Implicit Monte Carlo Method for Nonlinear Radiation Transport

J. Yuan, G.A. Moses

February 2006

UWFDM-1290

FUSION TECHNOLOGY INSTITUTE

UNIVERSITY OF WISCONSIN

MADISON WISCONSIN

Implicit Monte Carlo Method for Nonlinear Radiation Transport

J. Yuan, G.A. Moses
jyuan@cae.wisc.edu
moses@engr.wisc.edu

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

http://fti.neep.wisc.edu

February 2006

UWFDM-1290

Abstract

This report describes the physical model, algorithms and test examples in the module of implicit Monte Carlo (IMC) radiation transport for the hydrodynamic simulation code DRACO. The implementation has the following features: 1) The photon particles are equally created on multi-processors to ensure equal load balancing. Each processor is associated with a particle pool using a linked-list data structure so that storing and retrieving operations have clean and simple interfaces to the particle pool; 2) The Fleck and Cummings (FC) method of IMC is adopted in which the implicity is given by the parameter f; 3) The time independent (adiabatic) transport option is provided so that the photons are not censused when the hydro time is reached. This option may be adequate for conditions where the radiation is not strongly coupled with the plasma; 4) Up to 1000 energy groups can be used for the photon transport, since the speed of the calculation is not affected by the size and complexity of the tabulated cross section data; however, a large amount of memory is required. We have done a number of test problems to ensure the correctness of the implementation, such as the FC Marshak waves, the static problem, a hot square in a cold surrounding plasma, equilibrium distribution for infinite medium, etc. The IMC results from the above tests are physically reasonable and have been reported previously. In this report, we will focus on the FC Marshak wave problem to address several issues related to the IMC such as computer memory usage and timing in time dependent or adiabatic simulations, statistical noise and parallel speedup. Simulations for realistic ICF targets will be presented in other reports.

I. INTRODUCTION

In radiation hydrodynamics, the radiation transport is coupled with the fluid when the fluid temperature increases since the radiation energy density varies as the fourth power of the temperature. At a very high temperature, a condition that usually occurs in Inertial Confinement Fusion (ICF) studies, the energy density of the radiation field becomes sufficiently large that radiation transfer is a dominant heat transfer mechanism. Because of the complexity and difficulties in solving these coupled radiation and material equations using conventional deterministic approaches, the Monte Carlo method provides an accurate and highly feasible computational tool for these nonlinear radiation problems.

In the Monte Carlo method, photons are created in the mesh zones at the beginning of the time step according to the material thermal emission. Then they are followed through the zones, and heat the material according to the radiation absorption. The temperatures are updated at the end of the time step. This algorithm works when the radiation is weakly coupled with the material. However, when the radiation is strongly coupled with the material, this method will be unstable when the time steps increase to a point that at this time interval, the amount of the radiation and material energy exchange is able to change the material temperature by a large amount. If the temperature is used for the emission term at the beginning of the time step, the material does not radiate and only absorbs the radiation energy and thus the instability occurs.

The IMC method works by using the material equation to estimate the future material temperature, and using this estimate in the transport equation. In other words, in the integration of the material equation from t^n to t^{n+1}, the integrands are approximated by the mean-value theorem and the centered value $u_r(r,t)$ is approximated as a linear combination of t^n and t^{n+1} time step values. It is shown that it is equivalent to reducing the absorption opacity by a factor of f and adding an equal amount of isotropic scattering with effective scattering cross section $(1-f)\sigma_s$. When f is small (Δt is large), the photons being absorbed are quickly reemitted, and most

of the absorption opacity is replaced by an effective isotropic scattering opacity.

The IMC treatment of nonlinear radiation transport leads to a significant improvement in stability, accuracy and computational efficiency over the explicit conventional Monte Carlo treatment. For the basis of the IMC method, we closely follow the work of FC. However, we are handling 2D geometries, actually 3D tracking in a 2D cylindrical geometry while only 1D geometry is handled in the FC paper. In Section II, we give the coupled nonlinear radiation transport equations. In Section III, we introduce the FC IMC method and derive the IMC equations by replacing the exponentials in the emission term by the first-order expansion. In Section IV, we describe the IMC algorithm, code structure, data structure support, and input deck and its usage. Finally, in Section V, we give numerical results for the FC Marshak wave problem run in 2D.

II. NONLINEAR RADIATION TRANSPORT EQUATIONS

The time-dependent radiation transport equation including scattering is in the form [1]

$$\frac{1}{c}\frac{I(\vec{r},\vec{\Omega},\nu,t)}{\partial t}+\vec{\Omega}\cdot\nabla I(\vec{r},\vec{\Omega},\nu,t)+\mu_t(\nu)I(\vec{r},\vec{\Omega},\nu,t)=\mu_a(\nu)B(\nu) + \int\int\frac{\nu}{\nu'}\mu_s(\nu'\rightarrow\nu,\vec{\Omega}\cdot\vec{\Omega}')I(\vec{r},\vec{\Omega}',\nu',t)d\nu' d\vec{\Omega}', \tag{1}$$

where ν is the frequency, $I(\vec{r},\vec{\Omega},\nu,t)$ is the specific intensity and $B(\nu)$ is the Planck function. Since local thermodynamic equilibrium (LTE) is assumed in the above transport equation,

$$B(\nu)=\frac{2h\nu^3}{c^2}\frac{1}{e^{h\nu/kT}-1}, \tag{2}$$

$\mu_t(\nu)$ is the total attenuation coefficient

$$\mu_t(\nu)=\mu_a(\nu)+\int\int\mu_s(\nu\rightarrow\nu',\vec{\Omega}\cdot\vec{\Omega}')d\nu' d\vec{\Omega}', \tag{3}$$

and $\mu_a(\nu)$ is the absorption coefficient including induced effects,

$$\begin{aligned}\mu_a(\nu)&=\frac{\mu'_a(\nu)}{1+c^2B(\nu)/2h\nu^3}\\ &=\mu'_a(\nu)(1-e^{-h\nu/kT}).\end{aligned} \tag{4}$$

The transport equation is coupled to the energy balance material equation

$$\frac{\partial u_m(\vec{r},t)}{\partial t} = \int\int \mu_t(\nu) I(\vec{r},\vec{\Omega},\nu,t) d\nu d\vec{\Omega} - 4\pi \int \mu_a(\nu) B(\nu) d\nu$$
$$- \int\int\int\int \frac{\nu}{\nu'} \mu_s(\nu' \to \nu, \vec{\Omega}\cdot\vec{\Omega}') I(\vec{r},\vec{\Omega}',\nu',t) d\nu d\vec{\Omega} d\nu' d\vec{\Omega}' + S(\vec{r},t), \tag{5}$$

where $u_m(\vec{r},t)$ is the material energy density, which is related to the temperature $T(\vec{r},t)$ through the equation of state,

$$u_m(\vec{r},t) = \gamma(\vec{r},t) T(\vec{r},t), \tag{6}$$

where $\gamma(\vec{r},t)$ depends on the material properties such as pressure, specific heat capacity, etc.

