Monte Carlo charged-particle tracking and energy deposition on a Lagrangian mesh

J. Yuan,[1] G. A. Moses,[1] and P. W. McKenty[2]
[1]*Fusion Technology Institute, University of Wisconsin-Madison, 1500 Engineering Drive, Madison, Wisconsin 53706, USA*
[2]*Laboratory for Laser Energetics, University of Rochester, 250 East River Road, Rochester, New York 14623, USA*
(Received 23 September 2004; revised manuscript received 14 July 2005; published 10 October 2005)

A Monte Carlo algorithm for alpha particle tracking and energy deposition on a *RZ* cylindrical computational mesh in a Lagrangian hydrodynamics code used for inertial confinement fusion (ICF) simulations is presented. The straight line approximation is used to follow propagation of "Monte Carlo particles" which represent collections of alpha particles generated from thermonuclear deuterium-tritium (DT) reactions. Energy deposition in the plasma is modeled by the continuous slowing down approximation. The scheme addresses various aspects arising in the coupling of Monte Carlo tracking with Lagrangian hydrodynamics; such as non-orthogonal severely distorted mesh cells, particle relocation on the moving mesh and particle relocation after rezoning. A comparison with the flux-limited multi-group diffusion transport method is presented for a polar direct drive target design for the National Ignition Facility. Simulations show the Monte Carlo transport method predicts about 30 picosecond earlier ignition than predicted by the diffusion method, and generates higher hot spot temperature. Nearly linear speed-up is achieved for multi-processor parallel simulations.

DOI: 10.1103/PhysRevE.72.046706 PACS number(s): 02.70.−c, 52.57.−z

I. INTRODUCTION

Inertial confinement fusion (ICF) is an approach to controlled nuclear fusion that uses high power lasers or charged particle beams as drivers to create a high energy density plasma that reaches fusion conditions [1–3]. In the direct drive approach, the laser or particle beams are focused onto a spherical fuel pellet that contains a mixture of deuterium-tritium (DT) fuel. The outer portion of the pellet is ablated and accelerated outward in a spherically divergent flow, consequently driving the inner part of the pellet inward in a spherically convergent flow. This inwardly imploding DT fuel ideally reaches very high densities (on the order of 200 g/cm^3). However, the imploding plasma is also hydrodynamically unstable, so that any non-spherical perturbation introduced by non-uniformity in the driver intensity or the pellet fabrication will grow in time. Thus the highly compressed fuel will deviate to some degree from perfect sphericity. This final distorted fuel region must remain sufficiently spherical to avoid mixing of the DT fuel with surrounding pellet material such as Be or plastic, otherwise it will fail to compress and ignite. The carefully timed increasing driver intensity drives a final strong shock wave toward the origin of the spherically imploded fuel and this shock wave heats a small center portion ($<10\%$) of the fuel (central hot spot) to a temperature exceeding 4 keV. The combination of very high DT fuel density and the high central temperature results in nuclear fusion reactions between the deuterium and tritium nuclei, releasing a 14.1 MeV neutron and a 3.5 MeV alpha particle for each fusion event. The fusion hot spot plasma radius is comparable to the range of the 3.5 MeV alpha particles (0.3 g/cm^2) such that they slow down and lose an appreciable amount of energy in the hot plasma [4,5], thus "bootstrap heating" the plasma to yet higher temperature. A fusion burn wave is created that propagates outward into the surrounding dense, but cooler, fuel and heats this fuel to fusion conditions so it begins burning as well with temperature rising to 50–80 keV. This process of hot spot ignition and fusion burn occurs on a time scale of about 10–50 picoseconds (ps). In this short time, the fuel is held stationary by its own inertia, despite the enormous pressures being created. Thus the name inertial confinement fusion is given to this dynamic process.

A critical stage in this complex process is the central hot spot ignition phase. Significant fusion energy gain (>100) requires that a hot spot is formed and the fusion burn propagates into the surrounding cold fuel. High gain is necessary for potential fusion power plant applications. This important hot spot ignition process will be first tested on the National Ignition Facility currently under construction at the Lawrence Livermore National Laboratory [6]. There are unfortunately many pathways that lead to ignition failure. If the hot spot fuel has insufficient temperature after shock heating, then too few fusion reactions occur and bootstrap heating fails to materialize. If the fuel density (and thus charged particle stopping power) is too low, then alpha particle reaction products deposit their energy in a large volume of fuel and fail to raise the temperature sufficiently to create a burn wave. Making all of these processes occur at precisely the same time requires very precise physical tolerances on the driver power and the fuel pellet dimensions and sphericity. It is estimated from ICF implosion simulations that laser pulse shape tolerances of a few picoseconds and laser beam power balancing tolerances of a few percent between the hundreds of laser beams are necessary to achieve fusion ignition conditions.

These highly precise physical tolerances demand that the theoretical models and numerical schemes used to simulate the pellet implosion and fusion burn have a very high fidelity so that numerical modeling inaccuracies are significantly smaller than the true physical tolerances. The numerical simulation of inertial confinement fusion laser-pellet interaction, pellet implosion, and fusion burn is accomplished with complex multi-physics computer codes. The fundamental ba-

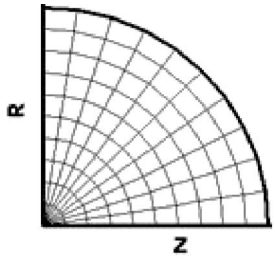

FIG. 1. Typical 2D computational mesh used in fusion target simulations.

sis of these codes is radiation-hydrodynamics. The physics of ICF radiation-hydrodynamics is the same as that used in astrophysics to simulate dynamic phenomena in stars [7]. The dense plasma is treated in the fluid approximation and the heat transfer is modeled by x-ray radiation transport. In the multi-physics nature of these codes, other physical phenomena, such as laser energy absorption and charged fusion reaction product transport and slowing down are coupled to the mass, momentum and energy conservation equations of hydrodynamics.

The computer codes used to simulate ICF experiments employ either finite difference or finite volume methods to solve Eulerian, Lagrangian, or arbitrary Lagrangian Euler (ALE) forms of the hydrodynamics equations. The Lagrangian form has an advantage in preserving material interfaces and is thus commonly used in ICF pellet implosion simulations. While the computational mesh remains stationary in the Eulerian form, vertices move with the local fluid velocity in the Lagrangian form and the ALE form. Such moving meshes add additional complexity to numerical modeling because the finite different cells can become distorted as they follow the flow of the ever-present hydrodynamic instabilities. A typical two-dimensional (2D) mesh used in fusion target simulations is shown in Fig. 1, which employs a *RZ* coordinate system to simulate a half sphere. Since it is a 2D mesh, the finite difference cells are quadrilaterals with four vertices and four edges rotated around the *Z* axis. The distorted shape of an individual cell under hydrodynamic motion may have three different forms, as shown in Figs. 2(a)–2(c). These are of the convex (regular) shape, the concave (banana) shape, and the tangled (bowtie) shape. When severe convex and concave shapes occur, the computational mesh must be rezoned in order for the simulation to proceed in a timely fashion without loss of numerical accuracy. This is accomplished by redefining the mesh to be more orthogonal and interpolating the original finite difference values onto the new mesh. The tangled shape results in negative cell volume and must be corrected by rezoning before continuing the simulation.

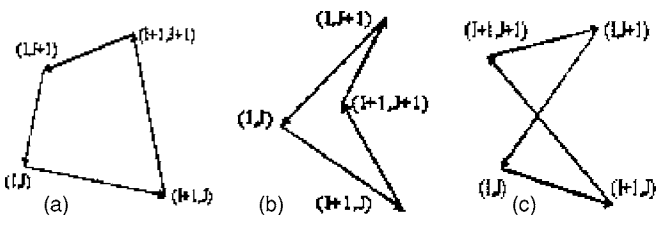

FIG. 2. Three different quadrilateral shapes under hydrodynamic flow. (a) Convex (regular), (b) Concave (banana), and (c) Tangled (bowtie).

