Verification and Analysis of the Radiation Transport Packages in the BUCKY 1-D Radiation-Hydrodynamics Code

Gregory A. Rochau

June 2003

UWFDM-1208

FUSION TECHNOLOGY INSTITUTE

UNIVERSITY OF WISCONSIN

MADISON WISCONSIN

Verification and Analysis of the Radiation Transport Packages in the BUCKY 1-D Radiation-Hydrodynamics Code

Gregory A. Rochau*

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

http://fti.neep.wisc.edu

June 2003

UWFDM-1208

*Sandia National Laboratory, Albuquerque NM 87185.

Contents

1 Introduction

The BUCKY radiation-hydrodynamics code has been used to model high energy-density plasmas over a wide range of plasma conditions. Depending on the properties of both the material(s) and the radiation field, different approximations to the Boltzmann transport equation can predict significantly different solutions for the spectral and spatial distribution of the radiation energy density. Therefore, properly modeling the dynamics of a plasma requires a good understanding of the different transport approximations, and a high confidence in the implementation of these approximations within the code.

The two radiation transport approximations in common use in BUCKY are flux-limited diffusion (FLD), and multi-angle short-characteristics (short-c). Both require a number of test problems to verify the accuracy of the finite difference equations as implemented in the code. This verification is accomplished in essentially three parts: First, a few simple test problems are solved analytically by both time-independent transport and diffusion, and are compared to BUCKY calculations using both FLD and short-c. Second, some analytic problems specific to the FLD equations are compared to BUCKY calculations to verify each of the terms specific to FLD (such as the flux-limiter). Third, two time-dependent benchmark problems, which are intended to verify both the radiation transport and the associated coupling between the radiation energy and the plasma energy, are compared to both FLD and short-c.

Most of these problems are only applicable to planar geometries, and therefore the majority of the discussion takes place in Cartesian coordinates. However, because the FLD equations are also implemented for cylindrical and spherical geometries, a few problems that are specific to FLD are also tested in these coordinate systems.

After completion of this test suite, one can have confidence that the finite difference equations in each transport approximation are properly implemented to solve the equations for which they are intended. One should note, however, that this says nothing of the applicability of each transport approximation to a particular problem. This is a much more complicated issue, and must usually be addressed on a case-by-case basis.

2 The Transport Equation

In a material with only isotropic elastic scattering and isotropic external sources, the radiation transport equation in 1-D Cartesian coordinates can be written as [1]:

$$\begin{aligned} \frac{1}{c}\frac{\partial I(r,t,\mu,\nu)}{\partial t} + \mu\frac{\partial I(r,t,\mu,\nu)}{\partial r} &= -\sigma_t(r,t,\nu)I(r,t,\mu,\nu) \\ &+ \frac{1}{2}\sigma_s(r,t,\nu)\int_{-1}^{1} I(r,t,\mu' \to \mu,\nu)d\mu' \\ &+ 2\pi\sigma_e(r,t,\nu)B_\nu(r,t,\nu) + 2\pi S(r,t,\nu) \end{aligned} \tag{1}$$

where

$I(r,t,\mu,\nu) \hat{=}$ the specific intensity in units of $\frac{J}{cm^2\, s\, \mu\, Hz}$,

$S(r,t,\nu) \hat{=}$ an external source term in units of $\frac{J}{cm^3\, s\, st\, Hz}$ [1],

$\mu \hat{=}$ the cosine of the angle between I and the unit vector in the r direction,

$c \hat{=}$ the speed of light in cm/s,

$\sigma_t \hat{=}$ the total opacity (absorption + scattering) in cm^{-1},

$\sigma_s \hat{=}$ the scattering opacity in cm^{-1},

$\sigma_e \hat{=}$ the emission opacity in cm^{-1},

$B_\nu(r,t,\nu) \hat{=}$ the Planck function in units of $\frac{J}{cm^2\, s\, st\, Hz}$ given by:

$$B_\nu = \frac{2}{h^3c^2}T_R^4\left(\frac{\left(\frac{h\nu}{T_R}\right)^3}{e^{\frac{h\nu}{T_R}} - 1}\right), \tag{2}$$

for T_R the radiation temperature in eV.

In BUCKY, some approximations to Eq. 1 can be solved by diffusion or short-characteristics.

[1] In many benchmark calculations, it may be more convenient to define a blackbody radiation temperature as the external source term. In this case, the external source term will have a functional form as in Eq. 2, and one must also define an artificial emission opacity, σ_x, in units of cm^{-1} (i.e. $S(r,t,\nu) = \sigma_x B_\nu(T)$).

2.1 Diffusion and Flux-Limited Diffusion

The simplest (and quickest) solution to Eq. 1 is the diffusion approximation. One way to derive the diffusion equation is to take the 0^{th} and 1^{st} moments of the transport equation:

$$\int_{-1}^{1}\left(\frac{1}{c}\frac{\partial I}{\partial t}+\mu\frac{\partial I}{\partial r}\right)d\mu=\int_{-1}^{1}\left(-\sigma_t I+\frac{1}{2}\sigma_s\int_{-1}^{1}Id\mu'+2\pi\sigma_e B_\nu+2\pi S\right)d\mu \tag{3}$$

$$\int_{-1}^{1}\mu\left(\frac{1}{c}\frac{\partial I}{\partial t}+\mu\frac{\partial I}{\partial r}\right)d\mu=\int_{-1}^{1}\mu\left(-\sigma_t I+\frac{1}{2}\sigma_s\int_{-1}^{1}Id\mu'+2\pi\sigma_e B_\nu+2\pi S\right)d\mu. \tag{4}$$

Defining $I_0=\int_{-1}^{1}Id\mu$ and $I_1=\int_{-1}^{1}\mu Id\mu$, and carrying out the integrations by assuming that all opacities are isotropic gives:

$$\frac{1}{c}\frac{\partial I_0}{\partial t}+\frac{\partial I_1}{\partial r}=-\sigma_a I_0+4\pi\sigma_e B_\nu+4\pi S \tag{5}$$

$$\frac{1}{c}\frac{\partial I_1}{\partial t}+\frac{\partial}{\partial r}fI_0=-\sigma_t I_1, \tag{6}$$

where f is the normalized Eddington factor defined by:

$$f=\frac{1}{I_0}\int_{-1}^{1}\mu\mu Id\mu. \tag{7}$$

In the diffusion approximation, it is assumed that the specific intensity has only a linear dependence on angle in the form [2]:

$$I(r,\mu,t)=\frac{1}{2}I_0+\frac{3}{2}\mu I_1, \tag{8}$$

so that the Eddington factor is evaluated as $f=\frac{1}{3}$. Finally, assuming that I_1 is steady state, then Eq. 5 and Eq. 6 can be combined to give the time-dependent 1-D diffusion equation:

$$\frac{\partial E}{\partial t}-\nabla cD\nabla E=-c\sigma_a E+4\pi\sigma_e B_\nu+4\pi S, \tag{9}$$

where $E=\frac{1}{c}I_0$ is the radiation energy density in units of J/cm^3, and $D=\frac{1}{3\sigma_t}$ is the classical diffusion coefficient. While, for simplicity, this equation was derived in planar geometry, it has the exact same form in any orthogonal coordinate system [2], and therefore can be applied in Cartesian, cylindrical, and spherical geometries.

Eq. 9 is simple to solve, and can provide an accurate description of the radiation field in materials that have very short optical depths (high opacities). However, for very long optical depths or very large gradients in the radiation energy density, it can instantaneously transport radiation everywhere. For example, in a cold, purely scattering medium with no external sources, Eq. 9 reduces to:

$$\frac{\partial E}{\partial t} = c\frac{\partial}{\partial r}D\frac{\partial E}{\partial r}. \tag{10}$$

If σ_t approaches 0, or $\frac{\partial E}{\partial r}$ approaches "∞", then $\frac{\partial E}{\partial t}$ ="∞", which propagates radiation everywhere instantaneously. To fix this, one can apply a flux-limiter to the diffusion coefficient so that, for $\left|\frac{\partial E}{\partial r}\right| >> \sigma_t$, the time-rate of change in the energy density evaluates to $\frac{\partial E}{\partial t} = c\frac{\partial E}{\partial r}$, which is the correct free-streaming limit.

There are many forms of the diffusion flux-limiter that have been proposed [3]. The four most commonly used are the SUM limiter:

$$D = \left[3\sigma_t + E^{-1}\left|\frac{\partial E}{\partial r}\right|\right]^{-1}, \tag{11}$$

the MAX limiter:

$$D = \left[\max\left(3\sigma_t, E^{-1}\left|\frac{\partial E}{\partial r}\right|\right)\right]^{-1}, \tag{12}$$

the Larsen limiter:

$$D = \left[(3\sigma_t)^n + \left(E^{-1}\left|\frac{\partial E}{\partial r}\right|\right)^n\right]^{-\frac{1}{n}}, \tag{13}$$

and the Simplified Levermore-Pomraning (L-P) limiter [4]:

$$D = \frac{1}{\sigma_t R}\left[\coth R - \frac{1}{R}\right], \qquad \text{for} \qquad R = \frac{\left|\frac{\partial E}{\partial r}\right|}{\sigma_t E}. \tag{14}$$

The effect that each of these limiters has on the diffusion coefficient is illustrated in Figure 1.

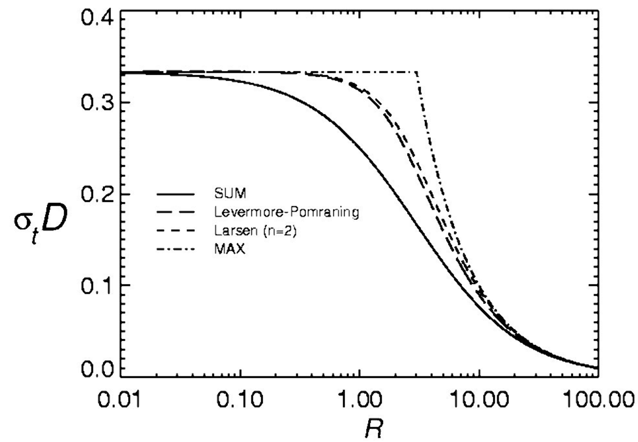


Figure 1. Flux-limited diffusion coefficients versus the scaled energy density gradient, $R = \frac{\left|\frac{\partial E}{\partial r}\right|}{\sigma_t E}$.

2.1.1 Diffusion Boundary Conditions

Obtaining a solution to Eq. 9 requires a definition of E on each boundary. The boundary conditions for diffusion can be defined through the incoming and outgoing partial flux ($\mathbf{F} = (F_{in} + F_{out})\hat{r}$) as:

$$F_{in} = -\int_{-1}^{0} \mu I d\mu \tag{15}$$

$$F_{out} = \int_{0}^{1} \mu I d\mu. \tag{16}$$

These can be solved by applying the diffusion approximation from Eq. 8 including the evaluation of I_1 from Eq. 6 to give:

$$\frac{1}{c}F_{in} = \frac{1}{4}E - (\hat{\mathbf{n}} \cdot \hat{\mathbf{r}})\frac{1}{2}D\frac{\partial E}{\partial r} \tag{17}$$

$$\frac{1}{c}F_{out} = \frac{1}{4}E + (\hat{\mathbf{n}} \cdot \hat{\mathbf{r}})\frac{1}{2}D\frac{\partial E}{\partial r}, \tag{18}$$

for $\hat{\mathbf{n}}$ the unit vector outward normal to the boundary surface.

There are many types of boundary conditions that are of interest in diffusion calculations. All can be prescribed by some combination of Eq. 17 and Eq. 18. To simplify this

Boundary Condition	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$
Dirichlet	1	0	E_0
Vacuum	$-1/2$	1	0
Source	$-1/2$	1	$-2\frac{1}{c}F_{\text{in}} = -\frac{2\pi}{c}B_\nu(T_{\text{source}})$
Reflection	0	1	0
Albedo	$\frac{1}{2}(\alpha-1)/(\alpha+1)$	1	0

Table 1. Coefficients for the diffusion boundary conditions [5]. α is the fraction of radiation reflected by the albedo boundary.

prescription, these equations can be generalized into a single expression that is valid for any boundary condition [5]:

$$\mathcal{A}E - (\hat{\mathbf{n}} \cdot \hat{\mathbf{r}})\mathcal{B}D\frac{\partial E}{\partial r} = \mathcal{C}. \tag{19}$$

Table 1 lists the factors $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ for Dirichlet, source, vacuum, and albedo boundary conditions.

2.1.2 Diffusion Finite Difference Equations

In order to solve Eq. 9 in BUCKY, it must be converted to Lagrangian coordinates and written in finite-difference form. Lagrangian coordinates are, by definition, in the reference frame of the fluid particle and must therefore automatically conserve mass. Thus, the conversion from Eulerian (observer) to Lagrangian (particle) coordinates can be written as:

$$dm = \rho(r)r^{\delta-1}dr, \tag{20}$$

where ρ is the fluid density and δ is a geometry-dependent factor which is 1 for planar geometry, 2 for cylindrical geometry, and 3 for spherical geometry.

Applying this conversion to Eq. 9 and re-arranging terms gives:

$$V\frac{\partial E}{\partial t} = \frac{\partial}{\partial m}\left(r^{\delta-1}Vc\frac{1}{3\sigma_t}\frac{\partial E}{\partial r}\right) - c\sigma_a E + 4\pi\sigma_e B_\nu + V4\pi S, \tag{21}$$

where σ_t, σ_a, and σ_e have been converted to units of $\frac{cm^2}{g}$, and V is the specific volume given by:

$$V = \frac{1}{\rho}. \tag{22}$$

This description is precise for static fluids, but requires a correction to account for a time-dependent zone thickness in a Lagrangian description where no particles are allowed to cross a zone boundary. This correction is derived from the first law of thermodynamics in the particle reference frame [6]:

$$\frac{\partial e_r}{\partial t} + P_r \frac{\partial V}{\partial t} = \dot{Q}_r, \tag{23}$$

where e_r is the specific radiation energy in units of $\frac{J}{g}$, P_r is the radiation pressure in $\frac{J}{cm^3}$, and $\dot{Q}_r$ is the heating term equivalent to everything on the right hand side of Eq. 21. Converting the specific radiation energy to the radiation energy density by $e_r = EV$ then gives:

$$V\frac{\partial E}{\partial t} + E\frac{\partial V}{\partial t} + P_r \frac{\partial V}{\partial t} = \dot{Q}_r. \tag{24}$$

Finally, inserting the classical form of the radiation pressure, $P_r = \frac{1}{3}E$, gives the full Lagrangian description of the radiation diffusion equation:

$$V\frac{\partial E}{\partial t} = \dot{Q}_r - \frac{4}{3}E\frac{\partial V}{\partial t}, \tag{25}$$

or expanding $\dot{Q}_r$ from Eq. 21:

$$V\frac{\partial E}{\partial t} = \frac{\partial}{\partial m}\left(r^{\delta-1} V c \frac{1}{3\sigma_t}\frac{\partial E}{\partial r}\right) - \frac{4}{3}E\frac{\partial V}{\partial t} - c\sigma_a E + 4\pi\sigma_e B_\nu + V 4\pi S. \tag{26}$$

As derived in the Lagrangian reference frame, this equation is applicable to planar, cylindrical, and spherical coordinates.

Solving Eq. 26 in BUCKY requires binning the photon energies into groups. This means making some choice about how to weight the opacities. Typically, this weighting is done by assuming the plasma to be at near LTE so that the radiation field is well-modeled as a Planckian distribution. Under this assumption, the three opacities in Eq. 26 can be

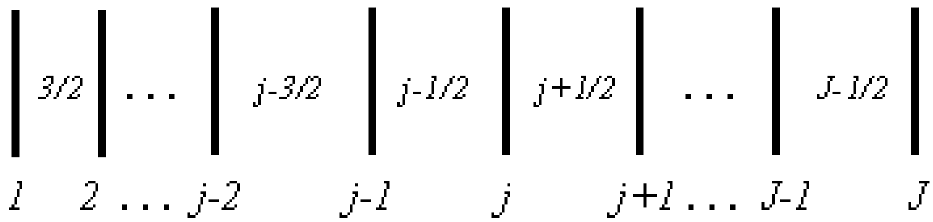


Figure 2. Finite difference grid in BUCKY for $J-1$ zones with J boundaries.

binned into the Planck emission opacity:

$$\sigma^g_{P,e} = \frac{\int_{\nu_g}^{\nu_{g+1}} \sigma_e B_\nu d\nu}{\int_{\nu_g}^{\nu_{g+1}} B_\nu d\nu}, \tag{27}$$

the Planck absorption opacity:

$$\sigma^g_{P,a} = \frac{\int_{\nu_g}^{\nu_{g+1}} \sigma_a B_\nu d\nu}{\int_{\nu_g}^{\nu_{g+1}} B_\nu d\nu}, \tag{28}$$

and the Rosseland opacity:

$$\frac{1}{\sigma^g_R} = \frac{\int_{\nu_g}^{\nu_{g+1}} \frac{1}{\sigma_t} B_\nu d\nu}{\int_{\nu_g}^{\nu_{g+1}} B_\nu d\nu}, \tag{29}$$

where ν_g are the group boundaries (in eV) for G total radiation groups. Then, the multi-group radiation diffusion equation is written as:

$$V\frac{\partial E^g}{\partial t} = \frac{\partial}{\partial m}\left(r^{\delta-1}\kappa^g_R \frac{\partial E^g}{\partial r}\right) - E^g \frac{4}{3}\frac{\partial V}{\partial t} - c\sigma^g_{P,a}E^g + 4\pi\sigma^g_{P,e}B^g_\nu + V4\pi S^g, \tag{30}$$

where each term has been integrated from ν_g to ν_{g+1}, κ^g_R is the radiation conductivity given by:

$$\kappa^g_R = \frac{cV}{3\sigma^g_R} = cD^g, \tag{31}$$

and the multi-group diffusion coefficient, D^g, can be flux-limited by any one of the flux-limiters listed in Section 2.1.

In BUCKY, the radiation energy densities are stored as zone-centered values. Therefore, given the finite grid shown in Figure 2, the finite difference form of Eq. 30 can be written as:

$$
\begin{aligned}
V_{j-\frac{1}{2}}^{n+\frac{1}{2}} \frac{E_{j-\frac{1}{2}}^{g,n+1} - E_{j-\frac{1}{2}}^{g,n}}{\Delta t^{n+\frac{1}{2}}} = & \frac{1}{\Delta m_{j-\frac{1}{2}}} \left[\frac{{r_j^{\delta-1}}^{n+\frac{1}{2}} \kappa_{R,j}^{g,n+\frac{1}{2}}}{\Delta r_j^{n+\frac{1}{2}}} \left(E_{j+\frac{1}{2}}^{g,n+1} - E_{j-\frac{1}{2}}^{g,n+1} \right) \right] \\
& - \frac{1}{\Delta m_{j-\frac{1}{2}}} \left[\frac{{r_{j-1}^{\delta-1}}^{n+\frac{1}{2}} \kappa_{R,j-1}^{g,n+\frac{1}{2}}}{\Delta r_{j-1}^{n+\frac{1}{2}}} \left(E_{j-\frac{1}{2}}^{g,n+1} - E_{j-\frac{3}{2}}^{g,n+1} \right) \right] \\
& - E_{j-\frac{1}{2}}^{g,n+1} \frac{4}{3} \dot{V}_{j-\frac{1}{2}}^{n+\frac{1}{2}} - c \sigma_{P,a_{j-\frac{1}{2}}}^{g,n+\frac{1}{2}} E_{j-\frac{1}{2}}^{g,n+1} + 4\pi \sigma_{P,e_{j-\frac{1}{2}}}^{g,n+\frac{1}{2}} B_{\nu,j-\frac{1}{2}}^{g,n+\frac{1}{2}} \\
& + V_{j-\frac{1}{2}}^{n+\frac{1}{2}} 4\pi S_{j-\frac{1}{2}}^{g,n+\frac{1}{2}},
\end{aligned} \tag{32}
$$

where n is the time index, and the work term, $\dot{V}_{j-\frac{1}{2}}^{n+\frac{1}{2}}$, is given by:

$$
\dot{V}_{j-\frac{1}{2}}^{n+\frac{1}{2}} = \frac{\left(r_j^{n+\frac{1}{2}}\right)^{\delta-1} u_j^{n+\frac{1}{2}} - \left(r_{j-1}^{n+\frac{1}{2}}\right)^{\delta-1} u_{j-1}^{n+\frac{1}{2}}}{\Delta m_{j-\frac{1}{2}}}, \tag{33}
$$

for $u_j^{n+\frac{1}{2}}$ the fluid velocity evaluated at time $n+\frac{1}{2}$.