The above equations are exact without any kind of approximation. The mathematical derivation of the basic equations in the IMC method starts with a crucial variable substitution, that is,

$$\frac{\partial u_m(\vec{r},t)}{\partial u_r(\vec{r},t)} = \frac{1}{\beta(\vec{r},t)}, \tag{7}$$

where $u_r(\vec{r},t)$ is the equilibrium radiation energy density defined in terms of the Planck function as

$$u_r(\vec{r},t) = \frac{4\pi}{c} \int B(\nu) d\nu = aT^4 \tag{8}$$

with

$$a = \frac{8k^4\pi^5}{15c^3h^3}. \tag{9}$$

Expressing the emission term in the transport equation in terms of the equilibrium radiation energy

$$\mu_a(\nu) B(\nu) = \frac{c}{4\pi} \mu_a(\nu) b_\nu u_r(\vec{r},t), \tag{10}$$

where b_ν is the normalized Planck function, and defining the Planck function weighted absorption coefficient as

$$\sigma_p(\vec{r},t) = \int b_\nu \mu_a(\nu) d\nu, \tag{11}$$

the transport and material equations used in the IMC method are as follows:

$$\frac{1}{c}\frac{I(\vec{r},\vec{\Omega},\nu,t)}{\partial t} + \vec{\Omega}\cdot\nabla I(\vec{r},\vec{\Omega},\nu,t) + \mu_t(\nu) I(\vec{r},\vec{\Omega},\nu,t) = \frac{c}{4\pi}\mu_a(\nu) b_\nu u_r(\vec{r},t)$$
$$+ \int\int \frac{\nu}{\nu'} \mu_s(\nu' \to \nu, \vec{\Omega}\cdot\vec{\Omega}') I(\vec{r},\vec{\Omega}',\nu',t) d\nu' d\vec{\Omega}', \tag{12}$$

$$\frac{1}{\beta(\vec{r},t)}\frac{\partial u_r(\vec{r},t)}{\partial t} = \int\int \mu_t(\nu) I(\vec{r},\vec{\Omega},\nu,t) d\nu d\vec{\Omega} - c\sigma_p(\vec{r},t) u_r(\vec{r},t) d\nu$$
$$- \int\int\int\int \frac{\nu}{\nu'} \mu_s(\nu' \to \nu, \vec{\Omega}\cdot\vec{\Omega}') I(\vec{r},\vec{\Omega}',\nu',t) d\nu d\vec{\Omega} d\nu' d\vec{\Omega}' + S(\vec{r},t). \quad (13)$$

With the relation between the material energy density and the temperature [Eq.(6)], and the relation between the material energy density and radiation energy density [Eq.(7)], the closure form of basic equations in the IMC method is defined.

III. THE IMC METHOD BY FLECK AND CUMMINGS

The IMC equations by FC are derived by making two distinct approximations. In the integration of the material equation from time t^n to t^{n+1}, the integrands are approximated by the mean-value theorem and the centered value $u_r(\vec{r},t)$ is approximated as a linear combination of beginning and ending time step values,

$$u_r(\vec{r},t) = \alpha u_r(\vec{r},t_{n+1} + (1-\alpha) u_r(\vec{r},t_n). \quad (14)$$

The above averaged radiation energy density is substituted back in the right-hand-side of the transport equation, and the second approximation is made there: the time-centered radiation energy density is replaced by its instantaneous value. In the absence of scattering, the resulting radiation transport equation has the form [2]

$$\frac{1}{c}\frac{I(\vec{r},\vec{\Omega},\nu,t)}{\partial t} + \vec{\Omega}\cdot\nabla I(\vec{r},\vec{\Omega},\nu,t) + \mu_a(\nu) I(\vec{r},\vec{\Omega},\nu,t) = f\frac{c}{4\pi}\mu_a(\nu) b_\nu u_r(\vec{r},t)$$
$$(1-f)\mu_a(\nu) b_\nu \frac{1}{4\pi\sigma_p}\left\{\int\int \mu_a(\nu') I(\vec{r},\vec{\Omega},\nu',t) d\nu' d\vec{\Omega} + S(\vec{r},t)\right\}, \quad (15)$$

where the parameter f is defined

$$f = \frac{1}{1+\alpha c\beta\sigma_p\Delta t}. \quad (16)$$

Another way to derive FC's IMC equation is to replace the exponentials in the emission term by the first-order expansion. Here is the summary of the derivation.

The solution of Eq. (13) is given by

$$u_r(\vec{r},t) = u_r(\vec{r},t^0) exp(-\int_{t_0}^{t} c\beta\sigma_p dt') + \int_{t_0}^{t} dt' \beta(\vec{r},t') exp(-\int_{t'}^{t} c\beta\sigma_p dt'')$$
$$\left\{\int\int \mu_a(\nu) I(\vec{r},\vec{\Omega},\nu,t') d\nu d\vec{\Omega} + S(\vec{r},t')\right\}, \quad (17)$$

where $u_r(\vec{r}, t^0)$ is an initial condition at t^0. The radiation transport equation is obtained in the absence of scattering by substituting the solution for $u_r(\vec{r}, t)$ in the source term,

$$\frac{1}{c}\frac{I(\vec{r}, \vec{\Omega}, \nu, t)}{\partial t} + \vec{\Omega} \cdot \nabla I(\vec{r}, \vec{\Omega}, \nu, t) + \mu_a(\nu) I(\vec{r}, \vec{\Omega}, \nu, t) = \frac{c}{4\pi}\mu_a(\nu) b_\nu \{u_r(\vec{r}, t^0) exp(-\int_{t_0}^{t} c\beta\sigma_p dt') + \int_{t_0}^{t} dt' \beta(\vec{r}, t') exp(-\int_{t'}^{t} c\beta\sigma_p dt'') \left[\int\int \mu_a(\nu) I(\vec{r}, \vec{\Omega}, \nu, t') d\nu d\vec{\Omega} + S(\vec{r}, t')\right]\}. \quad (18)$$

The emission source term consists of the intensity I, the external heat source S and the initial condition $u_r(\vec{r}, t^0)$. Consider the exponential $exp(-\int_{t_0}^{t} c\beta\sigma_p dt')$, assuming the product $\sigma\beta$ is almost constant in the time interval t^n and t^{n+1} and applying the mean-value theorem,

$$exp(-\int_{t_0}^{t} c\beta\sigma_p dt') = exp(-\alpha c\beta\sigma_p \Delta t), \quad (19)$$

where α is a time-centering parameter in the range of 0 and 1, which represents the degree of implicitness. Further assuming $\alpha c\beta\sigma_p\Delta t$ is a small value, the exponential can be transformed to the form

$$exp(-\alpha c\beta\sigma_p \Delta t) \approx \frac{1}{\alpha c\beta\sigma_p \Delta t}, \quad (20)$$

which is the f-parameter in Eq. (16). Applying the same argument in the second term on the right-hand-side of Eq. (18) and approximating $\int_{t_0}^{t} dt'\beta(\vec{r}, t')$ by $\beta\Delta t$, the same equation as Eq. (15) can be obtained.

IV. IMC CODE STRUCTURE AND DATA STRUCTURE SUPPORT

A. IMC Code Structure

The IMC has been implemented in the subtree of *rt_imc* with 14 new subroutines and several module subroutines. Other affected or modified subroutines include the initialization subroutines and thermal transport subroutines (see source codes for the details and the listing of IMC subroutines in Table 1).

The main structure of the radiation transport is as follows. In the subroutine *rtimc_control.f90*, The equilibrium radiation energy density is first updated according to Stefan's Law, then the IMC parameters such as β, f and Planck mean absorption cross section are calculated. Note the previous temperature is used for the evaluations. After these IMC parameters have been calculated, the photon particles are created randomly in the subroutine *rtimc_photon_source.f90*. The energy of volume source particles is given by

$$e_{j-1/2} = E_{j-1/2}/N_{j-1/2}, \tag{21}$$

while the energy of surface source particles is given by

$$e_s = \frac{ac}{4} T_0^4 \Delta t / N_s, \tag{22}$$

where a is the radiation constant, $N_{j-1/2}$ is the number of source particles created in zone j and N_s is the number of source particles created on the surface. $E_{j-1/2}$ is the source energy radiated in zone j, given by

$$E_{j-1/2} = f\sigma_p c u_{rj-1/2} \Delta x \Delta t + (1-f) S_{j-1/2} \Delta x \Delta t. \tag{23}$$

The code segment used to generate sampled photons for a surface source is listed in Figure 1, and the code segment for the volume source is listed in Figure 2.