The distortion of the hydrodynamic computational mesh creates complications in the derivation of deterministic finite difference algorithms for modeling the transport of particles such as x-ray photons and DT fusion reaction alpha particles on this mesh [8–11]. These algorithms have a figure of merit that measures their ability to preserve regular solution features such as plane waves crossing a distorted mesh that is not orthogonal to the isocontours of the solution. The degree to which the mesh does not distort the otherwise uniform solution and the convergence of the solution with refined mesh are features of these algorithms that researchers seek. The complexity of numerical solutions of the transport equation on distorted meshes as well as the substantial computational resources necessary for solving the transport equation in multiple dimensions often leads to the use of flux-limited multi-group diffusion as the method of choice for transport of x-rays and alpha particles in multi-physics ICF radiation-hydrodynamics simulation codes. Flux-limited diffusion has been shown to correctly predict general features of alpha particle transport and energy deposition and accurately preserve conservation of particle energy and number, but it fails to predict detailed features of energy deposition such as the "Bragg peak" that occurs at the end of the alpha particle range [12]. Furthermore, flux-limited diffusion is a finite difference method that suffers the peril of the distorted hydrodynamic mesh.

In this paper we describe a Monte Carlo method for the transport of DT fusion product alpha particles on a distorted hydrodynamic mesh. The Monte Carlo method [13] has the advantage that it is insensitive to the non-orthogonal nature of the mesh because the algorithm simply tracks alpha particles to the next cell edge that they intercept and this algorithm is independent of the cell shape. The Monte Carlo algorithm has the disadvantage that it is a non-deterministic method that relies on good statistical significance for accurate solutions. The challenge is to use sufficient numbers of "Monte Carlo particles" to obtain statistically significant results. This often requires substantial computational resources. This limitation is balanced by the fact that Monte Carlo algorithms are "embarrassingly parallel" and parallel computers can easily be employed.

The paper is organized as follows. In Sec. II, we present the Monte Carlo tracking algorithm. The initial sampling of particles, intersection with cell edges, particle relocation on the moving mesh and relocation after rezoning are discussed in each subsection. In Sec. III, we describe the alpha particle energy deposition, applying the same stopping power theory for both diffusion and Monte Carlo transport. Results of our Monte Carlo algorithm compared to flux-limited diffusion in a production ICF simulation code are given in Sec. IV. We use the baseline polar direct-drive target [14] proposed for testing on the National Ignition Facility as an example to examine the effect of using the Monte Carlo transport model. This will likely be the first experiment that tests the fidelity of the simulation methods used for DT fusion alpha particle transport and slowing down. Energy deposition, burn wave

propagation and ignition timing are compared side-by-side with the diffusion model [12,15]. The different results predicted by the Monte Carlo method compared to the diffusion method are discussed in Sec. V. Significant differences in simulation results between these two transport methods advocate for the use of the more physically accurate and mesh independent Monte Carlo method for such sensitive simulations. Finally, a note on the run time speed-up of the Monte Carlo algorithm for execution on a multi-processor parallel computer concludes the paper.

II. TRACKING ALGORITHM

Simulation of alpha particle transport, slowing down and tabulation of consequent energy deposition in the fusion plasma by the Monte Carlo method is based upon the idea that Monte Carlo particles can each approximate a large number of actual alpha particles (with appropriate statistical weight) and that tracking of the Monte Carlo particles is an accurate representation of the transport of all real particles. This approximation is improved as the number of Monte Carlo particles increases and thus the statistical weight of each becomes proportionately less. In the limit as the number of Monte Carlo particles approaches the number of real particles, the simulation becomes exact. We will refer to the Monte Carlo particles as particles in the following discussion. In practical terms, the number of particles that one chooses to track is determined by the amount of computational resources that are available and the degree of accuracy that is necessary. The Monte Carlo algorithm is independent of the number of particles; it is independently applied to each particle to track its history.

Radiation-hydrodynamics simulations are inherently time dependent and thus the alpha particle transport algorithm should be time dependent. We have implemented two approximations to time dependence for tracking the alpha particles. In the adiabatic approximation, all particles created on a given time step are tracked to the end of their life during that time step. This adiabatic approximation is valid when the hydrodynamic time step is comparable to the slowing down time of the alpha particles. A second approximation that we call the time dependent approximation is to track the alpha particles until they either slow down to thermal energy or they have transported for a time equal to the time step, whichever is a shorter time. If they have not slowed to thermal energy before the end of the time step, then information about this Monte Carlo particle is saved and the transport is restarted on the next time step. This second approximation is usually necessary for ICF burn calculations. It requires that a data structure to save particle information be constructed. This is discussed in Sec. II C.

To track the particles on the hydrodynamic computational mesh during a single time step, the Monte Carlo algorithm must randomly generate particles in the particle source cells, compute the distance to intersection with the next cell edge as particles traverse cells, compute the energy loss through slowing down in each cell traversed and tally the accumulated energy loss in each cell. Between time steps the algorithm must determine particle relocation in the cells after the mesh moves. In the following, we discuss each task individually.

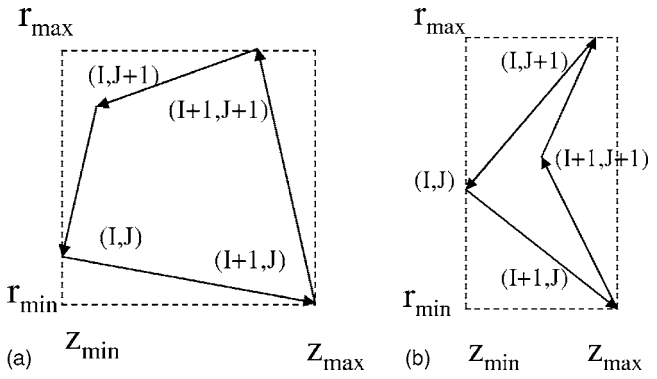

FIG. 3. Quadrilaterals are inscribed in rectangles. (a) Convex, (b) Concave.

A. Random sampling of particles in a cell

Random sampling of particles produced from fusion reactions in a cell is not straightforward in the 2D *RZ* cylindrical geometry. First of all, it is actually a three-dimensional (3D) problem. The position of a sampled particle is represented by Cartesian coordinates (x,y,z) within a torus enclosed by four conical segments defined by the quadrilateral cell rotated around the Z axis. Second, a geometric effect must be taken into account; that is, a uniform density of created particles implies that more total particles are created at larger radii within a cell. Third, the sampled particles on a cell edge must be associated with the indices of the cell into which they are traveling. Adjustment to the cell indices of the particle location is needed in order to ensure an intersection. This is illustrated at the end of this section. This ambiguity can cause problems with the tracking algorithm. Our algorithm allows particles to be generated on the cell edges but some algorithms do not [16]. This is necessary since the size of the final compressed DT fuel in a typical target implosion is very small (around 100 μm).