In addition, the radiation conductivity has a different implementation for each of the various forms of the flux-limiter. The finite-difference equations for each of these limiters are given for the SUM-limiter:

$$
\kappa_{R,j}^{g,n+\frac{1}{2}} = c \left[3\sigma_{R,j-\frac{1}{2}}^{g,n+\frac{1}{2}} V_{j-\frac{1}{2}}^{n+\frac{1}{2}} + 2\left(E_{j+\frac{1}{2}}^{g,n} + E_{j-\frac{1}{2}}^{g,n} \right)^{-1} \left| \frac{E_{j+\frac{1}{2}}^{g,n} - E_{j-\frac{1}{2}}^{g,n}}{\Delta r_j^{n+\frac{1}{2}}} \right| \right]^{-1} \tag{34}
$$

the MAX-limiter:

$$
\kappa_{R,j}^{g,n+\frac{1}{2}} = c \left[\max\left(3\sigma_{R,j-\frac{1}{2}}^{g,n+\frac{1}{2}} V_{j-\frac{1}{2}}^{n+\frac{1}{2}}, 2\left(E_{j+\frac{1}{2}}^{g,n} + E_{j-\frac{1}{2}}^{g,n} \right)^{-1} \left| \frac{E_{j+\frac{1}{2}}^{g,n} - E_{j-\frac{1}{2}}^{g,n}}{\Delta r_j^{n+\frac{1}{2}}} \right| \right) \right]^{-1} \tag{35}
$$

the Larsen-limiter:

$$
\kappa_{R,j}^{g,n+\frac{1}{2}} = c \left[\left(3\sigma_{R,j-\frac{1}{2}}^{g,n+\frac{1}{2}} V_{j-\frac{1}{2}}^{n+\frac{1}{2}} \right)^{n'} + \left(2\left(E_{j+\frac{1}{2}}^{g,n} + E_{j-\frac{1}{2}}^{g,n} \right)^{-1} \left| \frac{E_{j+\frac{1}{2}}^{g,n} - E_{j-\frac{1}{2}}^{g,n}}{\Delta r_j^{n+\frac{1}{2}}} \right| \right)^{n'} \right]^{-\frac{1}{n'}} \tag{36}
$$

and the approximate simplified Levermore-Pomraning-limiter:

$$
\begin{aligned}
\kappa_{R,j}^{g,n+\frac{1}{2}} = & c \frac{2 + R_j^{g,n+\frac{1}{2}}}{\sigma_{R,j-\frac{1}{2}}^{g,n+\frac{1}{2}} V_{j-\frac{1}{2}}^{n+\frac{1}{2}} \left[6 + 3R_j^{g,n+\frac{1}{2}} + \left(R_j^{g,n+\frac{1}{2}} \right)^2 \right]} \\
\text{for} \qquad & R_j^{g,n+\frac{1}{2}} = 2\left[\sigma_{R,j-\frac{1}{2}}^{g,n+\frac{1}{2}} V_{j-\frac{1}{2}}^{n+\frac{1}{2}} \left(E_{j+\frac{1}{2}}^{g,n} + E_{j-\frac{1}{2}}^{g,n} \right) \right]^{-1} \left| \frac{E_{j+\frac{1}{2}}^{g,n} - E_{j-\frac{1}{2}}^{g,n}}{\Delta r_j^{n+\frac{1}{2}}} \right|.
\end{aligned} \tag{37}
$$

For convenience, Eq. 32 can be reduced to [7]:

$$\alpha_{j-\frac{1}{2}}^{n+\frac{1}{2}}\left(E_{j-\frac{1}{2}}^{g,n+1}-E_{j-\frac{1}{2}}^{g,n}\right)=a_j^{g,n+\frac{1}{2}}\left(E_{j+\frac{1}{2}}^{g,n+1}-E_{j-\frac{1}{2}}^{g,n+1}\right)-a_{j-1}^{g,n+\frac{1}{2}}\left(E_{j-\frac{1}{2}}^{g,n+1}-E_{j-\frac{3}{2}}^{g,n+1}\right)$$
$$-\gamma_{j-\frac{1}{2}}^{n+\frac{1}{2}}E_{j-\frac{1}{2}}^{g,n+1}-\omega_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}E_{j-\frac{1}{2}}^{g,n+1}+\beta_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}, \tag{38}$$

by definition of the coefficients:

$$\alpha_{j-\frac{1}{2}}^{n+\frac{1}{2}}=V_{j-\frac{1}{2}}^{n+\frac{1}{2}}\frac{\Delta m_{j-\frac{1}{2}}}{\Delta t^{n+\frac{1}{2}}} \tag{39}$$

$$a_j^{g,n+\frac{1}{2}}=r_j^{\delta-1^{n+\frac{1}{2}}}\frac{\kappa_{R,j}^{g,n+\frac{1}{2}}}{\Delta r_j^{n+\frac{1}{2}}} \tag{40}$$

$$\gamma_{j-\frac{1}{2}}^{n+\frac{1}{2}}=\frac{4}{3}\dot{V}_{j-\frac{1}{2}}^{n+\frac{1}{2}}\Delta m_{j-\frac{1}{2}} \tag{41}$$

$$\omega_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}=c\sigma_{P,A_{j-\frac{1}{2}}}^{g,n+\frac{1}{2}}\Delta m_{j-\frac{1}{2}} \tag{42}$$

$$\beta_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}=4\pi\sigma_{P,e_{j-\frac{1}{2}}}^{g,n+\frac{1}{2}}B_{\nu,j-\frac{1}{2}}^{g,n+\frac{1}{2}}\Delta m_{j-\frac{1}{2}}+V_{j-\frac{1}{2}}^{n+\frac{1}{2}}4\pi S_{j-\frac{1}{2}}^{g,n+1}\Delta m_{j-\frac{1}{2}}. \tag{43}$$

Finally, collecting terms in Eq. 38 gives the tri-diagonal matrix equation for the radiation energy density at time $n+1$:

$$-A_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}E_{j+\frac{1}{2}}^{g,n+1}+B_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}E_{j-\frac{1}{2}}^{g,n+1}-C_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}E_{j-\frac{3}{2}}^{g,n+1}=D_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}, \tag{44}$$

where the matrix coefficients are given by:

$$A_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}=a_j^{g,n+\frac{1}{2}} \tag{45}$$

$$B_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}=\alpha_{j-\frac{1}{2}}^{n+\frac{1}{2}}+a_j^{g,n+\frac{1}{2}}+a_{j-1}^{g,n+\frac{1}{2}}+\gamma_{j-\frac{1}{2}}^{n+\frac{1}{2}}+\omega_{j-\frac{1}{2}}^{g,n+\frac{1}{2}} \tag{46}$$

$$C_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}=a_{j-1}^{g,n+\frac{1}{2}} \tag{47}$$

$$D_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}=\beta_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}+\alpha_{j-\frac{1}{2}}^{n+\frac{1}{2}}E_{j-\frac{1}{2}}^{g,n}. \tag{48}$$

It should be noted that each of these matrix coefficients are listed as being evaluated at time $n+\frac{1}{2}$. In reality, these coefficients depend on the energy density, which is not yet known at time $n+\frac{1}{2}$, so that they are actually evaluated based on the energy density at time n. This solution to the diffusion equation is therefore semi-implicit. In some instances, the solution can be made more implicit by iterating over a time step until these coefficients (or the radiation energy density itself) converge on the value at time $n+1$. However, there is no guarantee that the iteration will converge in every situation, and may occasionally lead to erroneous solutions. Additionally, because these equations are derived in a 1-D coordinate system, the constant Lagrangian mass term, $\Delta m_{j-\frac{1}{2}}$, is given in units of $\frac{g}{cm^2}cm^{\delta-1}$. Thus, the formulation of the diffusion equation given in Eq. 44 is applicable in planar, cylindrical, and spherical coordinates.

Because the radiation energy density, E, is a zone-centered quantity in BUCKY, then the matrix coefficients in Eq. 44 are only good for $3 \leq j \leq J-1$. The matrix values on the edges must therefore be evaluated using the boundary conditions from Eq. 19. Discretizing the boundary condition at $j=2$ and $j=J$ on the finite grid of Figure 2 gives:

$$\mathcal{A}_l E_1^{g,n+1} + \mathcal{B}_l \frac{1}{c} \kappa_{R,1}^{g,n+\frac{1}{2}} \left(\frac{E_{\frac{3}{2}}^{g,n+1} - E_1^{g,n}}{\Delta r_1^{n+\frac{1}{2}}} \right) = \mathcal{C}_l^{n+1} \tag{49}$$

$$\mathcal{A}_r E_J^{g,n+1} - \mathcal{B}_r \frac{1}{c} \kappa_{R,J}^{g,n+\frac{1}{2}} \left(\frac{E_J^{g,n+1} - E_{J-\frac{1}{2}}^{g,n}}{\Delta r_J^{n+\frac{1}{2}}} \right) = \mathcal{C}_r^{n+1}, \tag{50}$$

where Eq. 49 is applied on the left boundary $(j=1)$ and Eq. 50 is applied on the right boundary $(j=J)$. The radiation energy density on these boundaries $(E_1$ and $E_J)$ are defined on the first and last node (not the zone centers) so that $\kappa_{R,(1,J)}$ and $\Delta r_{(1,J)}$ are defined for:

$$\Delta r_1^{n+\frac{1}{2}} = \frac{1}{2}\left(r_2^{n+\frac{1}{2}} - r_1^{n+\frac{1}{2}} \right) \tag{51}$$

$$\Delta r_J^{n+\frac{1}{2}} = \frac{1}{2}\left(r_J^{n+\frac{1}{2}} - r_{J-1}^{n+\frac{1}{2}} \right). \tag{52}$$

Then, solving for the boundary values and plugging into Eq. 44 gives:

$$\begin{aligned} -A_{\frac{3}{2}}^{g,n+\frac{1}{2}} E_{\frac{5}{2}}^{g,n+1} + \left[B_{\frac{3}{2}}^{g,n+\frac{1}{2}} + a_1^{g,n+\frac{1}{2}} \frac{\mathcal{B}_l a_1^{g,n+\frac{1}{2}}}{c r_1^{\delta-1^{n+\frac{1}{2}}} \mathcal{A}_l - \mathcal{B}_l a_1^{g,n+\frac{1}{2}}} \right] E_{\frac{3}{2}}^{g,n+1} \\ = D_{\frac{3}{2}}^{g,n+\frac{1}{2}} + a_1^{g,n+\frac{1}{2}} \frac{c r_1^{\delta-1^{n+\frac{1}{2}}} \mathcal{C}_l^{n+1}}{r_1^{\delta-1^{n+\frac{1}{2}}} c\mathcal{A}_l - \mathcal{B}_l a_1^{g,n+\frac{1}{2}}} \end{aligned} \tag{53}$$

$$\left[B^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} + a^{g,n+\frac{1}{2}}_{J} \frac{\mathcal{B}_r a^{g,n+\frac{1}{2}}_{J}}{cr_J^{\delta-1^{n+\frac{1}{2}}} \mathcal{A}_r - \mathcal{B}_r a^{g,n+\frac{1}{2}}_{J}}\right] E^{g,n+1}_{J-\frac{1}{2}} - C^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} E^{g,n+1}_{J-\frac{3}{2}}$$
$$= D^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} + a^{g,n+\frac{1}{2}}_{J} \frac{cr_J^{\delta-1^{n+\frac{1}{2}}} \mathcal{C}_r^{n+1}}{cr_J^{\delta-1^{n+\frac{1}{2}}} \mathcal{A}_r - \mathcal{B}_r a^{g,n+\frac{1}{2}}_{J}}, \tag{54}$$

which implies that the matrix coefficients at $j = 2$ and $j = J$ are given by:

$$A^{g,n+\frac{1}{2}}_{\frac{3}{2}} = a^{g,n+\frac{1}{2}}_{2} \tag{55}$$

$$B^{g,n+\frac{1}{2}}_{\frac{3}{2}} = \alpha^{n+\frac{1}{2}}_{\frac{3}{2}} + a^{g,n+\frac{1}{2}}_{2} + a^{g,n+\frac{1}{2}}_{1} + \gamma^{n+\frac{1}{2}}_{\frac{3}{2}} + \omega^{g,n+\frac{1}{2}}_{\frac{3}{2}} + a^{g,n+\frac{1}{2}}_{1} \frac{\mathcal{B}_l a^{g,n+\frac{1}{2}}_{1}}{cr_1^{\delta-1^{n+\frac{1}{2}}} \mathcal{A}_l - \mathcal{B}_l a^{g,n+\frac{1}{2}}_{1}} \tag{56}$$

$$C^{g,n+\frac{1}{2}}_{\frac{3}{2}} = 0 \tag{57}$$

$$D^{g,n+\frac{1}{2}}_{\frac{3}{2}} = \beta^{g,n+\frac{1}{2}}_{\frac{3}{2}} + \alpha^{n+\frac{1}{2}}_{\frac{3}{2}} E^{g,n}_{\frac{3}{2}} + a^{g,n+\frac{1}{2}}_{1} \frac{cr_1^{\delta-1^{n+\frac{1}{2}}} \mathcal{C}_l^{n+1}}{cr_1^{\delta-1^{n+\frac{1}{2}}} \mathcal{A}_l - \mathcal{B}_l a^{g,n+\frac{1}{2}}_{1}} \tag{58}$$

and

$$A^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} = 0 \tag{59}$$

$$B^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} = \alpha^{n+\frac{1}{2}}_{J-\frac{1}{2}} + a^{g,n+\frac{1}{2}}_{J} + a^{g,n+\frac{1}{2}}_{J-1} + \gamma^{n+\frac{1}{2}}_{J-\frac{1}{2}} + \omega^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} + a^{g,n+\frac{1}{2}}_{J} \frac{\mathcal{B}_r a^{g,n+\frac{1}{2}}_{J}}{cr_J^{\delta-1^{n+\frac{1}{2}}} \mathcal{A}_r - \mathcal{B}_r a^{g,n+\frac{1}{2}}_{J}} \tag{60}$$

$$C^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} = a^{g,n+\frac{1}{2}}_{J-1} \tag{61}$$

$$D^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} = \beta^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} + \alpha^{n+\frac{1}{2}}_{J-\frac{1}{2}} E^{g,n}_{J-\frac{1}{2}} + a^{g,n+\frac{1}{2}}_{J} \frac{cr_J^{\delta-1^{n+\frac{1}{2}}} \mathcal{C}_r^{n+1}}{cr_J^{\delta-1^{n+\frac{1}{2}}} \mathcal{A}_r - \mathcal{B}_r a^{g,n+\frac{1}{2}}_{J}}. \tag{62}$$

The Thomas algorithm [8] can then be used to solve Eq. 44 by defining the forward-elimination variables EE and FF as:

$$EE^{g,n+\frac{1}{2}}_{\frac{1}{2}} = FF^{g,n+\frac{1}{2}}_{\frac{1}{2}} = 0 \tag{63}$$

$$EE^{g,n+\frac{1}{2}}_{j-\frac{1}{2}} = \frac{A^{g,n+\frac{1}{2}}_{j-\frac{1}{2}}}{B^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} - C^{g,n+\frac{1}{2}}_{j-\frac{1}{2}} EE^{g,n+\frac{1}{2}}_{j-\frac{3}{2}}} \quad \text{, for} \quad 2 \le j \le J \tag{64}$$

$$FF^{g,n+\frac{1}{2}}_{j-\frac{1}{2}} = \frac{D^{g,n+\frac{1}{2}}_{j-\frac{1}{2}} + C^{g,n+\frac{1}{2}}_{j-\frac{1}{2}} FF^{g,n+\frac{1}{2}}_{j-\frac{3}{2}}}{B^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} - C^{g,n+\frac{1}{2}}_{j-\frac{1}{2}} EE^{g,n+\frac{1}{2}}_{j-\frac{3}{2}}} \quad \text{, for} \quad 2 \le j \le J, \tag{65}$$

and then back-substituting to solve for the radiation energy density at time $n+1$ using the equations:

$$E^{g,n+1}_{J-\frac{1}{2}} = FF^{g,n+\frac{1}{2}}_{J-\frac{1}{2}} \tag{66}$$

$$E^{g,n+1}_{j-\frac{1}{2}} = EE^{g,n+\frac{1}{2}}_{j-\frac{1}{2}} E^{g,n+1}_{j+\frac{1}{2}} + FF^{g,n+\frac{1}{2}}_{j-\frac{1}{2}} \quad , \text{ for } \quad 2 \leq j \leq J-1. \tag{67}$$

The mapping of variable names in BUCKY to the various quantities listed throughout this section is shown in Table 2. Additionally, a flowchart of the subroutines in BUCKY for computing the flux-limited diffusion solution is shown in Figure 3, where a description of the calculations in each subroutine is listed in Table 3.

Variable	Type	Dimensions	Units	Description
erfd2a	R*8	G_{max}, J_{max}	$\frac{J}{cm^3 \text{ group}}$	$E^{g,n+1}_{j-\frac{1}{2}}$
erfd2c	R*8	G_{max}, J_{max}	$\frac{J}{cm^3 \text{ group}}$	$E^{g,n}_{j-\frac{1}{2}}$
srfd2b	R*8	G_{max}, J_{max}	$\frac{J}{g\, s \text{ group}}$	$4\pi\sigma^{g,n+\frac{1}{2}}_{P,e_{j-\frac{1}{2}}} B^{g,n+\frac{1}{2}}_{\nu,j-\frac{1}{2}}$
esfd2b	R*8	G_{max}, J_{max}	$\frac{J}{cm^3\, s \text{ group}}$	$4\pi S^{g,n+\frac{1}{2}}_{j-\frac{1}{2}}$
sr2b	R*8	G_{max}, J_{max}	$\frac{cm^2}{g}$	$\sigma^{g,n+\frac{1}{2}}_{R,j-\frac{1}{2}}$
sp2b	R*8	G_{max}, J_{max}	$\frac{cm^2}{g}$	$\sigma^{g,n+\frac{1}{2}}_{P,a_{j-\frac{1}{2}}}$
spe2b	R*8	G_{max}, J_{max}	$\frac{cm^2}{g}$	$\sigma^{g,n+\frac{1}{2}}_{P,e_{j-\frac{1}{2}}}$
ss2b	R*8	G_{max}, J_{max}	$\frac{1}{cm}$	$\sigma^{g,n+\frac{1}{2}}_{x_{j-\frac{1}{2}}}$
hnu1	R*8	$G_{max}+1$	eV	ν_g
xkrp1b	R*8	$J_{max}+1$	$\frac{cm^2}{s}$	$\kappa^{g,n+\frac{1}{2}}_{R,j}$
xkrm1b	R*8	$J_{max}+1$	$\frac{cm^2}{s}$	$\kappa^{g,n+\frac{1}{2}}_{R,j-1}$
dmass2	R*8	J_{max}	$\frac{g}{cm^{3-\delta}}$	$\Delta m_{j-\frac{1}{2}}$
v2b	R*8	J_{max}	$\frac{cm^3}{g}$	$V^{n+\frac{1}{2}}_{j-\frac{1}{2}}$
vdot2b	R*8	J_{max}	$\frac{cm^3}{g\, s}$	$\dot{V}^{n+\frac{1}{2}}_{j-\frac{1}{2}}$
rs1b	R*8	$J_{max}+1$	$cm^{\delta-1}$	$\left(r^{n+\frac{1}{2}}_j\right)^{\delta-1}$
dr2b	R*8	J_{max}	cm	$\Delta r^{n+\frac{1}{2}}_j$
al222b	R*8	J_{max}	$\frac{cm^\delta}{s}$	$\alpha^{n+\frac{1}{2}}_{j-\frac{1}{2}}$
aa221b	R*8	$J_{max}+1$	$\frac{cm^\delta}{s}$	$a^{g,n+\frac{1}{2}}_j$
gm222b	R*8	J_{max}	$\frac{cm^\delta}{s}$	$\gamma^{n+\frac{1}{2}}_{j-\frac{1}{2}}$
om222b	R*8	J_{max}	$\frac{cm^\delta}{s}$	$\omega^{g,n+\frac{1}{2}}_{j-\frac{1}{2}}$
bet22b	R*8	J_{max}	$\frac{J}{cm^{3-\delta}\, s}$	$\beta^{g,n+\frac{1}{2}}_{j-\frac{1}{2}}$
a22r	R*8	J_{max}	$\frac{cm^\delta}{s}$	$A^{g,n+\frac{1}{2}}_{j-\frac{1}{2}}$
b22	R*8	J_{max}	$\frac{cm^\delta}{s}$	$B^{g,n+\frac{1}{2}}_{j-\frac{1}{2}}$
c22r	R*8	J_{max}	$\frac{cm^\delta}{s}$	$C^{g,n+\frac{1}{2}}_{j-\frac{1}{2}}$
d2	R*8	J_{max}	$\frac{J}{cm^{3-\delta}\, s}$	$D^{g,n+\frac{1}{2}}_{j-\frac{1}{2}}$
dtb	R*8	1	s	$\Delta t^{n+\frac{1}{2}}$

Table 2. Radiation transport variables in BUCKY for flux-limited diffusion. J_{max} is the maximum allowed number of zones and G_{max} is the maximum allowed number of groups.

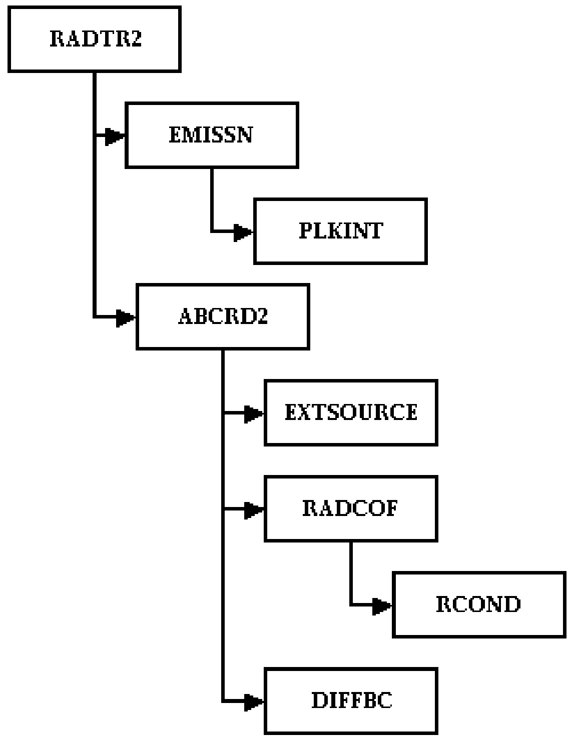


Figure 3. Flow diagram for BUCKY flux-limited diffusion subroutines.

Subroutine	Description
radtr2	Outer frequency loop Thomas back-substitution Calculate $E_j^{g,n+1}$
emissn	Calculate $4\pi\sigma_{P,e_{j-\frac{1}{2}}}^{g,n+\frac{1}{2}} B_{\nu,j-\frac{1}{2}}^{g,n+\frac{1}{2}}$
plkint	Calculate $\int_{\nu_l}^{\nu_l+1} \frac{x^3}{e^x-1}dx$
abcrd2	Calculate $A_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$, $B_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$, $C_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$, and $D_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$ Thomas forward elimination
extsource	Calculate $4\pi S_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$
radcof	Calculate $\alpha_{j-\frac{1}{2}}^{n+\frac{1}{2}}$, $a_j^{g,n+\frac{1}{2}}$, $\gamma_{j-\frac{1}{2}}^{n+\frac{1}{2}}$, $\omega_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$, and $\beta_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$
rcond	Calculate $\kappa_{R,j}^{g,n+\frac{1}{2}}$
diffbc	Calculate $A_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$, $B_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$, $C_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$, and $D_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$ for j=(2,J)

Table 3. Description of subroutines for BUCKY flux-limited diffusion.

2.2 Multi-Angle Short Characteristics

As mentioned in Section 2.1, flux-limited diffusion has the advantage of being a very efficient approximation to Eq. 1. However, while the flux-limiter expands the phase-space where diffusion may be applicable, it still has many limitations. Namely, if the conditions in a material are such that the scaled radiation energy density gradient (R in Figure 1) has the value $0.1 < R < 100$, then the diffusion length, D, is entirely dependent on an ad-hoc interpolation. Additionally, because diffusion is only applicable for near-isotropic radiation fields, then the value of E near the boundaries is usually incorrect. Therefore, a different transport approximation may be required in order to properly model the radiative transfer in some problems.