The full 3D Monte Carlo tracking is then followed after the sampled photons. The photon to be transported is retrieved from the photon pool by calling function *get_curr_pused_indx_pool*(). First, the correct zone index is searched near the neighbor zones to ensure the photon resides in the correct physical zone, then the photon travel distances are calculated for the boundary crossing, for a collision event and to census time. The distance to a boundary crossing is geometry dependent. Two kinds of geometry are supported in the current implementation, that is, planar 2D and cylindrical 2D, in subroutines *rtimc_planar_dist_2d.f*90 and *rtimc_cylind_dist_3d.f*90, respectively. The distance to a scattering collision is given by

$$d_c = \frac{|ln\xi|}{(1-f_{j-1/2})\sigma_\nu}, \tag{24}$$

and the distance to census is,

$$d_s = c(t^{n+1} - t). \tag{25}$$

If the photon is crossing the zone boundary or it is censused, the photon is advanced according to [2]

$$\begin{aligned} x^{'} &= x + \mu d, \\ t^{'} &= t + d/c, \\ \nu^{'} &= \nu, \\ \mu^{'} &= \mu, \\ E^{'} &= Ee^{-f\sigma_\nu d}. \end{aligned} \tag{26}$$

The photon's energy loss $\Delta E = E(1 - e^{-f\sigma_\nu d})$ is added to the total radiation energy in the zone. The cutoff photon energy is chosen as 1% of the photon's energy at birth. The photon is eliminated if its energy is lower than the cutoff energy and its remaining energy is deposited in the zone. If the photon is crossing the zone boundary, a new plasma condition is calculated, and the geometry routine is reentered. If the photon is censused, its data is stored in the photon pool for the next integration cycle.

If the photon experiences a scattering collision, $x^{'}, t^{'}$, $E^{'}$ and energy deposition are advanced as before. A different photon frequency $\nu^{'}$ is resampled from the volume source frequency distribution, and also a random direction is re-sampled. A new zone is then reentered.

Finally, the material temperature is updated according to the equation

$$T^{n+1}_{j-1/2} = T^{n}_{j-1/2} + \frac{1}{\beta(T^{n}_{j-1/2})} \left[E_{j-1/2}/\Delta x - f_{j-1/2} c \Delta t \sigma_{pj-1/2} u_{rj-1/2} + f_{j-1/2} S_{rj-1/2} \Delta t \right]. \tag{27}$$

Note that IMC solves the coupled equations of radiation transport and the material equation. As we know, the material equation is formulated as part of the thermal electron transport in DRACO. In order to incorporate the IMC, we use operator splitting, that is, the IMC radiation updates the electron temperature to an intermediate

value, and then the thermal transport updates this median value to n+1 time step without the radiation source.

An approximation, called adiabatic transport, has also been implemented in DRACO. In this approximation, the photon is transported time independently, that is, the photon will not be censused even if the hydro time step limit is reached. Under certain conditions, this approximation is valid if the radiation and material evolution is not strongly coupled. This assumption is based on the condition that during the photon life time the underlying plasma is under small changes in terms of temperature and density. The implementation of the adiabatic transport is similar to the above described time dependent transport, except that the photon travel distance is not bounded by the census distance. This approximation will significantly reduce the amount of memory used for storing the censused particles although the computing time is basically the same.

B. Data Structure Support

The basic data structure is in the module file *module_rtimc.f*90. It takes advantage of the Fortran 90 object data type capability. The photon type is defined as

```
type :: photon_t
    type(particle_t) :: particle
    real(real_kind) :: freq
end type photon_t
```

The *particle_t* object type is further defined as shown in Figure 3. This type abstracts the basic information for a particle. Actually, this abstract is also shared in the alpha charged particle transport.

One of the challenges for parallelization for the particle tracking time-dependently is how to store the tracking temporal particle information. Here is the data structure implemented in DRACO. A pool built with a linked list is associated with each processor and the capacity of each pool is a user-defined parameter. Retrieving and storing particles are manipulated through two linked lists, one for free indices and

the other for used indices. The advantage of doing so is it provides a simple and clean interface for connecting the particle pool. This method is also used in alpha particle transport. All the data structures are collected in the subtree code tree, such as *alpha_pool* and *photon_pool*. The operations that access the data pool are through the interfaces, as shown in the following.

public :: *init_photon_pool*

public :: *get_pfree_indx_pool*

ublic :: *put_pused_indx_pool*

public :: *put_pfree_indx_pool*

public :: *count_pfree_indx_pool*

public :: *count_pused_indx_pool*

public :: *print_pfree_indx_pool*

public :: *print_pused_indx_pool*

public :: *start_sweep_pused_indx_pool*

public :: *sweeping_pused_indx_pool*

public :: *is_end_pused_indx_pool*

public :: *get_curr_pused_indx_pool*

public :: *remove_curr_pused_indx_pool*

public :: *check_photon_pool*

public :: *photon_debug_plot*

A diagram of the photon pool allocated on parallel machines is shown in Figure 4.

C. IMC Input Deck and Usage

A separate input block is added in the *sim_input.txt* file besides several global radiation control parameters in input block *radtins*. Four input parameters in input block *radtins* are relevant and important for radiation IMC. In order to use IMC, the parameter *radiation_schema* must be set to ÍMC. The number of frequency groups is set by parameter *number_of_radiation_freq_groups*. The parameter *rad_trans_energy_limit* is used to decide whether the IMC radiation will be

invoked or not. Another opacity table format option is also added for the IMC. In order to use the radiation IMC, the *opacity_table_format* must be set as *YAC* because current LLE opacity files do not have the cross section data in them. To run the IMC, the cross section itself, the absorption probability and the mean Planck opacity are required. These data are calculated by the YAC code in a format that is slightly different from the standard LLE opacity format.

The separate input block *rtimcins* is exclusively for the radiation IMC. The size of the photon pool is set by parameter *photon_pool_capacity*. This will pre-allocate memory for the whole simulation on each processor. The photons are dynamically created and destroyed in the pool. The parameter *photon_pool_number* controls the number of newly created photons emitted in each time step on each processor. The size of *photon_pool_capacity* must be very much larger than *emit_photon_number* in order to run a large time step. For example, if *photon_pool_capacity* is set to 1,000,000, and *emit_photon_number* is set to 1,000 on 4 processors, the total number of 4,000,000 photons will be allocated on these 4 processors and total number of 4,000 photons will be emitted for each time step on these 4 processors. The estimated number of time steps that could be supported is about 1000. The parameter *fleck_alpha* is the implicity from Fleck and Cumming's method. The value should be set in the range 0.5 to 1. The default value is 1. The parameter *cv_type* is for the specific heat and the parameter *cs_type* is for the cross section type. For realistic calculations, *cv_type* and *cs_type* muse be set to zero. The boundary conditions are set from parameters

rtimc_bc_ism,

rtimc_bc_iep,

rtimc_bc_jsm,

rtimc_bc_jep.

The value of these parameters can be $'none'$ or $'zero'$, which have the same meaning as for the diffusion transport.

More detailed information about the IMC input deck is given in Tables 2-4.

V. NUMERICAL TESTS

We have done a number of test problems to ensure the correctness of the implementation, such as the FC Marshak waves, the static problem, a hot square in a cold surrounding plasma, equilibrium distribution for infinite medium, etc. In this report, we will only focus on the FC Marshak wave problem. We use this test problem to address several issues related to computer memory usage and timing in time dependent or adiabatic simulations, statistical noise and parallel speedup. Simulations for realistic ICF targets and comparison with the diffusion simulations will be presented in other reports.

In the FC Marshak wave problem, a slab with thickness of 4 cm is heated by a 1 keV blackbody source at the left side x=0, and the macroscopic cross section has an analytical form as given by

$$\sigma_v = \frac{27}{\nu^3}(1 - e^{-\nu/T}) \text{ cm}^{-1}, \tag{28}$$

where ν and temperature T are measured in keV. This cross section has a value of mean free path of 1 cm for ν=3 keV, which is approximately the frequency for which a 1 keV blackbody spectrum peaks. The temperatures in all volume zones are set equal to 10 eV initially to keep σ_p finite. The value of specific heat is taken to be $0.5917aT_0^3$. We run this 1D problem in the 2D DRACO mode. To compare with the results in FC's paper, we average the results along I-lines.

We have varied the number of sampled photons and the mesh size to study the influences on the temperature distribution. Figure 5 shows the effect of using a small number of sampled photons and a large number of sampled photons. We can see the result from a large number of photons has much smoother curves due to the reduction of statistical errors. Comparing with FC's paper for time $ct = 6$ cm, the agreement is very good.

