Extension of the Modeling of Collisional Blooming and Straggling of the Electron Beam in the Fast Ignition Scenario

Roy Hanna
Williamsville South High School
Advisor: Jacques Delettrez

Laboratory for Laser Energetics
University of Rochester
Summer High School Research Program
2007

Abstract

When an electron beam is fired into an imploding target in order to initiate a fast-ignition-type propagating burn, spreading of the electrons occurs within the beam. The spread in the direction of motion (straggling) and the spread perpendicular to the direction of motion (blooming), as well as the energy lost as a result of these phenomena, can be defined by three equations. Previously, a program was written that added the effects of blooming, straggling, and energy deposition to an existing straight-line model. This model functioned by initially creating a single electron which was then split into several simulated electrons of different directions and energies as the probability of spreading increased. However, this method resulted in a too-large number of simulated electrons, consuming large amounts of computer memory. The program was amended by the addition of a protocol to combine simulated electrons that are in close proximity, effectively treating them as one entity and allowing for greater efficiency. This algorithm has been successful in shortening runtime and decreasing memory requirements and was tested for realistic implosion conditions in which the plasma density is nonuniform.

Introduction

Ignition in conventional inertial confinement fusion occurs when fusion products created in the central hot-spot enter surrounding high-density low-temperature fuel, initiating a propagating burn. An alternative method for achieving a propagating burn has been proposed. This entails firing a relativistic beam of electrons into the high-density cold fuel, initiating the propagating burn. The path of these beams moving through plasma must be modeled in order to understand the physics behind fast ignition.

Current simulations provide for straight-line electron movement in which electrons have no deviation from uniform, horizontal travel. Electrons undergo straggling (uneven penetration depths) and blooming (spreading of the electron beam) in reality due to collisions with plasma in the environment.[1,2] A model was recently developed to include the effects of blooming and straggling.[3] This report describes how this previously created model was improved to increase efficiency and accuracy as well as to allow for simulation of electron beams in sloped plasma fields.

Previously existing model

In this report a method for accounting for blooming and straggling is described, along with how that method was improved. As the electron beam travels through the plasma in the target, collisions occur between electrons and the plasma, causing changes in electron energy (due to energy deposition) and velocity. This is described by the following expression, which is used in the existing model:

$$\frac{dE}{ds} = -\frac{2\pi r_0^2 m_0 c^2 n_i Z}{\beta^2}\left[\ln\left(\frac{(\gamma-1)\lambda_D}{2\sqrt{2\gamma}r_0}\right)^2 + 1 + \frac{1}{8}\left(\frac{\gamma-1}{\gamma}\right)^2 - \left(\frac{2\gamma-1}{\gamma}\right)\ln 2 + \ln\left(\frac{1.123\beta}{\sqrt{2kT_e/m_0c^2}}\right)^2\right], \quad (1)$$

where r_0 is the classical electron radius, m_0 the electron mass, c the speed of light, n_i the background ion density, Z the background average ion charge, T_e the background electron temperature, k the Boltzmann constant, γ the relativistic energy divided by m_0c^2, β the relativistic velocity and λ_D is the Debye length. This equation determines the amount of energy lost per distance the electron traveled (dE/ds). Previously, blooming

and straggling were added to the initial model. Blooming is defined as the deviation perpendicular to the direction of movement and is described by the equation:

$$\langle y^2 \rangle = \frac{2}{3} \int_{E_0}^{E} \langle P_1(\cos\theta) \rangle \left(\frac{dE'}{ds} \right)^{-1} \left(\int_{E_0}^{E'} \frac{1 + 2\langle P_2(\cos\theta) \rangle}{\langle P_1(\cos\theta) \rangle} \left(\frac{dE''}{ds} \right)^{-1} dE'' \right) dE'. \qquad (2)$$

where $P_1(\cos\theta)$ and $P_2(\cos\theta)$ are Legendre polynomials, dE/ds is the amount of energy lost per distance the electron traveled, and E is the energy of the electron.

