Radiation Diffusion in DRACO using Kershaw Difference Scheme

G.A. Moses and J. Yuan

September 2003

UWFDM-1213

FUSION TECHNOLOGY INSTITUTE

UNIVERSITY OF WISCONSIN

MADISON WISCONSIN

Radiation Diffusion in DRACO using Kershaw Difference Scheme

G.A. Moses and J. Yuan

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

http://fti.neep.wisc.edu

September 2003

UWFDM-1213

1. Introduction

Radiation hydrodynamics is the study of very high energy density plasma fluid systems where heat is transferred predominantly by the transfer of x-ray radiation. These systems are found in astrophysics and on earth in high energy density plasmas like those found in laser fusion targets [1]. The mathematical treatment of radiation hydrodynamics is extremely complex because it is a combination of fluid dynamics and radiative transfer. Each of these subjects themselves is very involved and in combination the problem offers no useful analytical mathematical solutions. Therefore one must immediately resort to numerical solutions to very nonlinear sets of coupled partial differential equations. The opacities used as coefficients in these equations are nonlinear functions of the plasma energy density (density and temperature), rendering the solutions even more difficult to compute. Because one is forced to simultaneously simulate the fluid motion under the influence of radiative heat transfer and the heat transfer in moving background plasma, computer resources limit the choice of radiative transfer model. For this reason, the diffusion approximation is often chosen over more accurate transport approximations to model the radiative transfer, even in circumstances where the diffusion approximation is not strictly valid. Therefore the diffusion approximation is modified with a “fix-up” called a flux limit.

The equations of radiation hydrodynamics can be written as [2]

$$\frac{\partial I}{c\partial t} - \Omega \cdot \nabla I = S(\nu) - \sigma(\nu) I + \int_0^\infty d\nu' \int_{4\pi} d\Omega' \frac{\nu}{\nu'} \sigma_s(\nu' - \nu, \Omega' \cdot \Omega) I(\nu', \Omega') \tag{1.1}$$

where we have omitted the fluid equations for the conservation of mass and momentum and only included the conservation of energy equation, because this is the fluid equation that couples directly to the radiative transfer equation. In this report we will focus only on the solution of the radiative transfer equation in the diffusion approximation.

The radiative transfer equation is treated in the multi-frequency approximation where the frequency dependence of the specific intensity is treated in a group structure with average opacities computed for each group. Each of these group equations is uncoupled from the others. In fact they are coupled through the emission term that appears in both the radiative transfer equations and the plasma energy density equation. This coupling is broken using an explicit treatment of the solution of these equations, that is the radiative transfer equations are updated from t^n to t^{n+1} using values of plasma parameters at t^n. The plasma energy is updated using an emission term evaluated at t^n and an absorption term evaluated at t^{n+1}. This time centering inconsistency allows sequential solution of these equations, but introduces errors and possibly numerical instabilities that must be controlled.

Taking moments of the radiative transfer equations with respect to their angular dependence and truncating after zeroth and first moments produces k equations and k+1 unknown functions. This can also be viewed as expanding the specific intensity in a power series in the angular variable and retaining only the zeroth and first order terms. In either case, assuming a transport law that relates the gradient of the radiation energy density to the radiation flux in the form of Fick's Law closes the resulting radiation energy density equation:

$$\vec{F} = -D\nabla I\,, \tag{1.2}$$

where D is the flux-limited diffusion coefficient. The diffusion form of the radiative transfer equation is [2]

$$\frac{\partial I_g}{c\partial t} - \nabla \cdot D\nabla I_g = 4\pi\sigma_a' B_\nu - \sigma_a' I_g \quad . \tag{1.3}$$

It is this equation, in multi-frequency form, that is solved in DRACO. The emission and the absorption terms that couple the radiation diffusion equations to the plasma energy density

equation are simply cell-wise quantities. There is no spatial finite differencing process involved in representing these quantities.

The major challenge in solving these equations is the spatial finite difference approximation to the diffusion term. This is the term that transports the radiation from one cell to another. Furthermore, DRACO [3] is a 2D lagrangian code, so the finite difference mesh is composed of quadrilateral cells in a structured mesh. By this we mean that cells map to a regular logical index mesh, but the geometric relationship of a cell and its surrounding cells can be arbitrarily distorted as shown in Figure 1. DRACO also uses either x-y or r-z coordinates. For x-y coordinates and orthogonal cell boundaries there are numerous finite difference approximations to the so-called diffusion operator. For r-z coordinates and arbitrary quadrilateral cells, there are few published finite difference schemes.

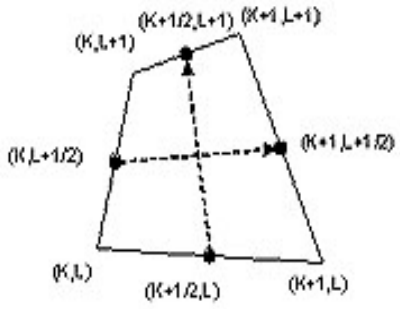


Fig. 1.1 Logical coordinates and quadrant numbering for a quadrilateral cell on a logically rectangular mesh.

David Kershaw developed the finite difference scheme that is best documented in the published literature and we hereafter refer to this as the “Kershaw scheme” [4].

In Section 2, we give a description of the Kershaw scheme by following his paper. We also discuss the boundary condition treatment and the different types of the flux limiters. Detailed information about the matrix and the corresponding symbols appearing in the diffusion equation is also given. In Section 3, we present the results of simple test problems in the contour graphs. We

also compare the numerical results to the exact solutions for several problems that have analytical solutions. In Section 4, we list some variables related to the code implementation and the input options for users.

2. Kershaw difference scheme for the diffusion equation on an arbitrary r-z quadrilateral grid

2.1 Differencing the diffusion operator [4]

To difference the diffusion operator, one observes from Fig. 1.1 that

$$f = f(R(K,L), Z(K,L)) \quad . \tag{2.1}$$

R and *Z* are the familiar Eulerian coordinates, *R* being either Cartesian or cylindrical and *Z* always being Cartesian. (The following derivations will be carried out in cylindrical coordinates.) *K* and *L* are the mesh point indices coordinates, with *K = 1, …, KMAX* and *L = 1, …, LMAX*.

The expression $\nabla \cdot D\nabla f$ in cylindrical coordinates is:

$$\frac{1}{R}\frac{d}{dR} DR \frac{df}{dR} + \frac{d}{dZ} D \frac{df}{dZ} \quad . \tag{2.2}$$

If *K* and *L* are treated as continuous variables of *R* and *Z*, one can transform the coordinates from *(R,Z)* to *(K,L)* using the relationships

$$\begin{aligned} \frac{df}{dR} &= \frac{df}{dK}\frac{dK}{dR} + \frac{df}{dL}\frac{dL}{dR} \\ \\ \frac{df}{dZ} &= \frac{df}{dK}\frac{dK}{dZ} + \frac{df}{dL}\frac{dL}{dZ} \end{aligned} \quad . \tag{2.3}$$

We can define the area Jacobian of the cell as

$$j = \frac{d(R,Z)}{d(K,L)} = \frac{dR}{dK}\cdot\frac{dZ}{dL} - \frac{dZ}{dK}\cdot\frac{dR}{dL} \quad . \tag{2.4}$$

We define

$$R_K = \begin{pmatrix} \dfrac{dR}{dK} \\ \dfrac{dZ}{dK} \end{pmatrix} \qquad R_L = \begin{pmatrix} \dfrac{dR}{dL} \\ \dfrac{dZ}{dL} \end{pmatrix} \tag{2.5}$$

Then working through the partial derivatives, one gets

$$\begin{aligned} \nabla \cdot (D\nabla f) = \frac{1}{Rj}\left\{ \frac{d}{dK}\left[\frac{DR(R_L)^2}{j}\frac{df}{dK} \right] + \frac{d}{dL}\left[\frac{DR(R_K)^2}{j}\frac{df}{dL} \right] \right\} \\ - \frac{1}{Rj}\left\{ \frac{d}{dK}\left[\frac{DR(R_K \bullet R_L)}{j}\frac{df}{dL} \right] + \frac{d}{dL}\left[\frac{DR(R_K \bullet R_L)}{j}\frac{df}{dK} \right] \right\} \end{aligned} . \tag{2.6}$$

The difference approximation to this equation is derived using a technique based on a variational formulation of the solution. Using the identity

$$\nabla \cdot (fD\nabla f) = f\nabla \cdot D\nabla f = D(\nabla f)^2$$

one can write the equation

$$\int RdRdZf\nabla \cdot (D\nabla f) = \int RdRdZ\nabla \cdot (fD\nabla f) - \int RdRdZD(\nabla f)^2 \tag{2.7}$$

$$\int RdRdZf\nabla \cdot (D\nabla f) = \int RdRdZ\nabla \cdot (fD\nabla f) - \int RdRdZD(\nabla f)^2$$

$$\int RdRdZ\nabla \cdot (fD\nabla f) = 0,$$

since by the divergence theorem

$$\int RdRdZ\nabla \cdot (fD\nabla f) = \int fDdS \cdot \nabla f$$

and on the boundary of the problem, it is assumed that $f = 0$ or $dS \cdot \nabla f = 0$. Occasionally one has $f = f_B \neq 0$ on the boundary. When this occurs, the terms in $(Af)_{K,L}$ that involve f_B are now known and can be moved to the right-hand side of Eq. 2.5 when it is finite differenced. Then the remaining matrix *A*, which operates on the vector of unknowns, is the same as if $f = 0$ on the boundary.

Then Eq. (2.7) is written

$$\int RdRdZf\nabla\cdot(D\nabla f) = -\int RdRdZD(\nabla f)^2. \tag{2.8}$$

The finite difference analogue of the differentials in Eq. (2.8) is

$$\sum_{K,L} f_{K,L}(Af)_{K,L}V_{K,L} = -\sum_{K,L}\left|Bf\right|^2_{K,L} \tag{2.9}$$

where

$$V_{K,L} = Rj_{K,L} = \text{ zone volume} / 2\pi \quad .$$

and *A* is the matrix to be determined. The matrix *B* will be used to define *A*.