If it is assumed that the radiation field is steady state over a particular time-step, and that scattering is not an important contribution to the transport dynamics, then Eq. 1 can be simplified to:

$$\mu \frac{\partial I(r,t,\mu,\nu)}{\partial r} = -\sigma_a(r,t,\nu) I(r,t,\mu,\nu) + 2\pi\sigma_e(r,t,\nu) B_\nu(r,t,\nu) + 2\pi S(r,t,\nu). \tag{68}$$

Defining the monochromatic optical depth, τ, as:

$$\partial\tau = \sigma_a \partial r, \tag{69}$$

then Eq. 68 can be transformed to optical depth space as:

$$\mu\sigma_a \frac{\partial I}{\partial \tau} = -\sigma_a I + 2\pi\sigma_e B_\nu + 2\pi S. \tag{70}$$

Or, multiplying through by $(\mu\sigma_a)^{-1} e^{\frac{\tau}{\mu}}$ gives:

$$\frac{\partial}{\partial \tau}\left(I e^{\frac{\tau}{\mu}}\right) = \frac{2\pi}{\mu\sigma_a}\left(\sigma_e B_\nu + S\right) e^{\frac{\tau}{\mu}}. \tag{71}$$

Separating this equation into outward $(0 < \mu \leq 1)$ and inward $(-1 \leq \mu < 0)$ going rays, and integrating along characteristics (paths at some angle μ) from some nearby point in

	i	w_i	μ_i
$N = 2$	1	0.5000000000	0.2113248654
	2	0.5000000000	0.7886751346
$N = 5$	1	0.1184634425	0.0469100770
	2	0.2393143352	0.2307653449
	3	0.2844444444	0.5000000000
	4	0.2393143352	0.7692346551
	5	0.1184634425	0.9530899230

Table 4. Integration angle cosines and weights for multi-angle short-characteristics in BUCKY [10].

the slab (denoted by τ_{k-1} and τ_{k+1}) gives:

$$0 < \mu \leq 1: \quad \int_{I^+(\tau_{k-1})e^{\frac{\tau_{k-1}}{\mu}}}^{I^+(\tau_k)e^{\frac{\tau_k}{\mu}}} d\left(I' e^{\frac{\tau'}{\mu}}\right) = \frac{2\pi}{\mu} \int_{\tau_{k-1}}^{\tau_k} \frac{1}{\sigma_a} \left(\sigma_e B_\nu + S\right) e^{\frac{\tau'}{\mu}} d\tau' \tag{72}$$

$$-1 \leq \mu < 0: \quad \int_{I^-(\tau_{k+1})e^{\frac{\tau_{k+1}}{\mu}}}^{I^-(\tau_k)e^{\frac{\tau_k}{\mu}}} d\left(I' e^{\frac{\tau'}{\mu}}\right) = \frac{2\pi}{\mu} \int_{\tau_{k+1}}^{\tau_k} \frac{1}{\sigma_a} \left(\sigma_e B_\nu + S\right) e^{\frac{\tau'}{\mu}} d\tau'. \tag{73}$$

Finally, carrying out the integrals at discrete values of μ gives the analytic equations of multi-angle short-characteristics [9] at the point τ_k:

$$I_i^+(\tau_k) = I_i^+(\tau_{k-1}) e^{-\frac{(\tau_k - \tau_{k-1})}{\mu_i}} + \frac{2\pi}{\mu_i} \int_{\tau_{k-1}}^{\tau_k} \frac{1}{\sigma_a} \left(\sigma_e B_\nu + S\right) e^{\frac{-(\tau_k - \tau')}{\mu_i}} d\tau' \tag{74}$$

$$I_i^-(\tau_k) = I_i^-(\tau_{k+1}) e^{-\frac{(\tau_{k+1} - \tau_k)}{|\mu_i|}} - \frac{2\pi}{|\mu_i|} \int_{\tau_{k+1}}^{\tau_k} \frac{1}{\sigma_a} \left(\sigma_e B_\nu + S\right) e^{\frac{-(\tau' - \tau_k)}{|\mu_i|}} d\tau', \tag{75}$$

such that the radiation energy density can be computed as:

$$E(\tau_k, \nu) = \frac{1}{c} \sum_{i=1}^{N} w_i \left[I_i^+(\tau_k, \nu) + I_i^-(\tau_k, \nu) \right], \tag{76}$$

where N is the total number of angles computed in each direction. Table 4 lists the angle cosines and corresponding integration weights for $N = 2$ and $N = 5$ [10].

It should be noted that, unlike the implementation of the diffusion equation in BUCKY, the method of short-characteristics described here is only applicable in planar geometry, and is only derived for time-independent radiation transport. This places serious restrictions on the usefulness of short-characteristics for many problems, and thought should be given to its applicability before evoking it for a particular simulation.

2.2.1 Short-Characteristics Finite Difference Equations

In order to solve Eq. 74 and Eq. 75 in BUCKY, they must be cast onto a finite difference grid. However, this requires making a choice about how to transform the finite grid from the spatial distribution in Figure 2 to an optical depth distribution. Because the radiation energy density is stored as a zone centered quantity in BUCKY, the choice has been made to have twice the number of grid points in optical depth space as in position space (so that solutions exist on each zone center). The transform is then given by a finite differencing of Eq. 69 at time $n+\frac{1}{2}$ as:

$$\tau_1 = 0. \tau_k^{g,n+\frac{1}{2}} - \tau_{k-2}^{g,n+\frac{1}{2}} = \sigma_{P,a_{j-\frac{1}{2}}}^{g,n+\frac{1}{2}} \left(r_j^{n+\frac{1}{2}} - r_{j-1}^{n+\frac{1}{2}} \right), \quad 2 \le j \le J \tag{77}$$

$$\tau_{k-1}^{g,n+\frac{1}{2}} = \frac{1}{2} \left(\tau_{k+1}^{g,n+\frac{1}{2}} - \tau_{k-2}^{g,n+\frac{1}{2}} \right) + \tau_{k-2}, \tag{78}$$

where k is the optical depth grid index given by $k = 2j-1$, and the opacity has been grouped as in Eq. 28 and assumed to be constant across zone j [2].

Given the finite difference grid defined by Eq. 77, Eq. 74 and Eq. 75 can be integrated between radiation group boundaries and written in finite difference form as:

$$I_{i,k}^{+g,n+1} = I_{i,k-1}^{+g,n+1} e^{-\Delta \tau_{i,k-1}^{g,n+\frac{1}{2}}} + \int_0^{\Delta \tau_{i,k-1}^{g,n+\frac{1}{2}}} S_T^{g,n+\frac{1}{2}} e^{-\Delta \tau'} d\Delta \tau' \tag{79}$$

$$I_{i,k}^{-g,n+1} = I_{i,k+1}^{-g,n+1} e^{-\Delta \tau_{i,k}^{g,n+\frac{1}{2}}} + \int_0^{\Delta \tau_{i,k}^{g,n+\frac{1}{2}}} S_T^{g,n+\frac{1}{2}} e^{-\Delta \tau'} d\Delta \tau', \tag{80}$$

where the optical depth interval, $\Delta\tau$, is defined as:

$$\Delta \tau_{i,k}^{g,n+\frac{1}{2}} = \frac{(\tau_{k+1}^{g,n+\frac{1}{2}} - \tau_k^{g,n+\frac{1}{2}})}{|\mu_i|}, \tag{81}$$

and the total source function, S_T, is defined as:

$$S_T^{g,n+\frac{1}{2}} = \frac{2\pi}{\sigma_{P,a}^{g,n+\frac{1}{2}}} \left(\sigma_{P,e}^{g,n+\frac{1}{2}} B_\nu^{g,n+\frac{1}{2}} + S^{g,n+\frac{1}{2}} \right), \tag{82}$$

[2] Because $\sigma_{P,a_{j-\frac{1}{2}}}^{g,n+\frac{1}{2}}$ is likely to be different between radiation groups, then the separation between nodes on the optical depth grid is likely to be different for each group.

	Linear Interpolation	**Quadratic Interpolation**
α_k^+	$e_{0,k} - \frac{e_{1,k}}{\Delta\tau_{k-1}}$	$e_{0,k} + \frac{e_{2,k} - (\Delta\tau_k + 2\Delta\tau_{k-1})e_{1,k}}{\Delta\tau_{k-1}(\Delta\tau_k + \Delta\tau_{k-1})}$
β_k^+	$\frac{e_{1,k}}{\Delta\tau_{k-1}}$	$\frac{(\Delta\tau_k + \Delta\tau_{k-1})e_{1,k} - e_{2,k}}{\Delta\tau_{k-1}\Delta\tau_k}$
γ_k^+	0	$\frac{e_{2,k} - \Delta\tau_{k-1}e_{1,k}}{\Delta\tau_k(\Delta\tau_k + \Delta\tau_{k-1})}$
α_k^-	0	$\frac{e_{2,k+1} - \Delta\tau_k e_{1,k+1}}{\Delta\tau_{k-1}(\Delta\tau_k + \Delta\tau_{k-1})}$
β_k^-	$\frac{e_{1,k+1}}{\Delta\tau_k}$	$\frac{(\Delta\tau_k + \Delta\tau_{k-1})e_{1,k+1} - e_{2,k+1}}{\Delta\tau_{k-1}\Delta\tau_k}$
γ_k^-	$e_{0,k+1} - \frac{e_{1,k+1}}{\Delta\tau_k}$	$e_{0,k+1} + \frac{e_{2,k+1} - (\Delta\tau_{k-1} + 2\Delta\tau_k)e_{1,k+1}}{\Delta\tau_k(\Delta\tau_k + \Delta\tau_{k-1})}$

Table 5. Coefficients for solving the source integrals for linear or quadratic interpolation of S_T in the multi-angle short-characteristics equations [9].

for $\sigma_{P,e}$ defined as in Eq. 27.

The integrals in Eq. 79 and Eq. 80 can be evaluated by assuming either a linear or quadratic variation of S_T (in optical depth space). In either case, the solution can be written as a three coefficient evaluation:

$$\int^{\pm} S_T e^{-\Delta\tau'} d\Delta\tau' = \alpha_k^{\pm} S_{T,k-1} + \beta_k^{\pm} S_{T,k} + \gamma_k^{\pm} S_{T,k+1}, \tag{83}$$

where the subscripts, i, g, and n have been suppressed for clarity. Table 5 lists the values of these coefficients for both linear and quadratic interpolations of S_T, where the exponential functions, $e_{(0,1,2),k}$ are given by [9]:

$$e_{0,k} = 1 - e^{-\Delta\tau_{k-1}}$$

$$e_{1,k} = \Delta\tau_{k-1} - e_{0,k}$$

$$e_{2,k} = (\Delta\tau_{k-1})^2 - 2e_{1,k}.$$

Therefore, the full set of $(4J-2)$ finite difference equations for multi-angle short-characteristics are written as:

$$I_{i,1}^{+g,n+1} = I_{bc,i}^{+g,n+1} \tag{84}$$

$$I_{i,K}^{-g,n+1} = I_{bc,i}^{-g,n+1} \tag{85}$$

	Source	Vacuum	Periodic	Albedo
$I_{bc,i}^{\pm g,n+1}$	$2\pi B_\nu^{g,n+\frac{1}{2}}(T_R)$	0	$I_{i,\frac{1}{K}}^{\mp g,n+1}$	$\frac{\alpha}{N}\sum_{i=1}^{N} I_{i,\frac{1}{K}}^{\mp g,n+1}$

Table 6. Boundary conditions for the partial specific intensity in the finite difference equations for multi-angle short-characteristics. α is the albedo, and T_R is the radiation temperature specified on the boundary.

$$I_{i,k}^{+g,n+1} = I_{i,k-1}^{+g,n+1}\, e^{-\Delta\tau_{i,k-1}^{g,n+\frac{1}{2}}} + \alpha_{i,k}^{+g,n+\frac{1}{2}}\, S_{T,k-1}^{g,n+\frac{1}{2}} + \beta_{i,k}^{+g,n+\frac{1}{2}}\, S_{T,k}^{g,n+\frac{1}{2}} + \gamma_{i,k}^{+g,n+\frac{1}{2}}\, S_{T,k+1}^{g,n+\frac{1}{2}} \tag{86}$$

$$I_{i,k}^{-g,n+1} = I_{i,k+1}^{-g,n+1}\, e^{-\Delta\tau_{i,k}^{g,n+\frac{1}{2}}} + \alpha_{i,k}^{-g,n+\frac{1}{2}}\, S_{T,k-1}^{g,n+\frac{1}{2}} + \beta_{i,k}^{-g,n+\frac{1}{2}}\, S_{T,k}^{g,n+\frac{1}{2}} + \gamma_{i,k}^{-g,n+\frac{1}{2}}\, S_{T,k+1}^{g,n+\frac{1}{2}}, \tag{87}$$

where K is the maximum grid index ($K = 2J-1$), and $I_{bc,i}^{\pm g,n+1}$ are the boundary conditions as listed in Table 6.

Once the values of $I_{i,k}^{\pm g,n+1}$ are known at every value of i and k, then the radiation energy density can be computed at each zone center from Eq. 76 as:

$$E_{j-\frac{1}{2}}^{g,n+1} = \frac{1}{c}\sum_{i=1}^{N} w_i \left[I_{i,2j-2}^{+g,n+1} + I_{i,2j-2}^{-g,n+1}\right], \quad 2 \le j \le J. \tag{88}$$

Alternatively, in order to better conserve flux at the zone boundaries for significantly large values of $\Delta\tau$, the energy density can be computed by integrating Eq. 70 over μ using the calculated values of $I_i^{\pm}$ to evaluate the streaming term. Then, the finite difference equations can be written on the Lagrangian grid in Figure 2 as:

$$\begin{aligned} E_{j-\frac{1}{2}}^{g,n+1} =& \frac{4\pi}{c\sigma_{P,a_{j-\frac{1}{2}}}^{g,n+\frac{1}{2}}} \left(\sigma_{P,e_{j-\frac{1}{2}}}^{g,n+\frac{1}{2}} B_{\nu,j-\frac{1}{2}}^{g,n+\frac{1}{2}} + S_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}\right) \\ & - \sum_{i=1}^{N} w_i\mu_i \left[\frac{\left(I_{i,2j-1}^{+g,n+1} - I_{i,2j-3}^{+g,n+1}\right) + \left(I_{i,2j-3}^{-g,n+1} - I_{i,2j-1}^{-g,n+1}\right)}{c\left(\tau_{2j-1}^{g,n+\frac{1}{2}} - \tau_{2j-3}^{g,n+\frac{1}{2}}\right)}\right], \quad 2 \le j \le J. \end{aligned} \tag{89}$$

The mapping of variable names in BUCKY to the various quantities listed above is shown in Table 7. Additionally, a flowchart of the short-characteristics subroutines in BUCKY is shown in Figure 4, where a description of the calculations in each subroutine is listed in Table 8.

Variable	Type	Dimensions	Units	Description
erfd2a	R*8	G_{max}, J_{max}	$\frac{J}{cm^3 \text{ group}}$	$E^{g,n+1}_{j-\frac{1}{2}}$
srfd2b	R*8	G_{max}, J_{max}	$\frac{J}{g\, s \text{ group}}$	$4\pi\sigma^{g,n+\frac{1}{2}}_{P,e_{j-\frac{1}{2}}} B^{g,n+\frac{1}{2}}_{\nu,j-\frac{1}{2}}$
esfd2b	R*8	G_{max}, J_{max}	$\frac{J}{cm^3\, s \text{ group}}$	$4\pi S^{g,n+\frac{1}{2}}_{j-\frac{1}{2}}$
sp2b	R*8	G_{max}, J_{max}	$\frac{cm^2}{g}$	$V^{n+\frac{1}{2}}_{j-\frac{1}{2}} \sigma^{g,n+\frac{1}{2}}_{P,a_{j-\frac{1}{2}}}$
spe2b	R*8	G_{max}, J_{max}	$\frac{cm^2}{g}$	$V^{n+\frac{1}{2}}_{j-\frac{1}{2}} \sigma^{g,n+\frac{1}{2}}_{P,e_{j-\frac{1}{2}}}$
ss2b	R*8	G_{max}, J_{max}	$\frac{1}{cm}$	$\sigma^{g,n+\frac{1}{2}}_{x_{j-\frac{1}{2}}}$
hnu1	R*8	$G_{max}+1$	eV	ν_g
sourcefn	R*8	$2J_{max}-1$	$\frac{J}{cm^2\, s\, sr \text{ group}}$	$\frac{1}{2\pi} S^{g,n+\frac{1}{2}}_{T_k}$
simins	R*8	$2J_{max}-1$	$\frac{J}{cm^2\, s\, sr \text{ group}}$	$\frac{1}{2\pi} I^{+g,n+1}_{i,k}$
siplus	R*8	$2J_{max}-1$	$\frac{J}{cm^2\, s\, sr \text{ group}}$	$\frac{1}{2\pi} I^{-g,n+1}_{i,k}$
dtau	R*8	$2J_{max}-2$	$-$	$\tau^{g,n+\frac{1}{2}}_{k} - \tau^{g,n+\frac{1}{2}}_{k-1}$
dtaumu	R*8	$2J_{max}-2$	$-$	$\Delta\tau^{g,n+\frac{1}{2}}_{i,k}$
alpham	R*8	$2J_{max}-1$	$-$	$\alpha^{+g,n+\frac{1}{2}}_{i,k}$
betam	R*8	$2J_{max}-1$	$-$	$\beta^{+g,n+\frac{1}{2}}_{i,k}$
gammam	R*8	$2J_{max}-1$	$-$	$\gamma^{+g,n+\frac{1}{2}}_{i,k}$
alphap	R*8	$2J_{max}-1$	$-$	$\alpha^{-g,n+\frac{1}{2}}_{i,k}$
betap	R*8	$2J_{max}-1$	$-$	$\beta^{-g,n+\frac{1}{2}}_{i,k}$
gammap	R*8	$2J_{max}-1$	$-$	$\gamma^{-g,n+\frac{1}{2}}_{i,k}$
wtangl	R*8	5	$-$	w_i
xmu	R*8	5	$-$	$\lvert\mu_i\rvert$
dr2b	R*8	J_{max}	cm	$r^{n+\frac{1}{2}}_{j} - r^{n+\frac{1}{2}}_{j-1}$

Table 7. Radiation transport variables in BUCKY for multi-angle short-characteristics. J_{max} is the maximum allowed number of zones and G_{max} is the maximum allowed number of groups.

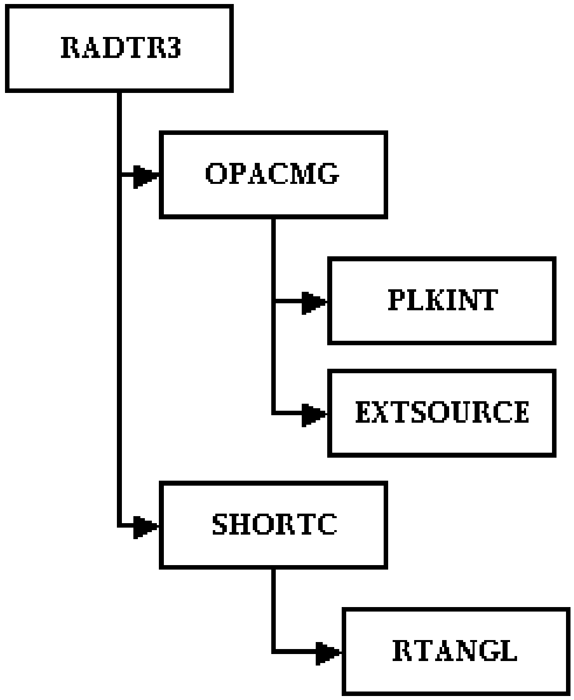


Figure 4. Flow diagram for BUCKY multi-angle short-characteristics subroutines.

Subroutine	Description
radtr3	Outer frequency loop
opacmg	Set-up optical depth grid (Calculate $\tau_k^{g,n+\frac{1}{2}} - \tau_{k-1}^{g,n+\frac{1}{2}}$) Calculate $\frac{1}{2\pi} S_{T_k}^{g,n+\frac{1}{2}}$
plkint	Calculate $\int_{\nu_l}^{\nu_l+1} \frac{x^3}{e^x-1} dx$
extsource	Calculate $4\pi S_{j-\frac{1}{2}}^{g,n+\frac{1}{2}}$
shortc	Calculate $\frac{1}{2\pi} I_{i,k}^{\pm g,n+1}$ Calculate $E_{j-\frac{1}{2}}^{g,n+1}$
rtangl	Define w_i and $\lvert\mu_i\rvert$

Table 8. Description of subroutines for BUCKY multi-angle short-characteristics.

3 Analytic Solutions for Transport and Diffusion

There are a few simple problems in cartesian coordinates where both the steady-state transport equation and the steady-state diffusion equation can be solved analytically. These problems are a good place to start in verifying the finite difference equations described in Section 2 because they are also instructive in comparing diffusive solutions to true transport solutions.