Straggling is defined as the deviation parallel to the direction of movement (slowing down) and is given by the equation:

$$\langle x^2 \rangle = \frac{2}{3} \int_{E_0}^{E} \langle P_1(\cos\theta) \rangle \left(\frac{dE'}{ds} \right)^{-1} \left(\int_{E_0}^{E'} \frac{1 - \langle P_2(\cos\theta) \rangle}{\langle P_1(\cos\theta) \rangle} \left(\frac{dE''}{ds} \right)^{-1} dE'' \right) dE', \qquad (3)$$

where $P_1(\cos\theta)$ and $P_2(\cos\theta)$ are Legendre polynomials, dE/ds is the amount of energy lost per distance the electron traveled, and E is the energy of the electron.

This model functioned by initiating a simulation with one electron, representing the entire electron beam. As the electron moved through the plasma field, the calculated probability that it had deviated (through either blooming or straggling) increased, until it reached such a level that the simulation split the electron into slightly offset three daughter electrons, which were each run separately (see Fig. 1). The program continued to function until all electrons had reached negligible energy levels. Splitting would occur millions of times before program completion, especially in more complex plasma densities, so memory consumption became a problem using this model. Also, the

existing model did not easily allow variable-density plasma fields to be simulated. Both of these problems had to be rectified to allow this model to effectively simulate realistic plasma conditions.

Procedure

A combining algorithm was added to the existing model in order to decrease memory consumption. The medium of movement had previously been divided into grids in order to facilitate movement procedures and to allow for varying plasma densities, so a subroutine was introduced to the program to combine electrons that shared a single grid square (Fig. 2). The model functioned by creating a queue of electrons, and running the simulation of each one individually until it split or ran out of energy; daughter electrons were added to the end of the queue, the current electron was deleted and the next began running. This created problems because only the current electron could be referenced or modified while the program was running – no nearby electrons could even be detected. It was therefore difficult to combine electrons. A modification was put in place to create a loop structure within the program procedure: instead of creating daughter electrons immediately, the data associated with them was saved in the grid squares where they were created. When the queue of electrons to simulate was exhausted, the program then processed each grid square, combining each electron contained therein and then creating a new electron from the new data, which was then placed in the queue. A rule was put in place so that sister electrons (those spawned from the same parent) could not combine (which would undo the effects of blooming and straggling). After being combined, electrons would continue to bloom and straggle until all energy was exhausted.

The original simulation had not been optimized or tested for use with variable-density-plasma environments. Sloped plasma fields could be put in place, but only through the use of equations to determine conditions in each grid square (so that experimental data could not be used to define each square; rather, conditions were limited to configurations that could be expressed using polynomials). The ability to simulate more realistic conditions was necessary. The initial environment generation subroutine was modified to allow data input, and another separate program was written to convert polar coordinate data about imploding targets to a format the simulation program could recognize (essentially a conversion to Cartesian coordinates). The program was tested with a variable-density-plasma field using data from a target implosion (Fig. 3).

Results

Through the use of the combining algorithm, runtime memory use was significantly decreased. The number of electrons stored in memory increased exponentially and without limit in the original program, but the number of electrons was capped at 2143 (a number found to balance accuracy and efficiency) in the new program. Figure 4 compares the number of electrons simulated in the two programs as a function of loops completed (a new loop was considered to have begun when the daughter electrons that were created by the initial electron of the previous loop, had begun running). It is obvious from this graph and accompanying data table that memory consumption in the new model was significantly lower than that of the old model; the program runs much more quickly. There is a loss in accuracy, but it is small enough that it can be discounted – depending on the exact settings used, the results are within a few percent of those from the original model.