2.2 Definition of the elements of the finite difference operator

The form of *(Bf)* is chosen so that:

$$-\int RdRdZD(\nabla f)^2 \approx -\frac{1}{4}\sum_{K,L}\sum_{i=1}^{4}\left|B^i f\right|^2_{K,L} \quad . \tag{2.10}$$

The reason for choosing to divide by 1/4 is not immediately obvious, but will become apparent later on. The left hand side integral is transformed to *(K,L)* coordinates (again treating *K* and *L* as continuous variables) to get

$$-\int RdRdZD(\nabla f)^2 = -\int dKdL\left[\left(\frac{DR}{j}\right)^{\frac{1}{2}}\left(R_L\frac{d}{dK} - R_K\frac{d}{dL}\right)f\right]^2 \quad . \tag{2.11}$$

From this, one can define:

$$(Bf)_{K,L} \approx (\frac{DR}{j})^{1/2}(R_L\frac{df}{dK} - R_K\frac{df}{dL}) \quad . \tag{2.12}$$

This can be written:

$$(Bf)_{K,L} \simeq k_{K,L}R_L - \ell_{K,L}R_K$$

where

$$k_{K,L} \simeq \left(\frac{DR}{j}\right)^{\frac{1}{2}} \frac{df}{dK}$$
$$\ell_{K,L} \simeq \left(\frac{DR}{j}\right)^{\frac{1}{2}} \frac{df}{dL} \quad . \tag{2.13}$$

k and ℓ are face centered quantities because they involve $\frac{df}{dK}$ and $\frac{df}{dL}$, so they can be differenced

$$k_{K,L} = \Sigma_{K,L}(f_{K+1,L} - f_{K,L)})$$
$$\ell_{K,L} = \Lambda_{K,L}(f_{K,L+1} - f_{K,L}) \tag{2.14}$$

where $\Sigma_{K,L} = (DR/j)^{1/2}$ and is suitably averaged between the zones *(K,L)* and *(K+1,L)*, and $\Lambda_{K,L} = (DR/j)^{1/2}$ is suitably averaged between the zones *(K,L)* and *(K,L+1)*. Since *K* and *L* are actually discrete variables, the partial derivatives have been differenced as

$$\frac{df}{dK} = \frac{f_{K+1,L} - f_{K,L}}{(K+1) - K} = f_{K+1,L} - f_{K,L}$$
$$\frac{df}{dL} = \frac{f_{K+L+1} - f_{K,L}}{(L+1) - L} = f_{K,L+1} - f_{K,L} \quad . \tag{2.15}$$

One choice for a suitable $\Sigma_{K,L}$ and $\Lambda_{K,L}$ is given by

$$\Sigma^2_{K,L} = \frac{(R_{K,L} + R_{K,L-1})}{\frac{j_{K,L}}{D_{K,L}} + \frac{j_{K+1,L}}{D_{K+1,L}}} \qquad \Lambda^2_{K,L} = \frac{(R_{K,L} + R_{K,L+1})}{\frac{j_{K,L}}{D_{K,L}} + \frac{j_{K,L+1}}{D_{K,L+1}}} \quad . \tag{2.16}$$

The terms $k_{K,L}$ and $\ell_{K,L}$ are represented in diagram form in Figs. 2.1a,b.

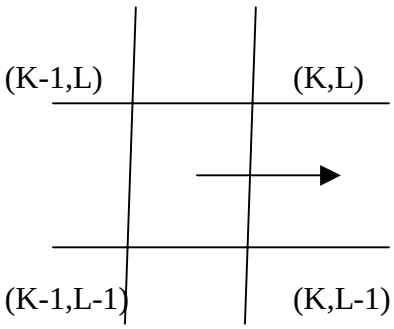


Fig. 2.1a. $k_{K,L}$

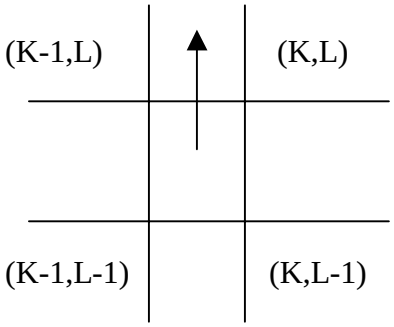


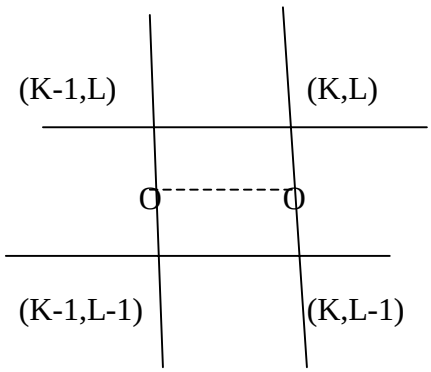
Fig. 2.2b. $\ell_{K,L}$

Remember that

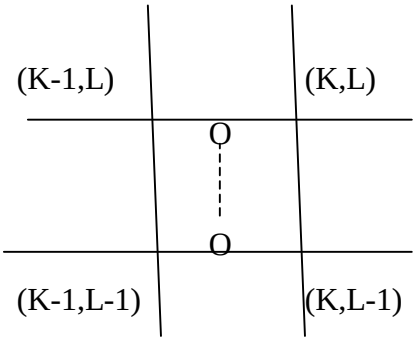
$$R_K = \begin{pmatrix} \frac{dR}{dK} \\ \frac{dZ}{dK} \end{pmatrix} \qquad R_Z = \begin{pmatrix} \frac{dR}{dL} \\ \frac{dZ}{dL} \end{pmatrix}$$

and let $R_{K,L} = (R_{K,L}, Z_{K,L})$, then $(R_K)_{K,L}$ and $(R_L)_{K,L}$ are differenced

$$\left(R_K\right)_{K,L} = \frac{\left(R_{K,L} + R_{K,L-1}\right)}{2} - \frac{\left(R_{K-1,L} + R_{K-1,L-1}\right)}{2} \tag{2.17}$$



$$(R_L)_{K,L} = \frac{\left(R_{K,L} + R_{K-1,L}\right)}{2} - \frac{\left(R_{K,L-1} + R_{K-1,L-1}\right)}{2} \tag{2.18}$$

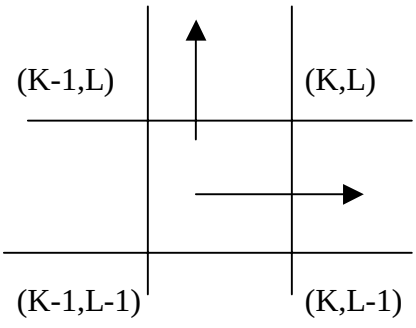


Then $(B^1 f)$ is chosen to be

$$\left(B^1 f\right)_{K,L} = k_{K,L}\left(R_L\right)_{K,L} - \ell_{K,L}\left(R_K\right)_{K,L} \tag{2.19}$$

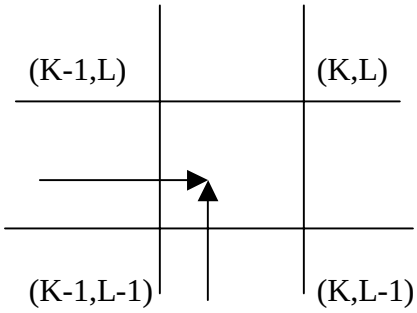
$$\left(B^1 f\right)_{K,L} = \sum_{K,L}\left(f_{K+1,L} - f_{K,L}\right)\left[\frac{\left(R_{K,L} + R_{K-1,L}\right)}{2} - \frac{\left(R_{K,L-1} + R_{K-1,L-1}\right)}{2}\right]$$
$$-\Lambda_{K,L}\left(f_{K,L+1} - f_{K,L}\right)\left[\frac{\left(R_{K,L} + R_{K,L-1}\right)}{2} - \frac{\left(R_{K-1,L} + R_{K-1,L-1}\right)}{2}\right] \tag{2.20a}$$

$(Bf)_{K,L}$ is also a zone-centered quantity and combining Figs. 2.1a and 2.1b gives

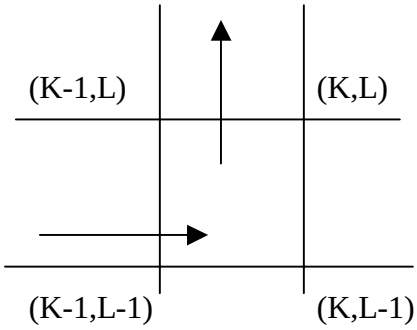


$(B^1 f)$

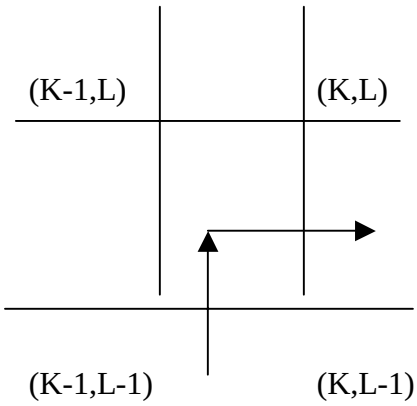
There are three other equally valid choices for $(B^i f)$



$$\left(B^2 f\right) = k_{K,L}(R_L)_{K,L} - \ell_{K,L-1}\left(R_K\right)_{K,L} \tag{2.20b}$$



$$\left(B^3 f\right) = k_{K-1,L}(R_L)_{K,L} - \ell_{K,L}\left(R_K\right)_{K,L} \tag{2.20c}$$



$$\left(B^4 f\right) = k_{K,L}(R_L)_{K,L} - \ell_{K,L-1}\left(R_K\right)_{K,L} \tag{2.20d}$$

Since any of these is equally valid, an average of the four is taken

$$-\int RdRdZD(\nabla f)^2 \approx -\frac{1}{4}\sum_{K,L}\sum_{i=1}^{4}\left|B^i f\right|_{K,L}^2 = \sum_{K,L} f_{K,L}(Af)_{K,L}V_{K,L} \tag{2.21}$$

then

$$-\frac{1}{4}\sum_{K,L}\sum_{i=1}^{4}\left(B^i f\right)_{K,L}^2 = \sum_{K,L} f_{K,L}\left(Af\right)_{K,L} V_{K,L}$$
$$-\frac{1}{4}\sum_{i=1}^{4}\left(B^i f\right)^2 = f_{K,L}\left(Af\right)_{K,L} V_{K,L} \tag{2.22}$$

and finally,

$$-\frac{1}{4}\sum_{i=1}^{4}\left(B^i\right)^T\left(B^i\right) = VA \quad . \tag{2.23}$$

