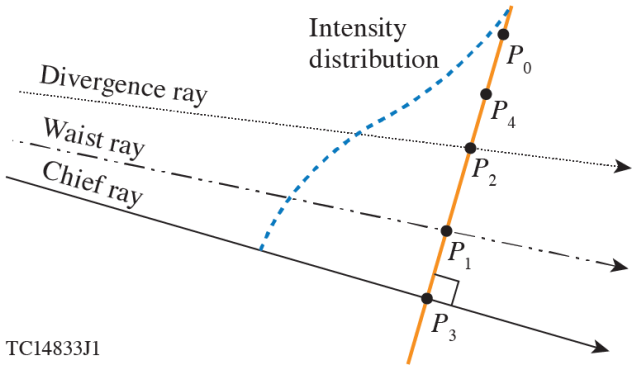


FIG. 10. Determining the intensity at an arbitrary point. Adapted from [8].

P_0, we simply calculate the distances from P_3 to P_0, P_1, and P_2 along the orange line (which we denote as p_0, p_1, and p_2, respectively) and then use the following formula:[8]

$$E(p_0) = \frac{w_0}{p_4} \exp\left[-\left(\frac{p_0}{p_4}\right)^2\right], \tag{17}$$

where

$$p_4 = \sqrt{p_1^2 + p_2^2}. \tag{18}$$

The intensity can then be calculated by squaring the electric field.

A. Gaussian Beams in the Diffraction Limit

Consider the case where the power distribution of a laser can be described using the Gaussian function. Such a beam remains Gaussian throughout its path, and the Gaussian function

solves the wave equation, making it an ideal candidate for modeling purposes. The transverse profile of the intensity of a Gaussian beam can be described as follows, using *r*–*z* coordinate axes:[7]

$$I(r,z) = \frac{2P}{\pi w^2(z)} \exp\left\{-2\left[\frac{r}{w(z)}\right]^2\right\}, \tag{19}$$

where

$$p = \frac{\pi w_0^2}{2} I_0, \tag{20}$$

I_0 is the initial beam intensity and w_0 is the initial beam waist, and

$$w(z) = w_0\sqrt{1+\left(z/z_{\mathrm{R}}\right)^2}, \tag{21}$$

where z_{R} is the Rayleigh range defined by

$$z_{\mathrm{R}} = \frac{\pi w_0^2}{\lambda}. \tag{22}$$

The intensity function is plotted in Fig. 11(a).