We use the rejection sampling method for the random particle sampling in a cell. As shown in Fig. 3, the quadrilateral of either regular shape or concave shape is enclosed in a rectangle with the maximum Z and R length as the width and height of the rectangle. The reject sampling method first samples particles from the rectangle, for which it has an analytic solution. The random selection of radius r in the rectangle is given by

$$r^2 = r_{\min}^2 + \xi(r_{\max}^2 - r_{\min}^2), \tag{1}$$

where $r_{\min}$ and $r_{\max}$ represent the minimum and maximum radius of the quadrilateral, respectively. The variable ξ is a random number between 0 and 1. The x and y coordinates of the particle position can thus be sampled from a random azimuthal angle as

$$x = r\cos(\vartheta)$$

$$y = r\sin(\vartheta), \tag{2}$$

where ϑ is the azimuthal angle which is selected randomly from 0 to 2π. The z coordinate of the particle position is simply given by,

$$z = z_{\min} + \xi(z_{\max} - z_{\min}), \tag{3}$$

where $z_{\min}$ and $z_{\max}$ represent the minimum and maximum z components of the quadrilateral, respectively. The second step in the rejection sampling method is to determine whether this sampled particle from the rectangle region is in the quadrilateral. If the particle is outside the quadrilateral, the sampled particle is rejected. We design a method to detect whether a particle is in a quadrilateral by splitting the quadrilateral into two triangles. The particle-in-triangle (PIT) test is performed for each of the two triangles. A point is inside of a triangle if and only if the summation of the areas of three sub-triangles created by connecting the point with three vertices of the triangle is equal to the area of the triangle. In order to include the quadrilateral concave shape as well as the convex shape in the algorithm, the splitting line is first determined to connect two opposite vertices of the quadrilateral. For quadrilaterals with regular shape, either pair of two opposite vertices can be used as the splitting line. For quadrilaterals with concave shape, the vertex with internal angle greater than 180 degrees must be chosen for the splitting line. This vertex can be found by performing the PIT test since this vertex is inside of the triangle constituted by the other three vertices. The sampled particle location is tested with the PIT test. If it is successful in either of the two triangles, the particle is inside of the quadrilateral or on the edge of the quadrilateral. Otherwise, the sampled particle is rejected.

The accepted particle is also assigned the quadrilateral's computational mesh indices. In the rare occurrence of a particle located on a cell edge, the cell indices may need adjustment because the isotropically sampled direction of travel may associate the shared edge with the adjoining cell. The cell to which the particle is assigned is always the cell into which it is traveling. Care must be taken for edge particles because failure to assign the correct cell indices can lead to "lost particles" and errors in the particle census and deposition results. The algorithm used to handle the assignment of the indices for the particle sampled on the edge is as follows.

(i) Determine the inward normal direction vector of the edge. As shown in Sec. II C, the normal line of the conical segment surface constructed by the edge is $(\partial f/\partial x, \partial f/\partial y, \partial f/\partial z)$, which is $(2x, 2y, -2k[r_1 + k(z - z_1)])$ and f is the equation function of the conical segment defined in Eq. (5). However, the normal line has no direction. To assign the inward direction to the normal line, a reference point inside of the quadrilateral is needed. We choose the centroid of the quadrilateral as the reference point.

(ii) The cosine of the angle between the normal line vector and the vector from the particle position to the reference point is calculated. If the cosine is less than zero, the direction of the normal line vector is reversed to use as the inward normal direction vector.

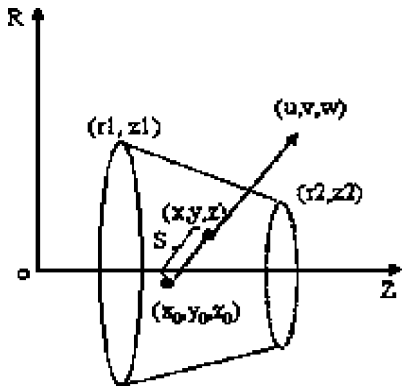

FIG. 4. Intersection with one edge of a quadrilateral by the particle straight line trajectory.

(iii) The cosine of the angle between the inward normal direction vector and the particle direction is calculated. If the cosine is less than zero, the assigned indices need to be adjusted according to the location of the edge. For example, if it is the edge $(I, J+1; I, J)$, the indices of the particle are changed from (I, J) to indices $(I-1, J)$.

B. Intersection with cell edges

To calculate the energy deposition in a cell, the particle travel distance in the cell is required. To determine the point at which a particle passes from one computational cell to another, the intersection with the cell edge is calculated. In the *RZ* cylindrical geometry, this requires the solution of two coupled equations consisting of the equation of a line in 3D and a conical segment equation. Again, each quadrilateral in the mesh actually represents a volume enclosed by four conical segments defined by the edges. Unlike the situation in *XY* planar geometry in which we can determine the intersection using only the angular information without actual computation of the intersection, there is no shortcut for the 2D *RZ* cylindrical geometry to the best of our knowledge. Computation of the intersections for each of the four sides must be performed in order to determine the intersected edge. This is actually the workhorse operation in the whole Monte Carlo tracking algorithm, consuming the greatest amount of computer time.

As shown in Fig. 4, the line in 3D is represented by the starting point (x_0, y_0, z_0) with direction cosine vector (u, v, w). One edge of a quadrilateral is connected by point (r_1, z_1) and point (r_2, z_2). This edge is rotated about the Z axis and thus forms a conical segment. The intersection point of the line and conical segment is (x, y, z) and the distance from the starting point to the intersection is represented by the symbol S. The equation of the particle trajectory line is

$$\frac{x - x_0}{u} = \frac{y - y_0}{v} = \frac{z - z_0}{w} = S, \tag{4}$$

and the equation of the conical segment with ending point (r_1, z_1) and point (r_2, z_2) is

$$x^2 + y^2 - [r_1 + k(z - z_1)]^2 = 0, \tag{5}$$

where $k \equiv (r_2 - r_1)/(z_2 - z_1)$.

The solution of the two-coupled equations is based upon the locations of the two points (r_1,z_1) and (r_2,z_2).

(a) When the two points (r_1,z_1) and (r_2,z_2) are the same, that is, $r_1=r_2$ and $z_1=z_2$, there is one potential solution $S=(z_1-z_0)/w$ only if the solution satisfies the equation $(x_0+Su)^2+(y_0+Sv)^2=r_1^2$, otherwise, there is no solution.

(b) When $r_1=r_2$ and $z_1\neq z_2$, it is a cylinder with radius r_1, and there are two solutions if $w\neq\pm1$. These two solutions are

$$S=\frac{-(x_0u+y_0v)\pm\sqrt{(x_0u+y_0v)^2-(1-w^2)(r_0-r_1)^2}}{1-w^2}, \tag{6}$$

where $r_0^2\equiv x_0^2+y_0^2$. Positive solutions are selected. If there are two positive solutions, the smaller one is selected. There is no solution when $w=\pm1$, which means the particle trajectory is parallel to the Z axis.

(c) When $z_1=z_2$ and $r_1\neq r_2$, the edge is a plane disk parallel to the r direction. There is a solution only if $w\neq0$, which is $S=(z_1-z_2)/w$.

(d) If none of the above cases apply, the distance to the cell edge is obtained by solving the following quadratic equation,

$$S^2(1-w^2-k^2w^2)+S[2x_0u+2y_0v-2kw(r_1+k(z_0-z_1)\ )+r_0^2-(r_1+k(z_0-z_1))^2=0, \tag{7}$$

where $k\equiv(r_2-r_1)/(z_2-z_1)$ and $r_0^2\equiv x_0^2+y_0^2$. The acceptable solutions are positive ones.

The intersection with a quadrilateral edge results in eight possible solutions, with two possible solutions for each edge. The rules required to select the unique and correct solution are (1) it must be positive, (2) it must be smallest, and (3) the Z component of the intersection point must be in the range (z_1,z_2).