Eq. (2.20a) and the counterparts for (B^2F), (B^3f) and (B^4f) are substituted into Eq. (2.23). On the left side, the coefficients of $f_{K,L}f_{K',L'}$ are collected and equated with the coefficients of $f_{K,L}f_{K',L'}$ on the right side to get

$$V_{K,L}A_{(K,L),(K,L)} = -\sigma_{(K,L)} - \sigma_{(K-1,L)} - \lambda_{(K,L)} - \lambda_{(K,L-1)} + \frac{1}{2}\left(\rho^1_{(K,L)} + \rho^2_{(K,L)} - \rho^3_{(K+1,L)} - \rho^4_{(K,L)}\right) \tag{2.24a}$$

$$V_{K,L}A_{(K,L),(K+1,L)} = \sigma_{(K,L)} - \frac{1}{4}\left(\rho^1_{(K,L)} + \rho^2_{(K+1,L)} - \rho^3_{(K+1,L)} - \rho^4_{(K,L)}\right) \tag{2.24b}$$

$$V_{K,L}A_{(K,L),(K,L+1)} = \lambda_{(K,L)} - \frac{1}{4}\left(\rho^1_{(K,L)} + \rho^2_{(K,L+1)} - \rho^3_{(K,L)} - \rho^4_{(K,L+1)}\right) \tag{2.24c}$$

$$V_{K,L}A_{(K,L),(K+1,L+1)} = -\frac{1}{4}\left(\rho^3_{(K+1,L)} + \rho^4_{(K,L+1)}\right) \tag{2.24d}$$

$$V_{K,L}A_{(K,L),(K-1,L+1)} = \frac{1}{4}\left(\rho^1_{(K-1,L)} + \rho^2_{(K,L+1)}\right) \tag{2.24e}$$

where

$$
\begin{aligned}
\sigma_{(K,L)} &= \Sigma^2_{K,L}\,\frac{\left(R_L\right)^2_{K,L}+\left(R_L\right)^2_{K+1,L}}{2}\\
\lambda_{(K,L)} &= \Lambda^2_{K,L}\,\frac{\left(R_K\right)^2_{K,L}+\left(R_K\right)^2_{K,L+1}}{2}\\
\rho^1_{(K,L)} &= \Sigma_{(K,L)}\Lambda_{(K,L)}C_{(K,L)}\\
\rho^2_{(K,L)} &= \Sigma_{(K-1,L)}\Lambda_{(K,L-1)}C_{(K,L)}\\
\rho^3_{(K,L)} &= \Sigma_{(K-1,L)}\Lambda_{(K,L)}C_{(K,L)}\\
\rho^4_{(K,L)} &= \Sigma_{(K,L)}\Lambda_{(K,L-1)}C_{(K,L)}
\end{aligned}
\qquad (2.25)
$$

and

$$
C_{K,L} = (R_K)_{K,L}\cdot(R_L)_{K,L} \quad . \qquad (2.26)
$$

Since *VA* is symmetric

$$
V_{K',L'}A_{(K',L'),(K,L)} = V_{K,L}A_{(K,L),(K',L')} \qquad (2.27)
$$

and all elements of *A* not defined by Eqs. 2.24a-e to 2.26 are zero.

2.3 Some derivations related to the code implementation

As shown before, the diffusion equation including the absorption and emission terms is:

$$
\frac{\partial I}{c\,\partial t} - \nabla\cdot D\,\nabla I = 4\pi\sigma_a' B_\nu - \sigma_a' I \quad . \qquad (2.28)
$$

Using I=cE, and multiplying by the volume, the equation becomes

$$
V\,\frac{\partial E}{\partial t} - Vc\,\nabla\cdot D\,\nabla E = V\,4\pi\sigma_a' B_\nu - Vc\,\sigma_a' E \qquad (2.29)
$$

Using the implicit differencing scheme for the time variable,

$$
\begin{aligned}
&V\frac{E^{n+1}-E^n}{\Delta t} - Vc\nabla\cdot D\nabla E = V\,4\pi\sigma_a' B_\nu - Vc\,\sigma_a' E^{n+1}\\
&V(E^{n+1}-E^n) = \Delta t Vc\nabla\cdot D\nabla E + \Delta t V\,4\pi\sigma_a' B_\nu - \Delta t Vc\,\sigma_a' E^{n+1}
\end{aligned}
\qquad (2.30)
$$

Replace $\nabla\cdot D\nabla E$ by **A**E as shown before,

$$
V\,(E^{n+1}-E^n) = \Delta t VcAE^{n+1} + \Delta t V\,4\pi\sigma_a' B_\nu - \Delta t Vc\,\sigma_a' E^{n+1} \qquad (2.31)
$$

Rearrange the above equation:

$$\{V - c\Delta t(VA - V\sigma_a')\}E^{n+1} = VE^n + \Delta tV\,4\pi\sigma_a' B_\nu \tag{2.32}$$

From the Kershaw scheme, we know the spatial differential matrix Ã=VA, and the volume for the last time step is V^n,

$$\{V^{n+1} - c\Delta t(\tilde{A} - V\sigma_a')\}E^{n+1} = V^n E^n + \Delta tV^{n+1}\,4\pi\sigma_a' B_\nu \tag{2.33}$$

Therefore, the diagonal elements and the nondiagonal elements are,

Diagonal = $V_{i,i}^{n+1} - c\Delta t(\tilde{A}_{i,i} - V_{i,i}\sigma_a')$

Nondiagonal = $-c\Delta t\tilde{A}_{i,j}$

Further define for convenience:

$$\overline{\overline{A}} = V^{n+1} - c\Delta t(\tilde{A} - V\sigma_a') \tag{2.34}$$

The last equation becomes,

$$\overline{\overline{A}}E^{n+1} = V^n E^n + \Delta tV^{n+1}\,4\pi\sigma_a' B_\nu \quad . \tag{2.35}$$

Define $\Delta E^{n+1} = E^{n+1} - E^n$,

$$\overline{\overline{A}}\Delta E^{n+1} = V^n E^n - \overline{\overline{A}}E^n + \Delta tV^{n+1}\,4\pi\sigma_a' B_\nu \quad . \tag{2.36}$$

This is the final form that is solved by DRACO radiation diffusion transport module. As we can see, the actual result solved by the linear equation solver is the change of the radiation energy density, rather than the radiation energy density itself.

2.4 Variables in the code

In the situation that users want to understand how the scheme is actually implemented, we give brief descriptions about the variables in the code and their corresponding symbols in the equation. In the radiation module in the DRACO code, variable cmac_14 represents the diagonal elements of the matrix, and other 8 variables cmac_10-13, cmac_15-18 represent the nondiagonal

parts of the matrix. The code segment of implementation of the above equations in DRACO is listed as follows,

```
 cmac_14(is:ie,js:je,1,igr) = cmac_14(is:ie,js:je,1,igr)
                    - opacity_zone(is:ie,js:je,1,igr,iemiss)*vol(is:ie,js:je,1)
cmac_10(is:ie,js:je,1,igr)  = cmac_10(is:ie,js:je,1,igr) *const1
cmac_11(is:ie,js:je,1,igr)  = cmac_11(is:ie,js:je,1,igr) *const1
cmac_12(is:ie,js:je,1,igr)  = cmac_12(is:ie,js:je,1,igr) *const1
cmac_13(is:ie,js:je,1,igr)  = cmac_13(is:ie,js:je,1,igr) *const1
cmac_14(is:ie,js:je,1,igr)  = cmac_14(is:ie,js:je,1,igr) *const1
cmac_15(is:ie,js:je,1,igr)  = cmac_15(is:ie,js:je,1,igr) *const1
cmac_16(is:ie,js:je,1,igr)  = cmac_16(is:ie,js:je,1,igr) *const1
cmac_17(is:ie,js:je,1,igr)  = cmac_17(is:ie,js:je,1,igr) *const1
cmac_18(is:ie,js:je,1,igr)  = cmac_18(is:ie,js:je,1,igr) *const1
cmac_rhs(is:ie,js:je,1,igr) = cmac_rhs(is:ie,js:je,1,igr) +
                    radiation_energy_emission(is:ie,js:je,1,igr)*dt*vol(is:ie,js:je,1)
cmac_14(is:ie,js:je,1,igr) = vol(is:ie,js:je,1) - cmac_14(is:ie,js:je,1,igr)
cmac_rhs(is:ie,js:je,1,igr) = cmac_rhs(is:ie,js:je,1,igr) +
    vollast(is:ie,js:je,1)*radiation_energy_density(is:ie,js:je,1,igr)
cmac_rhs(is:ie,js:je,1,igr) = cmac_rhs(is:ie,js:je,1,igr)
  + cmac_10(is:ie,js:je,1,igr)*radiation_energy_density(is-1:ie-1,js-1:je-1,1,igr)
  + cmac_11(is:ie,js:je,1,igr)*radiation_energy_density(is  :ie  ,js-1:je-1,1,igr)
  + cmac_12(is:ie,js:je,1,igr)*radiation_energy_density(is+1:ie+1,js-1:je-1,1,igr)
  + cmac_13(is:ie,js:je,1,igr)*radiation_energy_density(is-1:ie-1,js  :je  ,1,igr)
  - cmac_14(is:ie,js:je,1,igr)*radiation_energy_density(is  :ie  ,js  :je  ,1,igr)
  + cmac_15(is:ie,js:je,1,igr)*radiation_energy_density(is+1:ie+1,js  :je  ,1,igr)
  + cmac_16(is:ie,js:je,1,igr)*radiation_energy_density(is-1:ie-1,js+1:je+1,1,igr)
  + cmac_17(is:ie,js:je,1,igr)*radiation_energy_density(is  :ie  ,js+1:je+1,1,igr)
  + cmac_18(is:ie,js:je,1,igr)*radiation_energy_density(is+1:ie+1,js+1:je+1,1,igr)
```

The variables in the code corresponding to the symbols in the equation are listed in the table,

vol	$V_{i,i}^{n+1}$
const1	$c\Delta t$
vollast	$V_{i,i}^{n}$
opacity_zone	$\sigma_a^{'}$
radiation_energy_emission	$4\pi\sigma_a^{'}B_\nu$
cmac	$\overline{\overline{A}}$

We also want to point out the difference between the current implementation of the above eaquation and Verdon's implementation. (We don't know exactly Verdon's scheme. We can only tell the slight implementation difference from the code segment.)