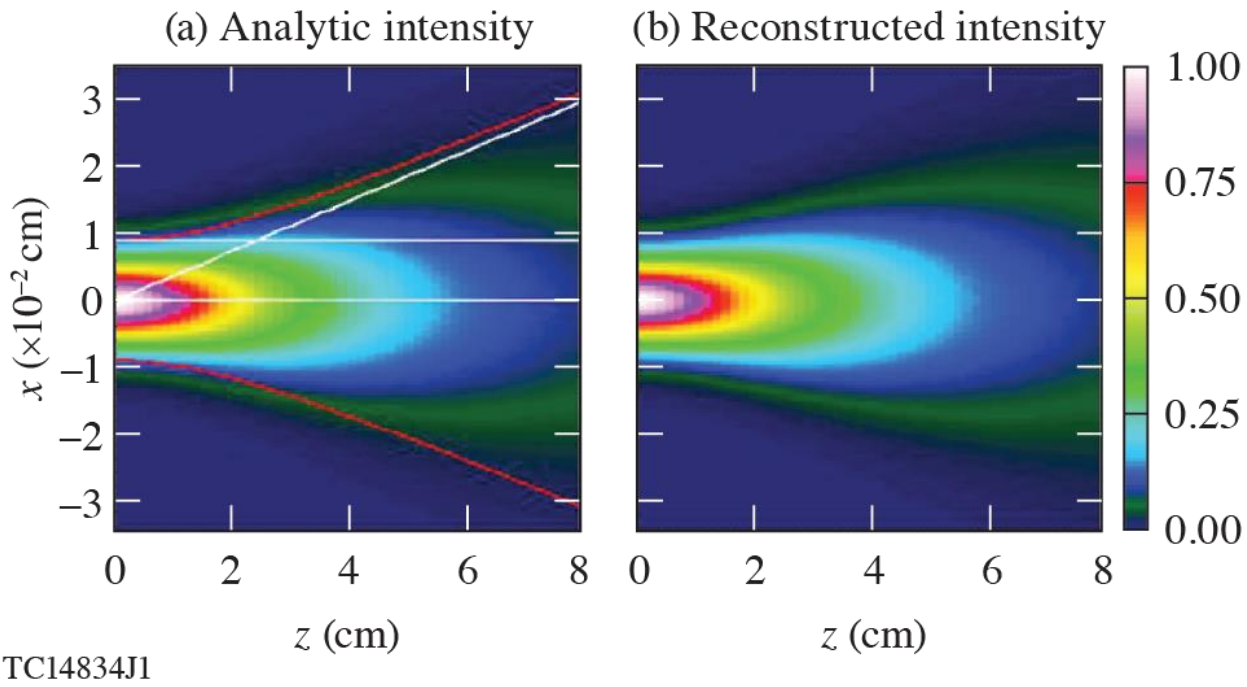


FIG. 11. Analytic formula versus complex ray tracing for a diffraction-limited Gaussian beam in vacuum. The white lines are three of the complex rays (chief, waist, and divergence) and the red lines denote the beam waist.

To model a Gaussian beam via standard ray tracing, we must statistically weight the powers of each ray, with the rays closest to the center receiving the highest powers, in the case that the rays are spaced equally apart (however, standard ray tracing does not properly model the behavior near the focal plane, as it does not include diffraction). The power distribution is Gaussian and is given by the following, where x is the ray position relative to the beam center:

$$P = \exp\left\{-\left[\left(\frac{x}{\sigma}\right)^2\right]^{n/2}\right\}. \tag{23}$$

A standard Gaussian beam has $n = 2$, whereas the OMEGA laser uses $n = 4$ to 5, a super-Gaussian.

Equations (17) and (18) were used to create a complex ray-tracing program. The test case of a Gaussian beam in vacuum is shown in Fig. 11; this plot was easy to create because all the perpendicular lines were vertical and lined up with the grid.

When a beam refracts (such as in plasma), however, or its propagation direction does not line up with the grid axes, then a different method must be used. Two methods were attempted, and the results are discussed below.

B. Cell-by-Cell Approach

In this method of complex ray tracing, we trace all five rays at the beginning and then go cell by cell along the trajectories to calculate the intensity for all the cells. To do this, we must trace the perpendiculars from each cell (P_0) to the chief ray so that we may determine distances along this perpendicular. Figure 12 shows the perpendiculars drawn in red. Note that in this figure, P_0 is taken to be at the corner of each cell, rather than the center; as the number of cells increases, this discrepancy is negligible.