Figure 5 depicts the output of the program in graphical form for a simulation of an electron beam moving through the variable-density plasma profile shown in Figure 3. The electron beam enters from the left side of the figure, and the deposition of energy on the grid is shown through the use of different colors. In the first 0.005 cm, there is little spread (because plasma density is low, resulting in little blooming and straggling), but high levels of energy deposition (because large numbers of electrons travel through that region). A sharp increase in the width of the energy deposition region (depicted in brighter colors) is evident at approximately 0.005 cm; this is the point where the electron beam encounters a quick increase in plasma density. This is in keeping with the precept that deviation is caused by collisions with plasma particles. However, there isn't a similar increase in spread evident at 0.0105 cm. This could be because there is so little energy left in the beam at that point that not much deviation occurs, but it could also result from a decrease in data integrity from continual recombining. The figure also shows a "corner structure" at the first deposition region. This may be due to the large distances between blooming/straggling events. There is also a focusing-type structure after 0.013 cm, which occurs because, as distance traveled increases, electrons slowly lose energy and stop. The outermost electrons, with the least energy, stop first, leaving the core. There is little spreading of the electron beam after 0.005 cm because the electrons continue moving parallel with the original beam after blooming. In realistic conditions, they would begin to move at an angle to the original beam. This must be rectified in order to create more realistic simulations.

Future Work

Adding parallelism to decrease runtime of the program is a possibility. Each electron being run is simulated completely independently, so, using the current combination parameters, up to 2143 independent threads could be used to simulate electrons. This would create a much faster simulation. Additionally, because memory consumption is now less of a concern, the parameters for simulation of blooming and straggling could be set to allow more frequent adjustment with less significant changes in order to increase the accuracy of the simulation. This would mitigate problems such as the corner structure found in Figure 5. There is one major shortcoming in the current simulation: after blooming occurs within electron beams, they continue moving in parallel with the original beam. Realistically, they should move at an angle for the rest of the simulation. This must be changed in order to ensure more accurate simulations.

Conclusion

A method for combining electrons in order to decrease memory consumption and runtime has been added to an existing model which simulates the progression of electrons within the fast-ignition approach to inertial confinement fusion. This method functions by combining electrons in close spatial proximity, and only suffers from a small loss of accuracy. Additionally, the program was refined to allow for variable-density-plasma environments to be simulated.

References

1. C. K. Li and R. D. Petrasso, Phys. Rev. E 73, 016402 (2006).

2. C. K. Li and R. D. Petrasso, Phys. Rev. E 70, 067401 (2004).

3. E Dobson, “Modeling Collisional Blooming and Straggling of the Electron Beam in the Fast Ignition Scenario”, LLE High School Program Project Report (2006).

Acknowledgements

I would like to thank my advisor, Dr. Jacques Delettrez, for his guidance, support, and advice throughout this project. I would also like to thank Dr. Christian Stoeckl for his assistance in the final stages of the project and in the creation of the symposium presentation. Many thanks also to Dr. Stephen Craxton for his time and energy in directing the High School Summer Research Program at the Laboratory.