C. Particle travel history

Once a particle is launched in a cell, its trajectory is constructed by a straight line which encounters one of the cell's edges. This straight line approximation to the particle trajectory is discussed in Sec. III. The particle trajectory is terminated under one of the following three circumstances, (1) the particle travels out of a user specified region of interest, (2) the particle energy falls below a user specified cut-off energy, or (3) the particle travel time is larger than the hydrodynamic time step if the time dependent tracking is applied. The alternative tracking approach is adiabatic, in which the particle continues to propagate until one of the first two conditions are met. The time dependent tracking method requires additional computer memory to store the previous incomplete trajectories so that they can resume on the next hydrodynamic time step. In our implementation, a two-dimensional array is created for this purpose, one dimension is for the particle number and the other is for the time step history number. A particle element in the two dimensional array is represented by a Fortran 90 data type which has position, direction, energy, statistical weight, and other statistical variables as its data members. The history number equals one for adiabatic tracking. For time dependent tracking, a rule must be applied to free a history slot so that it can be reused if the needed history index is larger than the total available number of histories in the data structure. The history slot marked as reusable either (1) contains the fewest particles or (2) is the oldest, depending upon user input. The particles in the chosen history slot are forced to propagate continually until either (1) they are out of the region or (2) their energy is below the cut-off energy. The escape boundary or the boundary of the user specified region can be set as the outermost surface or a predefined radius. For example, in the simulation of a laser driven target implosion, we can set the region of interest radius as the interface between the DT fuel and the ablation layer under the assumption that the alpha particles propagating beyond this radius have little contribution to the fusion burn. Particles with energy less than the cut-off energy (e.g., 0.01 MeV) are considered as thermalized and their energies are deposited locally.

D. Moving mesh, rezoning, and restart

A Monte Carlo charged particle deposition package using the above algorithm has been implemented in the ICF simulation code, DRACO, developed by the Laboratory for Laser Energetics (LLE) Ref. [17]. DRACO is a Lagrangian radiation-hydrodynamics code designed to run in a multidimensional geometry. It includes important ICF physics such as energy exchange among the fields, refractive laser ray tracing, classical ion and electron conductivity, multigroup radiation diffusion, and multi-group charged particle diffusion deposition [18–20]. Rezoning is included in the code to keep the computational mesh from severe distortion, and material interfaces are tracked in the mixed material cells. For the adiabatic tracking method, the particles are tracked to their end-of-life on each time step so there is no complication with the moving mesh. However, for the time dependent tracking method, the hydrodynamic time step serves as a clock for the particle tracking. The particles stop traveling temporarily when their propagation time is equal to the current hydrodynamic time step. At the beginning of the next time step or after mesh rezoning, the cell indices of the particles' locations need to be readjusted because the mesh moves, but the particles remain fixed in space. The correct cell indices of particle location are required to find the intersection with the next cell edge and also for the correct use of the material properties in computing the particle slowing down. Finding the new indices is greatly simplified by the fact that numerical stability required by the Courant-Friedrichs-Lewy (CFL) condition restricts a cell to move for a distance less than one cell length. This is also true for mesh rezoning which relocates the cell vertices to reconstruct a more regular mesh. In this case, the relocation is limited to one cell to minimize the re-flux numerical diffusion of hydrodynamic field variables associated with the rezoning [21]. This restriction greatly simplifies the cell index search algorithm and reduces the computing time, since only the four neighbor cells need to be searched.

Very often, hydrodynamics codes need to restart from previous runs. To implement this capability, data information is

periodically written from memory to hard disk at checkpoint time steps. However, it is impractical to write out all detailed Monte Carlo particle data structures during checkpointing since it significantly increases the execution time and uses too much disk space. This can be seen from an example. Assuming one million particles are followed on each time step and assuming 50-bytes of storage for each particle, the total storage size to be written is about 50 Mbytes per history. When particles from all histories are included, the total storage may grow to more than several gigabytes in the particle history data structure. One way to overcome this difficulty is to force all live particles to deposit their remaining energy according to the adiabatic method by propagating until either they are out of the region or their energies drop below the cut-off energy on the time step before the operation to write data at the checkpoint is taken. In this way the particle history data structure need not be written. However, detailed time dependent information is lost. Another approach is to statistically build an energy distribution function and a direction distribution function from the data structure of live particles, and write these distribution functions for each cell to disk storage instead. When the simulation resumes from the previous run, the Monte Carlo particles are dynamically sampled from these distribution functions. This approach uses less disk space than writing the entire data structure and preserves some limited information about the energy and direction distribution.

III. ENERGY DEPOSITION BY STOPPING POWER

There are two major approaches to the theoretical calculation of charged particle stopping power. One is based on Bohr's theory which is dependent on the impact parameter between the trajectory particle and the target atom in the classical mechanic limit combined with the Bethe-Bloch equation which is dependent on momentum transfer from the particle to the target in the quantum mechanic limit. The other approach is based on the Fokker-Planck equation to evaluate the collision term of the Boltzmann equation [22]. In our work, we use the formulas derived by Li and Petrasso [23], which properly treat the effects of large-angle scattering as well as small-angle collisions by retaining the third-order term in the Taylor expansion of the collision operator in the Fokker-Planck equation.

The relationship between the stopping power and the energy loss rate is

$$\left.\frac{dE}{ds}\right|_{e/i} = \frac{E_t}{v_t} v_\varepsilon^{e/i} \tag{8}$$

where E_t is the trajectory particle energy, v_t is the particle velocity, and $v_\varepsilon^{e/i}$ is the energy loss rate to background electrons and ions, which is given by

$$\nu_\varepsilon^{e/i} = \frac{2e_t^2 \ln \Lambda_b^2}{m_t v_t^3} \omega_{pf}^2 G(\mu), \tag{9}$$

where e_t (e_f) is the trajectory (field) particle charge, $m_t(m_f)$ is the trajectory (field) particle mass), ω_{pf} is the plasma frequency and

$$G(\mu) = \mu - m_f \mu' / m_t + m_f(\mu + \mu')/(m_t \ln \Lambda_b). \tag{10}$$

$\mu = (2/\sqrt{\pi}) \int_0^{x_{t/f}} e^{-\xi} \sqrt{\xi} d\xi$ is the incomplete Gamma integral, $x_{t/f} = v_t^2 / v_f^2$ where $v_f^2 = 2kT_f/m_f$. For the Coulomb logarithm $\ln \Lambda_b$ for electrons we use the formula by Skupsky [24],

$$\ln \Lambda_{RPA} = 0.5[\ln(1 + \Lambda_s^2(0.37 + 0.44\eta^2) - 1\ ], \tag{11}$$

which is obtained from the random-phase-approximation form of the quantum-mechanical dielectric function. Λ_s is the standard Coulomb logarithm argument ($=12mT/\hbar^2 k_D^2$). The effect of electron degeneracy η is calculated through the relation of the Fermi integral and the electron number density,

$$n = 4\pi/h^3 (2mT)^{3/2} F_{1/2}(\eta). \tag{12}$$

Note in the case of alpha particle and field electron interactions, $G(\mu)$ approaches $\frac{2}{3}x^{3/2}$ because of the small mass ratio ($\approx 10^{-4}$) and $x \ll 1$. Thus, v_ε^e is independent of the trajectory particle energy E_t.

Assuming the stopping power is constant for a small traveling distance ΔS, the final energy becomes

$$E = E_i - \frac{dE}{ds} \Delta S$$

The traveling time is solved from $\tau = \int_{E_i}^{E} (dE/dt)^{-1} dE$, which can be approximated by Ref. [12]

$$\tau = \frac{2}{3} \frac{1}{v_\varepsilon^e} \ln \frac{v_\varepsilon^e E^{3/2} + A_i}{v_\varepsilon^e E_i^{3/2} + A_i}, \tag{13}$$

where $A_i = e_t^2 \ln \Lambda_i \omega_{pi}^2 G^i(\mu) \sqrt{m_t/2}$.