The stability of the IMC solutions is illustrated by increasing the time step as shown in Figure 6. For the time step of 1×10^{-2} shakes (10^{-8} s), initially the

temperature near the source is slighly higher, however after the time $ct = 15$ cm the solutions become correct. For a larger time step of 2×10^{-2} shakes, the solution is incorrect near the source although the remainder of the curve looks good.

We have also studied the influence of zone sizes on the temperature distribution as done in the FC paper. We can see that decreasing the zone size does not improve the accuracy of the solution but instead introduces a slight spatial fluctuation, as shown in Figure 7. This observation also agrees well with that shown in the FC paper.

The temperature contours for this 1D problem run in 2D at three different times (0.6 ns, 1 ns and 3 ns) are shown as 3D surfaces in Figure 8. The run uses 4 cpus and each cpu transports 50,000 photon particles. The time step is set to 1.e-11 s and the code runs up to 3 ns. The statistical noise is obvious at the wavefront. However, the overall statistical error is less than 3% as shown in Figure 9. For some regions, the statistical error is around 1%.

We estimate the parallel speedup by running the problem on 1, 2 and 4 processors on UR-LLE parallel machines. The time per cycle on 1, 2, and 4 processors are 1.964 second, 1.011 second and 0.502 second, respectively. Therefore, the speedup factors for 2 and 4 processors are 1.94 and 3.91, respectively. We can see the speedup is almost linear with the number of processors.

Comparisons with the results from the adiabatic approximation are shown in Figures 10-12 for 0.6 ns, 1 ns and 3 ns, respectively. We can see the adiabatic approximation gives incorrect answers for earlier times; however, it converges to the correct solution at the equilibirum state for later times.

VI. CONCLUSIONS AND FUTURE WORK

There are some other variant forms of the IMC method [3] [4] [5], for example, Gentle uses an implicit Monte Carlo diffusion method by solving the diffusion matrix with Monte Carlo method, random walk method using a confined diffusive sphere, discrete diffusion Monte Carlo solving diffusion on some regions and symbolic Implicit method. These are topics we may study further.

Acknowledgments

This work was performed under the auspices of the U.S. DOE for the Laboratory for Laser Energetics.

---

[1] G. C. Pomraning, *The Equations of Radiation Hydrodynamics* (Pergamon Press, Oxford, 1973).

[2] J. J. A. Fleck and J. D. Cummings, J. Comp. Phy. **8**, 313 (1971).

[3] N. Gentile, J. Comp. Phy. **172**, 543 (2001).

[4] J.J.A. Fleck and E. Canfield, J. Comp. Phy. **54**, 508 (1984).

[5] T. Nkaoua, J. Sci. Stat. Comput. **12**, 505 (1991).

FIG. 1: The code segment used to generate sampled photons for a surface source

```
pdx = 0
num_check = 0
eng_check = MC_0
do k=ks, ke
   do j=js, je
      if( surface_number_r(j) <= mc_0 ) cycle
      wgtde = energy_surface_r(j)/surface_number_r(j)
      do i=1, surface_number_r(j)

         pdx = pdx + 1
         if( mod(pdx, total_mpi_procs) == id_mpi_rank ) then

            indx = get_pfree_indx_pool()
            if( indx <= 0 ) stop "Not enough photon pool allocation"
            photon => photon_pool(indx)
            part => photon%particle
            cell_indx = cell_indx_t(ie,j,k)
            call rtimc_sample_surface(1, photon, cell_indx)
            part%energy = wgtde
            part%energy_at_birth = part%energy
            part%tstep_indx = ntstep

            call put_pused_indx_pool(indx)
            num_check = num_check + 1
            eng_check = eng_check + part%energy
         end if
      end do
   end do
end do
```

FIG. 2: The code segment used to generate sampled photons for volume sources

```
do k=ks, ke
   do j=js, je
      do i=is, ie
         cellnum = cell_number(i,j,k)
         if( cellnum <= mc_0 ) cycle
         wgtde = energy_cell(i,j,k)/cellnum
         do n=1, cellnum

            pdx = pdx + 1
            if( mod(pdx, total_mpi_procs) == id_mpi_rank ) then

               cell_indx = cell_indx_t(i,j,k)

               indx = get_pfree_indx_pool()
               if( indx <= 0 ) stop "Not enough photon pool allocation"
               photon => photon_pool(indx)
               part => photon%particle
               call rtimc_sample_cell(photon, cell_indx)
               part%energy = wgtde

               part%energy_at_birth = wgtde
               part%tstep_indx = ntstep

               call put_pused_indx_pool(indx)

               num_check = num_check + 1
               eng_check = eng_check + part%energy

               if( plt_radpow ) then
                  ergs_freq = photon%freq * ev_to_ergs
                  nfg = 1
                  do ng=1, number_of_radiation_freq_groups
                     if(  ergs_freq >= freq_pts_radt(ng) .and. &
                              ergs_freq < freq_pts_radt(ng+1) ) then
                        nfg = ng
                        exit
                     end if
                  end do

                  radiation_energy_emission(i,j,k, nfg) = &
                     radiation_energy_emission(i,j,k, nfg) + wgtde

               end if

            end if

         end do
      end do
   end do
end do
```

FIG. 3: The data structure for particles in Fortran 90

```
type :: direction_t
   real(real_kind)    :: cosx       ! to x axis
   real(real_kind)    :: cosy       ! to y axis
   real(real_kind)    :: cosz       ! to z axis
end type direction_t

type :: position_t
   real(real_kind)    :: x
   real(real_kind)    :: y
   real(real_kind)    :: z
end type position_t

type :: cell_indx_t
   integer(int_kind)  :: i
   integer(int_kind)  :: j
   integer(int_kind)  :: k
end type cell_indx_t

type :: vertex_t
   integer(int_kind)  :: i
   integer(int_kind)  :: j
   integer(int_kind)  :: k
end type vertex_t

type :: edge_t
   type(vertex_t)     :: point1
   type(vertex_t)     :: point2
end type

type :: particle_t
   type(position_t)   :: position
   type(direction_t)  :: direction
   type(cell_indx_t)  :: cell

   integer            :: loc_edge

   real(real_kind)    :: energy
   real(real_kind)    :: weight
   real(real_kind)    :: time

   real(real_kind)    :: energy_at_birth

   integer(int_kind)  :: tstep_indx

end type particle_t
```