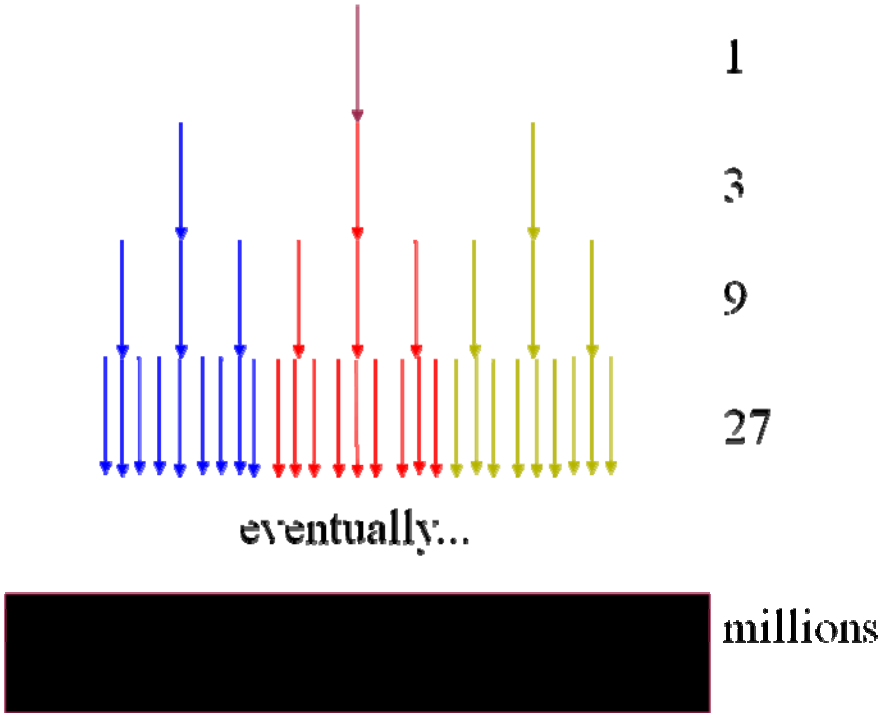


Figure 1: Spreading of electrons, as conducted in the original program

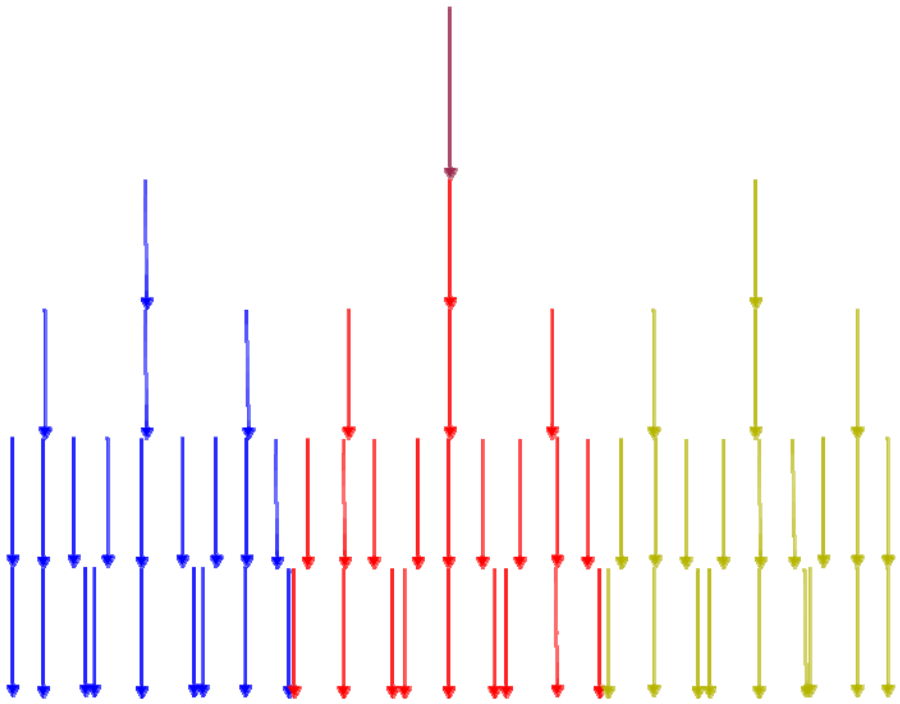


Figure 2: Recombination of electrons using new combining algorithm. The bottom row of arrows depicts pairs of electrons that have been combined, reducing memory consumption. As the simulation presently occurs, horizontal movement does not enter into calculations after initial blooming – all electrons remain parallel in movement.