The Verdon's code segment is:

```
cmac_14(i,j,1,igr) = dt*cmac_14(i,j,1,igr)-(const3*alpha)-(const4*alpha)
cmac_14(i,j,1,igr) = cmac_14(i,j,1,igr)*vol(i,j,1)
cmac_17(i,j,1,igr) = cmac_17(i,j,1,igr)*const5
cmac_18(i,j,1,igr) = cmac_18(i,j,1,igr)*const5
cmac_15(i,j,1,igr) = cmac_15(i,j,1,igr)*const5
cmac_12(i,j,1,igr) = cmac_12(i,j,1,igr)*const5
cmac_11(i,j,1,igr) = cmac_11(i,j,1,igr)*const5
cmac_10(i,j,1,igr) = cmac_10(i,j,1,igr)*const5
cmac_13(i,j,1,igr) = cmac_13(i,j,1,igr)*const5
cmac_16(i,j,1,igr) = cmac_16(i,j,1,igr)*const5
cmac_rhs(i,j,1,igr) = radiation_energy_emission(i,j,1,igr)*const5

cmac_rhs(i,j,1,igr) = cmac_rhs(i,j,1,igr) +
  ((cmac_14(i,j,1,igr)*radiation_energy_density(i  ,j  ,1,igr) +
    cmac_17(i,j,1,igr)*radiation_energy_density(i  ,j+1,1,igr) +
    cmac_18(i,j,1,igr)*radiation_energy_density(i+1,j+1,1,igr) +
    cmac_15(i,j,1,igr)*radiation_energy_density(i+1,j  ,1,igr) +
    cmac_12(i,j,1,igr)*radiation_energy_density(i+1,j-1,1,igr) +
    cmac_11(i,j,1,igr)*radiation_energy_density(i  ,j-1,1,igr) +
    cmac_10(i,j,1,igr)*radiation_energy_density(i-1,j-1,1,igr) +
    cmac_13(i,j,1,igr)*radiation_energy_density(i-1,j  ,1,igr) +
    cmac_16(i,j,1,igr)*radiation_energy_density(i-1,j+1,1,igr)) * rtalphai)
    cmac_14(i,j,1,igr)    = vollast(i,j,1)-cmac_14(i,j,1,igr)
```

From this code, we guess Verdon followed the derivation like this:

From Equation (2.32):

$$\{V^{n+1} - c\Delta t(\tilde{A} - V\sigma_a^{'})\}E^{n+1} = V^n E^n + \Delta t V^{n+1} 4\pi\sigma_a^{'} B_\nu \ . \tag{2.37}$$

Write $c\Delta t(\tilde{A} - V\sigma_a^{'})$ as $\ddot{A}$ (we use $\ddot{A}$ to distinguish from the above matrix symbol),

$$V^n E^{n+1} - \ddot{A}E^{n+1} = V^n E^n + \Delta t V^{n+1} 4\pi\sigma_a^{'} B_\nu \ . \tag{2.38}$$

Note V^{n+1} in Equation (2.37) changes to V^n in Equation (2.38), and we also define $\Delta E^{n+1} = E^{n+1} - E^n$,

$$(V^n - \vec{\vec{A}})\Delta E^{n+1} = \vec{\vec{A}} E^n + \Delta t V^{n+1} 4\pi\sigma_a' B_\nu \quad . \tag{2.39}$$

Therefore, the difference comes from the treatment of the volume and the definition of the intermediate matrix $\vec{\vec{A}}$.

2.5 Boundary Conditions

The logical mesh consists of two types of zones: physical and vacuum or ghost zones. The vacuum zones completely surround the physical zones. With the use of vacuum zones, it is possible to have irregular-shaped physical zone boundaries while still maintaining the rectangular logical mesh (i.e., for a given *K = 1, …, KMAX* and *L = 1, …, LMAX*, *KMAX* and *LMAX* are constant throughout the mesh).

The two types of boundary conditions considered are the Dirichlet condition or "escape" boundary

$$f = f_B$$

where f_B is a specified flux in the vacuum, and the reflective or "no escape" boundary

$$dS \cdot \nabla f = 0.$$

To implement these boundary conditions, it is necessary to revise the definitions of $\Sigma_{K,L}$ and $\Lambda_{K,L}$ on the boundaries. The revisions depend upon the type of zone it is and whether the neighboring zones are vacuum or physical zones. The following rules are used to determine the components $\Sigma_{K,L}$ and $\Lambda_{K,L}$. The rules are defined for $\Sigma_{K,L}$, which connects the zone *(K,L)* with the zone *(K+1,L).* An analogous sent of rules applies for $\Lambda_{K,L}$, which connects the zone *(K,L)* with the zone *(K,L+1).*

1) If *(K,L)* and *(K+1,L)* are both physical zones $\Sigma_{K,L}\ (R_L)_{K,L}$ and $\Sigma_{K,L}\ (R_L)_{K+1,L}$ are defined as previously (using Eqs. 2.16). This is corresponding to the physical region for the problem.

2) If *(K,L)* and *(K+1,L)* are both vacuum zones, $\Sigma_{K,L} = 0$.

3) If *(K,L)* is a physical zone and *(K+1,L)* is a vacuum zone and the face separating them has a "no escape" boundary condition, $n \cdot \nabla f = 0$, where n is the unit normal to the vacuum surface then $\Sigma_{K,L} = 0$ and $(R_K)_{K,L}$ is replaced by

$$n\,(n \cdot (R_K)_{K,L})$$

4) If *(K,L)* is a physical zone and *(K+1,L)* is a vacuum zone and the face separating them has an "escape" boundary condition, then $\Sigma_{K,L}\ (R_L)_{K,L}$ remains defined as previously (Eqs. 2.6), but now in the expression

$$\Sigma_{K,L}\ (R_L)_{K+1,L},$$

we use

$$(R_L)_{K+1,L} = R_{K,L} - R_{K,L-1},$$

since $R_{K+1,L}$ and $R_{K+1,L-1}$ do not exist, and $(R_K)_{K,L}$ is replaced by

$$n\,(n \cdot (R_K)_{K,L})$$

n is the unit normal to the vacuum surface. $\Sigma_{K,L}$ is now $\left(DR/j\right)^{1/2}$ suitably averaged between zone *(K,L)* and the vacuum, but since *D* is not defined in the vacuum zone, $\Sigma_{K,L}$ is defined by

$$\Sigma_{K,L}^2 = \left(\frac{D_{K,L} R_{K,L}}{j_{K,L}}\right) \cdot \frac{1}{2}$$

The above cases (3) and (4) correspond to the right-side boundary for the problem.

5) If *(K,L)* is a vacuum zone and *(K+1,L)* is a physical zone and the face separating them has a "no escape" boundary condition, $n \cdot \nabla f = 0$, then $\Sigma_{K,L} = 0$ and $(R_K)_{K+1,L}$ is replaced by

$$n\,(n \cdot (R_K)_{K+1,L})$$

6) If *(K,L)* is a vacuum zone and *(K+1,L)* is a physical zone and the face separating them has an "escape" boundary condition, $f = f_B$, then in the expression

$$\Sigma_{K,L}(R_L)_{K,L},$$

we use

$$(R_L)_{K,L} = R_{K,L} - R_{K,L-1}$$

$(R_K)_{K,L}$ is replaced by

$$n\,(n \cdot (R_K)_{K+1,L})$$

The above cases (5) and (6) correspond to the left-side boundary for the problem.

7) If *(K,L)* is a physical zone and *(K,L+1)* is a vacuum zone and the face separating them has a "no escape" boundary condition, $n \cdot \nabla f = 0$, where n is the unit normal to the vacuum surface then $\Lambda_{K,L} = 0$ and $(R_L)_{K,L}$ is replaced by

$$n\,(n \cdot (R_L)_{K,L})$$

8) If *(K,L)* is a physical zone and *(K,L+1)* is a vacuum zone and the face separating them has an "escape" boundary condition, then $\Lambda_{K,L}\ (R_K)_{K,L}$ remains defined as previously (Eqs. 2.6), but now in the expression

$$\Lambda_{K,L}\ (R_K)_{K,L+1},$$

we use

$$(R_K)_{K,L+1} = R_{K,L} - R_{K-1,L},$$

since $R_{K,L+1}$ and $R_{K-1,L+1}$ do not exist, and $(R_L)_{K,L}$ is replaced by

$$n\,(n \cdot (R_L)_{K,L})$$

n is the unit normal to the vacuum surface. $\Lambda_{K,L}$ is now $(DR/j)^{1/2}$ suitably averaged between zone *(K,L)* and the vacuum, but since *D* is not defined in the vacuum zone, $\Lambda_{K,L}$ is defined by

$$\Lambda^2_{K,L} = \left(\frac{D_{K,L} R_{K,L}}{j_{K,L}} \right) \cdot \frac{1}{2}$$

The above cases (7) and (8) correspond to the top-side boundary for the problem.

9) If *(K,L)* is a vacuum zone and *(K,L+1)* is a physical zone and the face separating them has a "no escape" boundary condition, $n \cdot \nabla f = 0$, then $\Lambda_{K,L} = 0$ and $(R_L)_{K,L+1}$ is replaced by

$$n\,(n \cdot (R_L)_{K,L+1})$$

10) If *(K,L)* is a vacuum zone and *(K,L+1)* is a physical zone and the face separating them has an "escape" boundary condition, $f = f_B$, then in the expression

$$\Lambda_{K,L}(R_L)_{K,L},$$

we use

$$(R_L)_{K,L} = R_{K,L} - R_{K,L-1}$$

and $(R_L)_{K,L}$ is replaced by

$$n\,(n \cdot (R_L)_{K,L})$$

The above cases (9) and (10) correspond to the bottom-side boundary for the problem.

In summary, the treatment of the nonescape boundary condition is much easier than the free surface boundary condition since we know the values at the boundary because of the reflection and adiabatic property of the boundary. However, for the free escape boundary condition, we don't know the exact solution at the boundary. The best we can do is assume at some extrapolated distance the value of f goes to zero, as described in Milne's problem [5]. In DRACO, we use this approximation. We use the right-side boundary as an example (Fig. 2.2)

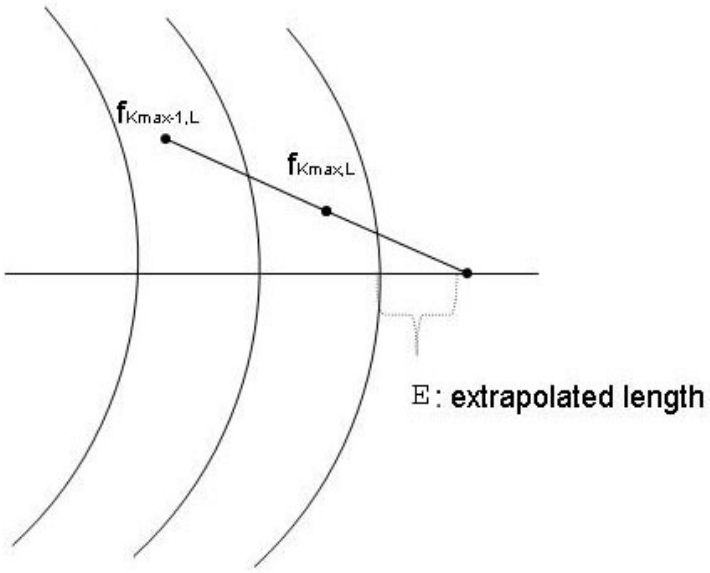


Fig. 2.2. Milne right-side boundary condition.

E = 0.71λ, where λ is the transport mean free path. In addition, the calculation of $\Sigma_{K,L}$ (Eq. 2.16) which is averaged between the last zone and the vacuum zone is changed to include the area defined by the extrapolated length as follows

$$\Sigma_{K,L}^{2} = \frac{R_{K,L} \cdot D_{K,L}}{A_{K,L}}$$

and the Jacobian area $A_{K,L} = j_{K,L} + A_{Ex}$, where A_{Ex} is the Jacobian area determined by the extrapolated length.