FIG. 4: Diagram of the photon pool allocated on each parallel machine

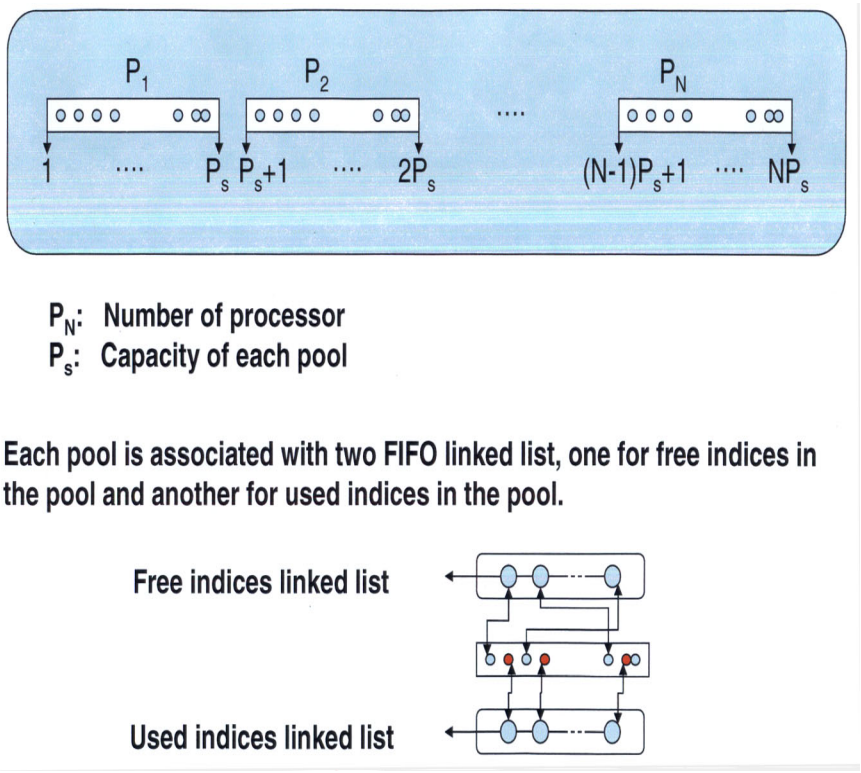

FIG. 5: Comparison with FC's result and effect of the number of sampled photons

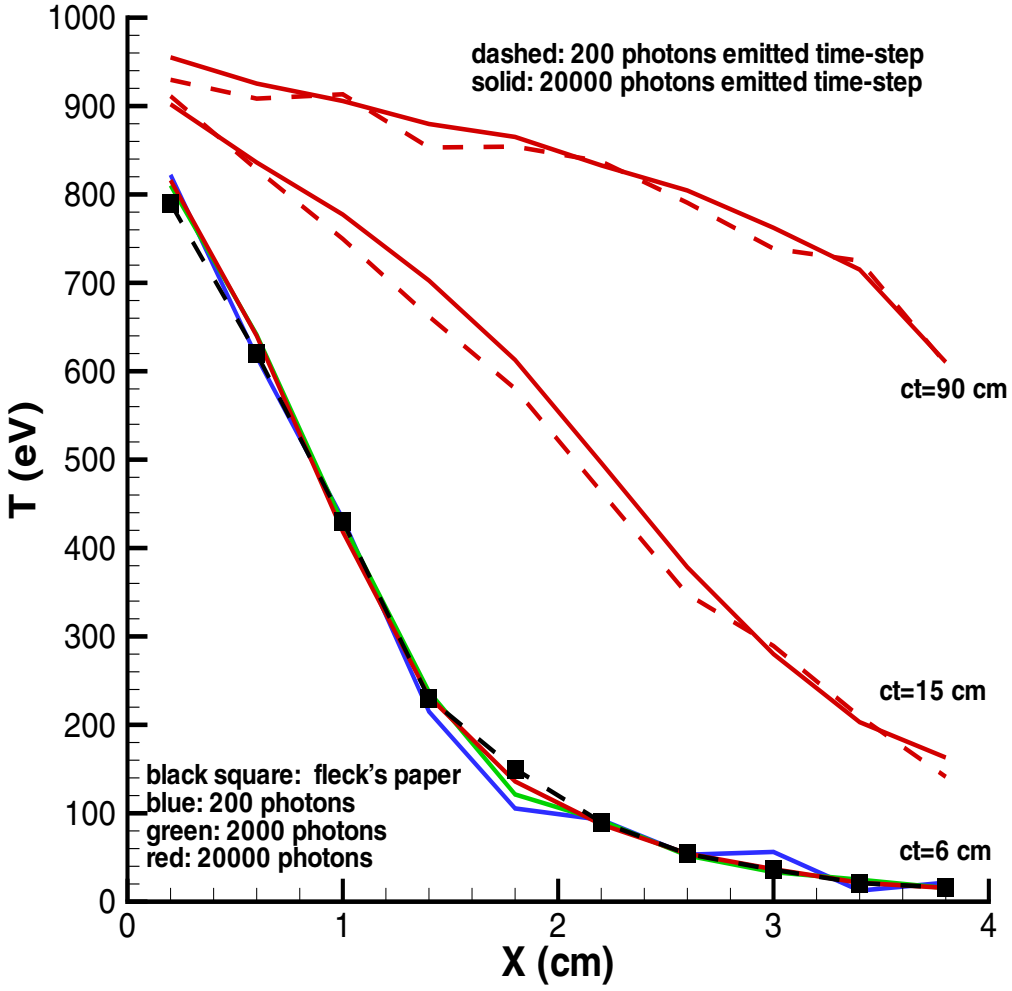

FIG. 6: Comparison between IMC temperatures for different time steps

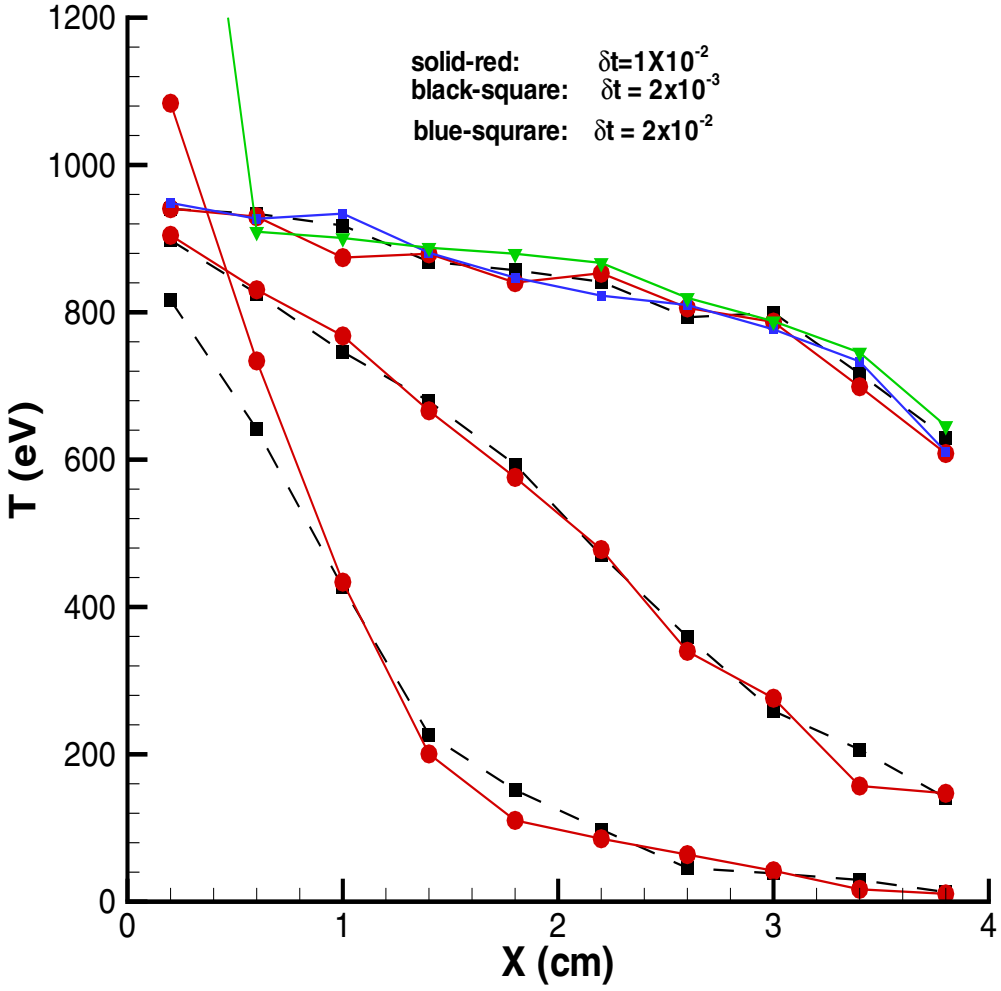

FIG. 7: Comparison between IMC temperatures for different spatial sizes

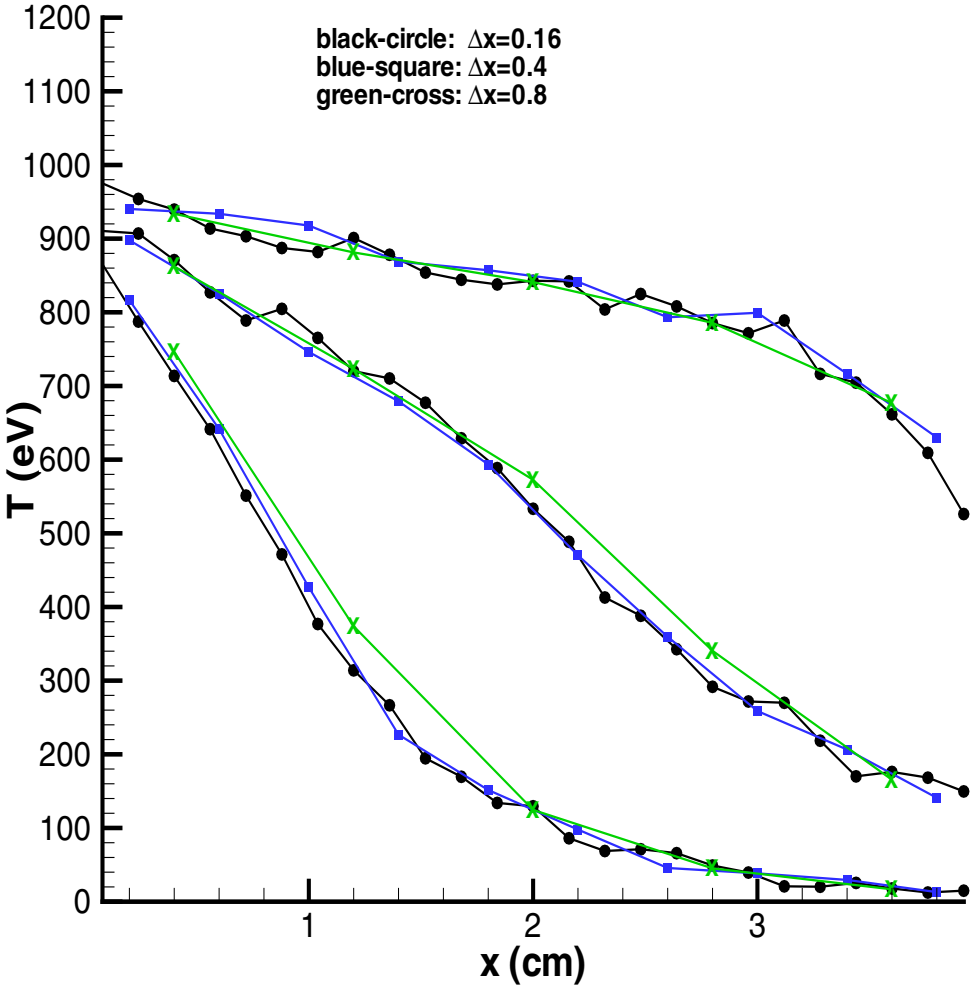

FIG. 8: The 3D temperature contour surfaces for the FC 1D problem run in 2D at three different times (0.6 ns, 1 ns and 3 ns)