All of the problems described in this section assume a purely absorbing cold (non-radiating) slab of thickness $x = X$ in 1-D cartesian coordinates. Under these circumstances, the steady-state monoenergetic transport equation can be simplified from Eq. 68 to give:

$$\frac{\partial}{\partial x}\left(I(x)e^{\frac{\sigma_a x}{\mu}}\right) = \frac{2\pi}{\mu}S(x)e^{\frac{\sigma_a x}{\mu}}. \tag{90}$$

Breaking the specific intensity into forward and backward propagating rays, and integrating over x gives:

$$0 < \mu \leq 1: \qquad I^{+}(x,\mu) = I^{+}(0,\mu)e^{\frac{-\sigma_a x}{\mu}} + \left[\frac{2\pi}{\mu}\int_0^x S(x')e^{\frac{\sigma_a x'}{\mu}}dx'\right]e^{\frac{-\sigma_a x}{\mu}} \tag{91}$$

$$-1 \leq \mu < 0: \qquad I^{-}(x,\mu) = I^{-}(X,\mu)e^{\frac{\sigma_a (X-x)}{\mu}} + \left[\frac{2\pi}{\mu}\int_X^x S(x')e^{\frac{\sigma_a x'}{\mu}}dx'\right]e^{\frac{-\sigma_a x}{\mu}}, \tag{92}$$

where it has been assumed that σ_a is constant throughout the slab. Furthermore, if it is assumed that the source function, S, can be described by an N^{th} order polynomial of the form:

$$S(x) = \sum_{i=0}^{N} c_{S,i}x^i, \tag{93}$$

then the integrals can be evaluated to give:

$$0 < \mu \leq 1: \qquad I^{+}(x,\mu) = I^{+}(0,\mu)e^{\frac{-\sigma_a x}{\mu}} + \frac{2\pi}{\sigma_a}\sum_{i=0}^{N} c_{S,i}\gamma_i^{+}(x,\mu) \tag{94}$$

$$-1 \leq \mu < 0: \qquad I^{-}(x,\mu) = I^{-}(X,\mu)e^{\frac{\sigma_a (X-x)}{\mu}} + \frac{2\pi}{\sigma_a}\sum_{i=0}^{N} c_{S,i}\gamma_i^{-}(x,\mu), \tag{95}$$

where the coefficients $\gamma_i^{\pm}$ are given by:

$$
\begin{aligned}
\gamma_0^{+} &= [1 - e^{\frac{-\sigma_a x}{\mu}}] \\
\gamma_i^{+} &= [x^i - i\frac{\mu}{\sigma_a}\gamma_{i-1}^{+}] \\
\gamma_0^{-} &= [1 - e^{\frac{\sigma_a (X-x)}{\mu}}] \\
\gamma_i^{-} &= [x^i - X^i e^{\frac{\sigma_a (X-x)}{\mu}} - i\frac{\mu}{\sigma_a}\gamma_{i-1}^{-}].
\end{aligned}
\tag{96}
$$

Finally, if the boundary values, $I^{+}(0)$ and $I^{-}(X)$, are independent of μ, then integrating Eq. 94 and Eq. 95 over μ gives the radiation energy density as:

$$
\begin{aligned}
E(x) &= \frac{1}{c}\left[\int_{-1}^{0} I^{-}(x,\mu)d\mu + \int_{0}^{+1} I^{+}(x,\mu)d\mu\right] \\
&= \frac{1}{c}\left[I^{+}(0)E_2(\sigma_a x) + I^{-}(X)E_2(\sigma_a(X-x))\right] + \frac{2\pi}{c\sigma_a}\sum_{i=0}^{N} c_{S,i}\left[\epsilon_i^{+}(x) + \epsilon_i^{-}(x)\right],
\end{aligned}
\tag{97}
$$

where the coefficients $\epsilon_i^{\pm}$ are given by:

$$
\begin{aligned}
\epsilon_i^{+} &= \sum_{n=0}^{i} \frac{i!}{(i-n)!}\frac{1}{\sigma_a^n}\frac{x^{i-n}}{n+1}(-1)^n + (-1)^{i+1}\frac{i!}{\sigma_a^i}E_{i+2}(\sigma_a x) \\
\epsilon_i^{-} &= \sum_{n=0}^{i} \frac{i!}{(i-n)!}\frac{1}{\sigma_a^n}\left[\frac{x^{i-n}}{n+1} - X^{i-n}E_{n+2}(\sigma_a(X-x))\right],
\end{aligned}
\tag{98}
$$

and the functions $E_n(\sigma_a x)$ and $E_n(\sigma_a(X-x))$ belong to the general family of functions called the exponential integrals given by:

$$
E_n(x) = x^{n-1}\int_x^{\infty}\frac{1}{u^n}e^{-u}du. \tag{99}
$$

If either of the boundary values depend on μ (as in the case of an albedo boundary condition), then the boundary terms in Eq. 97 must be integrated independently. In this case, the equation for the radiation energy density is given as:

$$
E(x) = \frac{1}{c}\left[\int_0^1 I^{+}(0,\mu')e^{\frac{-\sigma_a x}{\mu'}}d\mu' + \int_{-1}^{0} I^{-}(X,\mu')e^{\frac{\sigma_a(X-x)}{\mu'}}d\mu'\right] + \frac{2\pi}{c\sigma_a}\sum_{i=0}^{N} c_{S,i}\left[\epsilon_i^{+}(x) + \epsilon_i^{-}(x)\right]. \tag{100}
$$

Therefore, given the boundary conditions at $x = (0, X)$ and the spatial variation in the external source term, $S(x)$, then either Eq. 97 or Eq. 100 provides the general solution for the steady-state radiation energy density in a cold purely absorbing 1-D slab.

An analytic solution to this same problem can be defined for diffusion by simplifying Eq. 9 to give:

$$\frac{\partial^2 E}{\partial x^2} - \frac{\sigma_a}{D} E = -\frac{4\pi}{cD} S. \tag{101}$$

If it is again assumed that the opacities are constant through the slab, and that the source function, S, can be described by an N^{th} order polynomial as in Eq. 93, then Eq. 101 can be solved by the superposition approach to give:

$$E(x) = ae^{\lambda x} + be^{-\lambda x} + \sum_{i=0}^{N} c_i x^i, \tag{102}$$

where λ is the inverse diffusion length, $\lambda = \sqrt{\frac{\sigma_a}{D}}$, and the coeffecients c_i are determined by:

$$\begin{aligned}
c_N &= \frac{4\pi}{c\sigma_a} c_{S,N} \\
c_{N-1} &= \frac{4\pi}{c\sigma_a} c_{S,N-1} \\
c_i &= \frac{4\pi}{c\sigma_a} c_{S,i} + \frac{D}{\sigma_a}(i+2)(i+1)c_{i+2}, \qquad 0 \le i \le N-2.
\end{aligned} \tag{103}$$

The coeffecients of the homogeneous solution (a and b) must be determined from the coupled set of boundary conditions defined in Section 2.1.1. Thus, plugging Eq. 102 into Eq. 19 gives:

$$a[\mathcal{A}_l + \mathcal{B}_l D\lambda] + b[\mathcal{A}_l - \mathcal{B}_l D\lambda] = \mathcal{C}_l - (\mathcal{A}_l c_0 + \mathcal{B}_l D c_1) \tag{104}$$

$$\begin{aligned}
&a[\mathcal{A}_r e^{\lambda X} - \mathcal{B}_r D\lambda e^{\lambda X}] + b[\mathcal{A}_r e^{-\lambda X} + \mathcal{B}_r D\lambda e^{-\lambda X}] \\
&\qquad\qquad = \mathcal{C}_r + \left[\sum_{i=1}^{N} (i\mathcal{B}_r D - \mathcal{A}_r X) c_i X^{i-1} - \mathcal{A}_r c_0\right],
\end{aligned} \tag{105}$$

where Eq. 104 is applied at the left boundary ($x = 0$) and Eq. 105 is applied at the right boundary ($x = X$). Then, solving these for the coefficients a and b gives:

$$a = \frac{\mathcal{C}_l - (\mathcal{A}_l c_0 + \mathcal{B}_l D c_1) - b[\mathcal{A}_l - \mathcal{B}_l D\lambda]}{\mathcal{A}_l + \mathcal{B}_l D\lambda} \tag{106}$$

$$\begin{aligned}
b =& \frac{\left[\mathcal{C}_r + \left(\sum_{i=1}^{N} (i\mathcal{B}_r D - \mathcal{A}_r X) c_i X^{i-1} - \mathcal{A}_r c_0\right)\right] (\mathcal{A}_l + \mathcal{B}_l D\lambda)}{(\mathcal{A}_l + \mathcal{B}_l D\lambda)(\mathcal{A}_r + \mathcal{B}_r D\lambda) e^{-\lambda X} - (\mathcal{A}_l - \mathcal{B}_l D\lambda)(\mathcal{A}_r - \mathcal{B}_r D\lambda) e^{\lambda X}} \\
& - \frac{[\mathcal{C}_l - (\mathcal{A}_l c_0 + \mathcal{B}_l D c_1)] (\mathcal{A}_r - \mathcal{B}_r D\lambda) e^{\lambda X}}{(\mathcal{A}_l + \mathcal{B}_l D\lambda)(\mathcal{A}_r + \mathcal{B}_r D\lambda) e^{-\lambda X} - (\mathcal{A}_l - \mathcal{B}_l D\lambda)(\mathcal{A}_r - \mathcal{B}_r D\lambda) e^{\lambda X}}.
\end{aligned} \tag{107}$$

Therefore, the analytic solution to the diffusion equation for the steady-state radiation energy density in a cold purely absorbing 1-D slab is given by Eq. 102 where the coefficients are described by Eq. 103, 106, and 107.

3.1 Source and Vacuum Boundaries with No External Sources

The simplest case to consider in solving the equations in Section 3 is a cold slab with no external sources, where a radiation temperature source is applied on one boundary and a vacuum condition on the other boundary. Under these conditions, Eq. 97 reduces to:

$$E(x) = \frac{1}{c}I^{+}(0)E_2(\sigma_a x) = \frac{4\pi^5}{15h^3c^3}T_0^4 E_2(\sigma_a x), \tag{108}$$

where T_0 is the radiation temperature applied at the left boundary. Similarly for diffusion, Eq. 102 can be reduced to:

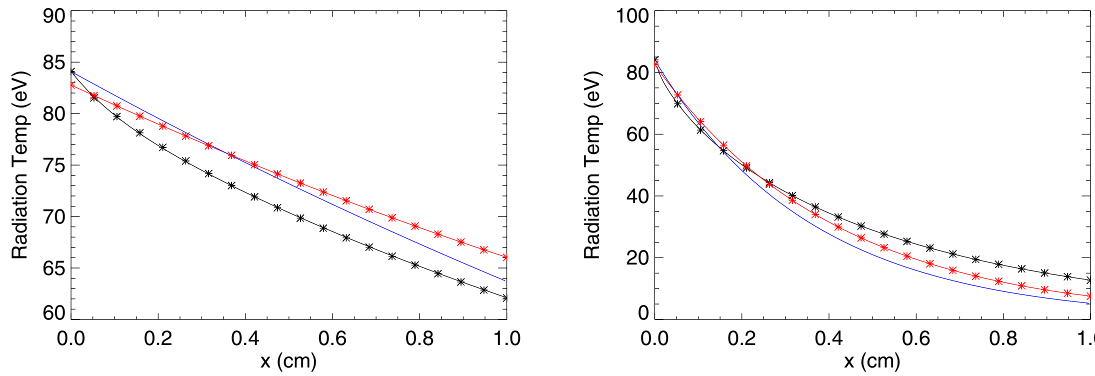
$$E(x) = \frac{4\pi^5}{15h^3c^3}T_0^4 \frac{\left[\left(\frac{1}{2} - D\lambda\right)e^{\lambda(x-X)} - \left(\frac{1}{2} + D\lambda\right)e^{\lambda(X-x)}\right]}{\left(D\lambda - \frac{1}{2}\right)^2 e^{-\lambda X} - \left(D\lambda + \frac{1}{2}\right)^2 e^{\lambda X}}. \tag{109}$$

For convenience in comparison to BUCKY output, the radiation energy density can then be converted to an effective radiation temperature, T_r, by:

$$T_r(x) = \left(\frac{c}{4}\frac{E(x)}{\sigma_{SB}}\right)^{\frac{1}{4}}, \tag{110}$$

where σ_{SB} is the Stephan-Boltzmann constant.

Figure 5 shows the solutions of Eq. 108 and Eq. 109 in comparison to that calculated by BUCKY for short-characteristics, diffusion, and flux-limited diffusion. The values for each of the variables in the equations are shown in Table 9. This comparison is done for two different opacities. In Figure 5(a), one mean free path is approximately 1.8 times the thickness of the slab. In this case, the distribution of radiation as calculated by the diffusion solution is significantly different than that calculated by the transport solution, and diffusion overpredicts the amount of radiation everywhere in the slab. This is not surprising since this problem violates most of the assumptions in the derivation of the



(a) (b)

Figure 5. Comparison between the analytic transport (Eq. 108) (black stars) and diffusion (Eq. 109) (red stars) solutions to those calculated by BUCKY for short-characteristics (black line), diffusion (red line), and flux-limited diffusion (blue line). All calculations are done assuming no external sources, source and vacuum conditions on the left and right boundaries respectively, and an absorption opacity of (a) $0.5558\ cm^{-1}$ and (b) $5.558\ cm^{-1}$.

diffusion equation. The average R value (as shown in Figure 1) throughout the slab is 2.17 (calculated from the transport solution). Under these conditions, one would expect the flux-limited diffusion solution to be a better approximation to the true transport characteristics (as evidenced by the figure).

In Figure 5(b), one mean free path is approximately 0.18 times the thickness of the slab. In this case, the diffusion approximation does a much better job of capturing the true radiation distribution. The average R value for this radiation field is 1.35, which is only a modest difference from that in case (a). Surprisingly, Figure 5(b) indicates that, for these conditions, the flux-limiter restricts the radiation too much, and actually looks less like the transport solution than pure diffusion. However, because this problem is calculated for a purely absorbing, nonradiating slab, it still violates the assumption in diffusion that requires the radiation field to be nearly isotropic. The primary points are that: the finite difference equations in BUCKY properly reproduce the analytic results, diffusion looks much more like transport when the optical depths are small compared to the size of the slab, and that flux-limited diffusion is not always better than pure diffusion.

	X	T_0	σ_a	σ_t
Value	$1.0cm$	$100.0eV$	$0.5558\ cm^{-1}$ or $5.558\ cm^{-1}$	σ_a

Table 9. Values used for each variable in comparing BUCKY short-characteristics and diffusion to the analytic equations.

3.2 Vacuum Boundaries With a Linear External Source

A slightly more complicated solution to the equations in Section 3 is to consider the case of a cold slab with vacuum boundaries, and a linearly dependent external source. If the source term has the form:

$$S(x) = S_0(1 - \frac{x}{a}), \tag{111}$$

then Eq. 97 reduces to:

$$E(x) = \frac{1}{2c\sigma_a} S_0 \left[2\left(1 - \frac{x}{X}\right) - E_2(\sigma_a x) + \frac{1}{\sigma_a X}\left[E_3(\sigma_a(X - x)) - E_3(\sigma_a x)\right]\right], \tag{112}$$

and Eq. 102 reduces to:

$$E(x) = ae^{\lambda x} + be^{-\lambda x} + \frac{4\pi}{c\sigma_a} S_0 \left(1 - \frac{x}{a}\right), \tag{113}$$

where the coefficients are given by:

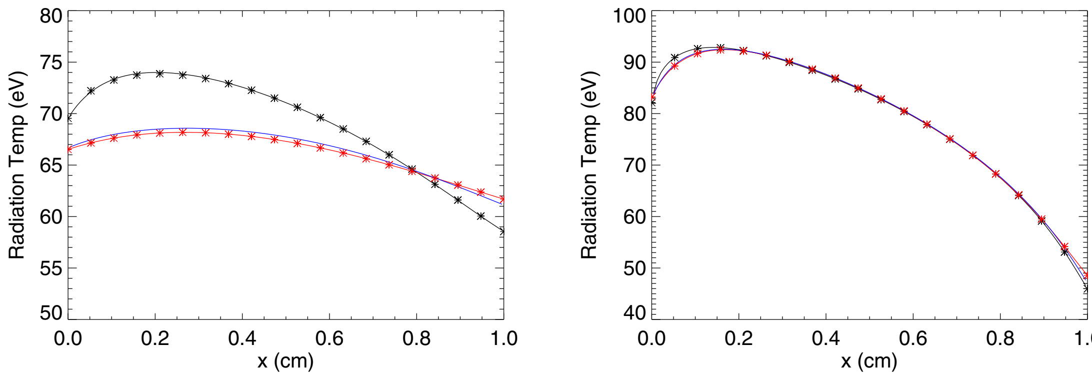
$$\begin{aligned} a &= -\frac{4\pi}{c\sigma_a} S_0 \frac{\left(\frac{D}{X} + \frac{1}{2}\right)\left(\frac{1}{2} - D\lambda\right)\right) e^{-\lambda X} + \frac{D}{X}\left(\frac{1}{2} + D\lambda\right)}{\left(D\lambda - \frac{1}{2}\right)^2 e^{-\lambda X} - \left(D\lambda + \frac{1}{2}\right)^2 e^{\lambda X}} \\ b &= \frac{4\pi}{c\sigma_a} S_0 \frac{\left(\frac{D}{X} + \frac{1}{2}\right)\left(\frac{1}{2} + D\lambda\right)\right) e^{\lambda X} + \frac{D}{X}\left(\frac{1}{2} - D\lambda\right)}{\left(D\lambda - \frac{1}{2}\right)^2 e^{-\lambda X} - \left(D\lambda + \frac{1}{2}\right)^2 e^{\lambda X}}. \end{aligned}$$

Assuming that the imposed external source function has a blackbody distribution, then S_0 in Eq. 111 can be described by:

$$S_0 = \sigma_x \frac{2\pi^4}{15h^3c^2} T_0^4, \tag{114}$$

where σ_x is an artificial emission opacity [3].

[3] In BUCKY, this artificial emission opacity is assigned as a zone dependent value of the form: $\sigma_x = \sigma_a\left(1 - \frac{x}{X}\right)$, so that the external source function is conveniently defined as in Eq. 111



(a) (b)

Figure 6. Comparison between the analytic transport (Eq. 112) (black stars) and diffusion (Eq. 113) (red stars) solutions to those calculated by BUCKY for short-characteristics (black line), diffusion (red line), and flux-limited diffusion (blue line). All calculations are done assuming a linear external source, vacuum conditions on both boundaries, and an absorption opacity of (a) $0.5558\ cm^{-1}$ and (b) $5.558\ cm^{-1}$.

The comparisons between BUCKY and the analytic results in Eq. 112 and Eq. 113 are shown in Figure 6 for the same set of values listed in Table 9. In each case, there is very good agreement between the BUCKY calculated results and the analytic solutions. In addition, Figure 6(b) shows that diffusion is a good approximation to true transport when one mean free path is much less than the total thickness of the slab. This is not surprising since the external source function is isotropic, and meets the primary criteria in the derivation of the diffusion equation (with the exception of the value near the boundary where radiation is allowed to escape). The agreement between diffusion and transport is not nearly as good in Figure 6(a) where one mean free path is 2.2 times the thickness of the slab. Even though the external source function is isotropic, the low opacity allows the radiation to stream to the boundaries resulting in a significant nonisotropic component to the radiation flow. It is also worth noting that, in each of these cases, the flux-limiter provides no significant benefit over pure diffusion. The calculated R values for Figure 6(a) and (b) are 2.3 and 0.1 respectively.

3.3 An External Source with a Source Boundary Condition

A more realistic case to consider in the comparison between transport and diffusion is that of a distributed external source function with a source boundary condition applied on one side. This may be thought of as a model for a sample that is being radiatively heated by a nearby source. To identify a realistic source function, Eq. 97 is iterated by initially assuming a cold material, and then fitting a polynomial to the resulting radiation distribution. This polynomial is then applied as the external source function for the next iteration, and the process is continued until a 'convergence' of the polynomial fit is achieved. The result is essentially modeling a sample that has come to equilibrium with the driving radiation source.

For the optically thin case ($\sigma_a = 0.5558\ cm^{-1}$), assuming a constant blackbody source on the left boundary at a temperature of $100\ eV$, the resulting external source function is represented by a 4^{th} order polynomial of the form:

$$S(x) = (12.617 - 8.4037x + 4.0473x^2 - 2.6986x^3 + 0.0047624x^4) * 1.e11 \qquad \frac{J}{cm^3\ s\ sr}. \tag{115}$$

Likewise for the optically thick case ($\sigma_a = 5.5558\ cm^{-1}$), the source function is represented by:

$$S(x) = (165.44 - 163.75x + 37.471x^2 - 22.651x^3 - 0.50728x^4) * 1.e11 \qquad \frac{J}{cm^3\ s\ sr}. \tag{116}$$

These polynomials are then interpolated onto the BUCKY finite difference grid (again using the artificial emission opacity to distribute the source), and calculated for short-characteristics, diffusion, and flux-limited diffusion. The results are shown in Figure 7.

The first thing to notice about this figure is that the diffusion solution looks very much like true transport. This is especially true in Figure 7(b) where one mean free path is much less than the thickness of the slab. This is a good illustration of why diffusion is such a popular way of computing the radiation transport. In plasmas driven by a steady state external radiation source, diffusion is a good approximation to true transport over a wide

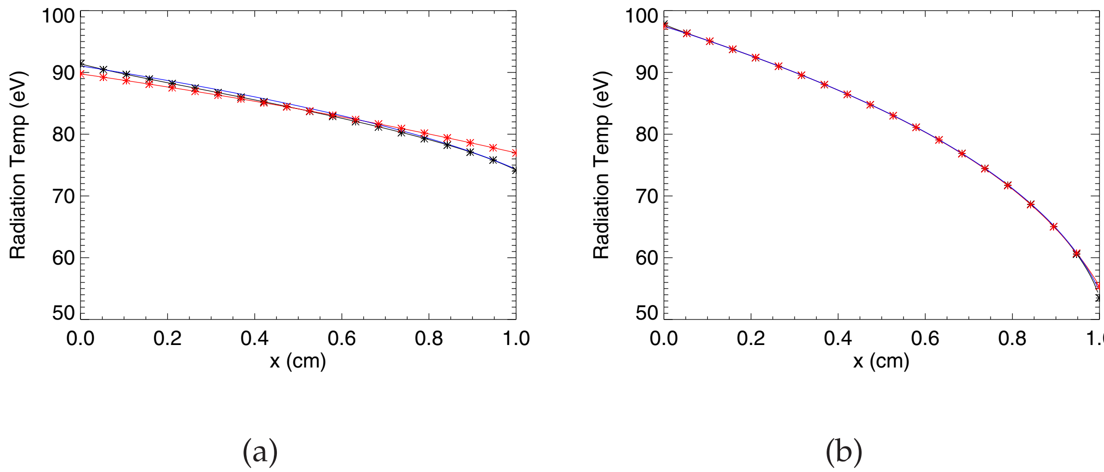


Figure 7. Comparison between the analytic transport (Eq. 112) (black stars) and diffusion (Eq. 113) (red stars) solutions to those calculated by BUCKY for short-characteristics (black line), diffusion (red line), and flux-limited diffusion (blue line). All calculations assume an external source function given by (a) Eq. 115 and (b) Eq. 116, source and vacuum conditions on the left and right boundaries respectively, and an absorption/emission opacity of (a) $0.5558\ cm^{-1}$ and (b) $5.558\ cm^{-1}$.

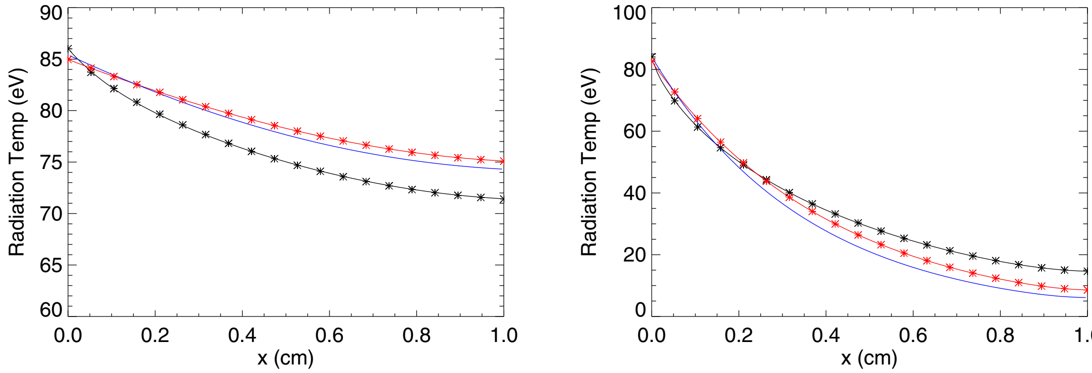
range of optical depths when the plasma temperature has enough time to equilibrate with the driving radiation source. The simple reason for this is that, at any particular point in the slab, the plasma is isotropically radiating at the same intensity as the anisotropic component of the radiation field that is contributed from the source applied at the boundary. Thus, the total radiation field has only a weakly anisotropic component, and therefore satisfies the primary assumptions in the derivation of the diffusion equation.