2.6 Flux limiter

When dealing with the moment of the transport equation, there always appears one unknown quantity. To close the set of equations, an approximation is needed to break the chain [5]. The P1 equation is obtained by setting the radiation pressure equal to one-third of the energy density. It has first-order accuracy in the optically thick limit, while in the optically thin limit it has an incorrect propagation velocity. The diffusion equation makes a further approximation to the P1 equation by omitting the gradient of the flux such that the cutoff equation is in the form of the

Fick's law. Morel [6] has shown in the diffusive limit the diffusion equation has the first-order accuracy comparable with the P1 equation. However, in the optically thin limit, this equation has an infinite propagation velocity. To correct this problem, the flux limiter is always introduced.

In DRACO, four options are available [7] beside one option (flux_limiter_type=0) that has no flux limiter included.

a) Option 1

Option 1 uses the summation form of the transport cross section and the gradient. It is called the "sum" flux limiter.

$$D = \frac{1}{3\sigma_{tr} + \frac{\delta}{E}|\nabla E|} \quad .$$

The factor δ is an adjustable parameter for which we set a default value of one. In the optically thick limit, the gradient of the energy density is small compared to the transport mean free path and therefore the equation tends to be diffusive. In the thin limit, the gradient dominates the denominator of the diffusion coefficient and the flow is limited to the speed of light, as desired. The introduction of the gradient leads the equation to be nonlinear in the energy density. Morel has shown this form is only zero order accurate in the thick limit but gets the correct approximation in the optically thin limit.

b) Option 2

Larsen [7] has suggested a modified form which retains the first-order accuracy in the optically thick limit:

$$D = \left[\frac{1}{(3\sigma_{tr})^n + (\frac{\delta}{E}|\nabla E|)^n} \right]^{1/n} \quad .$$

In DRACO, we use n=2.

c) Option 3

The maximum of the transport cross section and the gradient is used. It is called the “max” flux limiter. The form also retains the first-order accuracy in the thick limit when the cross section is larger than the gradient term. However, this flux limiter has discontinuous derivatives.

$$D = \frac{1}{\max(3\sigma_{tr}, \frac{\delta}{E}|\nabla E|)}$$

d) Option 4

Another flux limiter is proposed by Levermore and Pomraning (LP) [8] by solving exactly a particular transport problem. Olson has shown this flux limiter exhibits some nonphysical behavior. A simplified form of this kind of limiter is suggested by Zimmerman:

$$D = \frac{\xi(\beta)}{\sigma_{tr}}$$

$$\xi(\beta) = \left[\coth(\beta) - \frac{1}{\beta} \right] \frac{1}{\beta}$$

$$\beta = \frac{|\nabla E|}{\sigma_{tr} E} \quad .$$

These options can be selected by setting the “flux_limiter_type” in the input parameters.

3. Test problems

We have done a series of tests to look at the correctness of the implementation by running the code in different geometries and different meshes. Two kinds of meshes are used in the test problem as shown in Fig. 3.1 and Fig. 3.2. Both meshes are also applied to the planar and cylindrical geometries. All of the simple test problems were run with graphics. Figures 3.3-3.17 illustrate the time evolution of the radiation energy under various boundary conditions and meshes. The graphical results are easily understandable with the brief graphic description.

To test the accuracy of the scheme, we use three simple problems that have an exact analytical solution. Figure 3.18 shows the result for the problem that has the left boundary condition with the energy density equal to zero and the right boundary condition with the energy

density equal to 8. There is an exact solution for this problem in one dimension, that is, f = 8Z. We simulate the one dimension by setting the reflective boundary condition for both top and bottom boundaries. We can see the numerical result agrees with the exact solution very well. The second problem we solved is the wave propagation with a delta function source in the middle of the one-dimensional plane. To be more specific, the diffusion equation under this case is [9]

$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial x} D \frac{\partial T}{\partial x}$$

with the initial condition of T(x/2,0) = Qδ(x/2). The solution to the problem for the constant diffusion coefficient is

$$T = \frac{Q}{\sqrt{4\pi D t}} \exp(-\frac{(x - x_0/2)^2}{4Dt}).$$

The simulation starts with a distribution of the above form at some initial time. Figure 3.19 shows the comparison of the exact solution and the numerical result. While Fig. 3.19 shows the result for the linear conduction, Fig. 3.20 shows the result for the nonlinear wave conduction. The diffusion equation for this case is

$$\frac{\partial T}{\partial t} = a \frac{\partial}{\partial x} T^5 \frac{\partial T}{\partial x}.$$

The diffusion equation is nonlinear because the diffusion coefficient is a function of T. The coefficient has also very large gradient since it is a function of T to the fifth power. Again, the delta source Q is added at the middle of the plane at the initial time. The exact solution for this case [9] is

$$T = 1.12 \frac{Q}{2x_f} (1 - \frac{x^2}{x_f^2})^{1/5}$$

$$x_f = 0.77 (aQ^n t)^{\frac{1}{(n+2)}} \quad .$$

We compare the numerical result with the exact solution in Fig. 3.20. Again, they agree very well except that there is a little discrepancy at the wave front. This results from the average treatment of the diffusion coefficient at the cell face center, as in Eq. (2.16).

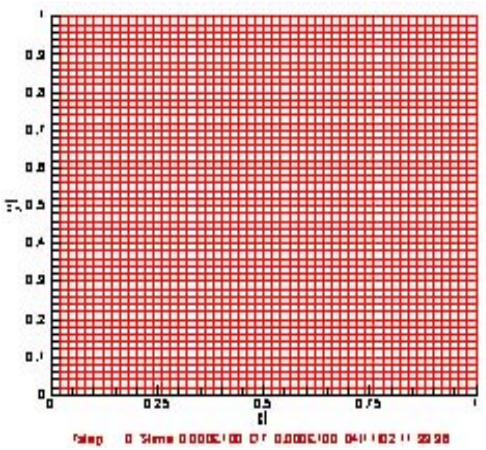


Fig. 3.1. Orthogonal mesh (Omesh) (50 by 50).

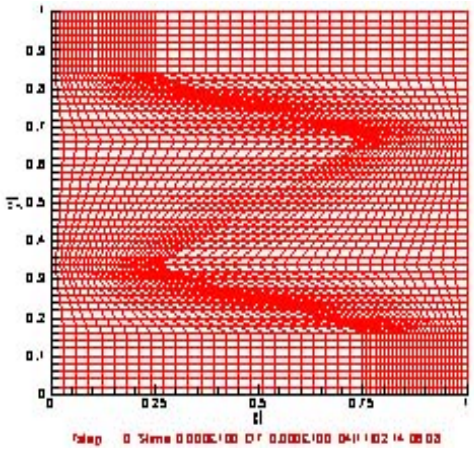


Fig. 3.2. Distorted mesh (Zmesh) (50 by 50).

Notation used in the figure captions:

Omesh:	orthogonal mesh as Fig. 3.1
Zmesh:	distorted mesh as Fig. 3.2
Planar:	planar geometry
Cyd:	cylindrical geometry
R:	right
L:	left
T:	top
B:	bottom
Escape bc:	escape boundary condition (f=0), otherwise, adiabatic boundary condition.

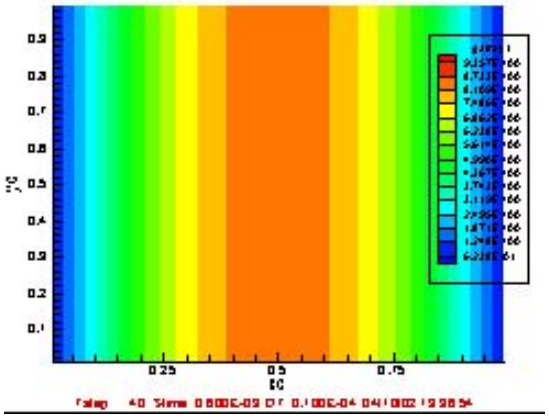


Fig. 3.3. Omesh, planar, L&R escape bc.

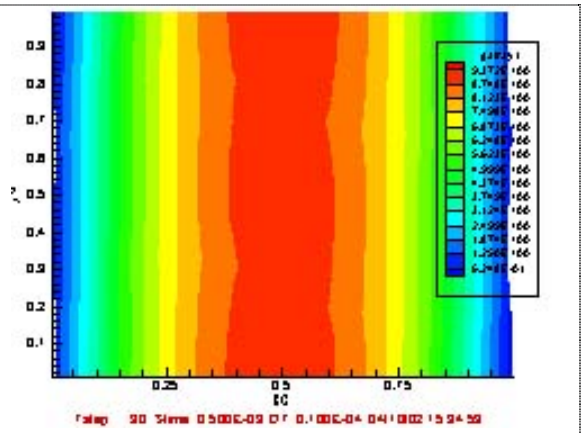


Fig. 3.4. Zmesh, planar, L&R escape bc.

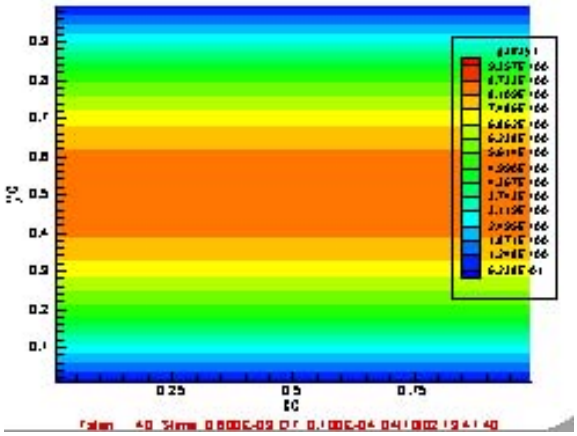


Fig. 3.5. Omesh, planar, T&B escape bc.

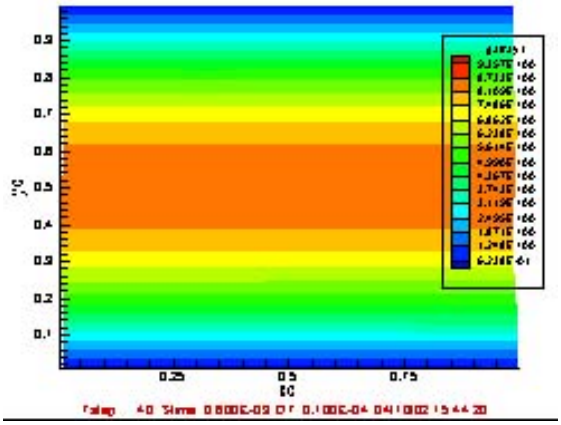


Fig. 3.6. Zmesh, planar, T&B escape bc.