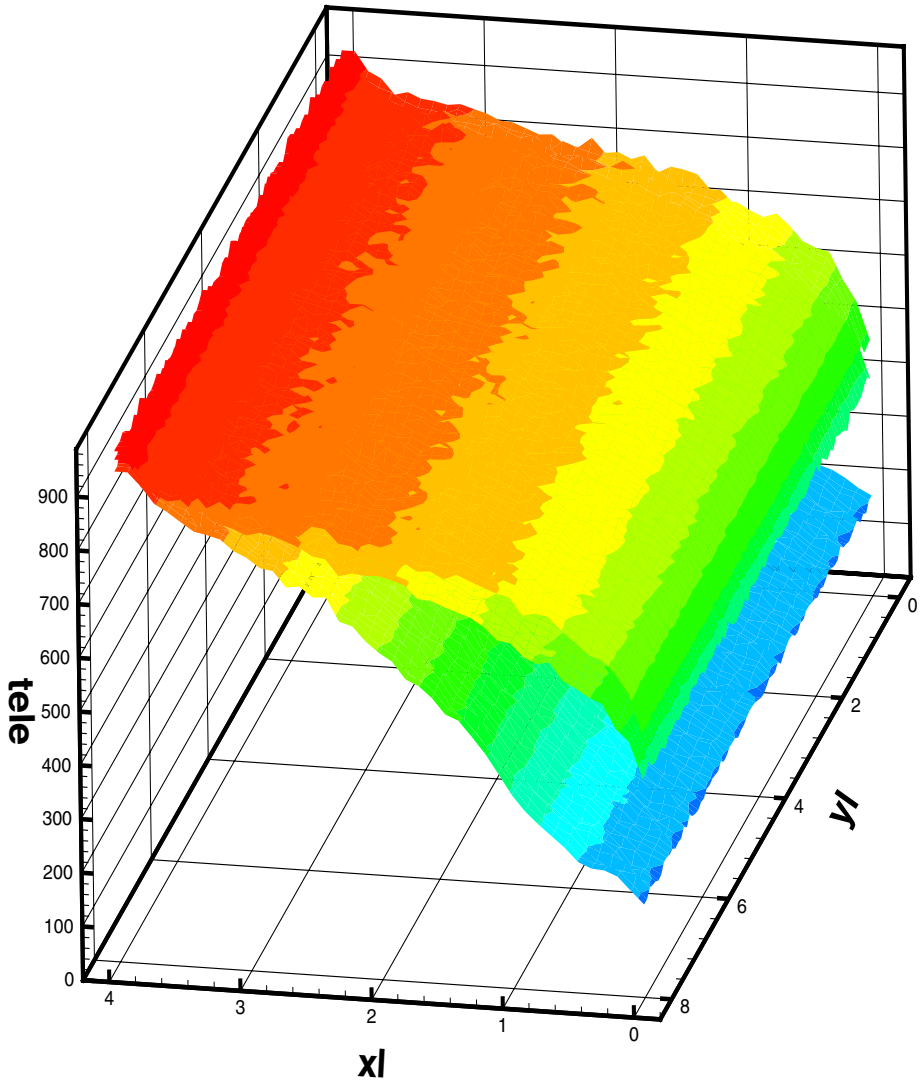

FIG. 9: Statistical error for the FC problem run in 2D

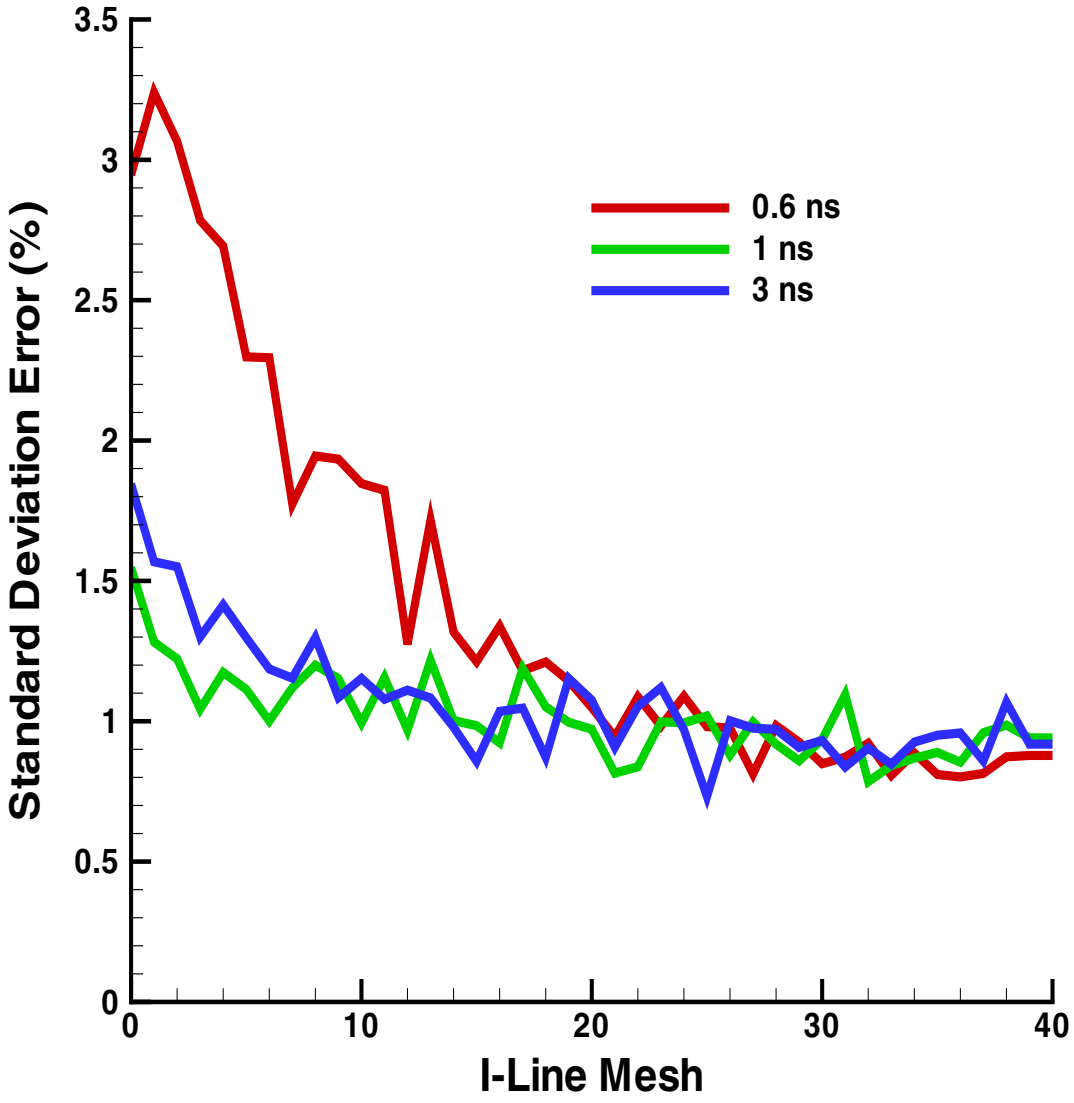

FIG. 10: Comparison with the adiabat approximation at 0.6 ns

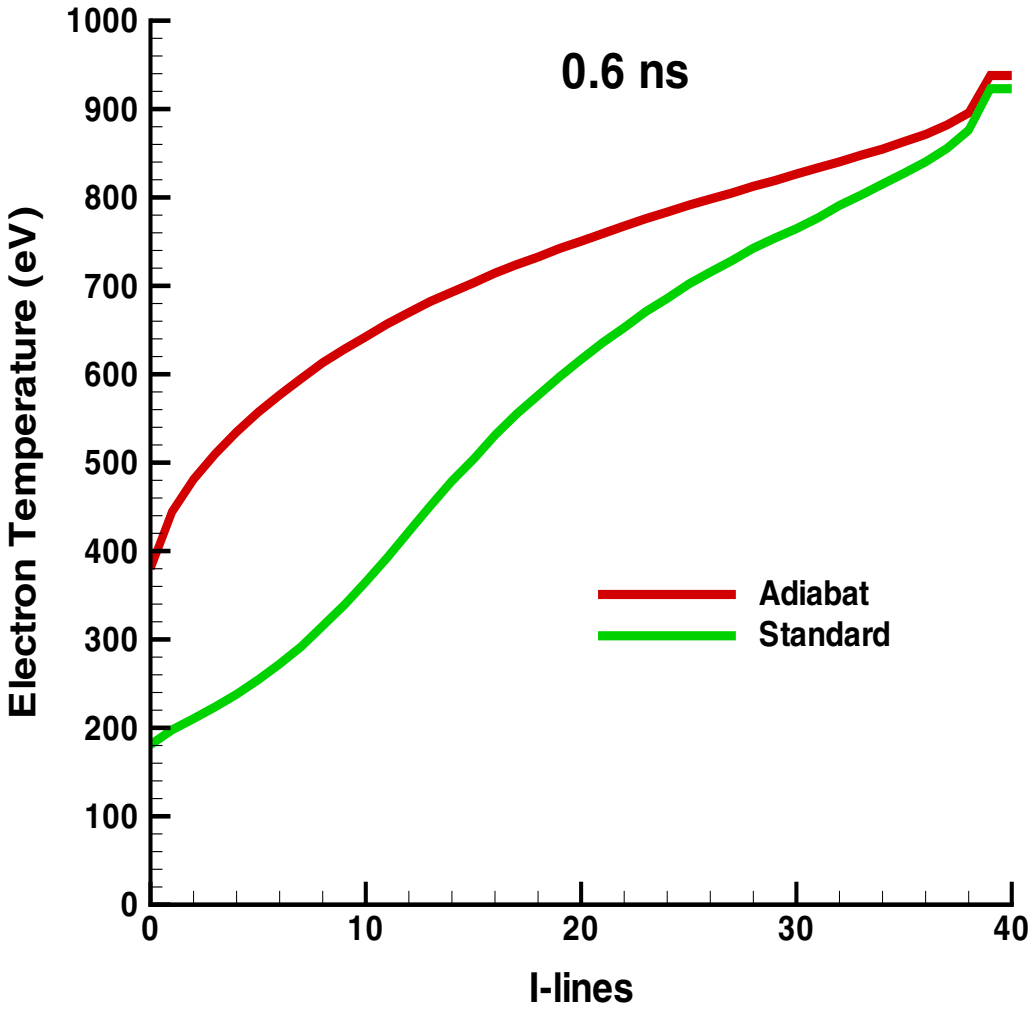

FIG. 11: Comparison with the adiabat approximation at 1 ns

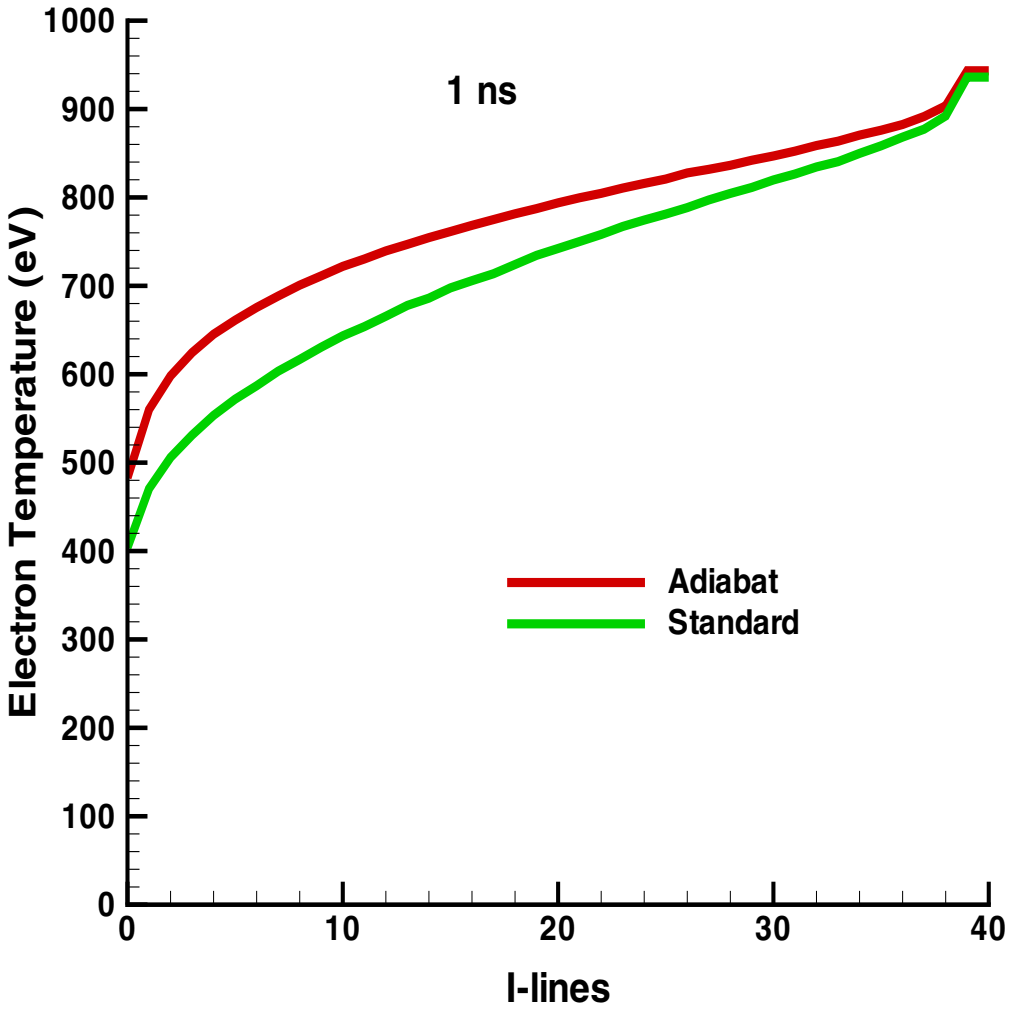

FIG. 12: Comparison with the adiabat approximation at 3 ns

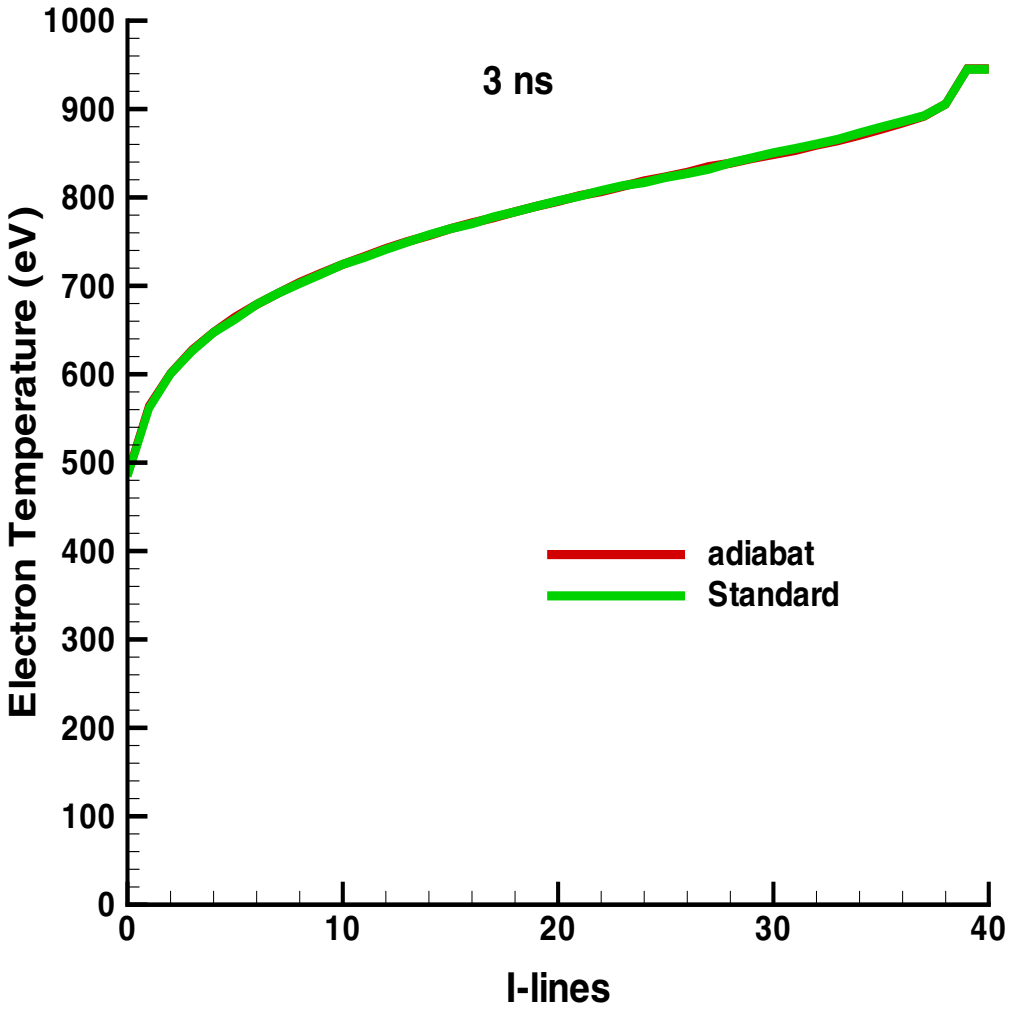

TABLE 1: Radiation IMC Subroutines

module_pfree_indx	data structure for a free queue element
module_pfree_indx_queue	data structure for a free queue
module_pfree_indx_queue_op	operations upon a free queue
module_pused_indx	data structure for a used queue element
module_pused_indx_queue	data structure for a used queue
module_pused_indx_queue_op	operations upon a used queue
module_photon_pool	data structure and operations for a photon pool
module_rtimc	module file for radiation IMC
module_particle	module file for particle transport
rtimc_cens_dist	calculate the travel distance to the census time
rtimc_coll_dist_2d	calculate the collision distance if a collison event occurs
rtimc_control	control subroutine for radiation IMC
rtimc_cross_section	calculate photon cross sections
rtimc_csetc_update	update Planck mean absorption cross section
	and other IMC parameters, such as f and β
rtimc_cylind_dist_3d	calculate the travel distance to the boundary condition
	for the cylindrical geometry
rtimc_energy_update	update material energy after the photon transport
rtimc_photon_pool_energy	check total photon energy in the pool