The relative contribution to the alpha particle slowing from the electrons and ions of the background plasma demonstrates that the stopping from the electrons dominates at high alpha particle energies (Fig. 6.12 of Ref. [2]). This validates the straight line slowing down approximation since the slowing ions suffer no significant deflection by colliding with electrons. Large angle scattering or dispersion occurs only near the end of the particle trajectory to thermalization when the velocity of alpha particles decreases and the stopping from the ions dominates.

IV. RESULTS FOR NIF POLAR DIRECT DRIVE TARGET IGNITION

The example we use to investigate the effect of the Monte Carlo alpha particle transport algorithm is the so called baseline polar direct-drive NIF design [14]. This design employs a cryogenic DT-shell target with a thin polymer ablator surrounding the DT ice shell. The isentrope of the ablation surface and the fuel is controlled by a preheat shock. Figures 5(a) and 5(b) shows the target specification and the laser pulse shape. In order to test the scheme with a distorted mesh, a non-uniform laser irradiation is applied at l mode equal to 6 with the amplitude of 1% variation. The simulation includes laser ray tracing, hydrodynamics, thermal conduction, and burn wave propagation. Calculations are conducted using both the diffusion method [12,15] and the

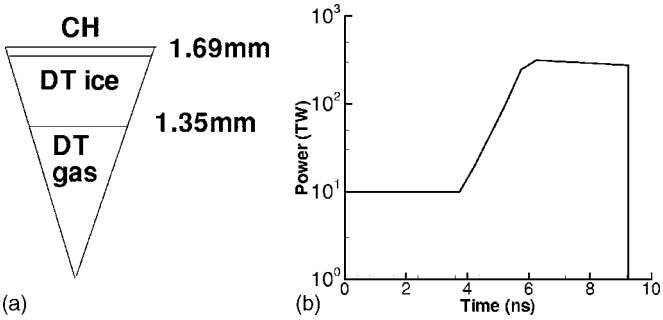

FIG. 5. The baseline "all-DT" 1.5 MJ target design. (a) The target specification and (b) the laser pulse shape.

Monte Carlo tracking method. Side-by-side comparisons are made for the results from these two methods.

First, we look at the difference of alpha particle energy deposition from the two different options in the Monte Carlo transport method; that is, the time dependent transport and the adiabatic transport approximation. In the time dependent method, the hydrodynamic time step acts as a clock for particles, while in the adiabatic method, the particles deposit all their energy in the current time step. From Fig. 6, it can be seen that the temporal and spatial integrated energy deposition for both methods are very similar. The latter method gives a slightly larger energy deposition than the former. The variation of energy deposition at each time interval results from different time steps used for the numerical stability.

Comparison between the Monte Carlo and diffusion transport for the same quantities as above is also plotted in Fig. 7. The spatially integrated current time energy deposition (the lower two lines in Fig. 7) is normalized to the same time step interval. The figure shows that the energy depositions before 9.55 nanoseconds (ns) are close. For example, at 9.38 ns the

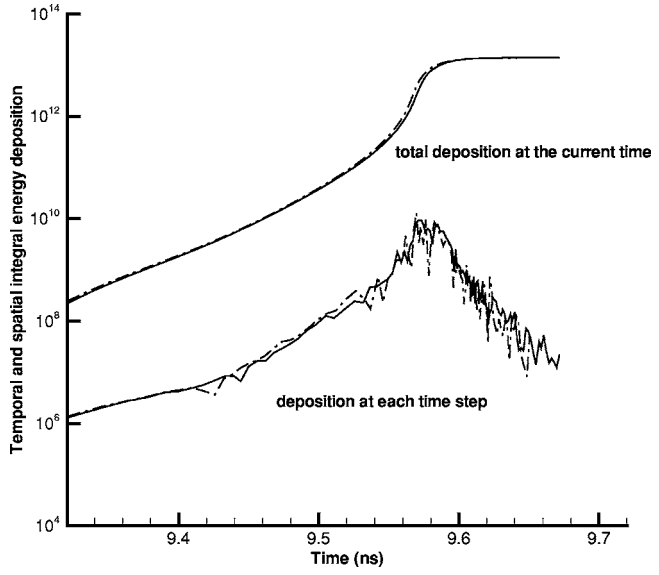

FIG. 6. A comparison between the time dependent and the adiabatic Monte Carlo transport. Dash-dotted lines: adiabatic. Solid lines: time dependent. Upper two lines: total energy deposition up to the current time step. Lower two lines: energy deposition at the current time step.

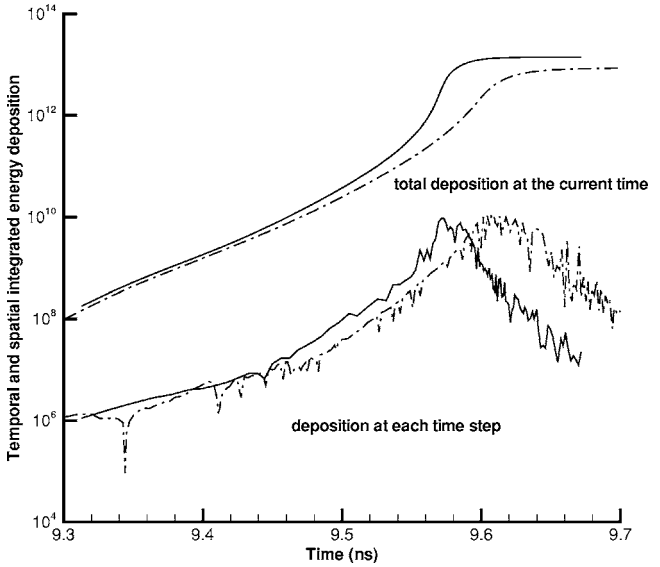

FIG. 7. A comparison between the time dependent Monte Carlo method and the flux-limited diffusion method. Dash-dotted lines: diffusion. Solid lines: Monte Carlo. Upper two lines: total energy deposition up to the current time step. Lower two lines: energy deposition at the current time step. The peak energy deposition time: Monte Carlo (9.56–9.59 ns), diffusion (9.6–9.65 ns).

total temporal and spatial integrated energy for the Monte Carlo method is 1.2×10^9 ergs while it is 0.93×10^9 ergs for the diffusion method, and thus the energy deposition from the diffusion method is 20% lower than that from the Monte Carlo method. After 9.55 ns, the energy deposition using the Monte Carlo method is significantly larger than that using the diffusion method. It lasts for 50 ps to 9.6 ns before the energy deposition decreases. The peak deposition using the diffusion method lasts longer comparing with the Monte Carlo method, which is about 80 ps. Therefore, the Monte Carlo method gives higher heating rate than the diffusion method. The other salient feature is the Monte Carlo method predicts earlier ignition than the diffusion method since the maximum energy deposition using the Monte Carlo method occurs earlier than using the diffusion method. This can be seen in detail in the later discussion. The ion temperature contours shown in Figs. 8(a) and 8(b) at 9.55 ns explain the higher heating rate for the Monte Carlo method. As a result of 20% more energy deposition, the Monte Carlo method drives higher ion temperature in the hot spot, which is 20 keV in comparison with 13 keV from the diffusion method. Note the size of the hot spot is quite similar.