3.4 A Boundary Source and an Albedo Boundary Condition

One final problem that can be applied to both short-characteristics and diffusion is intended to test the implementation of the albedo boundary condition. In this simple problem, a cold slab with no external source term has a source condition applied on the left boundary and an albedo condition applied to the right boundary.

Under these circumstances, Eq. 97 reduces to:

$$E(x) = \frac{4\pi^5}{15h^3c^3} T_0^4 \left[E_2(\sigma_a x) + \alpha E_2(\sigma_a X) E_2(\sigma_a (X - x)) \right] \tag{117}$$



(a) (b)

Figure 8. Comparison between the analytic transport (Eq. 117) (black stars) and diffusion (Eq. 118) (red stars) solutions to those calculated by BUCKY for short-characteristics (black line), diffusion (red line), and flux-limited diffusion (blue line). All calculations are done assuming no external sources, source and albedo ($\alpha = 0.75$) conditions on the left and right boundaries respectively, and an absorption opacity of (a) $0.5558\ cm^{-1}$ and (b) $5.558\ cm^{-1}$.

where α is the albedo of the boundary at $x = X$. Similarly for diffusion, Eq. 102 can be reduced to:

$$E(x) = ae^{\lambda x} + be^{-\lambda x}, \tag{118}$$

for the coefficients a and b given by:

$$a = \frac{b\left(D\lambda + \frac{1}{2}\right) - \frac{4\pi^5}{15h^3c^3}T_0^4}{\left(D\lambda - \frac{1}{2}\right)}$$

$$b = \frac{4\pi^5}{15h^3c^3}T_0^4 \frac{\left(\frac{1}{2}\frac{\alpha-1}{\alpha+1} - D\lambda\right)e^{\lambda X}}{\left(D\lambda - \frac{1}{2}\right)\left(D\lambda + \frac{1}{2}\frac{\alpha-1}{\alpha+1}\right)e^{-\lambda X} - \left(D\lambda + \frac{1}{2}\right)\left(D\lambda - \frac{1}{2}\frac{\alpha-1}{\alpha+1}\right)e^{\lambda X}}.$$

The comparison between these equations and the BUCKY calculated result is shown in Figure 8 for the values in Table 9 and an albedo of $\alpha = 0.75$. The results look very much like those from Section 3.1 except that the radiation temperature is elevated due to the radiation energy that is reflected at the right boundary.

4 Solutions Specific to the Diffusion Equation

The problems in Section 3 nearly provide a complete benchmarking of the steady-state diffusion equations as implemented in BUCKY. However, because Eq. 26 contains a term that is dependent on the coordinate system, the diffusion equations in BUCKY must also be verified in cylindrical and spherical coordinates.

4.1 Steady-State Diffusion in Cylindrical Coordinates

Assuming that there are no external source functions, Eq. 101 can be rewritten in cylindrical coordinates as:

$$\rho\frac{\partial^2 E(\rho)}{\partial \rho^2} + \frac{\partial E(\rho)}{\partial \rho} = \rho\frac{\sigma_a}{D}E(\rho), \tag{119}$$

where ρ is the radial coordinate ($\rho = \sqrt{x^2+y^2}$). The solution to this equation is given by [11]:

$$E(\rho) = b'\,\mathrm{I}_0(\lambda\rho), \tag{120}$$

where I_0 is the modified Bessel function of the first kind. Plugging this into the general boundary condition in Eq. 19 at $\rho = \rho_{\text{max}}$ and solving for b' then gives:

$$E(\rho) = \frac{\mathcal{C}}{\mathcal{A}\,\mathrm{I}_0(\lambda\rho_{\text{max}}) - \mathcal{B}D\lambda\,\mathrm{I}_1(\lambda\rho_{\text{max}})}\,\mathrm{I}_0(\lambda\rho). \tag{121}$$

In the case of a source boundary condition applied at $\rho = \rho_{\text{max}}$, Eq. 121 can be written as:

$$E(\rho) = \frac{4\pi^5}{15h^3c^3}T_0^4\frac{\mathrm{I}_0(\lambda\rho)}{\frac{1}{2}\,\mathrm{I}_0(\lambda\rho_{\text{max}}) + D\lambda\,\mathrm{I}_1(\lambda\rho_{\text{max}})}. \tag{122}$$

Figure 9 shows the comparison between this analytic result and BUCKY calculated diffusion for a boundary temperature of $T_0 = 100\ eV$ applied at a maximum radius of $\rho_{\text{max}} = 0.5643\,cm$. The material is assumed to both absorb and scatter radiation with opacities of $\sigma_a = 0.5558\ cm^{-1}$ and $\sigma_s = 5.558\ cm^{-1}$ respectively ($\sigma_t = \sigma_a + \sigma_s$). As evidenced by the figure, the BUCKY calculated result compares well with the analytic solution.

4.2 Steady-State Diffusion in Spherical Coordinates

Again assuming that there are no external source functions, Eq. 101 can be rewritten in spherical coordinates as:

$$r^2\frac{\partial^2 E(r)}{\partial r^2} + 2r\frac{\partial E(r)}{\partial r} = r^2\frac{\sigma_a}{D}E(r), \tag{123}$$

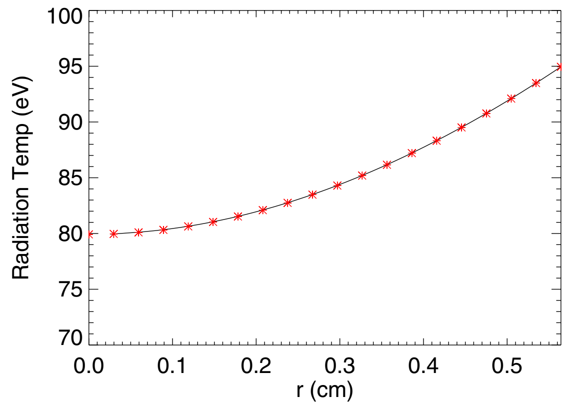


Figure 9. Comparison between BUCKY calculated diffusion (solid line) and the steady state analytic result for cylindrical coordinates in Eq. 122 (red stars) where the absorption and scattering opacities are given by $\sigma_a = 0.5558\ cm^{-1}$ and $\sigma_s = 5.558\ cm^{-1}$ respectively.

where r is the radial coordinate ($r = \sqrt{x^2 + y^2 + z^2}$). The solution to this equation is given by [11]:

$$E(\rho) = b' \frac{\sinh(\lambda r)}{r} \ . \tag{124}$$

Plugging this into the general boundary condition in Eq. 19 at $r = r_{\max}$ and solving for b' then gives:

$$E(r) = \left[\frac{\mathcal{C} r_{\max}}{\mathcal{A} \sinh(\lambda r_{\max}) - \mathcal{B} D \left[\lambda \cosh(\lambda r_{\max}) - \frac{\sinh(\lambda r_{\max})}{r_{\max}} \right]} \right] \frac{\sinh(\lambda r)}{r}. \tag{125}$$

In the case of a source boundary condition applied at $r = r_{\max}$, Eq. 125 can be written as:

$$E(\rho) = \frac{4\pi^5}{15 h^3 c^3} T_0^4 \left[\frac{r_{\max}}{\frac{1}{2} \sinh(\lambda r_{\max}) D \left[\lambda \cosh(\lambda r_{\max}) - \frac{\sinh(\lambda r_{\max})}{r_{\max}} \right]} \right] \frac{\sinh(\lambda r)}{r}. \tag{126}$$

Figure 10 shows the comparison between this analytic result and BUCKY calculated diffusion for a boundary temperature of $T_0 = 100\ eV$ applied at a maximum radius of $r_{\max} = 0.6204\ cm$. The material is assumed to both absorb and scatter radiation with opacities of $\sigma_a = 0.5558\ cm^{-1}$ and $\sigma_s = 5.558\ cm^{-1}$ respectively ($\sigma_t = \sigma_a + \sigma_s$). As evidenced by the figure, the BUCKY calculated result again compares well with the analytic solution.

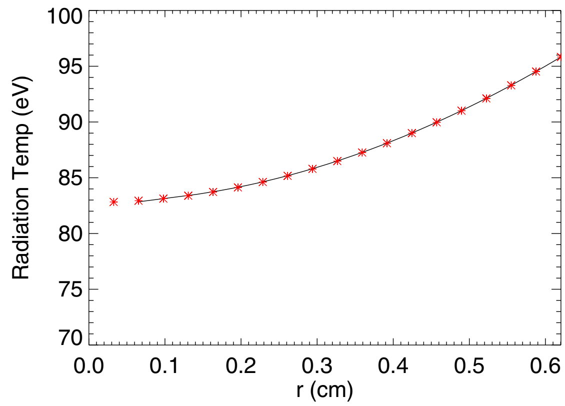


Figure 10. Comparison between BUCKY calculated diffusion (solid line) and the steady state analytic result for spherical coordinates in Eq. 126 (red stars) where the absorption and scattering opacities are given by $\sigma_a = 0.5558\ cm^{-1}$ and $\sigma_s = 5.558\ cm^{-1}$ respectively.

4.3 Flux-Limiters

As was demonstrated in Section 3, the flux-limiter in the diffusion coefficient can significantly alter the radiation profile calculated by diffusion. Thus, it is important to benchmark the implementation of each flux-limiter. However, because the flux-limiter makes the diffusion equation nonlinear, this is somewhat difficult to accomplish by attempting a direct analytic solution to the flux-limited diffusion equation(s). Instead, one can manufacture a solution for the radiation energy density distribution, and then plug the solution into Eq. 101 to determine the external source function that will produce that radiation distribution. This initial source function can then be input into BUCKY as an initial condition, and the resulting radiation energy density checked to verify the reproduction of the manufactured solution.

Assuming for simplicity that the test slab is cold, the flux-limited diffusion equation can be written as:

$$\frac{\partial}{\partial x} D(x) \frac{\partial E(x)}{\partial x} = \sigma_a E(x) - \frac{4\pi}{c} S(x). \tag{127}$$

Furthermore, assuming that the solution for some source function $S(x)$ has a linear distribution of the form:

$$E(x) = ax + b, \tag{128}$$

the coefficients a and b are dictated by the conditions of the radiation field at the boundaries. Thus, plugging Eq. 128 into Eq. 19 assuming Dirichlet conditions at the left and right boundary gives:

$$\begin{aligned} b &= \frac{4\pi}{c} B_\nu(T_L) \\ a &= \frac{1}{X}\frac{4\pi}{c}[B_\nu(T_R) - B_\nu(T_L)], \end{aligned} \tag{129}$$

where T_L and T_R are the radiation temperatures at the left and right boundaries respectively, and X is the total thickness of the slab.

Solving Eq. 127 for each of the flux-limiters in Section 2.1 gives different source functions for:

the SUM Limiter:

$$S(x) = \left[\sigma_a(ax+b) - \frac{|a|a^2}{\left(3\sigma_t(ax+b) + |a|\right)^2}\right]\frac{c}{4\pi}, \tag{130}$$

the MAX Limiter:

$$S(x) = \begin{array}{ll} \sigma_a(ax+b)\frac{c}{4\pi}, & 3\sigma_t > \frac{|a|}{ax+b} \\ \left[\sigma_a(ax+b) - \frac{a^2}{|a|}\right]\frac{c}{4\pi}, & 3\sigma_t < \frac{|a|}{ax+b} \end{array} \tag{131}$$

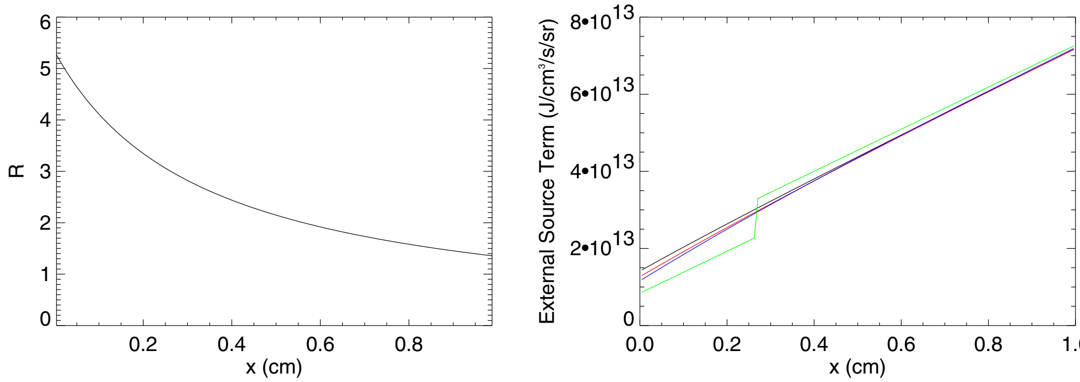
the Larsen Limiter:

$$S(x) = \left[\sigma_a(ax+b) - \frac{|a|^n a^2}{\left[(3\sigma_t)^n + \left(\frac{|a|}{ax+b}\right)^n\right]^{\frac{1}{n}+1}(ax+b)^{n+1}}\right]\frac{c}{4\pi}, \tag{132}$$

and the Simplified Levermore-Pomraning Limiter:

$$S(x) = \left[\sigma_a(ax+b) - \frac{a^4\left(a + 4\sigma_t(ax+b)\right)}{\left(a^2 + 3\sigma_t a(ax+b) + 6\sigma_t^2(ax+b)^2\right)^2}\right]\frac{c}{4\pi}. \tag{133}$$

Figure 11(a) shows the range of R values throughout a $1.0\ cm$ thick slab for a total opacity of $0.5558\ cm^{-1}$, and a fixed temperature on the left and right boundaries of $100\ eV$ and $141.42\ eV$ respectively. According to Figure 1, this range is within the region where all



(a) (b)

Figure 11. (a)Calculated R values for an assumed energy density of $E = B_\nu(100\ eV) + B_\nu(131.61\ eV)x$ and a total opacity of $0.5558\ cm^{-1}$. (b)External source functions for the SUM-limiter (black), the Levermore-Pomraning-limiter (red), the Larsen-limiter (blue), and the MAX-limiter (green).

the flux-limiters have a significant influence on the diffusion coefficient, and is therefore an acceptable place to test the implementation of the various flux-limiters. Figure 11(b) shows the plots of the calculated source functions for each of these limiters. In addition to the parameters listed above, these calculations assume an absorption opacity of $5.558\ cm^{-1}$ (a factor of 10 higher than the total opacity) in order to keep the source function positive.

The relative errors between the BUCKY solutions and the linear radiation energy density in Eq. 128 are shown in Figures 12(a) and (b). In each case, the BUCKY calculated solution is taken after 100 cycles. The solutions using the SUM-, Larsen-, and Levermore-Pomraning-limiters as shown in 12(a) all agree to better than 0.04%. However, the MAX-limiter shown in 12(b) has maximum errors up to 1%. This is an artifact of the discontinuity that exists in the form of the MAX-limited diffusion coefficient, and is a good reason to avoid this form of the flux-limiter.

While BUCKY reproduces the expected solutions rather well, these cases only test the implementation of the numerics in the interior of the slab. Because the test cases

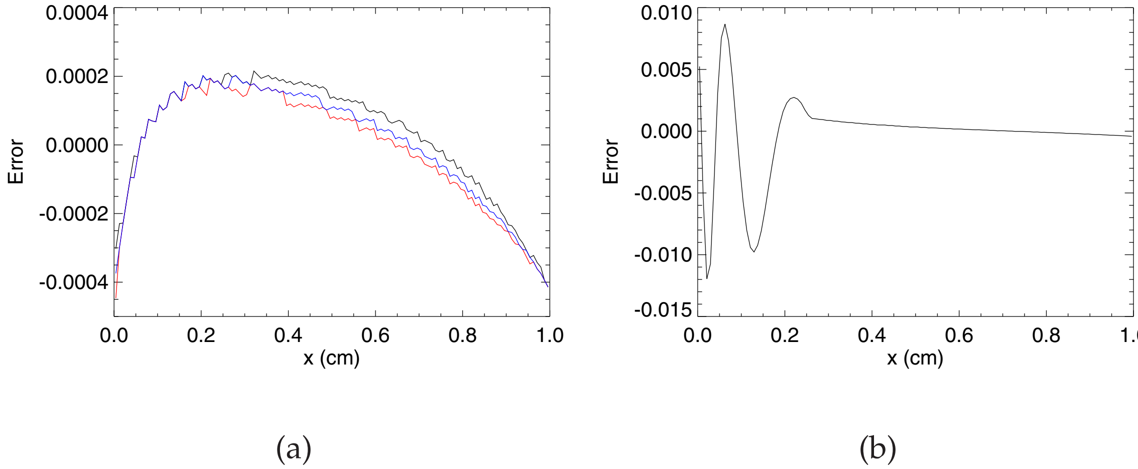


Figure 12. Relative errors between the assumed radiation energy density in Eq. 128 and that calculated by BUCKY for the (a) SUM (black), Larsen n=2 (red), and Levermore-Pomraning-limiters (blue), and (b) that calculated for the MAX-limiter.

assumed Dirichlet conditions on each boundary, the value of the radiation energy density on the left and right boundaries are well fixed and therefore not very demanding on the numerics. A more realistic case to consider would be that of a source condition on the left boundary of the slab and a vacuum condition on the right boundary. Solving Eq. 19 for these boundary conditions then gives the values of the coefficients in Eq. 128 as:

$$\begin{aligned} b &= \frac{4\pi}{c} B_\nu(T_L) \frac{1 + 3\sigma_T X}{3(1 + \sigma_T X)} \\ a &= -b/X. \end{aligned} \tag{134}$$

The comparison between the BUCKY calculated results (using the SUM-limiter) and the assumed form of the radiation energy density (using the coefficients in Eq. 134) is shown in Figure 13 (where $\sigma_a = \sigma_T = 0.5558\ cm^{-1}$ and $T_L = 100\ eV$). In this case, the BUCKY calculation never settles on a single solution, but rather oscillates between 10 different distributions (5 of which are shown in the figure). This numerical periodicity occurs because the gradients at each edge are calculated based on the result of the calculation from the previous cycle. However, the boundary value of the energy density is calculated based on the gradient at each edge on the current cycle. Because the diffusion

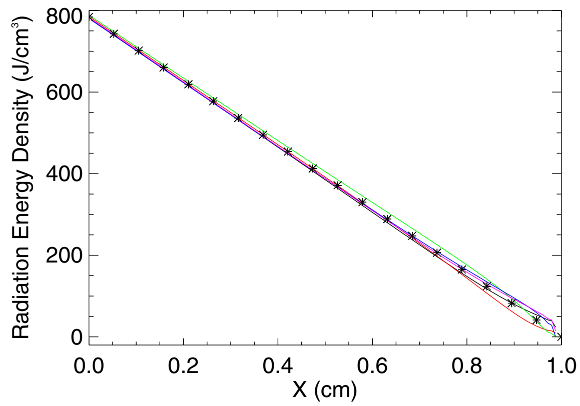


Figure 13. Comparison between the BUCKY calculated radiation energy density for the SUM-limiter (solid lines) and the assumed value using the coefficients in Eq. 134 (stars). The calculations assume a source condition on the left boundary and a vacuum condition on the right boundary.

equation is elliptical, the values at each boundary affect the values throughout the entire sample, thereby changing the gradient at each boundary (and thus the calculation of the boundary value). This leads to a periodic solution which, in this case, has a period of 10. According to Figure 13, these solutions oscillate around the assumed solution, but have relative errors up to 35%. This is a problem inherent with the first order implementation of the finite-differencing scheme, and may only be fixed by using a higher order (nonlinear) solver.

4.4 Time Dependent Solutions

In addition to (all) the analytic steady state solutions presented up to this point, the implementation of the time-dependency of the diffusion equation in BUCKY also requires verification. Because of the difficulty in solving Eq. 9 for real geometries, these solutions are all calculated for an infinite medium (thereby permitting the application of Fourier transform methods).

4.4.1 Planar Geometry

In the case of an infinitesimally thin, steady state planar source in a medium with constant opacities, the diffusion equation can be written as:

$$\frac{\partial E(x,t)}{\partial t} - \frac{c}{3\sigma_t}\frac{\partial^2 E(x,t)}{\partial x^2} = -\sigma_a E(x,t), \tag{135}$$

under the initial condition:

$$E(x,t_0) = E_0\delta(x - x_0), \tag{136}$$

where x_0 is the position of the radiation source, t_0 is the time when the source is turned on (and off), and E_0 is the total energy (in $\frac{J}{cm^2}$). Eq. 135 can be solved in $E(k,s)$ space by taking the Laplace $(t \to s)$ and Fourier $(x \to k)$ transforms to give [11]:

$$E(k,s) = \frac{E_0}{\sqrt{2\pi}\left(s + \frac{ck^2}{3\sigma_t} + c\sigma_a\right)}. \tag{137}$$

The inverse transforms then yield the analytic, time-dependent radiation energy density distribution as:

$$E(x,t) = E_0\sqrt{\frac{3\sigma_t}{4\pi ct}}e^{-c\sigma_a(t-t_0)}e^{-\frac{3\sigma_t(x-x_0)^2}{4ct}}. \tag{138}$$

Figure 14 shows the comparison between the BUCKY calculated results and the solution to Eq. 138 at times of $1\ ps$, $10\ ps$, $20\ ps$, and $30\ ps$. Each calculation assumes $\sigma_a = \sigma_t = 5.558\ cm^{-1}$, $t_0 = 0\ s$, and $E_0 = 13751.9\ \frac{J}{cm^2}$. In BUCKY, the source is seeded with the analytical distribution at a time of $1\ ps$, and is centered on an initial source position of $x_0 = 50.0\ cm$. The source input is done this way to avoid complications associated with trying to model a delta function in time and space as a finite value in BUCKY. According to Figure 14, the BUCKY calculation compares well to the analytic results at each time.

4.4.2 Spherical Geometry

Following the analysis by Brunner [11], the solution to the infinite planar source solution in Eq. 138 can be transformed to a point source solution by:

$$E_{\text{point}}(r,t) = -\frac{1}{2\pi r}\frac{\partial E_{\text{plane}}}{\partial x}|_{x=r}. \tag{139}$$

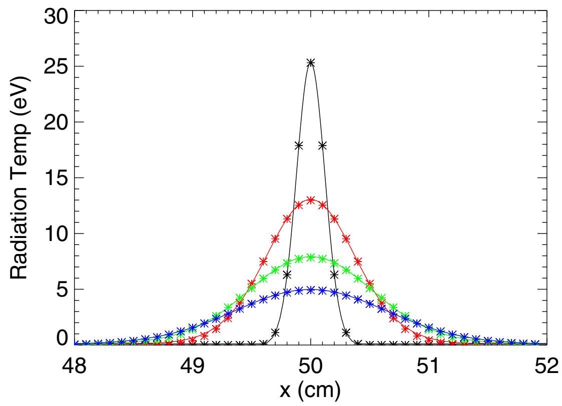


Figure 14. Time-dependent radiation temperature in planar geometry as calculated by BUCKY (solid lines) and the analytic result in Eq. 138 (stars) at times of $1\ ps$ (black), $10\ ps$ (red), $20\ ps$ (green), and $30\ ps$ (blue).