rtimc_photon_source	create randomly sampled photons from surface and volume
rtimc_photon_track	calculate the tracking trajectory for each photon
rtimc_planar_dist_2d	calculate the travel distance to the boundary condition
	for the planar 2D geometry
rtimc_sample_cell	sample photons in volume
sample_blackbody_planck	sample photon frequency from a blackbody distribution
rtimc_sample_frequency	sample frequency from a volume photon
rtimc_sample_surface	sample random photon from a surface

TABLE 2: Radiation IMC Namelist Options in Input Block radtins

Variables	Default	Dimension	Meaning
radiation_schema	Kershaw	1	Option for radiation transport method, set to 'IMC' for IMC transport
number_of_radiation _freq_groups	4	1	set the number according to available data files
rad_trans_energy_limit	0.	1	transport radiation if the radiation energy density is greater than this value
opacity_table_format	'LLE'	1	set to 'YAC' if radiation IMC is used The corresponding data file name is yac_imc_#group_element.txt

TABLE 3: Radiation IMC Namelist Options in Input Block rtimcins

Variables	Default	Dimension	Meaning
emit_photon_number	100	1	Number of sampled photons at each time step
photon_pool_capacity	100	1	total storage allocated on each processor. If the number of requested photons is greater , than this value the program will stop abnormally.
rtimc_method	'standard'	1	If it is set to 'standard', the program will run time dependent transport, meaning the photons will be censused If it is set to 'adiabat', the program will run time independent transport