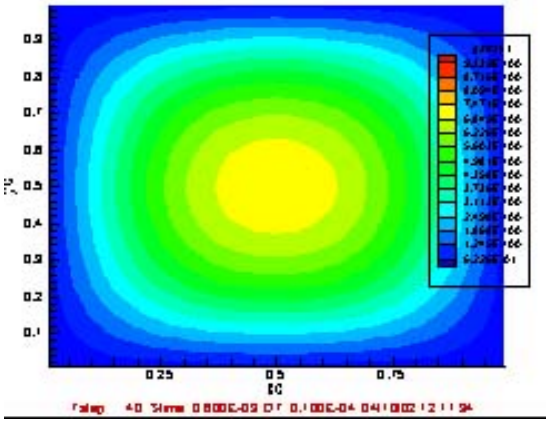


Fig. 3.7. Omesh, planar, T&B&L&R escape bc.

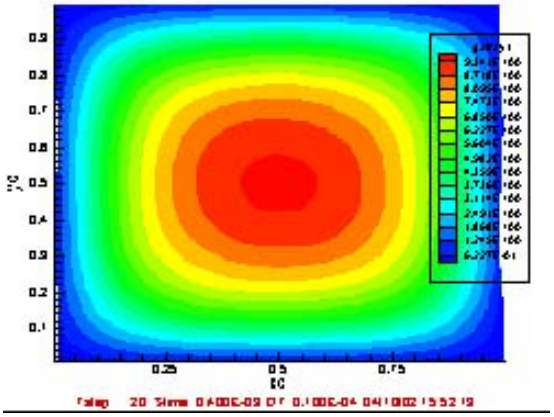


Fig. 3.8. Zmesh, planar, T&B&L&R escape bc.

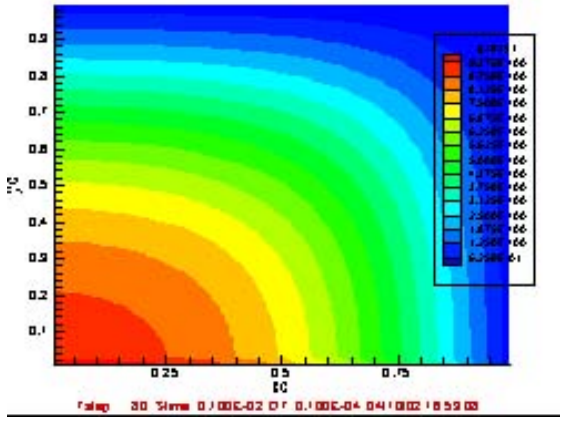


Fig. 3.9. Omesh, cyd, T&R escape bc.

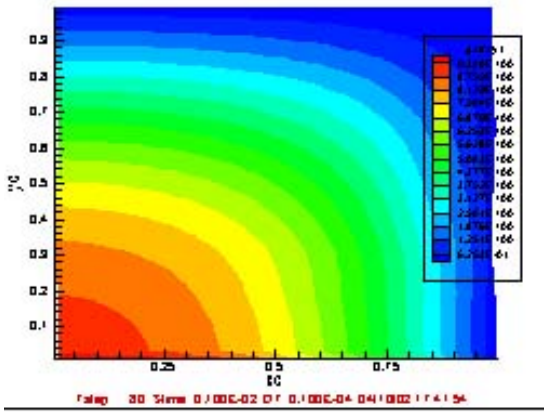


Fig. 3.10. Zmesh, cyd, T&R escape bc.

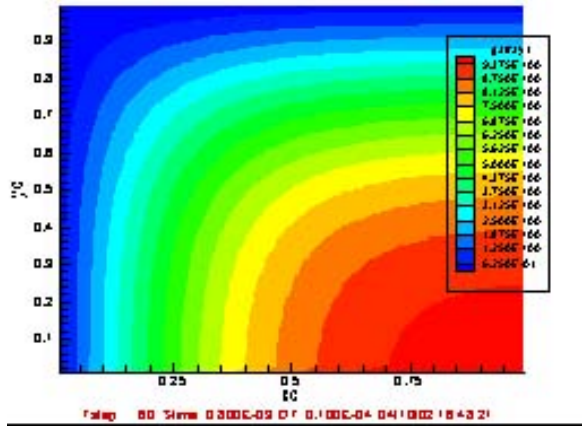


Fig. 3.11. Omesh, cyd, T&L escape bc.

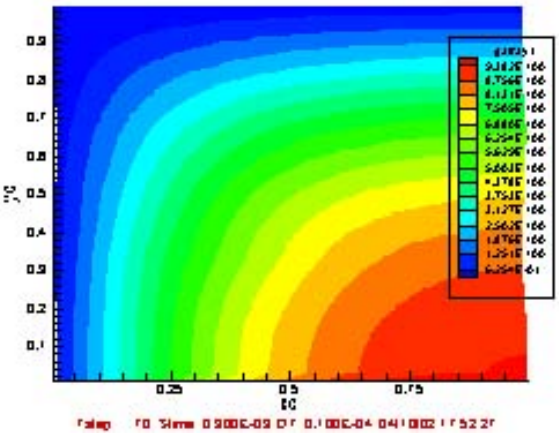


Fig. 3.12. Zmesh, cyd, T&L escape bc.

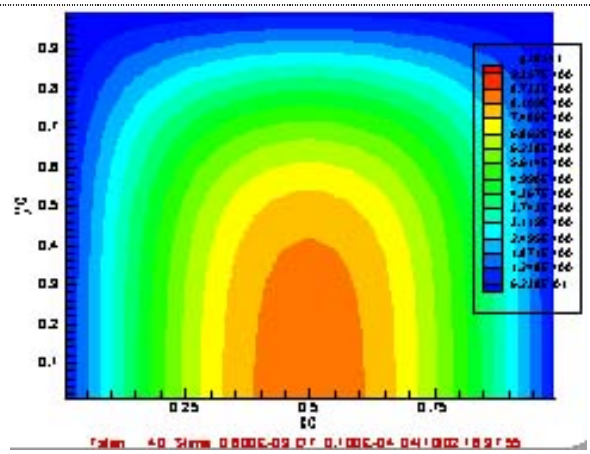


Fig. 3.13. Omesh, cyd, T&L&R escape bc.

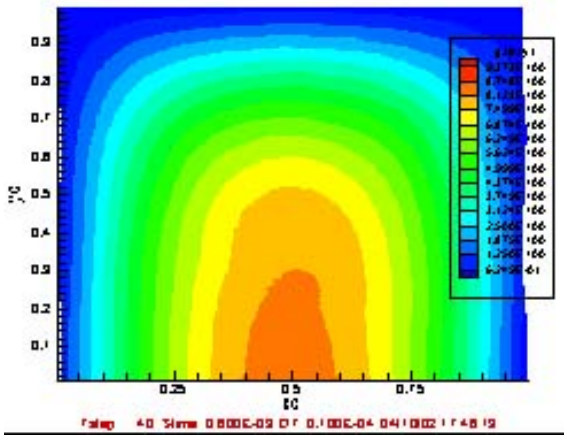


Fig. 3.14. Zmesh, cyd, T&L&R escape bc.

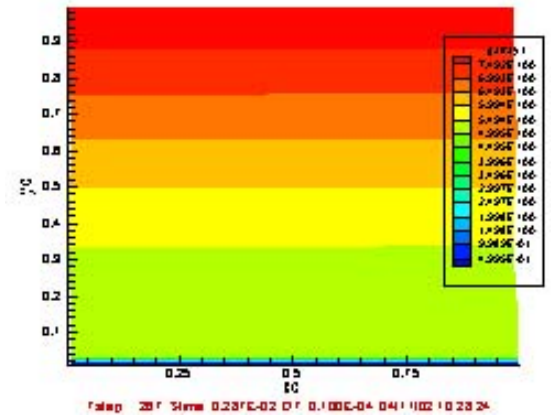


Fig. 3.15. Zmesh, cyd, f=8(T).

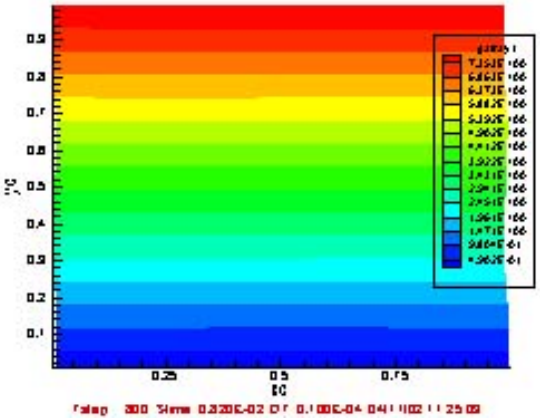


Fig. 3.16. Zmesh, planar, f=0(B) f=8(T).

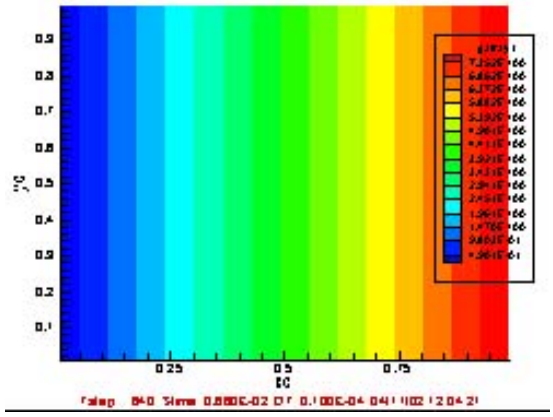


Fig. 3.17. Omesh, planar, f=0(L) f=8(R).

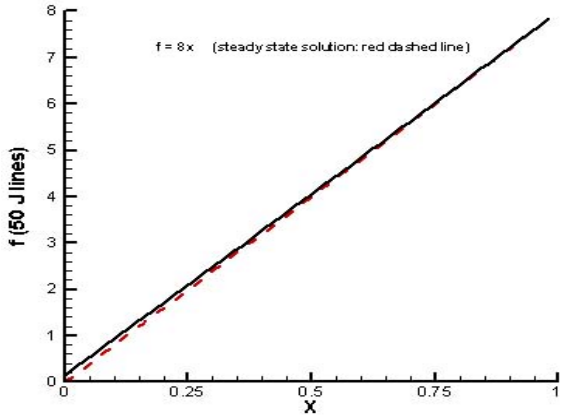


Fig. 3.18. Comparison with exact solution for Fig. 3.17.