The resulting equation for the time-dependent radiation energy density in spherical geometry can be written as:

$$E(r,t) = E_0 \left[\frac{3\sigma_t}{4\pi ct}\right]^{3/2} e^{-c\sigma_a(t-t_0)} e^{-\frac{3\sigma_t r^2}{4ct}}, \tag{140}$$

where E_0 is now given in units of J, and it has been assumed that the initial source location is $r_0 = 0$.

Figure 15 shows the comparison between the BUCKY calculated results and the solution to Eq. 140 at times of $1\ ps$, $10\ ps$, $20\ ps$, and $30\ ps$. The BUCKY calculation is again seeded with the analytical energy density at a time of $1\ ps$, for a total initial energy of $E_0 = 13751.9\ J$.

4.4.3 Cylindrical Geometry

Finally, verifying the time-dependence in cylindrical geometry simply requires defining an infinite line source. This can be done by integrating Eq. 140 over the line as:

$$E_{\text{line}}(\rho,t) = \int_{-\infty}^{\infty} E_{\text{point}}(\sqrt{\rho^2+z^2},t)dz. \tag{141}$$

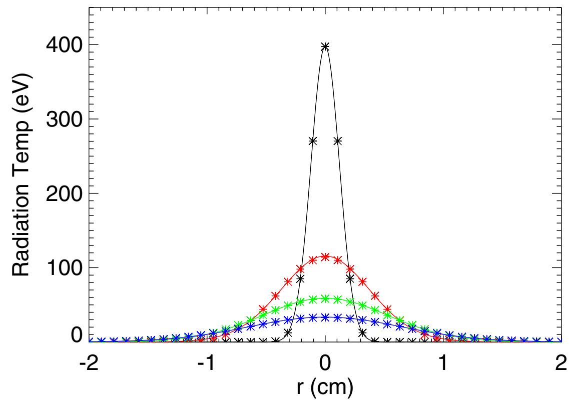


Figure 15. Time-dependent radiation temperature in spherical geometry as calculated by BUCKY (solid lines) and the analytic result in Eq. 140 (stars) at times of $1\ ps$ (black), $10\ ps$ (red), $20\ ps$ (green), and $30\ ps$ (blue).

Then, the solution for the case of an infinite line source in cylindrical geometry can be written as:

$$E(\rho,t) = E_0 \frac{3\sigma_t}{4\pi ct} e^{-c\sigma_a(t-t_0)} e^{-\frac{3\sigma_t \rho^2}{4ct}}, \tag{142}$$

for E_0 the total initial energy now given units of $\frac{J}{cm}$.

Figure 16 shows the comparison between the BUCKY calculated results and the solution to Eq. 142 at times of $1\ ps$, $10\ ps$, $20\ ps$, and $30\ ps$. The BUCKY calculation is again seeded with the analytical energy density at a time of $1\ ps$, for a total initial energy of $E_0 = 13751.9\ \frac{J}{cm}$. As in each of the comparisons above, the solutions calculated in BUCKY compare very well with the analytic results.

5 Radiatively Heated Plasmas

One final class of problems which requires proper verification is the case of a radiatively heated plasma. Any transport code which is intended to model the radiation conditions inside a plasma with real temperature dependent material properties must include an energy conservation equation which couples the plasma conditions to the radiation field. This coupling is accomplished through the radiative emission and absorption terms that

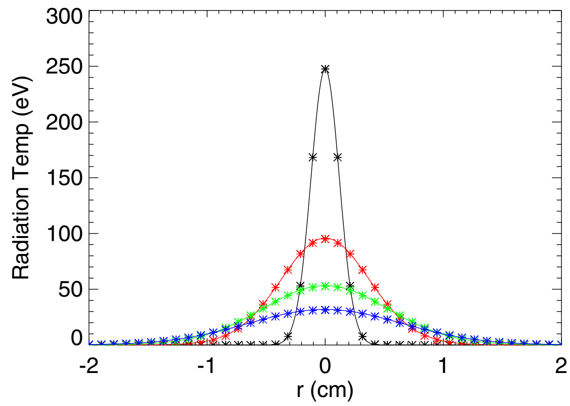


Figure 16. Time-dependent radiation temperature in cylindrical geometry as calculated by BUCKY (solid lines) and the analytic result in Eq. 142 (stars) at times of $1\ ps$ (black), $10\ ps$ (red), $20\ ps$ (green), and $30\ ps$ (blue).

appear in each equation. In BUCKY, the conservation of energy is expressed as a temperature diffusion equation, and is written in Lagrangian coordinates as [7]:

$$c_v \frac{\partial T_p}{\partial t} = \frac{\partial}{\partial m}\left(r^{\delta-1}\kappa_p \frac{\partial T_p}{\partial r}\right) - \left[\frac{\partial E_p}{\partial V} + P\right]\frac{\partial V}{\partial t} T_p + A - J + S_p, \tag{143}$$

where T_p is the plasma temperature, c_v is the heat capacity, κ_p is the plasma thermal conductivity, P is the plasma pressure, S_p is an external source term, and A and J are the radiation absorption and emission terms respectively. As in the radiation transport equation, the radiation absorption and emission are given by:

$$A = c\sigma_{P,a} E \tag{144}$$

$$J = 4\pi\sigma_{P,e}\int_0^\infty B_\nu(T_p)\,d\nu. \tag{145}$$

In order to calculate an analytic solution to the coupled set of equations (between the radiation transport and the energy conservation equations), Eq. 143 is typically simplified by assuming that thermal conductivity is negligible ($\kappa_p = 0$), the plasma is static ($\partial V/\partial t = 0$), and there are no external sources ($S_p = 0$). Then, if it is assumed that the heat capacity is proportional to the plasma temperature to the third power by:

$$c_v = \alpha T^3, \tag{146}$$

then Eq. 143 can be written as:

$$\frac{\partial T_p(r,t)}{\partial t} = \frac{1}{\alpha T^3}\left(c\sigma_{P,a}E(r,t) - \sigma_{P,e}\sigma_{SB}T_p^4\right), \tag{147}$$

where σ_{SB} is the Stephan-Boltzmann constant ($= 1.02825 * 10^5 J/cm^2/s/eV^4$), and $E(r,t)$ is the radiation energy density given by either Eq. 9 or Eq. 76. These equations are solved by Su and Olson for both a Marshak wave in a semi-infinite slab [12] and a time-dependent finite source in an infinite slab [13].

5.1 The Marshak Wave Problem

The Marshak wave problem is a classic benchmark for radiation transport codes. The premise is very simple: An isotropic radiation source condition is placed on the boundary of an initially cold, semi-infinite slab. The radiation from the boundary source penetrates and heats the material, which itself radiates isotropically at the local plasma temperature. The result is two well-defined, propagating wavefronts corresponding to the penetrating radiation and thermal energy. These wavefronts eventually coalesce, and the total energy wave propagates deep into the plasma.

This problem has been solved analytically in the single group radiation diffusion approximation by Su and Olson [12]. Their solution is expressed as a function of 4 dimensionless variables given in the nomenclature of this document as:

$$x = \sqrt{3}\sigma' r \tag{148}$$

$$\tau = \left(\frac{16\sigma_{SB}\sigma'}{\alpha}\right)t \tag{149}$$

$$u(x,\tau) = \left(\frac{c}{4}\right)\left[\frac{E(r,t)}{\sigma_{SB}T_0^4}\right] \tag{150}$$

$$v(x,\tau) = \left[\frac{T_p}{T_0}\right]^4, \tag{151}$$

where $\sigma'/\rho = \sigma_R = \sigma_{P,e} = \sigma_{P,a}$, T_0 is the radiation temperature of the boundary source, and $E(r,t)$ is given by Eq. 9. After a great deal of mathematics, their solutions for the dependent variables, u and v, are expressed as:

$$u(x,\tau) = 1 - \frac{2\sqrt{3}}{\pi}\int_0^1 e^{-\tau\eta^2}\left[\frac{\sin[x\gamma_1(\eta)+\theta_1(\eta)]}{\eta\sqrt{3+4\gamma_1^2(\eta)}}\right]d\eta$$
$$- \frac{\sqrt{3}}{\pi}e^{-\tau}\int_0^1 e^{-\tau/\epsilon\eta}\sqrt{\epsilon+\frac{1}{1-\eta^2}}\left[\frac{\sin[x\gamma_2(\eta)+\theta_2(\eta)]}{\eta(1+\epsilon\eta)\sqrt{3+4\gamma_2^2(\eta)}}\right]d\eta \tag{152}$$

$$v(x,\tau) = \int_0^\tau e^{-(\tau-\tau')}u(x,\tau')d\tau', \tag{153}$$

where

$$\gamma_1(\eta) = \eta\sqrt{\epsilon+\frac{1}{1-\eta^2}} \tag{154}$$

$$\gamma_2(\eta) = \sqrt{(1-\eta)\left(\epsilon+\frac{1}{\eta}\right)} \tag{155}$$

$$\theta_n(\eta) = \cos^{-1}\sqrt{\frac{3}{3+4\gamma_n^2(\eta)}}, n = 1, 2 \tag{156}$$

for the transport parameter $\epsilon = 16\sigma_{SB}/\alpha c$. These integrals must then be solved numerically for some value of ϵ. Table 10 lists the calculated values of Eq. 152 and 153 for $\epsilon = 0.1,\ (\alpha = 160\sigma_{SB}/c)$.

Figure 17(a) and (b) show the comparison between the analytic calculations and the conditions simulated by BUCKY. As evidenced by the figure, the BUCKY calculations compare very well to the analytic results at the plotted times of $\tau = 0.1, 1$, and 10.

5.2 Non-Equilibrium Transport in an Infinite Medium

Su and Olson have defined a second benchmark problem for non-equilibrium radiative transfer where they have constructed analytic solutions for both radiation diffusion and true transport [13]. In this problem, a finite radiation source in a region $-x_0 \leq x \leq x_0$ is active for a time $0 \leq \tau \leq \tau_0$ within an infinite slab. The solutions are expressed in the same dimensionless variables given in Eq. 148-151 except that T_0 is now the temperature of the isotropic blackbody source. These problems are rather complex, and the reader is directed to the reference for the derivation and final expression of the solutions. Table 11

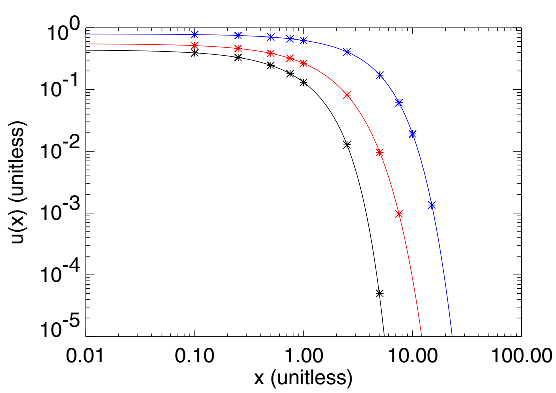
x	τ=0.01	0.1	1	10
0	0.23997	0.43876	055182	0.79720
0.1	0.17979	0.39240	0.51412	0.77644
0.25	0.11006	0.33075	0.46198	0.75004
0.5	0.04104	0.24629	0.38541	0.70679
0.75	0.01214	0.18087	0.32046	0.66458
1	0.00268	0.13089	0.26564	0.62353
2.5		0.01274	0.08147	0.40703
5			0.00961	0.17142
7.5			0.00097	0.06123
10				0.01909
15				0.00135

$u(x, \tau)$

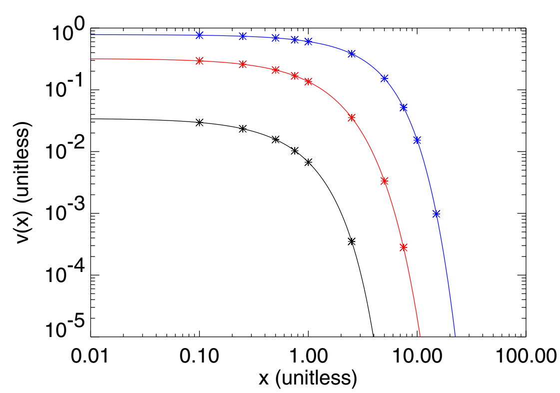
x	τ=0.01	0.1	1	10
0	0.00170	0.03446	0.32030	0.78318
0.1	0.00110	0.02955	0.29429	0.76448
0.25	0.00055	0.02339	0.25915	0.73676
0.5	0.00012	0.01566	0.20925	0.69139
0.75		0.01030	0.16862	0.64730
1		0.00672	0.13563	0.60461
2.5		0.00035	0.03539	0.38320
5			0.00334	0.15285
7.5			0.00028	0.05166
10				0.01527
15				0.00098

$v(x, \tau)$

Table 10. Analytic solutions to the Su and Olson Marshak wave problem for $\epsilon = 0.1$ [12]



(a)



(b)

Figure 17. Comparison between a BUCKY simulation (lines) and the analytic calculations (stars) of the scaled (a) radiation energy, u, and (b) plasma energy, v, for the Su and Olson Marshak wave problem [12] at times of $\tau = 0.1$ (black), 1 (red), and 10 (blue).

and 12 list the analytic evaluations of the diffusion and transport solutions respectively for a source with $\epsilon = 1$, $\tau_0 = 10$, and $x_0 = 0.5$.

The comparison between the analytic solutions and those calculated by BUCKY for the radiation diffusion case are shown in Figure 18(a) and (b). As evidenced by the figure, BUCKY compares well at each time. Because Su and Olson have also provided solutions for the case of true radiation transport, this problem provides a unique opportunity to investigate the accuracy of flux-limited diffusion. Figure 19(a) shows the comparison between the analytic solutions for the transport case, and those calculated by flux-limited diffusion (Levermore-Pomraning limiter). Figure 19(b) shows the comparison between the analytic solution at a time of $\tau = 1.0$, and the flux-limited diffusion solution for each flux-limiter. Clearly, flux-limited diffusion does a decent job of approximating the analytic result. Also plotted in Figure 19(b) is the short-characteristics solution to this problem at a time of $\tau = 1.0$. Under these circumstances, the short-characteristics solution transports radiation far too quickly. This is not surprising since the Su and Olson benchmark is intentionally a time-dependent problem, and the implementation of short-characteristics in BUCKY is time-independent. However, this serves as a reminder that, although short-characteristics is a much better approximation to true transport in problems with slowly varying radiation fields, there are some instances when flux-limited diffusion will provide a more accurate result.

x	τ=0.1	1	10	100
0.01000	0.09403	0.50359	1.86585	0.35365
0.10000	0.09326	0.49716	1.85424	0.35360
0.31623	0.08230	0.43743	1.74866	0.35309
0.50000	0.04766	0.33271	1.57237	0.35225
0.75000	0.00755	0.18879	1.29758	0.35051
1.00000	0.00064	0.10150	1.06011	0.34809
1.33352		0.04060	0.79696	0.34382
1.77828		0.01011	0.52980	0.33636
3.16228		0.00003	0.12187	0.30185
5.62341			0.00445	0.21453
10.0000				0.07351

$u(x, \tau)$

x	τ=0.1	1	10	100
0.01000	0.00466	0.21859	1.75359	0.35554
0.10000	0.00464	0.21565	1.74218	0.35548
0.31623	0.00424	0.18765	1.63837	0.35497
0.50000	0.00234	0.13590	1.46494	0.35411
0.75000	0.00023	0.06746	1.19584	0.35235
1.00000		0.03173	0.96571	0.34988
1.33352		0.01063	0.71412	0.34555
1.77828		0.00210	0.46369	0.33797
3.16228			0.09834	0.30294
5.62341			0.00306	0.21452
10.0000				0.07269

$v(x, \tau)$

Table 11. Analytic radiation diffusion solutions to the Su and Olson non-equilibrium transport problem in an infinite medium for $\epsilon = 1, \tau_0 = 10,$ and $x_0 = 0.5$ [13].

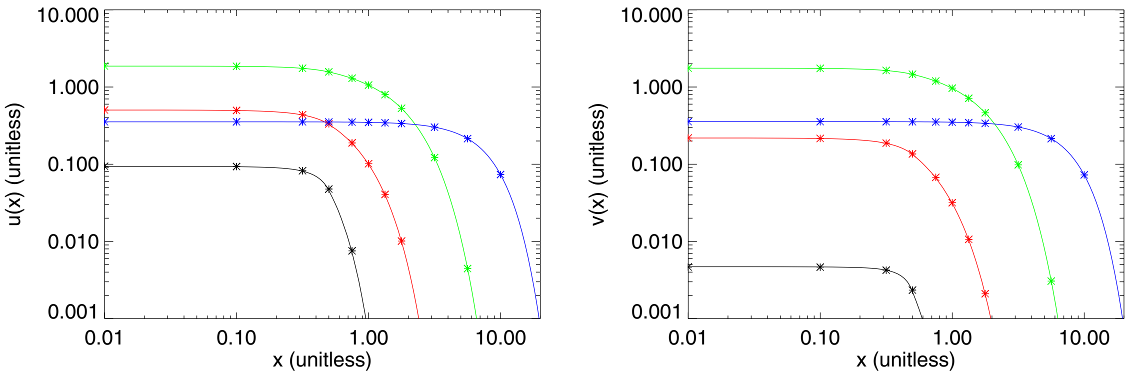
x	τ=0.1	1	10	100
0.01000	0.09531	0.64308	2.23575	0.35720
0.10000	0.09531	0.63585	2.21944	0.35714
0.31623	0.09529	0.56187	2.06448	0.35664
0.50000	0.04765	0.35801	1.73178	0.35574
0.75000		0.11430	1.27398	0.35393
1.00000		0.03648	0.98782	0.35141
1.33352		0.00291	0.70822	0.34697
1.77828			0.45016	0.33924
3.16228			0.09673	0.30346
5.62341			0.00375	0.21382
10.0000				0.07200

$u(x, \tau)$

x	τ=0.1	1	10	100
0.01000	0.00468	0.27126	2.11186	0.35914
0.10000	0.00468	0.26839	2.09585	0.35908
0.31623	0.00468	0.23978	1.94365	0.35854
0.50000	0.00234	0.14187	1.61536	0.35766
0.75000		0.03014	1.16591	0.35581
1.00000		0.00625	0.88992	0.35326
1.33352		0.00017	0.62521	0.34875
1.77828			0.38688	0.34086
3.16228			0.07642	0.30517
5.62341			0.00253	0.21377
10.0000				0.07122

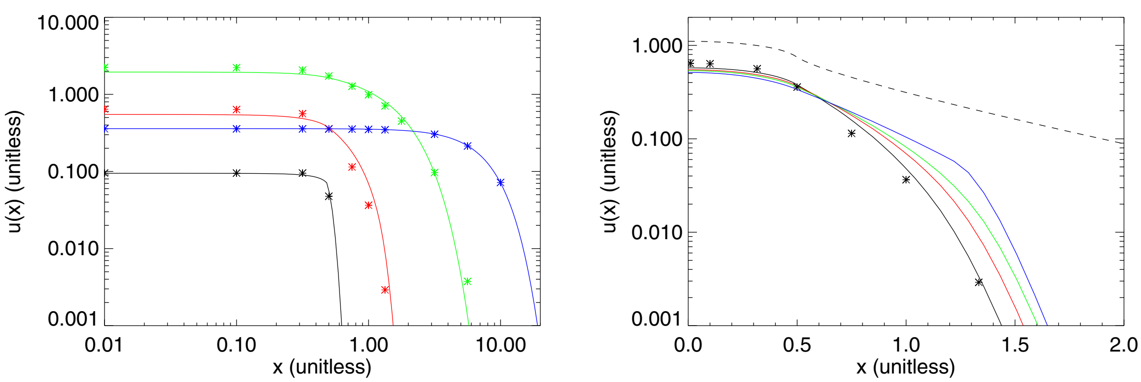
$v(x, \tau)$

Table 12. Analytic radiation transport solutions to the Su and Olson non-equilibrium transport problem in an infinite medium for $\epsilon = 1, \tau_0 = 10,$ and $x_0 = 0.5$.



(a) (b)

Figure 18. Comparison between a BUCKY simulation (lines) and the analytic calculations (stars) of the scaled (a) radiation energy, u, and (b) plasma energy, v, for the diffusion solution to the Su and Olson non-equilibrium transport problem at times of $\tau = 0.1$ (black), 1 (red), 10 (green), and 100 (blue).



(a) (b)

Figure 19. (a) Comparison between a BUCKY simulation using the Levermore-Pomraning version of flux-limited diffusion (lines) and the analytic calculations (stars) of the scaled radiation energy, u, for the transport solution to the Su and Olson non-equilibrium transport problem at times of $\tau = 0.1$ (black), 1 (red), 10 (green), and 100 (blue). (b) Comparison between the analytic calculation (stars) and a BUCKY simulation using the (a) SUM-limiter, (b) Levermore-Pomraning-limiter, (c) Larsen-limiter (n=2), and (d) MAX-limiter at a time of $\tau = 1$. Also shown is a BUCKY calculation using short-characteristics (dashed).

References

[1] D. Mihalas, *Stellar Atmospheres* (W.H. Freeman and Co., San Francisco, 1978).

[2] E.E. Lewis and W.F. Miller Jr., *Computational Methods of Neutron Transport* (American Nuclear Society, La Grange Park, 1993).

[3] G.L. Olson, L.H. Auer, and M.L. Hall, "Diffusion, P1, and other approximate forms of radiation transport," J. Quant. Spectrosc. Radiat. Transfer **64**, p. 619 (2000).

[4] C.D. Levermore and G.C. Pomraning, "A flux-limited diffusion theory," Astrophys. J., **248**, p. 321 (1981).

[5] T.A. Brunner, "Forms of approximate radiation transport," Sandia National Laboratory report, SAND2002-3739P.

[6] G.A. Moses, Private Communication.

[7] J.J. MacFarlane, G.A. Moses, and R.R. Peterson, 'BUCKY - A 1-D radiation hydrodynamics code for simulating inertial confinemnent fusion high energy density plasmas', University of Wisconsin Fusion Technology Institute report, UWFDM-984.

[8] J.D. Hoffman, *Numerical Methods for Engineers and Scientists* (McGraw-Hill, New York, 1992).

[9] G.L. Olson and P.B. Kunasz, "Short characteristic solution of the non-LTE line transfer problem by operator perturbation-I. The one-dimensional slab," J. Quant. Spectrosc. Radiat. Transfer, **38** No. 5, p. 325 (1987).

[10] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions, Applied Mathematics Series No. 55*, National Bureau of Standards, Washington D.C. (1964).