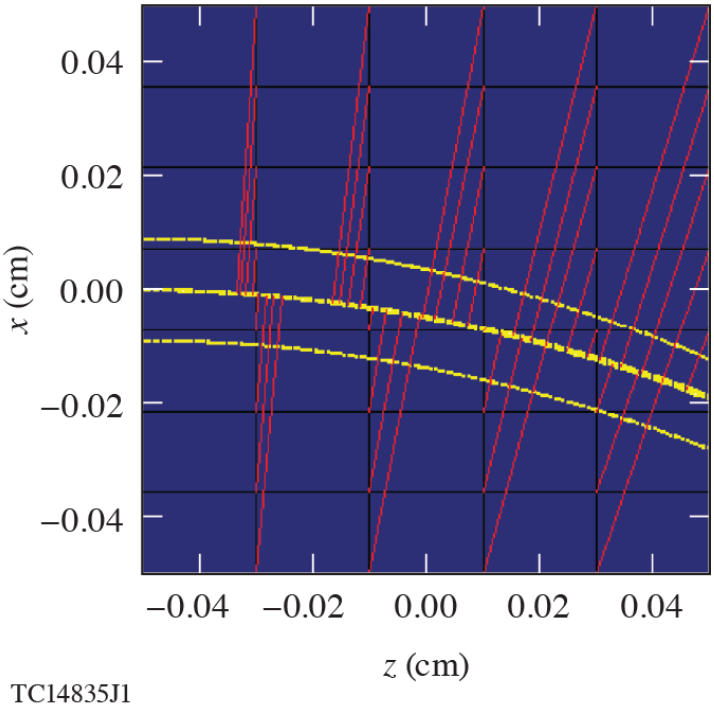


FIG. 12. Perpendicular lines (shown in red) drawn from each cell to the chief ray. The yellow lines show the five complex rays; at this scale, the chief ray and divergent rays are extremely close together. The colors of the density profile have been removed for clarity.

The following method is used to find the perpendiculars. Our first task is to determine the point on the chief ray P_3 that defines the perpendicular line. To do this, we move along the time

steps of the chief ray and for each point P on the chief ray's trajectory, we calculate the chief ray's directional vector **k** as well as the vector connecting P_0 to P. We take the dot product of these two vectors and save it to compare with the next point P. Two vectors are perpendicular if their dot product is zero; since the two vectors in question are never truly perpendicular, we wish to find the P that makes them closest to perpendicular; that is, the P that produces a dot product closest to zero. We stop the search when the dot product changes from negative to positive or vice versa (i.e., "crosses" zero) and then choose the one with the smaller absolute value.

While this method is very accurate, it is very slow when the number of cells becomes large. In addition, it requires us to move along each of the rays for each cell calculation. To fix these issues, we chose a new approach.

C. Update Outward Along the Chief Ray

This improved algorithm starts by tracing the waist and divergence rays; then, while the chief ray is traced, we update the intensities outward from the chief ray and use the energy deposition method again to "deposit" intensity. For each point on the chief ray, we find the equation of the line perpendicular to the chief ray's **k** vector and determine where it intersects the waist and divergence rays. To do this, we find the waist/divergence ray's **k** vector at each time, thereby creating another linear equation (the equation of the line tangent to the ray's trajectory), which we solve in tandem with the equation of the line perpendicular to the chief ray. If the solution to this system of equations is located on the waist/divergence ray, we have found the correct point of intersection and simply determine its distance from the chief ray. To speed up the process, we start with the point at the same time step as the current chief ray's time step and then move outward.

Once the distances p_1 and p_2 are determined, we return to the chief ray and move outward along the perpendicular line. At each point, we calculate the intensity using Eq. (17), and then deposit that intensity using the energy deposition algorithm from above. Therefore, each cell may receive multiple intensity contributions which are summed at the end. For the case of a Gaussian beam refracting in plasma, as shown in Fig. 13, this method worked quickly and accurately. It ran 4× faster than the standard ray-tracing method and produced a smoother plot, which is also important for simulation accuracy. If we were to make a comparison based on the quality of the solution, we would have to use many more rays in the standard case, so the difference in speed would be even greater.

With these tools in hand, we attempted to reproduce a famous interference pattern in order to demonstrate the advanced capability of complex ray tracing.