photon_per_cell	0	1	Number of sampled photons for each computational zone if the value is set to non-zero
fleck_alpha	1.0	1	Fleck's f parameter with a range between 0.5 and 1.
cv_type	0	1	=1, specific heat for Fleck test problem =2, specific heat for Tophat problem =0 (default), real specific heat
rtimc_right_tbc	0.	1	Right boundary condition under blackbody irradiation, temperature in unit of eV

TABLE 4: Radiation IMC Namelist Options in Input Block rtimcins (continue)

Variables	Default	Dimension	Meaning
rtimc_left_tbc	0.	1	Leftt boundary condition under blackbody irradiation, temperature in unit of eV
rtimc_top_tbc	0.	1	Top boundary condition under blackbody irradiation, temperature in unit of eV
rtimc_bottom_tbc	0.	1	Bottom boundary condition under blackbody irradiation, temperature in unit of eV
rtimc_bc_ism	'none'	1	Left boundary condition, 'none' - escape boundary 'zero' - adiabatic boundary
rtimc_bc_iep	'none'	1	Right boundary condition, 'none' - escape boundary 'zero' - adiabatic boundary
rtimc_bc_jsm	'none'	1	Bottom boundary condition, 'none' - escape boundary 'zero' - adiabatic boundary
rtimc_bc_jep	'none'	1	Top boundary condition, 'none' - escape boundary 'zero' - adiabatic boundary
cs_type	0	1	=1, cross section for Fleck test problem =2, cross section for Fleck test problem temperature dependent case =3, artifical temperature dependent opacity only for test purpose =4, one group, set to Planck mean =0 (default), real cross section from tabulated cross section data