[11] T.A. Brunner, "Some analytic solutions to the diffusion equation," Internal Memo.

[12] B. Su and G.L. Olson, "Benchmark results for the non-equilibrium Marshak diffusion problem," J. Quant. Spectrosc. Radiat. Transfer, **56** No. 3, p. 337 (1996).

[13] B. Su and G.L. Olson, "An analytical benchmark for non-equilibrium radiative transfer in an isotropically scattering medium," Ann. Nucl. Energy, **24** No. 13, p. 1035 (1997).

A BUCKY Namelist Files

A.1 Source and Vacuum Boundaries With No External Sources (Section 3.1)

```
$input

nmax   = 100
tmax   = 1.0e-9
ta     = 0.0e-9
dtb    = 1.e-13
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 1
isw(35) = 0
isw(36) = 1
isw(37) = 0
isw(38) = 0
srccon(1) = 120*1.
iradbc = 1
irad = 3
nrtang = 5
nfg = 1
tbc=0.025
filerh(1)='benchmark_radbc'

isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+02
radcon(1,2)=3.346e+02
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9
idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 120
jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1
dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 121*0.1

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000
io_netcdf = 100

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.2 Vacuum Boudaries With a Linear External Source (Section 3.2)

```
$input

nmax   = 100
tmax   = 1.0e-9
ta     = 0.0e-9
dtb    = 1.e-13
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9)  = 1
isw(35) = -1
isw(36) = 1
isw(37) = 0
isw(38) = 1
srccon(1) = 120*1.
iradbc = 0
irad   = 2
nrtang = 5
nfg = 1
tbc=0.025
filerx(1)='Uniform_extsource'

ss2b(1) =  0.5535
ss2b(2) =  0.5488
ss2b(3) =  0.5442
ss2b(4) =  0.5396
ss2b(5) =  0.5349
ss2b(6) =  0.5303
ss2b(7) =  0.5257
ss2b(8) =  0.5210
ss2b(9) =  0.5164
ss2b(10)=  0.5118
ss2b(11) = 0.5071
ss2b(12) =  0.5025
ss2b(13) =  0.4979
ss2b(14) =  0.4932
ss2b(15) =  0.4886
ss2b(16) =  0.4840
ss2b(17) =  0.4793
ss2b(18) =  0.4747
ss2b(19) =  0.4701
ss2b(20) =  0.4654
ss2b(21) =  0.4608
ss2b(22) =  0.4562
ss2b(23) =  0.4515
ss2b(24) =  0.4469
ss2b(25) =  0.4423
ss2b(26) =  0.4376
ss2b(27) =  0.4330
ss2b(28) =  0.4284
ss2b(29) =  0.4237
ss2b(30) =  0.4191
ss2b(31) =  0.4145
ss2b(32) =  0.4098
ss2b(33) =  0.4052
ss2b(34) =  0.4006
ss2b(35) =  0.3959
ss2b(36) =  0.3913
ss2b(37) =  0.3867
ss2b(38) =  0.3820
ss2b(39) =  0.3774
ss2b(40) =  0.3728
ss2b(41) =  0.3681
ss2b(42) =  0.3635
ss2b(43) =  0.3589
ss2b(44) =  0.3542
ss2b(45) =  0.3496
ss2b(46) =  0.3450
ss2b(47) =  0.3403
ss2b(48) =  0.3357
ss2b(49) =  0.3311
ss2b(50) =  0.3264
ss2b(51) =  0.3218
ss2b(52) =  0.3172
ss2b(53) =  0.3126
ss2b(54) =  0.3079
ss2b(55) =  0.3033
ss2b(56) =  0.2987
ss2b(57) =  0.2940
ss2b(58) =  0.2894
ss2b(59) =  0.2848
ss2b(60) =  0.2801
ss2b(61) =  0.2755
ss2b(62) =  0.2709
ss2b(63) =  0.2662
ss2b(64) =  0.2616
ss2b(65) =  0.2570
ss2b(66) =  0.2523
ss2b(67) =  0.2477
ss2b(68) =  0.2431
ss2b(69) =  0.2384
ss2b(70) =  0.2338
```

```
ss2b(71) =  0.2292
ss2b(72) =  0.2245
ss2b(73) =  0.2199
ss2b(74) =  0.2153
ss2b(75) =  0.2106
ss2b(76) =  0.2060
ss2b(77) =  0.2014
ss2b(78) =  0.1968
ss2b(79) =  0.1921
ss2b(80) =  0.1875
ss2b(81) =  0.1829
ss2b(82) =  0.1782
ss2b(83) =  0.1736
ss2b(84) =  0.1690
ss2b(85) =  0.1643
ss2b(86) =  0.1597
ss2b(87) =  0.1551
ss2b(88) =  0.1504
ss2b(89) =  0.1458
ss2b(90) =  0.1412
ss2b(91) =  0.1365
ss2b(92) =  0.1319
ss2b(93) =  0.1273
ss2b(94) =  0.1226
ss2b(95) =  0.1180
ss2b(96) =  0.1134
ss2b(97) =  0.1087
ss2b(98) =  0.1041
ss2b(99) =  0.0995
ss2b(100) =  0.0948
ss2b(101) =  0.0902
ss2b(102) =  0.0856
ss2b(103) =  0.0809
ss2b(104) =  0.0763
ss2b(105) =  0.0717
ss2b(106) =  0.0670
ss2b(107) =  0.0624
ss2b(108) =  0.0578
ss2b(109) =  0.0531
ss2b(110) =  0.0485
ss2b(111) =  0.0439
ss2b(112) =  0.0392
ss2b(113) =  0.0346
ss2b(114) =  0.0300
ss2b(115) =  0.0253
ss2b(116) =  0.0207
ss2b(117) =  0.0161
ss2b(118) =  0.0114
ss2b(119) =  0.0068
ss2b(120) =  0.0022

isw(6) = 1


ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+02
radcon(1,2)=3.346e+02
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9


idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 120*0.1

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.3 An External Source With a Source Boundary Condition (Section 3.3)

```
$input

nmax   = 100
tmax   = 1.0e-9
ta     = 0.0e-9
dtb    = 1.e-13
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13


isw(9) = 1
isw(35) = -1
isw(36) = 1
isw(37) = 0
isw(38) = 1
srccon(1) = 120*1.
iradbc = 1
irad = 3
nrtang = 5
nfg = 1
tbc=0.025
filerh(1)='benchmark_radbc'
filerx(1)='Uniform_extsource'

ss2b(1) =   3.8451e-001
ss2b(2) =   3.8239e-001
ss2b(3) =   3.8028e-001
ss2b(4) =   3.7819e-001
ss2b(5) =   3.7612e-001
ss2b(6) =   3.7406e-001
ss2b(7) =   3.7202e-001
ss2b(8) =   3.6999e-001
ss2b(9) =   3.6798e-001
ss2b(10) =   3.6598e-001
ss2b(11) =   3.6400e-001
ss2b(12) =   3.6203e-001
ss2b(13) =   3.6007e-001
ss2b(14) =   3.5813e-001
ss2b(15) =   3.5620e-001
ss2b(16) =   3.5428e-001
ss2b(17) =   3.5238e-001
ss2b(18) =   3.5049e-001
ss2b(19) =   3.4861e-001
ss2b(20) =   3.4675e-001
ss2b(21) =   3.4489e-001
ss2b(22) =   3.4305e-001
ss2b(23) =   3.4122e-001
ss2b(24) =   3.3940e-001
ss2b(25) =   3.3759e-001
ss2b(26) =   3.3579e-001
ss2b(27) =   3.3400e-001
ss2b(28) =   3.3221e-001
ss2b(29) =   3.3044e-001
ss2b(30) =   3.2868e-001
ss2b(31) =   3.2692e-001
ss2b(32) =   3.2518e-001
ss2b(33) =   3.2344e-001
ss2b(34) =   3.2171e-001
ss2b(35) =   3.1999e-001
ss2b(36) =   3.1828e-001
ss2b(37) =   3.1657e-001
ss2b(38) =   3.1487e-001
ss2b(39) =   3.1317e-001
ss2b(40) =   3.1148e-001
ss2b(41) =   3.0980e-001
ss2b(42) =   3.0813e-001
ss2b(43) =   3.0646e-001
ss2b(44) =   3.0479e-001
ss2b(45) =   3.0313e-001
ss2b(46) =   3.0147e-001
ss2b(47) =   2.9982e-001
ss2b(48) =   2.9817e-001
ss2b(49) =   2.9652e-001
ss2b(50) =   2.9488e-001
ss2b(51) =   2.9324e-001
ss2b(52) =   2.9161e-001
ss2b(53) =   2.8997e-001
ss2b(54) =   2.8834e-001
ss2b(55) =   2.8671e-001
ss2b(56) =   2.8508e-001
ss2b(57) =   2.8345e-001
ss2b(58) =   2.8183e-001
ss2b(59) =   2.8020e-001
ss2b(60) =   2.7858e-001
ss2b(61) =   2.7695e-001
ss2b(62) =   2.7533e-001
ss2b(63) =   2.7370e-001
ss2b(64) =   2.7208e-001
ss2b(65) =   2.7045e-001
ss2b(66) =   2.6882e-001
ss2b(67) =   2.6719e-001
ss2b(68) =   2.6556e-001
ss2b(69) =   2.6393e-001
```

```
ss2b(70) =   2.6229e-001
ss2b(71) =   2.6065e-001
ss2b(72) =   2.5901e-001
ss2b(73) =   2.5737e-001
ss2b(74) =   2.5572e-001
ss2b(75) =   2.5407e-001
ss2b(76) =   2.5241e-001
ss2b(77) =   2.5075e-001
ss2b(78) =   2.4909e-001
ss2b(79) =   2.4742e-001
ss2b(80) =   2.4574e-001
ss2b(81) =   2.4406e-001
ss2b(82) =   2.4237e-001
ss2b(83) =   2.4068e-001
ss2b(84) =   2.3898e-001
ss2b(85) =   2.3727e-001
ss2b(86) =   2.3556e-001
ss2b(87) =   2.3384e-001
ss2b(88) =   2.3211e-001
ss2b(89) =   2.3037e-001
ss2b(90) =   2.2863e-001
ss2b(91) =   2.2688e-001
ss2b(92) =   2.2511e-001
ss2b(93) =   2.2334e-001
ss2b(94) =   2.2157e-001
ss2b(95) =   2.1978e-001
ss2b(96) =   2.1798e-001
ss2b(97) =   2.1617e-001
ss2b(98) =   2.1435e-001
ss2b(99) =   2.1252e-001
ss2b(100) =   2.1068e-001
ss2b(101) =   2.0883e-001
ss2b(102) =   2.0696e-001
ss2b(103) =   2.0509e-001
ss2b(104) =   2.0320e-001
ss2b(105) =   2.0130e-001
ss2b(106) =   1.9939e-001
ss2b(107) =   1.9746e-001
ss2b(108) =   1.9552e-001
ss2b(109) =   1.9357e-001
ss2b(110) =   1.9160e-001
ss2b(111) =   1.8962e-001
ss2b(112) =   1.8763e-001
ss2b(113) =   1.8562e-001
ss2b(114) =   1.8360e-001
ss2b(115) =   1.8156e-001
ss2b(116) =   1.7950e-001
ss2b(117) =   1.7743e-001
ss2b(118) =   1.7534e-001
ss2b(119) =   1.7324e-001
ss2b(120) =   1.7112e-001
```

```
isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+02
radcon(1,2)=3.346e+02
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 120*0.1

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.4 A Boundary Source and an Albedo Boundary Condition (Section 3.4)

```
$input

nmax   = 100
tmax   = 1.0e-9
ta     = 0.0e-9
dtb    = 1.e-13
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 3
isw(35) = -1
isw(36) = 1
isw(37) = 0
isw(38) = 0
iradbc = 1
irad   = 2
nrtang = 5
nfg = 1
tbc=0.025
filerh(1)='benchmark_radbc'
con(73)=0.75

isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+03
radcon(1,2)=3.346e+03
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9
idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 121*0.1

isw(66) = 1
io(1)   = 5*1000
iobin   = 1000
dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.5 Steady-State Diffusion in Cylindrical Coordinates (Section 4.1)

```
$input

nmax   = 100
tmax   = 10.0e-9
ta     = 0.0e-9
dtb    = 1.e-14
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-14
dtmax  = 1.e-14

isw(9) = 2
isw(35) = -1
isw(36) = 1
isw(37) = 0
isw(38) = 0
iradbc = -1
irad   = 2
nrtang = 5
nfg = 1
tbc=0.025
filerh(1)='benchmark_radbc'

isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.6806e+03
radcon(1,2)=3.346e+02
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 2
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 120*0.1

isw(66) = 1
io(1)   = 5*1000
iobin   = 1000

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.6 Steady-State Diffusion in Spherical Coordinates (Section 4.2)

```
$input

nmax   = 100
tmax   = 10.0e-9
ta     = 0.0e-9
dtb    = 1.e-14
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-14
dtmax  = 1.e-14

isw(9) = 2
isw(35) = -1
isw(36) = 1
isw(37) = 0
isw(38) = 0
iradbc = -1
irad = 2
nrtang = 5
nfg = 1
tbc=0.025
filerh(1)='benchmark_radbc'

isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.6806e+03
radcon(1,2)=3.346e+02
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 3
```

```
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 120*0.1

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.7 Flux-Limiters (Section 4.3)

A.7.1 Dirichlet Boundary Conditions: SUM-Limiter

```
$input

nmax   = 100
tmax   = 1.0e-9
ta     = 0.0e-9
dtb    = 1.e-13
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 1
isw(35) = 0
isw(36) = 1
isw(37) = 1
isw(38) = 1
srccon(1) = 120*1.
iradbc = 1
irad = 2
nrtang = 5
nfg = 1
tbc = 0.025
filerh(1)='benchmark_radbc'
filerx(1)='Uniform_extsource'
con(74)=1.4142
ibench(3)=1

ss2b(1) =  4.3989e+000
ss2b(2) =  4.5595e+000
ss2b(3) =  4.7195e+000
ss2b(4) =  4.8790e+000
ss2b(5) =  5.0379e+000
ss2b(6) =  5.1964e+000
ss2b(7) =  5.3543e+000
ss2b(8) =  5.5118e+000
ss2b(9) =  5.6688e+000
ss2b(10) =  5.8253e+000
ss2b(11) =  5.9815e+000
ss2b(12) =  6.1372e+000
ss2b(13) =  6.2925e+000
ss2b(14) =  6.4475e+000
ss2b(15) =  6.6020e+000
ss2b(16) =  6.7563e+000
ss2b(17) =  6.9100e+000
ss2b(18) =  7.0636e+000
ss2b(19) =  7.2168e+000
ss2b(20) =  7.3695e+000
ss2b(21) =  7.5221e+000
ss2b(22) =  7.6743e+000
ss2b(23) =  7.8263e+000
ss2b(24) =  7.9778e+000
ss2b(25) =  8.1293e+000
ss2b(26) =  8.2804e+000
ss2b(27) =  8.4312e+000
ss2b(28) =  8.5818e+000
ss2b(29) =  8.7321e+000
ss2b(30) =  8.8822e+000
ss2b(31) =  9.0322e+000
ss2b(32) =  9.1817e+000
ss2b(33) =  9.3312e+000
ss2b(34) =  9.4803e+000
ss2b(35) =  9.6294e+000
ss2b(36) =  9.7782e+000
ss2b(37) =  9.9267e+000
ss2b(38) =  1.0075e+001
ss2b(39) =  1.0223e+001
ss2b(40) =  1.0371e+001
ss2b(41) =  1.0519e+001
ss2b(42) =  1.0667e+001
ss2b(43) =  1.0814e+001
ss2b(44) =  1.0962e+001
ss2b(45) =  1.1109e+001
ss2b(46) =  1.1256e+001
ss2b(47) =  1.1403e+001
ss2b(48) =  1.1550e+001
ss2b(49) =  1.1696e+001
ss2b(50) =  1.1843e+001
ss2b(51) =  1.1989e+001
ss2b(52) =  1.2135e+001
ss2b(53) =  1.2281e+001
ss2b(54) =  1.2427e+001
ss2b(55) =  1.2573e+001
ss2b(56) =  1.2718e+001
ss2b(57) =  1.2864e+001
ss2b(58) =  1.3009e+001
ss2b(59) =  1.3155e+001
ss2b(60) =  1.3300e+001
ss2b(61) =  1.3445e+001
ss2b(62) =  1.3590e+001
ss2b(63) =  1.3735e+001
ss2b(64) =  1.3880e+001
ss2b(65) =  1.4024e+001
ss2b(66) =  1.4169e+001
ss2b(67) =  1.4313e+001
ss2b(68) =  1.4458e+001
```

```
ss2b(69) =  1.4602e+001
ss2b(70) =  1.4746e+001
ss2b(71) =  1.4890e+001
ss2b(72) =  1.5034e+001
ss2b(73) =  1.5178e+001
ss2b(74) =  1.5322e+001
ss2b(75) =  1.5466e+001
ss2b(76) =  1.5610e+001
ss2b(77) =  1.5754e+001
ss2b(78) =  1.5897e+001
ss2b(79) =  1.6041e+001
ss2b(80) =  1.6184e+001
ss2b(81) =  1.6327e+001
ss2b(82) =  1.6471e+001
ss2b(83) =  1.6614e+001
ss2b(84) =  1.6757e+001
ss2b(85) =  1.6900e+001
ss2b(86) =  1.7043e+001
ss2b(87) =  1.7186e+001
ss2b(88) =  1.7329e+001
ss2b(89) =  1.7472e+001
ss2b(90) =  1.7615e+001
ss2b(91) =  1.7758e+001
ss2b(92) =  1.7901e+001
ss2b(93) =  1.8043e+001
ss2b(94) =  1.8186e+001
ss2b(95) =  1.8329e+001
ss2b(96) =  1.8471e+001
ss2b(97) =  1.8613e+001
ss2b(98) =  1.8756e+001
ss2b(99) =  1.8898e+001
ss2b(100) =  1.9041e+001
ss2b(101) =  1.9183e+001
ss2b(102) =  1.9325e+001
ss2b(103) =  1.9467e+001
ss2b(104) =  1.9609e+001
ss2b(105) =  1.9751e+001
ss2b(106) =  1.9893e+001
ss2b(107) =  2.0035e+001
ss2b(108) =  2.0177e+001
ss2b(109) =  2.0319e+001
ss2b(110) =  2.0461e+001
ss2b(111) =  2.0603e+001
ss2b(112) =  2.0745e+001
ss2b(113) =  2.0887e+001
ss2b(114) =  2.1028e+001
ss2b(115) =  2.1170e+001
ss2b(116) =  2.1312e+001
ss2b(117) =  2.1453e+001
ss2b(118) =  2.1595e+001
ss2b(119) =  2.1737e+001
ss2b(120) =  2.1878e+001
```

```
isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+02
radcon(1,2)=3.346e+03
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 120*0.1

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.7.2 Dirichlet Boundary Conditions: MAX-Limiter

```
$input

nmax   = 100
tmax   = 1.0e-9
ta     = 0.0e-9
dtb    = 1.e-13
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 1
isw(35) = 1
isw(36) = 1
isw(37) = 1
isw(38) = 1
srccon(1) = 120*1.
iradbc = 1
irad = 2
nrtang = 5
nfg = 1
tbc = 0.025
filerh(1)='benchmark_radbc'
filerx(1)='Uniform_extsource'
con(74)=1.4142
ibench(3)=1

ss2b(1) =  2.6273e+000
ss2b(2) =  2.7663e+000
ss2b(3) =  2.9053e+000
ss2b(4) =  3.0443e+000
ss2b(5) =  3.1833e+000
ss2b(6) =  3.3223e+000
ss2b(7) =  3.4613e+000
ss2b(8) =  3.6003e+000
ss2b(9) =  3.7393e+000
ss2b(10) =  3.8783e+000
ss2b(11) =  4.0173e+000
ss2b(12) =  4.1563e+000
ss2b(13) =  4.2953e+000
ss2b(14) =  4.4344e+000
ss2b(15) =  4.5733e+000
ss2b(16) =  4.7124e+000
ss2b(17) =  4.8513e+000
ss2b(18) =  4.9904e+000
ss2b(19) =  5.1294e+000
ss2b(20) =  5.2683e+000
ss2b(21) =  5.4074e+000
ss2b(22) =  5.5463e+000
ss2b(23) =  5.6854e+000
ss2b(24) =  5.8243e+000
ss2b(25) =  5.9634e+000
ss2b(26) =  6.1024e+000
ss2b(27) =  6.2413e+000
ss2b(28) =  6.3804e+000
ss2b(29) =  6.5193e+000
ss2b(30) =  6.6584e+000
ss2b(31) =  6.7975e+000
ss2b(32) =  6.9364e+000
ss2b(33) =  1.0076e+001
ss2b(34) =  1.0215e+001
ss2b(35) =  1.0354e+001
ss2b(36) =  1.0493e+001
ss2b(37) =  1.0632e+001
ss2b(38) =  1.0771e+001
ss2b(39) =  1.0910e+001
ss2b(40) =  1.1049e+001
ss2b(41) =  1.1188e+001
ss2b(42) =  1.1327e+001
ss2b(43) =  1.1466e+001
ss2b(44) =  1.1605e+001
ss2b(45) =  1.1744e+001
ss2b(46) =  1.1883e+001
ss2b(47) =  1.2022e+001
ss2b(48) =  1.2161e+001
ss2b(49) =  1.2299e+001
ss2b(50) =  1.2439e+001
ss2b(51) =  1.2577e+001
ss2b(52) =  1.2716e+001
ss2b(53) =  1.2855e+001
ss2b(54) =  1.2994e+001
ss2b(55) =  1.3133e+001
ss2b(56) =  1.3272e+001
ss2b(57) =  1.3411e+001
ss2b(58) =  1.3550e+001
ss2b(59) =  1.3689e+001
ss2b(60) =  1.3828e+001
ss2b(61) =  1.3967e+001
ss2b(62) =  1.4106e+001
ss2b(63) =  1.4245e+001
ss2b(64) =  1.4384e+001
ss2b(65) =  1.4523e+001
ss2b(66) =  1.4662e+001
ss2b(67) =  1.4801e+001
ss2b(68) =  1.4940e+001
ss2b(69) =  1.5079e+001
ss2b(70) =  1.5218e+001
ss2b(71) =  1.5357e+001
ss2b(72) =  1.5496e+001
```