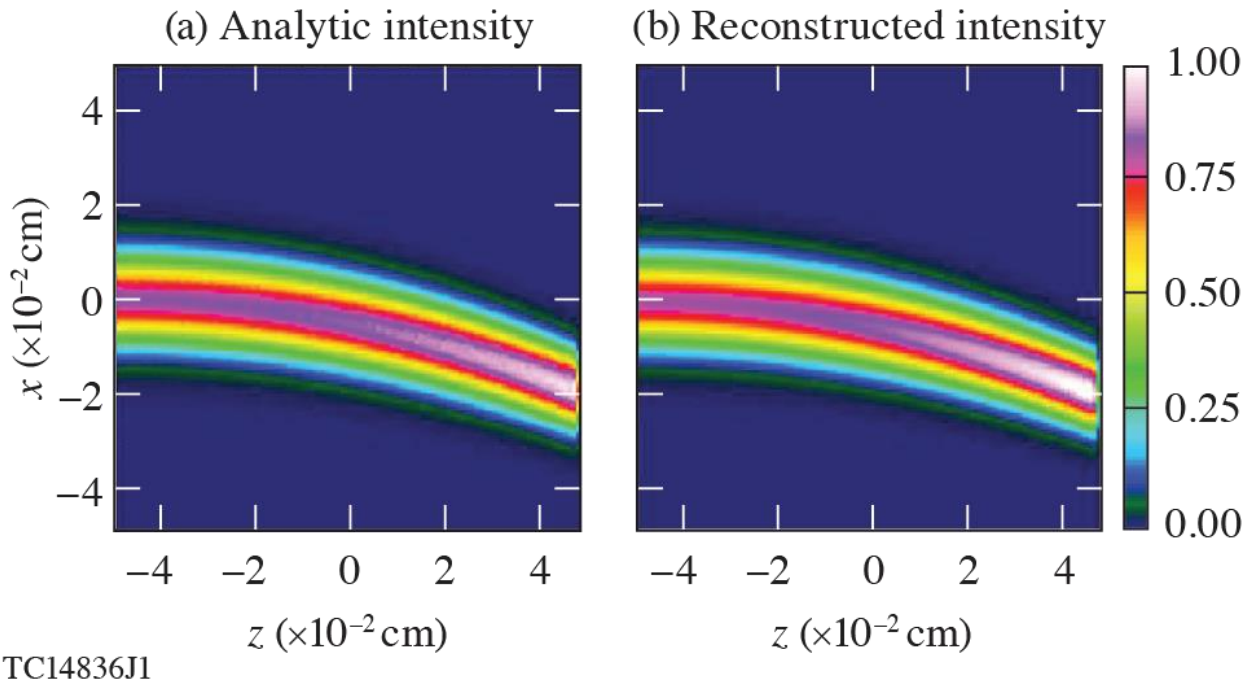


FIG. 13. (a) Standard ray tracing versus (b) complex ray tracing.

D. The Double-Slit Experiment

In 1801, Thomas Young performed the first double-slit experiment, in which he passed light through two small apertures, producing interference patterns on the viewing board. This showed that light has both particle and wave characteristics. Standard ray tracing cannot reproduce this phenomenon because it does not account for the wave nature of light and therefore does not include its phase.

With complex ray tracing, it is possible to include the phase of a laser beam by adding a phase term to Eq. (17):[2]

$$E\left(P_0\right)=\frac{w_0}{p_4}\exp\left[-\left(\frac{p_0}{p_4}\right)^2\right]\cos\left(-kz\right), \tag{24}$$

where

$$k=\frac{2\pi}{\lambda}. \tag{25}$$

We modeled Young's experiment by shooting two point sources spaced a small distance apart. Each point source was composed of 17 complex "beamlets," each of which was shot in a different direction to span a given angle. We time-averaged the intensities over a period to obtain an accurate cycle-averaged plot, shown in Fig. 14.