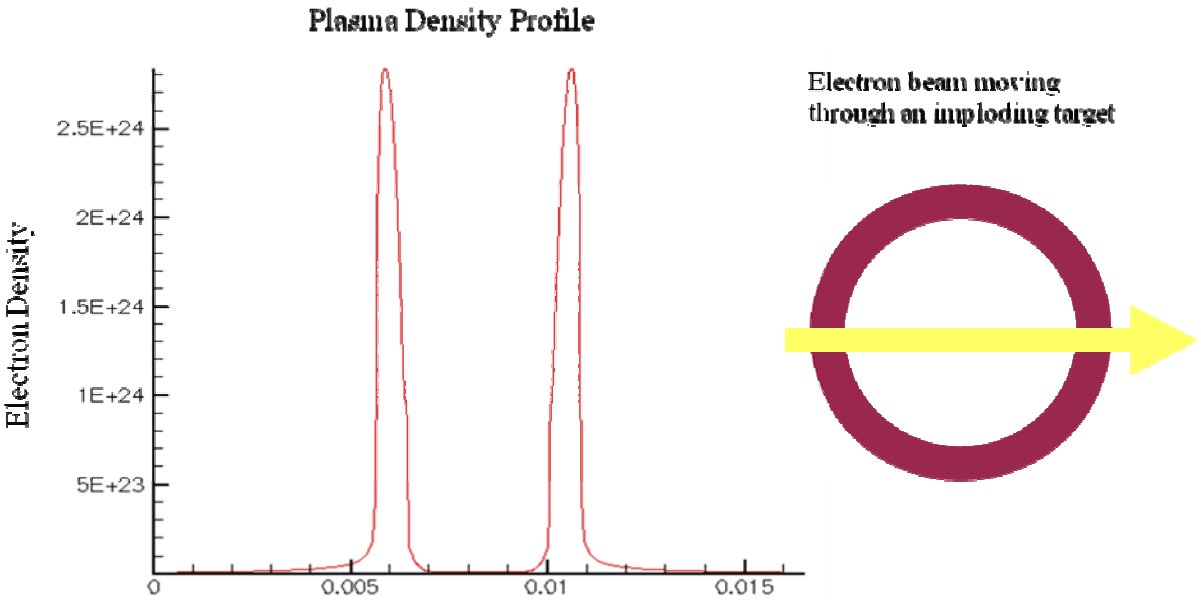
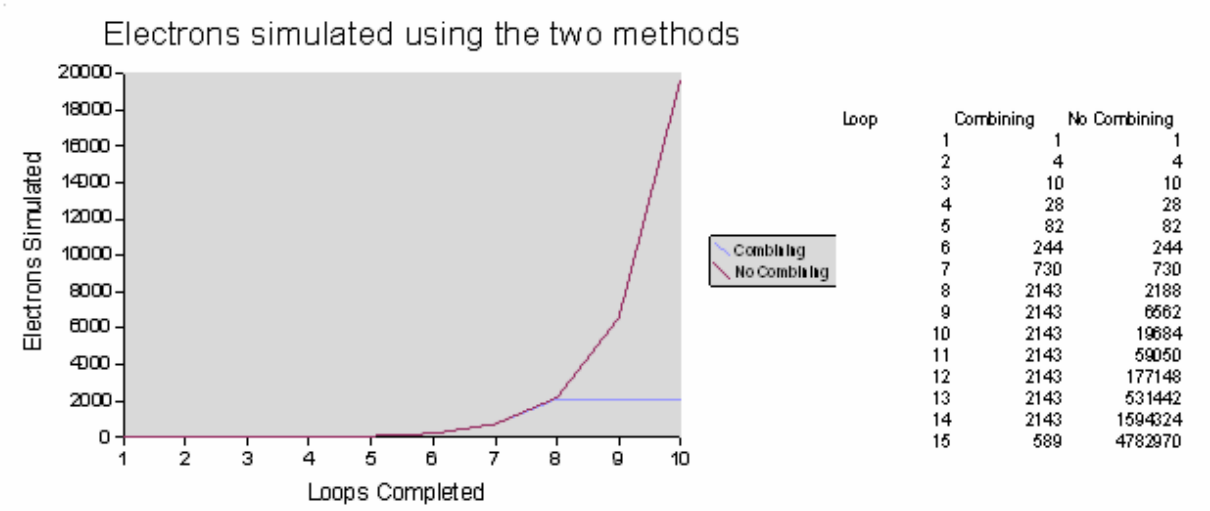


Figure 3: The new variable-density-plasma profile used in the simulation. The graph on the left and diagram on the right are two perspectives on the same thing – the red ring represents the increase in density which appears in cross-section in the graph.