Much closer investigation of the difference of energy deposition between the Monte Carlo method and the diffusion method is given in Fig. 9. The energy deposition is azimuthally averaged over constant mesh index in the azimuthal dimension. This averaging follows the contours created by the distorted mesh. We can see that at 9.50 ns they are very close. However, at 9.60 ns using the Monte Carlo method the ignition has already taken place and the alpha particles start to heat up the surrounding main fuel, while using the diffusion method the alpha particles still deposit energy at the center of the hot spot. At 9.67 ns, the thermal

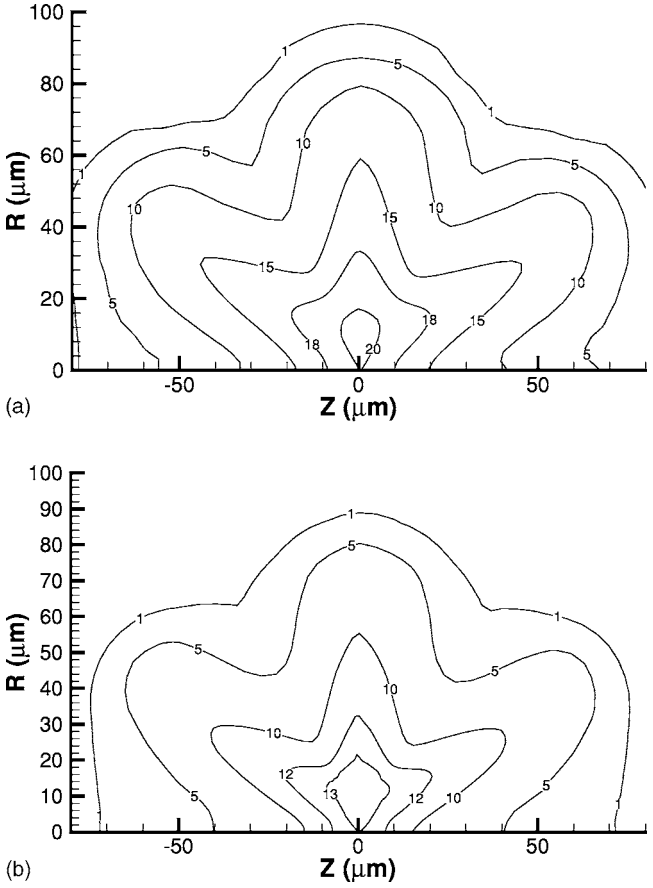

FIG. 8. The ion temperature contour at the compression stagnation phase (t=9.55 ns). (a) Monte Carlo and (b) diffusion. The temperature at the center of hot spot: Monte Carlo (20 keV) and diffusion (13 keV).

burn wave propagates outward further using the Monte Carlo method while the ignition only just starts when using the diffusion method. The resulting ion temperatures are shown in Fig. 10. As we can see, the ion temperature is increasing from 10 keV for both methods at 9.5 ns. The center of the hot spot is over 100 keV at 9.6 ns using the Monte Carlo method and the burn wave is developed at the radius of 200 μm and starts to propagate outward, while the compression is still on the way inward using the diffusion method. Using the Monte Carlo method, the center of the hot spot is cooling down at 9.67 ns as the result of the burn expansion and a larger thermal burn wave is developed from heating along the incoming main fuel. The main fuel is also ignited and a thermal wave is developed using the diffusion method. This offset in timing for ignition and burn waves is further shown in Fig. 11. The time at the end of compression and the beginning of heating the main fuel using the Monte Carlo method is 9.56 ns while it is 9.60 ns using the diffusion method. The thermal wave becomes obvious at 9.60 ns and amplifies at 9.67 ns for the Monte Carlo method while it is at 9.63 ns and 9.72 ns for the diffusion method, respectively. This clearly illustrates that the Monte Carlo method predicts the starting time of ignition about 30 ps earlier than the diffusion method predicts.

Finally, the gain factor is given in Fig. 12 for four different runs. Three Monte Carlo calculations are performed for

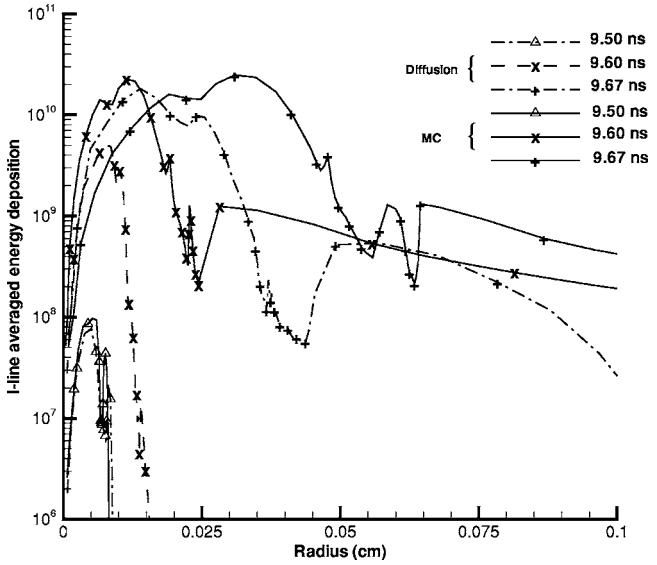

FIG. 9. A comparison of the averaged energy deposition between the Monte Carlo and diffusion method at different times, averaged over constant azimuthal mesh index (i-line).

the particle numbers of 10^4, 10^5, and 10^6 to investigate the convergence and study the statistical effect of the number of particles used. Note these are particles newly emitted from nuclear burn for each time step. The actual number of tracked particles is much larger since there are many census particles from previous time steps also included in the transport. It can be seen that the three calculations generate similar gains in the end, but the ignition time is different. From the three Monte Carlo runs, we observe a logarithmic convergence trend with differences in time of ignition varying constantly with each factor of ten in particle number. For example, the calculation with 10^4 particles predicts a delay ignition time for about 7 ps comparing with the calculation with 10^6 particles, and the calculation with 10^5 particles sits

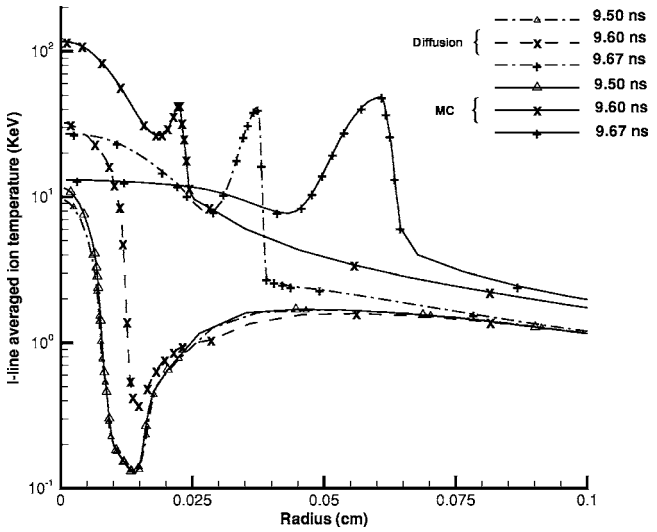

FIG. 10. A comparison of burn waves between the Monte Carlo and diffusion method at different times. Earlier ignition for Monte Carlo can be seen.