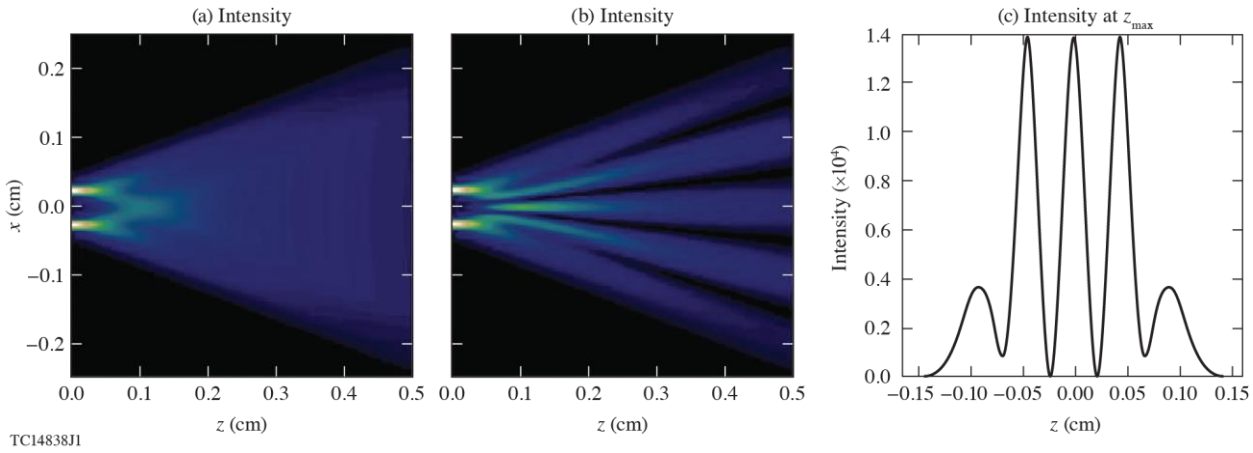


FIG. 14. Interference pattern for Young's double-slit experiment with complex ray tracing. (a) The point sources contain no phase information; therefore, there is no interference. (b) The inclusion of phase information makes the interference clearly visible. (c) The intensity profile taken at the maximum *z* value. The beam waist is 0.025 cm and the wavelength is 0.01 cm.

V. CONCLUSION

In this work, several programs were developed for implementation into the 3-D hybrid-fluid kinetic code *TriForce*. These programs include a standard ray-tracing program that traces bundles of rays to represent beams and can determine energy deposition, intensity, and electric field; a CBET feature that accurately models energy transfer between two beams in plasma; and a complex ray-tracing program that can perform the same tasks as standard ray tracing (but faster) as well as model interference. Attention was paid to speed and efficiency in the algorithms since they must run in *TriForce* without slowing it down. *TriForce* will be used in the future to design and interpret ICF experiments and other high-energy-density physics research experiments.

ACKNOWLEDGMENTS

I thank Dr. Adam Sefkow, my advisor, for his steadfast guidance throughout this project, and for teaching me what real research is all about. His knowledge and vision inspired me to tackle even the most difficult of problems. I also thank Dr. Russell Follett for sharing with me his CBET expertise. Many thanks to Dr. Stephen Craxton, the program director, for offering me this unique opportunity to perform research at the Laboratory for Laser Energetics. Finally, I would like to give a shoutout to my fellow interns (especially those stationed in the Annex) for their good sportsmanship.

REFERENCES

1. R. S. Craxton, K. S. Anderson, T. R. Boehly, V. N. Goncharov, D. R. Harding, J. P. Knauer, R. L. McCrory, P. W. McKenty, D. D. Meyerhofer, J. F. Myatt, A. J. Schmitt, J. D. Sethian, R. W. Short, S. Skupsky, W. Theobald, W. L. Kruer, K. Tanaka, R. Betti, T. J. B. Collins, J. A. Delettrez, S. X. Hu, J. A. Marozas, A. V. Maximov, D. T. Michel, P. B. Radha, S. P. Regan, T. C. Sangster, W. Seka, A. A. Solodov, J. M. Soures, C. Stoeckl, and J. D. Zuegel, Phys. Plasmas **22**, 110501 (2015).
2. E. R. Tracy, A. J. Brizard, A. S. Richardson, and A. N. Kaufman, *Ray Tracing and Beyond: Phase Space Methods in Plasma Wave Theory* (Cambridge University Press, Cambridge, England, 2014).
3. T. B. Kaiser, Phys. Rev. E **61**, 895 (2000).
4. F. F. Chen, *Introduction to Plasma Physics and Controlled Fusion*, 3rd ed. (Springer International Publishing, Cham, Switzerland, 2016).
5. J. F. Myatt, R. K. Follett, J. G. Shaw, D. H. Edgell, D. H. Froula, I. V. Igumenshchev, and V. N. Goncharov, Phys. Plasmas **24**, 056308 (2017).
6. R. K. Follett, J. G. Shaw, J. F. Myatt, V. N. Goncharov, D. H. Edgell, D. H. Froula, and J. P. Palastro, Phys. Rev. E **98**, 043202 (2018).
7. J. E. Harvey, R. G. Irvin, and R. N. Pfisterer, Opt. Eng. **54**, 035105 (2015).
8. H. Yu, G. Rossi, A. Braglia, and G. Perrone, Appl. Opt. **55**, 6530 (2016).