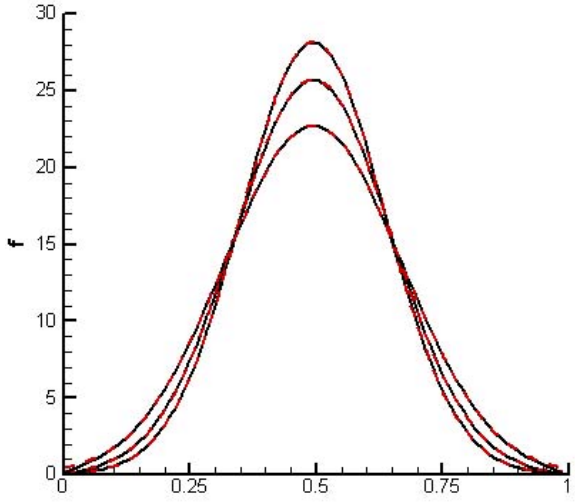


Fig. 3.19. Omesh, planar, Gaussian distribution test.

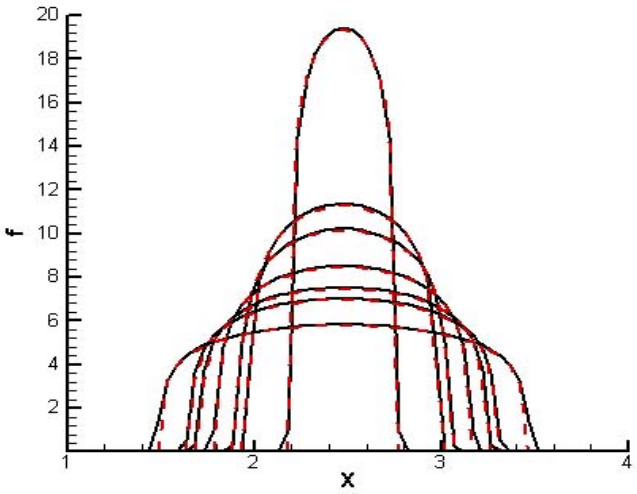


Fig. 3.20. Omesh, planar, nonlinear diffusion coef test.

4. Subroutines and namelist options

This section is devoted to describe the subroutines related to the radiation transport module in DRACO and the namelist options in the input file. We list the subroutines and their purpose in Table 4.1.

Figures 4.1-4 show the flow chart of radiation transport.

If only the radiation emission is considered, no radiation transport will be carried out. The energy contribution for electron source from radiation is simply from the radiation emission. Two diffusion schemes are available for building the 9-point diffusion matrix, that is, the Kershaw scheme and the Verdon scheme. After the matrix is set up, it is solved by the same solver routine "math_solver_rad_trans_2d".

Table 4.1 Subroutines in the radiation transport module

rad_trans_control	controls the radiation transport routines
rad_trans_emission_lte	determine the frequency dependent emission
rad_trans_opac_offtable_lte	determine the frequency dependent local thermal equilibrium opacities
rad_trans_opac_table_lte	get frequency dependent local thermal equilibrium opacities via table lookup
rad_trans_elec_srce_trms_2d	determines the electron source terms determined by the change of in the radiation field
elec_srce_trms_2d_no_transport	determines the electron source terms determined by the total emission
rad_gradient_2d	calculate the radiation energy gradient
math_solver_rad_trans_2d	solve for the change in radiation energy density with multiple solver options
rad_trans_coefs_2d	determines the matrix for the radiation diffusion under the old Verdon's scheme
rad_trans_coefs_2d_kershaw	determines the matrix for the radiation diffusion under the Kershaw's scheme

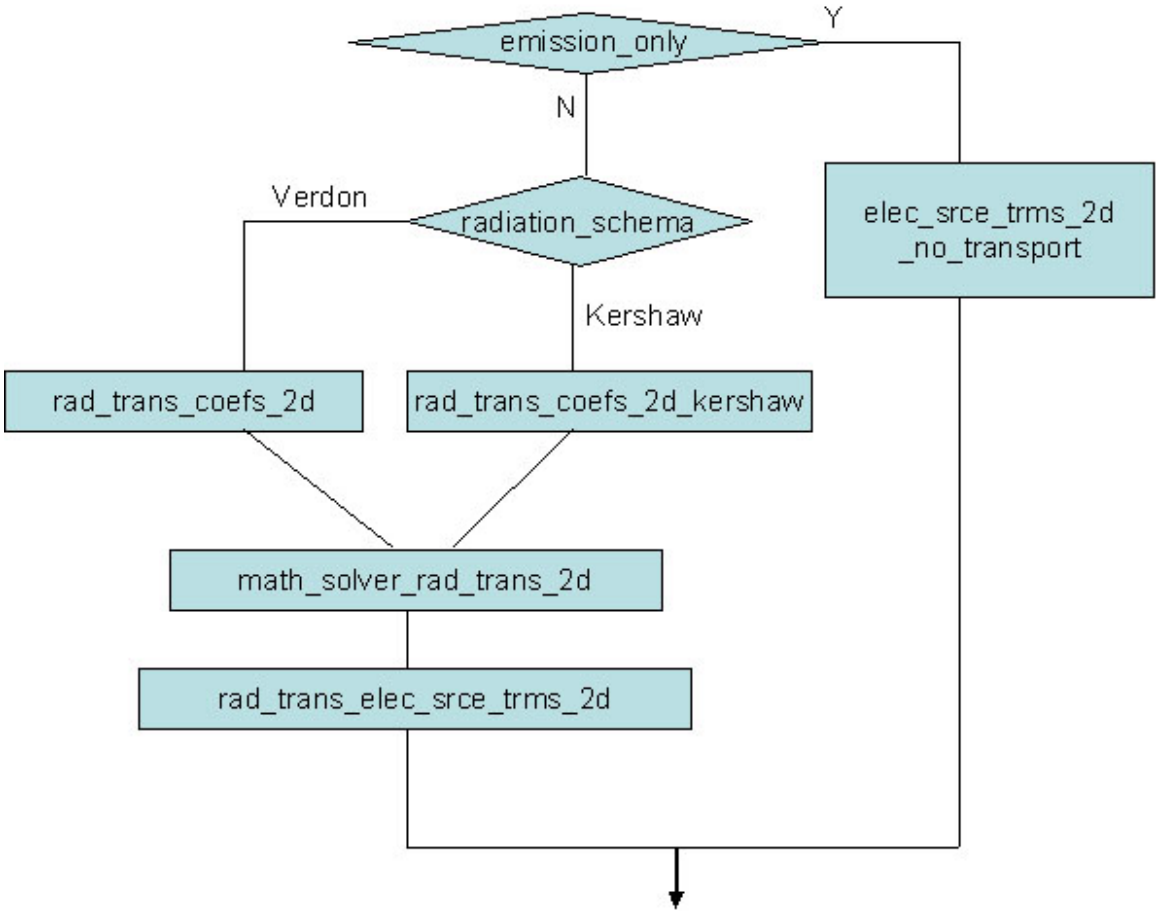


Fig. 4.1. Main control of the radiation transport.

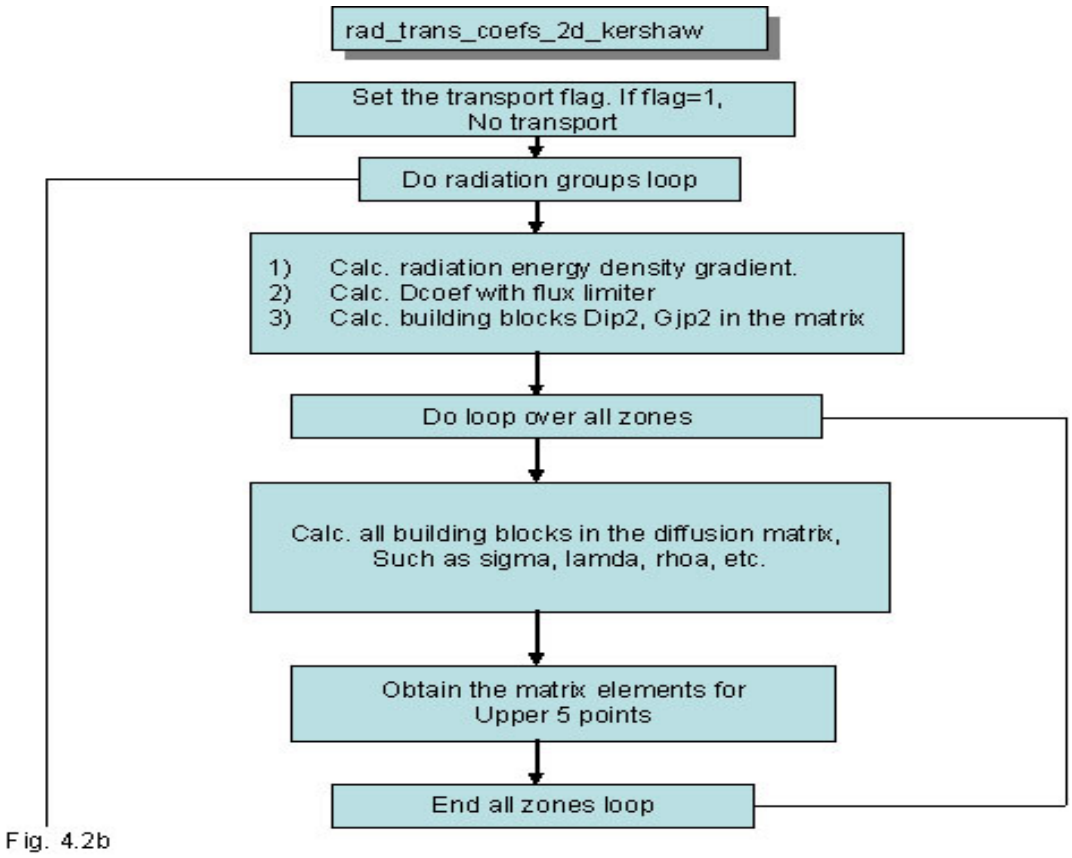


Fig. 4.2a. Flow chart of Kershaw scheme.

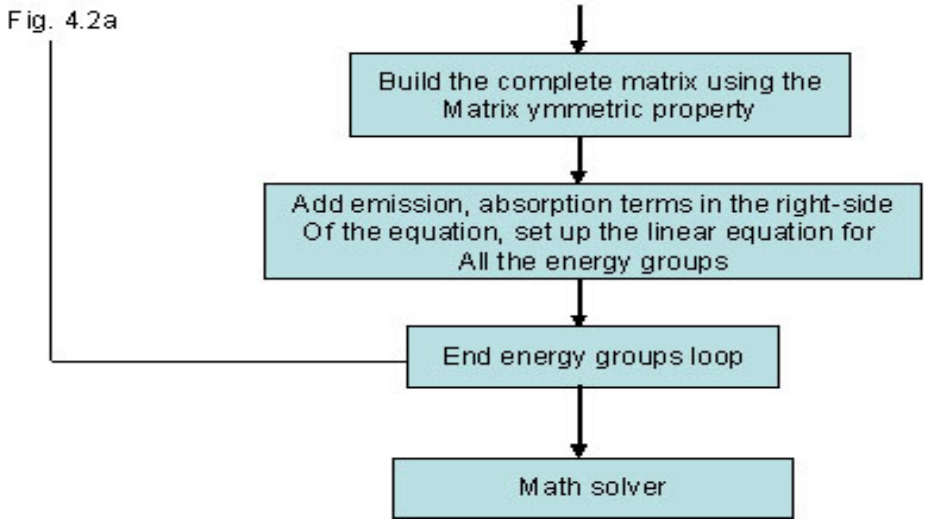


Fig. 4.2b. Flow chart of Kershaw scheme.