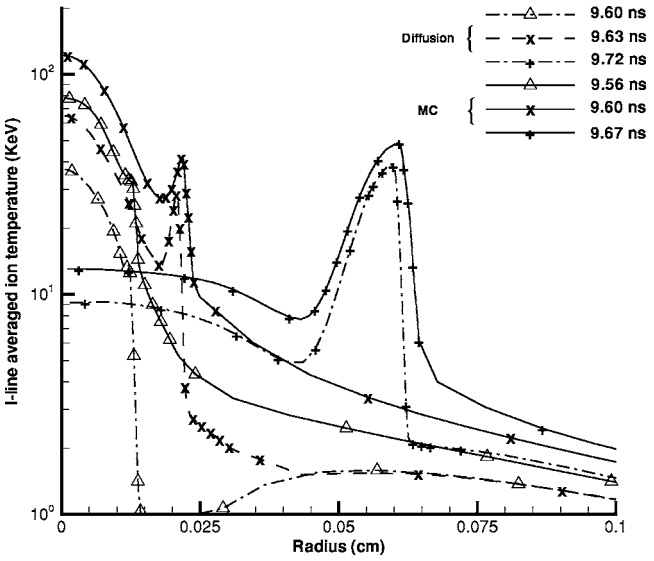

FIG. 11. The timing offset for ignition and burn wave by matching the time at which the target is ignited, the thermal burn wave is appearing and the strong burn wave propagates. Highest ion temperature at the center of hot spot achieved in the Monte Carlo method is 110 keV, while it is 80 keV in the diffusion method.

halfway between them. While this is not a rigorous convergence proof, the trend shows that the gap in ignition time between the Monte Carlo results and the diffusion result is increasing with increased number of Monte Carlo particles. In the time between 9.56 ns and 9.58 ns the gain factor differs by a factor of 2 by using different number of Monte Carlo particles. This shows the burn dynamics is very sensitive to the energy deposition. We postulate the statistical effect inherent in the Monte Carlo simulation is reflected in the energy deposition and therefore the plasma ion temperature.

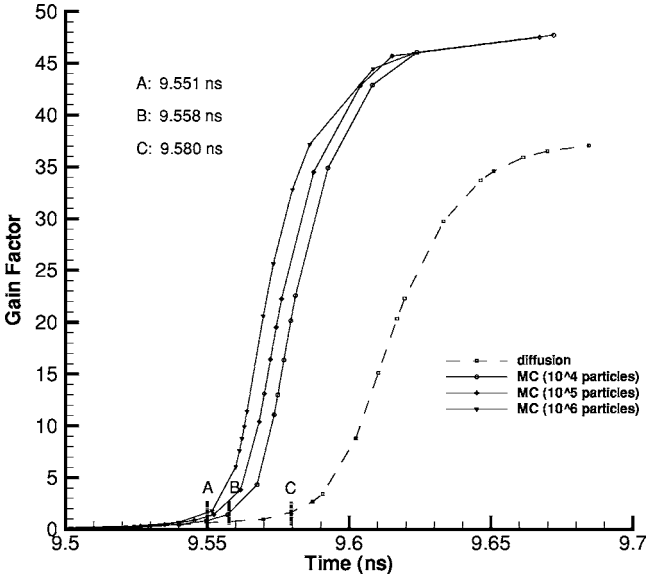

FIG. 12. A comparison of gain factor between the Monte Carlo and diffusion method. Ignition time: Monte Carlo with 10^6 particles (9.551 ns); Monte Carlo with 10^4 particles (9.558 ns); diffusion (9.58 ns).

A slightly different ion temperature at the ignition time may result in a larger difference in the reaction rates, which is fed into the next time cycle. The fourth calculation is performed using the diffusion method. The gain factor is 37, which is 21% lower than using the Monte Carlo method where it is 47.

V. SUMMARY AND DISCUSSION

A scheme for the Monte Carlo tracking of fusion burn product alpha particles on a Lagrangian mesh has been presented. Since the material properties such as temperature and density vary from cell to cell, detailed cell-by-cell tracking is required for energy deposition. The initial positions and directions are randomly and uniformly selected. The particle weights are assigned according to the number of generated alpha particles. Particle trajectories are obtained by solving for the intersection with the cell edges represented by a quadratic equation. The trajectories are assumed as straight lines since the charged alpha particles are highly energetic. Various aspects appearing in the coupling of Monte Carlo tracking with Lagrangian hydrodynamics, such as mesh moving and rezoning are discussed. The algorithm for determination of the particle location after the mesh moves or after rezoning is simplified by virtue of the CFL hydrodynamic stability requirement so that only neighbors are searched to locate the particle.

A test case is presented to study the effect on the target ignition by adopting the Monte Carlo method for alpha particle transport. In this test case, a non-uniform laser irradiation to the DT capsule is applied at l mode equal to 6 with amplitude of 1% variation. Comparing with the results from the alpha particle flux-limited diffusion transport method, we find the Monte Carlo energy deposition method predicts ignition time 30 ps earlier than the diffusion deposition method, and generates higher hot spot temperature. We observe the same effect when the uniform laser irradiation is used.

In a comparison between diffusion and Monte Carlo transport approximations the question of which is the intrinsically more appropriate model is always an issue. In the case of high energy alpha particle slowing down in DT plasmas with the temperature, density and spatial scale addressed here, the diffusion approximation is found to be an intrinsically less accurate approximation than Monte Carlo transport for the following reasons. The Monte Carlo approximation discussed in this paper directly simulates the angular and energy dependence of the alpha particles. The angular dependence is treated directly by choosing random directions of travel at birth from fusion. The energy dependence is treated using a continuous slowing down model that numerically mimics the continuous slowing down theory that underpins it. The transport itself is independent of the mesh distortion. The transport is approximated as straight lines, which accurately reflects the travel of the alpha particles so long as small angle scattering is the predominant energy loss mechanism. This is true until their velocity becomes comparable to the electron thermal velocity, at which point the alphas have lost a large percentage of their energy. Scattering of alphas

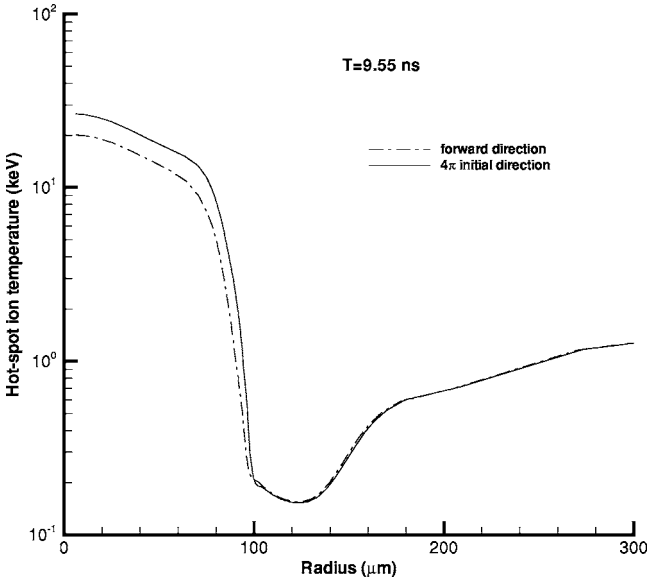

FIG. 13. A comparison of ignition ion temperatures at 9.55 ns using two different direction selection methods in the Monte Carlo transport. Dash-dotted line: particles are allowed to have only outward directions. Solid line: directions of particles are isotropic. The comparison is used to illustrate the importance of the inward traveling deposition which we suspect contributes to the higher ion temperature at the center of hot spot.