Loop	Combining	No Combining
1	1	1
2	4	4
3	10	10
4	28	28
5	82	82
6	244	244
7	730	730
8	2143	2188
9	2143	6562
10	2143	19684
11	2143	59050
12	2143	177148
13	2143	531442
14	2143	1594324
15	589	4782970

Figure 4: Graph of the number of electrons simulated using the two methods. The number of electrons simulated in the combining method is capped at 2143 (the optimal number). A "loop" is considered to have been completed when the daughter electrons of the electron that initiated the previous loop are processed.

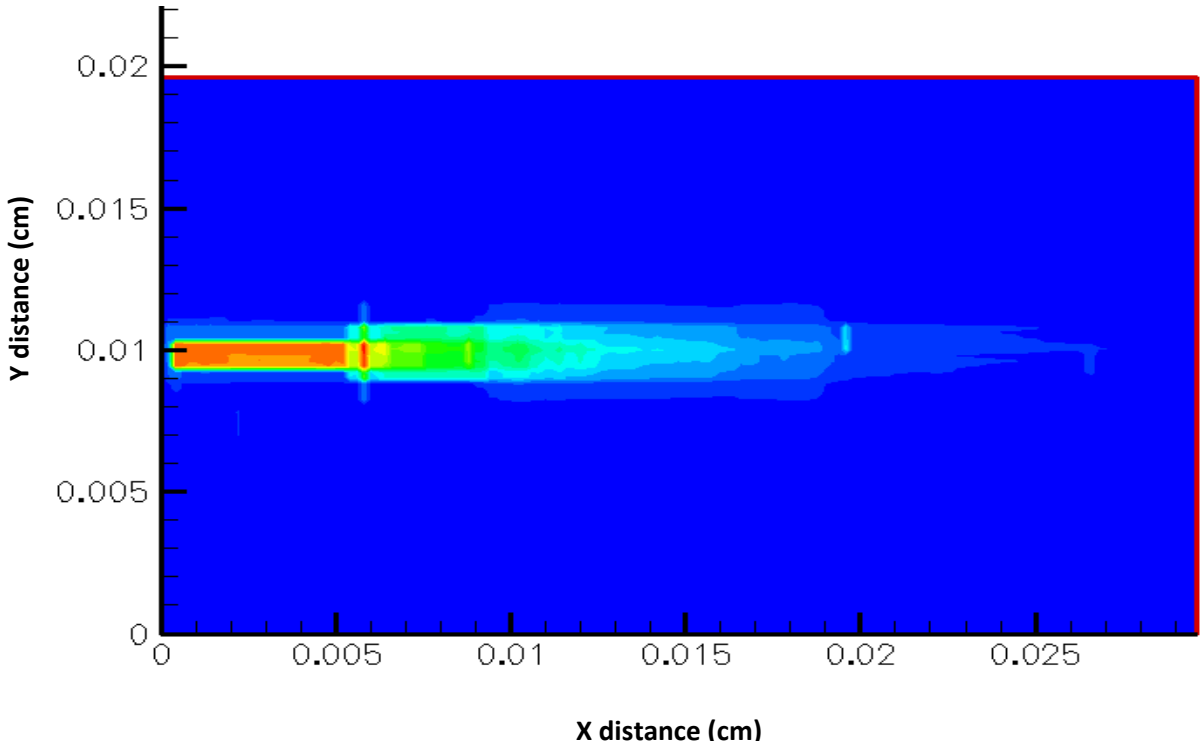


Figure 5: Graph of energy deposition of the electron beam as calculated by the model. The electron beam enters from the left of the figure, and moves through the profile shown in Fig. 3. High levels of energy deposition are shown with bright colors (red, yellow, and green), while low levels are shown with darker colors (shades of blue).