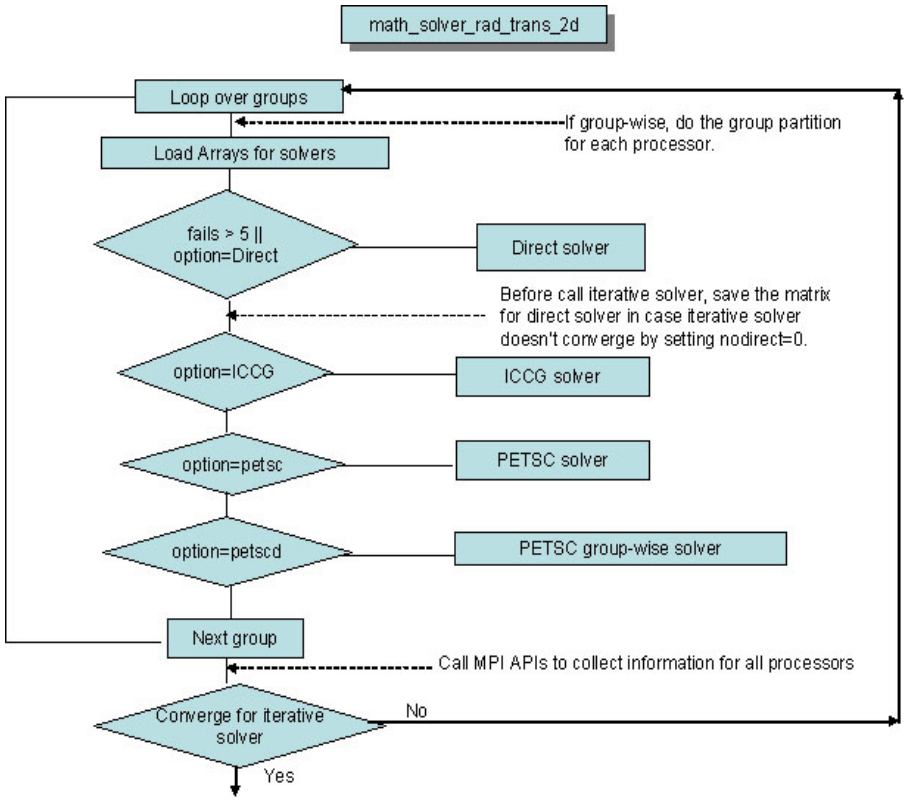


Fig. 4.3. Flow chart of math solver for the diffusion differencing linear equation.

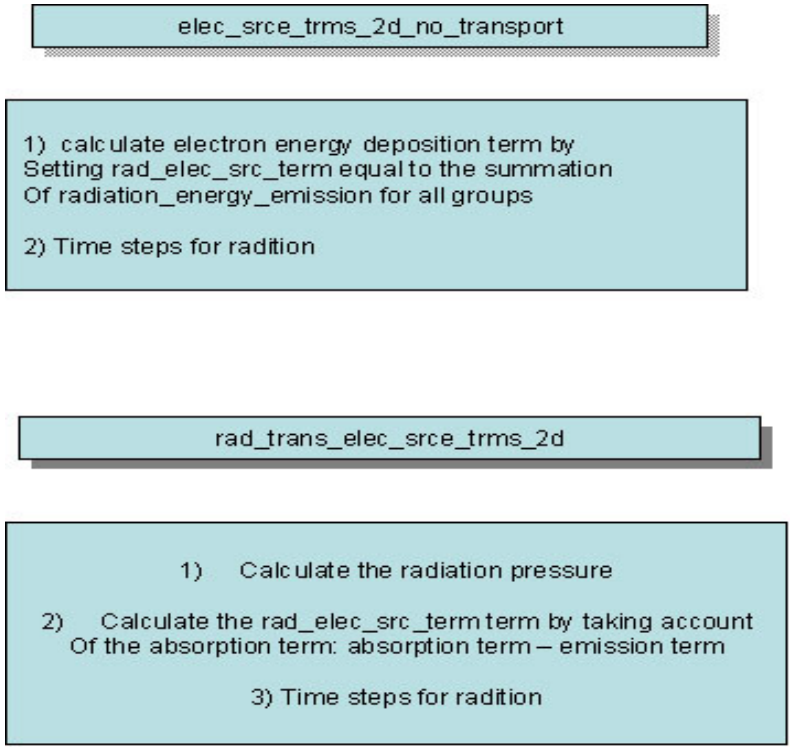


Fig. 4.4. Coupling of radiation and plasma. The radiation contributes the pressure to the hydrodynamics and energy (only emission energy or with the absorption energy) to the electron energy source term.

The namelist options are listed in Table 4.2

Variables	Def. value	Dim.	Meaning
opacity_table_format	lle	s	Opacity table format. Three options are available: “Wisconsin”, “TOPS”, “lle”.
rad_trans_bc_ism	zero	s	Left-side boundary condition. Two options are available: “zero”-no escape, reflective; “none”-free boundary, escape.
rad_trans_bc_iep	none	s	Right-side boundary condition. Options as in rad_trans_bc_ism.
rad_trans_bc_jsm	zero	s	Bottom-side boundary condition. Options as in rad_trans_bc_ism.
rad_trans_bc_jep	zero	s	Top-side boundary condition. Options as in rad_trans_bc_ism.
rad_trans_bc_ksm	zero	s	For three dimension problem. Not implemented yet.
rad_trans_bc_kep	zero	s	For three dimension problem. Not implemented yet.
number_of_radiation_freq_groups	0	s	Number of radiation frequency groups.
solver_option_rt	iccg	s	Choice of radiation diffusion matrix solver. Four options are available: “yale” – Yale direct solver; “iccg” - iccg iterative solver; “petsc” - parallel PETSC library [10] solver across all processors; “petscd” - parallel PETSC library solver group-wised.
petsc_ksp_type_rt	preonly	s	Krylov Subspace method choice. Twelve options are available: 'richardson' - Richardson method; 'chebychev' - Chebychev method; 'cg' - Conjugate Gradient method; 'bicg' - BiConjugate Gradient method; 'gmres' - Gen. Min. Residual method; 'bcgs' - BiCGSTAB method; 'cgs' - Con. Gradient sq. method; 'tfqmr' - Trans-free QMR 1 method; 'tcqmr' - Trans-free QMR 2 method; 'cr' - Conjugate Residual method;

			'lsqr' - Least Squares method; 'preonly' - Shell for no KSP method.
petsc_pc_type_rt	lu	s	Choice of preconditioner types. Thirteen options are available: 'jacobi' - Jacobi precondictioner; 'bjacobi' - Block Jacobi preconditioner; 'sor' - SOR method; 'eisenstat' - SOR w/Eisenstat preconditioner; 'icc' - Incomplete Cholesky preconditioner; 'ilu' - Incomplete LU preconditioner; 'asm' - Additive Schwarz preconditioner; 'sles' - Linear solver preconditioner; 'composite' - Combin. preconditioners; 'lu' - LU preconditioner; 'cholesky' - Cholesky preconditioner; 'none' - no preconditioner; 'shell' - Shell for user defined preconditioner.
radiation_schema	kershaw	s	Verdon's diffusion matrix scheme. Kershaw's diffusion matrix scheme.
verbose_rt	false	s	Prints solver iterations each time step.
mprtfrt	99999	s	Radiation transport information print frequency.
number_iters_limit_ solver_rt	1000	s	Maximum number of iterations for iterative solver.
number_iters_max_ solver_rt	1000	s	Maximum number iterations per cycle for radiation transport solver.
tot_its_group_rt	1000	1	total iterations per group for radiation.
solver_atol_rt	10	s	absolute size of residual norm for rt solver
solver_dtol_rt	1.0E100	s	Relative increase in the residual.
solver_rtol_rt	1.0E-06	s	Relative converge test.
rad_trans_energy_ limit	1.0E-10	s	radiation transport energy limit, if less than this value, no radiation transport.
flux_limiter_type	1	s	radiation transport flux_limiter_type. Five options are available: 0 – no limiter; 1 – sum limiter; 2 – quadrilateral limiter; 3 – max limiter; 4 – LP limiter.
emission_only	false	s	Only sink term for radiation. No radiation transport.

s denotes scalar value.

Acknowledgement

Support for this work was provided by the U.S. Department of Energy through the University of Rochester.

References

[1] John Lindl, “Development of the Indirect-Drive Approach to Inertial Confinement Fusion and the Target Physics Basis for Ignition and Gain”, Nov. 1995, Physics of Plasmas.

[2] G.C. Pomraning, “Radiation Hydrodynamics”, Pergamon Press, 1973.

D. Mihalas and B.W. Mihalas, “Foundations of Radiation Hydrodynamics”, Oxford University Press, 1984.

[3] D. Keller, T.J.B. Collins, J.A. Delettrez, P.W. McKenty, P.B. Radha, B. Whitney, and G.A. Moses, “DRACO – A New Multidimensional Hydrocode”, Bull. Am. Phys. Soc. 44(7), 37(1999).

[4] D.S. Kershaw, “Differencing of the Diffusion Equation in Lagrangian Hydrodynamic codes”, Journal of Computational Physics, 39, 375-395 (1981).

[5] Bell, G.I., and Glasstone, S., “Nuclear Reactor Theory”, Reinhold, 1970.

[6] J.E. Morel, “Asymptotic Accuracy of Diffusion, P1, and Flux-Limited Diffusion Theories in the Equilibrium Limit”, J.Q.S.R.T., 1999.

[7] G.L. Olson, L.H. Auer, M.L. Hall, “Diffusion, P1, and Other Approximate Forms of Radiation Transport”, LA-UR-99-471, 1999.

[8] C.D. Levermore, G.C. Pomraning, “A Flux-Limited Diffusion Theory”, Astrophysical J., 248, 321, 1981.

[9] Ya.B. Zeldovich, Yu.P. Raizer, “Physics of Shock Waves and High Temperature Hydrodynamics Phenomena”, Dover Publications.

[10] S. Balay, W. Fropp, L.C. McInnes, B. Smith, “PETSc Users Manual”, http://www.mcs.anl.gov/petsc.