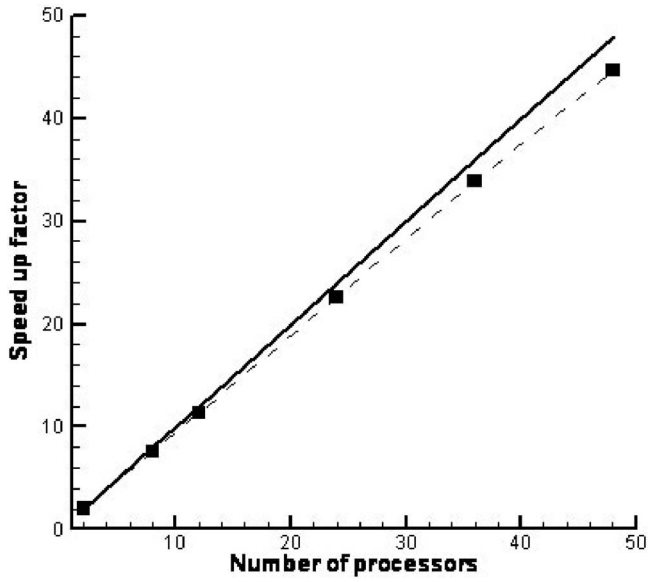

FIG. 14. The speed-up factor for the Monte Carlo transport running in parallel on a Linux cluster.

on ions contributes to large angle collisions, but these occur at the end of the alpha particle range and contribute little to the energy deposition. The most significant approximation in the Monte Carlo approach is of a statistical nature, and numerical tests show that enough Monte Carlo particles have been used in the reported simulations to support the conclusions of the paper.

In the diffusion approximation to the transport equation, the angular dependence of the alpha particle distribution function is first order in angle. The particles must be nearly isotropic. The isotropy of the particle distribution function in neutral particle transport is maintained by significant large angle scattering. This is lacking in alpha particle transport, where the particles travel in straight lines. Isotropy is often inferred by comparing the particle mean-free-path to relevant scale lengths of the transport medium. In the case of alpha particles, the range is approximately 0.35 g/cm^2 measured in density independent units. This is by definition the extent of the hot spot necessary for ignition. The total fuel radius is approximately 1 g/cm^2. Thus the system of relevance is only three times the range of the alpha particles and yet we are attempting to simulate the details of the alpha slowing down within this range. Finally, the diffusion approximation is augmented with so-called flux-limiting to address the breakdown of the approximation under the conditions found in burning DT plasma. The flux-limiting avoids causality violation where particles otherwise would propagate a signal at speeds faster than their maximum velocity, but it gives no credible information about the details of the transport. For these reasons we believe the only credible conclusion is that the Monte Carlo transport algorithm is giving more accurate transport and energy deposition results than the diffusion algorithm. Simple numerical tests of the algorithms confirm that the variations in energy deposition of the alpha particles over spatial scales of less than the alpha particle range are significantly different.

The reason that flux-limited diffusion is used in simulation codes rather than Monte Carlo transport is the substantial increase in computer time that is needed to execute the Monte Carlo algorithm in comparison to diffusion [25]. The point of this paper is to measure the differences in integrated target performance results (i.e., ignition) that are predicted using these two approximations that vary significantly in their accuracy of alpha particle transport. We see that the time of ignition (an important parameter in target design) differs significantly for the two transport models. The earlier ignition time may be explained by the fact that the alpha particles tend to lose their energy at the end of their range (the so called "Bragg peak") and thus heat the incoming main fuel sooner than the diffusion transport which incorrectly predicts a peak deposition in the middle of the range. For marginal ignition, $\rho R \approx 1$, energy deposition is sensitive to the transport model. For the case we studied in Sec. IV, the Monte Carlo method predicts 20% higher energy deposition to the hot spot plasma than the diffusion method, and thus generates higher hot spot ion temperature. Another reason that the Monte Carlo method gives higher hot spot temperature we can conjecture is that the particles having an inward direction travel a length of the diameter of the sphere. The longer the traveling length is, the more likely the particle deposits more of its energy. This is naturally simulated in the Monte Carlo method. To support this reasoning, we rerun the calculation with the particles having only outward directions. We expect that the ion temperature at the center of the hot spot would be less than the full 4π transport, since the particles will not enter the sphere defined by their initial radius. Figure 13 demonstrates this effect. At time 9.55 ns, the full 4π transport gives the temperature of the center of the hot

spot of about 28 keV while the only outward transport gives the temperature of 20 keV.

We conclude by discussing the simulation run time using the Monte Carlo transport method. An advantage of Monte Carlo particle transport is that the algorithm is "embarrassingly parallel." With negligible communication cost, the calculation can achieve linear speed-up. Figure 14 shows the linear speed-up factor by using up to 50 processors on a Linux cluster. The slight deviation from ideal linear speed-up is understandable since there are low bandwidth and low-speed switches in the cluster.

ACKNOWLEDGMENTS

One of the authors is grateful to S. Skupsky, P. B. Radha, and D. Keller for discussions. This work is performed under the auspices of the U.S.D.O.E by the Laboratory of Laser Energetics.

[1] J. Lindl, Phys. Plasmas **2**, 3933 (1995).
[2] J. J. Duderstadt, G. A. Moses, in *Inertial Confinement Fusion* (Wiley, New York, 1982).
[3] R. L. McCrory, Nucl. Fusion **44**, 123 (2004).
[4] J. S. Clarke, H. N. Fisher, and R. J. Mason, Phys. Rev. Lett. **30**, 89 (1973).
[5] G. S. Fraley, E. J. Linnebur, R. J. Mason, and R. L. Morse, Phys. Fluids **17**, 474 (1974).
[6] Inertial Confinement Fusion, ICF Annual Report, Lawrence Livermore National Laboratory.
[7] D. Mihalas, in *Stellar Atmospheres* (W. H. Freeman and Company, New York, 1978).
[8] J. E. Morel, R. M. Roberts, and M. J. Shashkiv, J. Comput. Phys. **144**, 17–51 (1998).
[9] D. S. Kershaw, J. Comput. Phys. **39**, 375 (1981).
[10] J. E. Morel, J. E. Dendy, Jr., M. Hall, and S. White, J. Comput. Phys. **103**, 286 (1992).
[11] I. Avavtsmark, T. Barkve, and T. Mannseth, J. Comput. Phys. **127**, 2 (1996).
[12] E. G. Corman, W. E. Loewe, G. E. Cooper, and A. M. Winslow, Nucl. Fusion **15**, 377 (1975).
[13] X. Berger, in *Monte-Carlo Calculation of the Penetration and Diffusion of Fast Charged Particles*, Methods in the computational Physics I (Academic Press, NY and London, 1963).
[14] P. W. McKenty, V. N. Goncharov, R. P. J. Town, S. Skupsky, R. Betti, and R. L. McCrory, Phys. Plasmas **8**, 2315 (2001).
[15] J. Yuan and G. A. Moses, *Two-Dimensional Multi-Group Diffusion Transport for Charged Particles in the ICF Target* UWFDM-1207 (2003).
[16] W. R. Martin (private communication).
[17] D. Keller, T. J. B. Collins, J. A. Delettrez, P. W. McKenty, P. B. Rahda, B. Whitney, and G. A. Moses, *DRACO-A New Multidimensional Hydrocode*, Bull. Am. Phys. Soc. 44, 3 (1999).
[18] R. L. McCrory and C. P. Verdon, *Computer Modeling and Simulation in Inertial Confinement Fusion*, Inertial Confinement Fusion, edited by A. Caruso and E. Sindoni, Verenna, 6–16 September (1988).
[19] T. J. B. Collins, *DRACO Laser Energy Deposition*, Laboratory Report 323 (2002).
[20] G. A. Moses and J. Yuan, *Radiation Diffusion in DRACO using Kershaw Difference Scheme*, UWFDM-1213, September (2003).
[21] P. B. Radha (private communication).
[22] I. P. Shkarovsky, T. W. Johnston, and M. P. Bachynski, in *The Particle Kinetics of Plasma* (Addison-Wesley, Reading, MA, 1966).
[23] Chi-Kang Li and R. D. Petrasso, Phys. Rev. Lett. **70**, 3059 (1993).
[24] S. Skupsky, Phys. Rev. A **16**, 727 (1977).
[25] G. A. Moses and J. Yuan, Bull. Am. Phys. Soc. RP110 **48**, No. 7 (2003).