```
ss2b(73) =  1.5635e+001
ss2b(74) =  1.5774e+001
ss2b(75) =  1.5913e+001
ss2b(76) =  1.6052e+001
ss2b(77) =  1.6191e+001
ss2b(78) =  1.6330e+001
ss2b(79) =  1.6469e+001
ss2b(80) =  1.6608e+001
ss2b(81) =  1.6747e+001
ss2b(82) =  1.6886e+001
ss2b(83) =  1.7025e+001
ss2b(84) =  1.7164e+001
ss2b(85) =  1.7303e+001
ss2b(86) =  1.7442e+001
ss2b(87) =  1.7581e+001
ss2b(88) =  1.7720e+001
ss2b(89) =  1.7859e+001
ss2b(90) =  1.7998e+001
ss2b(91) =  1.8137e+001
ss2b(92) =  1.8276e+001
ss2b(93) =  1.8415e+001
ss2b(94) =  1.8554e+001
ss2b(95) =  1.8693e+001
ss2b(96) =  1.8832e+001
ss2b(97) =  1.8971e+001
ss2b(98) =  1.9110e+001
ss2b(99) =  1.9249e+001
ss2b(100) =  1.9388e+001
ss2b(101) =  1.9527e+001
ss2b(102) =  1.9666e+001
ss2b(103) =  1.9805e+001
ss2b(104) =  1.9944e+001
ss2b(105) =  2.0083e+001
ss2b(106) =  2.0222e+001
ss2b(107) =  2.0361e+001
ss2b(108) =  2.0500e+001
ss2b(109) =  2.0639e+001
ss2b(110) =  2.0778e+001
ss2b(111) =  2.0917e+001
ss2b(112) =  2.1056e+001
ss2b(113) =  2.1195e+001
ss2b(114) =  2.1334e+001
ss2b(115) =  2.1473e+001
ss2b(116) =  2.1612e+001
ss2b(117) =  2.1751e+001
ss2b(118) =  2.1890e+001
ss2b(119) =  2.2029e+001
ss2b(120) =  2.2168e+001

isw(6) = 1
```

```
ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+02
radcon(1,2)=3.346e+03
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 120*0.1

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.7.3 Dirichlet Boundary Conditions: Larsen-Limiter

```
$input

nmax   = 100
tmax   = 1.0e-9
ta     = 0.0e-9
dtb    = 1.e-13
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 1
isw(35) = 2
isw(36) = 1
isw(37) = 1
isw(38) = 1
srccon(1) = 120*1.
iradbc = 1
irad = 2
nrtang = 5
nfg = 1
tbc = 0.025
filerh(1)='benchmark_radbc'
filerx(1)='Uniform_extsource'
con(72) = 2.
con(74)=1.4142
ibench(3)=1

ss2b(1) =  3.6416e+000
ss2b(2) =  3.8159e+000
ss2b(3) =  3.9900e+000
ss2b(4) =  4.1639e+000
ss2b(5) =  4.3374e+000
ss2b(6) =  4.5107e+000
ss2b(7) =  4.6837e+000
ss2b(8) =  4.8563e+000
ss2b(9) =  5.0286e+000
ss2b(10) =  5.2004e+000
ss2b(11) =  5.3719e+000
ss2b(12) =  5.5430e+000
ss2b(13) =  5.7136e+000
ss2b(14) =  5.8839e+000
ss2b(15) =  6.0535e+000
ss2b(16) =  6.2229e+000
ss2b(17) =  6.3915e+000
ss2b(18) =  6.5600e+000
ss2b(19) =  6.7279e+000
ss2b(20) =  6.8952e+000
ss2b(21) =  7.0621e+000
ss2b(22) =  7.2284e+000
ss2b(23) =  7.3944e+000
ss2b(24) =  7.5597e+000
ss2b(25) =  7.7247e+000
ss2b(26) =  7.8892e+000
ss2b(27) =  8.0530e+000
ss2b(28) =  8.2166e+000
ss2b(29) =  8.3794e+000
ss2b(30) =  8.5420e+000
ss2b(31) =  8.7041e+000
ss2b(32) =  8.8656e+000
ss2b(33) =  9.0268e+000
ss2b(34) =  9.1873e+000
ss2b(35) =  9.3476e+000
ss2b(36) =  9.5074e+000
ss2b(37) =  9.6666e+000
ss2b(38) =  9.8255e+000
ss2b(39) =  9.9839e+000
ss2b(40) =  1.0142e+001
ss2b(41) =  1.0300e+001
ss2b(42) =  1.0457e+001
ss2b(43) =  1.0613e+001
ss2b(44) =  1.0770e+001
ss2b(45) =  1.0926e+001
ss2b(46) =  1.1082e+001
ss2b(47) =  1.1237e+001
ss2b(48) =  1.1392e+001
ss2b(49) =  1.1546e+001
ss2b(50) =  1.1700e+001
ss2b(51) =  1.1854e+001
ss2b(52) =  1.2008e+001
ss2b(53) =  1.2161e+001
ss2b(54) =  1.2314e+001
ss2b(55) =  1.2466e+001
ss2b(56) =  1.2618e+001
ss2b(57) =  1.2770e+001
ss2b(58) =  1.2922e+001
ss2b(59) =  1.3074e+001
ss2b(60) =  1.3224e+001
ss2b(61) =  1.3375e+001
ss2b(62) =  1.3526e+001
ss2b(63) =  1.3676e+001
ss2b(64) =  1.3826e+001
ss2b(65) =  1.3976e+001
ss2b(66) =  1.4125e+001
ss2b(67) =  1.4275e+001
ss2b(68) =  1.4424e+001
ss2b(69) =  1.4573e+001
ss2b(70) =  1.4721e+001
ss2b(71) =  1.4870e+001
```

```
ss2b(72) =  1.5018e+001
ss2b(73) =  1.5166e+001
ss2b(74) =  1.5313e+001
ss2b(75) =  1.5461e+001
ss2b(76) =  1.5608e+001
ss2b(77) =  1.5756e+001
ss2b(78) =  1.5903e+001
ss2b(79) =  1.6049e+001
ss2b(80) =  1.6196e+001
ss2b(81) =  1.6342e+001
ss2b(82) =  1.6489e+001
ss2b(83) =  1.6635e+001
ss2b(84) =  1.6781e+001
ss2b(85) =  1.6927e+001
ss2b(86) =  1.7073e+001
ss2b(87) =  1.7218e+001
ss2b(88) =  1.7364e+001
ss2b(89) =  1.7509e+001
ss2b(90) =  1.7654e+001
ss2b(91) =  1.7799e+001
ss2b(92) =  1.7944e+001
ss2b(93) =  1.8089e+001
ss2b(94) =  1.8233e+001
ss2b(95) =  1.8378e+001
ss2b(96) =  1.8522e+001
ss2b(97) =  1.8667e+001
ss2b(98) =  1.8811e+001
ss2b(99) =  1.8955e+001
ss2b(100) =  1.9099e+001
ss2b(101) =  1.9243e+001
ss2b(102) =  1.9387e+001
ss2b(103) =  1.9530e+001
ss2b(104) =  1.9674e+001
ss2b(105) =  1.9817e+001
ss2b(106) =  1.9961e+001
ss2b(107) =  2.0104e+001
ss2b(108) =  2.0247e+001
ss2b(109) =  2.0390e+001
ss2b(110) =  2.0533e+001
ss2b(111) =  2.0676e+001
ss2b(112) =  2.0819e+001
ss2b(113) =  2.0962e+001
ss2b(114) =  2.1105e+001
ss2b(115) =  2.1247e+001
ss2b(116) =  2.1390e+001
ss2b(117) =  2.1532e+001
ss2b(118) =  2.1675e+001
ss2b(119) =  2.1817e+001
ss2b(120) =  2.1959e+001

isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+02
radcon(1,2)=3.346e+03
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 120*0.1

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.7.4 Dirichlet Boundary Conditions: Levermore-Pomraning-Limiter

```
$input

nmax   = 100
tmax   = 1.0e-9
ta     = 0.0e-9
dtb    = 1.e-13
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 1
isw(35) = 3
isw(36) = 1
isw(37) = 1
isw(38) = 1
srccon(1) = 120*1.
iradbc = 1
irad = 2
nrtang = 5
nfg = 1
tbc = 0.025
filerh(1)='benchmark_radbc'
filerx(1)='Uniform_extsource'
con(74)=1.4142
ibench(3)=1

ss2b(1) =  3.9585e+000
ss2b(2) =  4.1254e+000
ss2b(3) =  4.2919e+000
ss2b(4) =  4.4579e+000
ss2b(5) =  4.6235e+000
ss2b(6) =  4.7887e+000
ss2b(7) =  4.9535e+000
ss2b(8) =  5.1178e+000
ss2b(9) =  5.2817e+000
ss2b(10) =  5.4452e+000
ss2b(11) =  5.6083e+000
ss2b(12) =  5.7709e+000
ss2b(13) =  5.9332e+000
ss2b(14) =  6.0951e+000
ss2b(15) =  6.2565e+000
ss2b(16) =  6.4176e+000
ss2b(17) =  6.5781e+000
ss2b(18) =  6.7385e+000
ss2b(19) =  6.8985e+000
ss2b(20) =  7.0579e+000
ss2b(21) =  7.2171e+000
ss2b(22) =  7.3758e+000
ss2b(23) =  7.5343e+000
ss2b(24) =  7.6922e+000
ss2b(25) =  7.8500e+000
ss2b(26) =  8.0075e+000
ss2b(27) =  8.1644e+000
ss2b(28) =  8.3212e+000
ss2b(29) =  8.4774e+000
ss2b(30) =  8.6335e+000
ss2b(31) =  8.7894e+000
ss2b(32) =  8.9447e+000
ss2b(33) =  9.0999e+000
ss2b(34) =  9.2546e+000
ss2b(35) =  9.4092e+000
ss2b(36) =  9.5635e+000
ss2b(37) =  9.7173e+000
ss2b(38) =  9.8711e+000
ss2b(39) =  1.0024e+001
ss2b(40) =  1.0178e+001
ss2b(41) =  1.0331e+001
ss2b(42) =  1.0483e+001
ss2b(43) =  1.0635e+001
ss2b(44) =  1.0787e+001
ss2b(45) =  1.0939e+001
ss2b(46) =  1.1091e+001
ss2b(47) =  1.1242e+001
ss2b(48) =  1.1393e+001
ss2b(49) =  1.1544e+001
ss2b(50) =  1.1695e+001
ss2b(51) =  1.1845e+001
ss2b(52) =  1.1996e+001
ss2b(53) =  1.2146e+001
ss2b(54) =  1.2295e+001
ss2b(55) =  1.2445e+001
ss2b(56) =  1.2595e+001
ss2b(57) =  1.2744e+001
ss2b(58) =  1.2893e+001
ss2b(59) =  1.3042e+001
ss2b(60) =  1.3190e+001
ss2b(61) =  1.3339e+001
ss2b(62) =  1.3487e+001
ss2b(63) =  1.3635e+001
ss2b(64) =  1.3784e+001
ss2b(65) =  1.3931e+001
ss2b(66) =  1.4079e+001
ss2b(67) =  1.4227e+001
ss2b(68) =  1.4374e+001
ss2b(69) =  1.4521e+001
ss2b(70) =  1.4668e+001
ss2b(71) =  1.4815e+001
ss2b(72) =  1.4962e+001
```

```
ss2b(73) =  1.5109e+001
ss2b(74) =  1.5255e+001
ss2b(75) =  1.5402e+001
ss2b(76) =  1.5548e+001
ss2b(77) =  1.5694e+001
ss2b(78) =  1.5840e+001
ss2b(79) =  1.5986e+001
ss2b(80) =  1.6132e+001
ss2b(81) =  1.6277e+001
ss2b(82) =  1.6423e+001
ss2b(83) =  1.6568e+001
ss2b(84) =  1.6713e+001
ss2b(85) =  1.6859e+001
ss2b(86) =  1.7004e+001
ss2b(87) =  1.7149e+001
ss2b(88) =  1.7294e+001
ss2b(89) =  1.7438e+001
ss2b(90) =  1.7583e+001
ss2b(91) =  1.7728e+001
ss2b(92) =  1.7872e+001
ss2b(93) =  1.8017e+001
ss2b(94) =  1.8161e+001
ss2b(95) =  1.8305e+001
ss2b(96) =  1.8449e+001
ss2b(97) =  1.8593e+001
ss2b(98) =  1.8737e+001
ss2b(99) =  1.8881e+001
ss2b(100) =  1.9025e+001
ss2b(101) =  1.9169e+001
ss2b(102) =  1.9312e+001
ss2b(103) =  1.9456e+001
ss2b(104) =  1.9600e+001
ss2b(105) =  1.9743e+001
ss2b(106) =  1.9886e+001
ss2b(107) =  2.0030e+001
ss2b(108) =  2.0173e+001
ss2b(109) =  2.0316e+001
ss2b(110) =  2.0459e+001
ss2b(111) =  2.0602e+001
ss2b(112) =  2.0745e+001
ss2b(113) =  2.0888e+001
ss2b(114) =  2.1031e+001
ss2b(115) =  2.1174e+001
ss2b(116) =  2.1316e+001
ss2b(117) =  2.1459e+001
ss2b(118) =  2.1601e+001
ss2b(119) =  2.1744e+001
ss2b(120) =  2.1887e+001

isw(6) = 1
```

```
ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+02
radcon(1,2)=3.346e+03
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)    = 120*1
jmn(1)     = 1
jmx(1)     = 120
jzn1(1)    = 40
jzn3(1)    = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)    = 120*1.e21
do2b(1)    = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)    = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)    = 120*1.0
tn2c(1)    = 120*0.1
te2c(1)    = 120*0.1
tr2c(1)    = 120*0.1

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.7.5 Source and Vacuum Boundary Conditions: SUM-Limiter

```
$input

nmax   = 100
tmax   = 1.0e-9
ta     = 0.0e-9
dtb    = 1.e-13
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 1
isw(35) = 0
isw(36) = 1
isw(37) = 0
isw(38) = 1
srccon(1) = 120*1.
iradbc = 1
irad = 2
nrtang = 5
nfg = 1
tbc=0.025
filerh(1)='benchmark_radbc'
filerx(1)='Uniform_extsource'

ss2b(1) =   2.3557e-001
ss2b(2) =   2.3207e-001
ss2b(3) =   2.2856e-001
ss2b(4) =   2.2504e-001
ss2b(5) =   2.2150e-001
ss2b(6) =   2.1794e-001
ss2b(7) =   2.1437e-001
ss2b(8) =   2.1079e-001
ss2b(9) =   2.0719e-001
ss2b(10) =   2.0358e-001
ss2b(11) =   1.9995e-001
ss2b(12) =   1.9630e-001
ss2b(13) =   1.9263e-001
ss2b(14) =   1.8895e-001
ss2b(15) =   1.8526e-001
ss2b(16) =   1.8154e-001
ss2b(17) =   1.7781e-001
ss2b(18) =   1.7405e-001
ss2b(19) =   1.7028e-001
ss2b(20) =   1.6649e-001
ss2b(21) =   1.6267e-001
ss2b(22) =   1.5884e-001
ss2b(23) =   1.5499e-001
ss2b(24) =   1.5111e-001
ss2b(25) =   1.4721e-001
ss2b(26) =   1.4329e-001
ss2b(27) =   1.3935e-001
ss2b(28) =   1.3538e-001
ss2b(29) =   1.3139e-001
ss2b(30) =   1.2738e-001
ss2b(31) =   1.2333e-001
ss2b(32) =   1.1927e-001
ss2b(33) =   1.1517e-001
ss2b(34) =   1.1105e-001
ss2b(35) =   1.0690e-001
ss2b(36) =   1.0272e-001
ss2b(37) =   9.8511e-002
ss2b(38) =   9.4269e-002
ss2b(39) =   9.0001e-002
ss2b(40) =   8.5697e-002
ss2b(41) =   8.1359e-002
ss2b(42) =   7.6993e-002
ss2b(43) =   7.2592e-002
ss2b(44) =   6.8150e-002
ss2b(45) =   6.3677e-002
ss2b(46) =   5.9161e-002
ss2b(47) =   5.4612e-002
ss2b(48) =   5.0017e-002
ss2b(49) =   4.5388e-002
ss2b(50) =   4.0710e-002
ss2b(51) =   3.5996e-002
ss2b(52) =   3.1237e-002
ss2b(53) =   2.6426e-002
ss2b(54) =   2.1574e-002
ss2b(55) =   1.6669e-002
ss2b(56) =   1.1719e-002
ss2b(57) =   6.7119e-003
ss2b(58) =   1.6582e-003
ss2b(59) =  -3.4559e-003
ss2b(60) =  -8.6199e-003
ss2b(61) =  -1.3842e-002
ss2b(62) =  -1.9129e-002
ss2b(63) =  -2.4472e-002
ss2b(64) =  -2.9884e-002
ss2b(65) =  -3.5355e-002
ss2b(66) =  -4.0900e-002
ss2b(67) =  -4.6507e-002
ss2b(68) =  -5.2194e-002
ss2b(69) =  -5.7947e-002
ss2b(70) =  -6.3777e-002
ss2b(71) =  -6.9693e-002
ss2b(72) =  -7.5683e-002
ss2b(73) =  -8.1765e-002
ss2b(74) =  -8.7927e-002
```

```
ss2b(75) =  -9.4187e-002
ss2b(76) =  -1.0053e-001
ss2b(77) =  -1.0698e-001
ss2b(78) =  -1.1352e-001
ss2b(79) =  -1.2017e-001
ss2b(80) =  -1.2693e-001
ss2b(81) =  -1.3379e-001
ss2b(82) =  -1.4078e-001
ss2b(83) =  -1.4789e-001
ss2b(84) =  -1.5511e-001
ss2b(85) =  -1.6247e-001
ss2b(86) =  -1.6995e-001
ss2b(87) =  -1.7758e-001
ss2b(88) =  -1.8536e-001
ss2b(89) =  -1.9327e-001
ss2b(90) =  -2.0135e-001
ss2b(91) =  -2.0959e-001
ss2b(92) =  -2.1799e-001
ss2b(93) =  -2.2658e-001
ss2b(94) =  -2.3533e-001
ss2b(95) =  -2.4429e-001
ss2b(96) =  -2.5343e-001
ss2b(97) =  -2.6279e-001
ss2b(98) =  -2.7236e-001
ss2b(99) =  -2.8214e-001
ss2b(100) =  -2.9217e-001
ss2b(101) =  -3.0243e-001
ss2b(102) =  -3.1296e-001
ss2b(103) =  -3.2375e-001
ss2b(104) =  -3.3481e-001
ss2b(105) =  -3.4617e-001
ss2b(106) =  -3.5782e-001
ss2b(107) =  -3.6980e-001
ss2b(108) =  -3.8210e-001
ss2b(109) =  -3.9477e-001
ss2b(110) =  -4.0780e-001
ss2b(111) =  -4.2121e-001
ss2b(112) =  -4.3503e-001
ss2b(113) =  -4.4926e-001
ss2b(114) =  -4.6396e-001
ss2b(115) =  -4.7912e-001
ss2b(116) =  -4.9476e-001
ss2b(117) =  -5.1094e-001
ss2b(118) =  -5.2765e-001
ss2b(119) =  -5.4496e-001
ss2b(120) =  -5.6288e-001

isw(6) = 1

ideos(1)  =  3

idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+02
radcon(1,2)=3.346e+02
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 120

jmat(1)   = 120*1
jmn(1)    = 1
jmx(1)    = 120
jzn1(1)   = 40
jzn3(1)   = 40
zonfc1(1) = 0.
zonfc3(1) = 0.
regmas(1) = 1.661e-3
regms1(1) = 5.537e-4
regms3(1) = 5.537e-4

isw(3) = 1

dn2b(1)   = 120*1.e21
do2b(1)   = 120*1.e21
atw2b(1)  = 120*1.0
atwo(1)   = 120*1.0
atn2b(1)  = 120*1.0
zo2b(1)   = 120*1.0
tn2c(1)   = 120*0.1
te2c(1)   = 120*0.1
tr2c(1)   = 120*0.1

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000
io_netcdf = 100

dtpout(1) = 10.0e-9
tprbeg(1) = 0.0e-9
dtbout(1) = 0.0001e-9
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.8 Time Dependent Diffusion in Planar Coordinates (Section 4.4.1)

```
$input

nmax   = 10000
tmax   = 0.1e-9
ta     = 0.001e-9
dtb    = 1.e-16
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 1
isw(35) = -1
isw(36) = 0
isw(37) = 1
isw(38) = 0
iradbc = 0
irad = 2
nrtang = 5
nfg = 1
tbc=0.025

isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+03
radcon(1,2)=3.346e+03
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 240

jmat(1)   = 240*1
jmn(1)    = 1
jmx(1)    = 240
jzn1(1)   = 99
jzn3(1)   = 99
zonfc1(1) = -0.1
zonfc3(1) = -0.1
```

```
regmas(1) = 1.661e-1
regms1(1) = 8.3045e-2
regms3(1) = 8.3045e-2

isw(3) = 1

dn2b(1)   = 240*1.e21
do2b(1)   = 240*1.e21
atw2b(1)  = 240*1.0
atwo(1)   = 240*1.0
atn2b(1)  = 240*1.0
zo2b(1)   = 240*1.0
tn2c(1)   = 240*0.1
te2c(1)   = 240*0.1
tr2c(1)  =  46*1.0000e-001
tr2c(47) =  2.6610e-001
tr2c(48) =  6.4382e-001
tr2c(49) =  1.3334e+000
tr2c(50) =  2.3710e+000
tr2c(51) =  3.7731e+000
tr2c(52) =  5.5450e+000
tr2c(53) =  7.5062e+000
tr2c(54) =  9.6007e+000
tr2c(55) =  1.1700e+001
tr2c(56) =  1.3686e+001
tr2c(57) =  1.5472e+001
tr2c(58) =  1.7128e+001
tr2c(59) =  1.8618e+001
tr2c(60) =  1.9806e+001
tr2c(61) =  2.0925e+001
tr2c(62) =  2.1755e+001
tr2c(63) =  2.2519e+001
tr2c(64) =  2.3124e+001
tr2c(65) =  2.3592e+001
tr2c(66) =  2.4008e+001
tr2c(67) =  2.4315e+001
tr2c(68) =  2.4532e+001
tr2c(69) =  2.4724e+001
tr2c(70) =  2.4890e+001
tr2c(71) =  2.4996e+001
tr2c(72) =  2.5088e+001
tr2c(73) =  2.5164e+001
tr2c(74) =  2.5206e+001
tr2c(75) =  2.5256e+001
tr2c(76) =  2.5280e+001
tr2c(77) =  2.5290e+001
tr2c(78) =  2.5304e+001
tr2c(79) =  2.5308e+001
tr2c(80) =  2.5312e+001
```

```
tr2c(81) =  2.5311e+001
tr2c(82) =  2.5308e+001
tr2c(83) =  2.5304e+001
tr2c(84) =  2.5298e+001
tr2c(85) =  2.5298e+001
tr2c(86) =  2.5290e+001
tr2c(87) =  2.5280e+001
tr2c(88) =  2.5280e+001
tr2c(89) =  2.5269e+001
tr2c(90) =  2.5269e+001
tr2c(91) =  2.5256e+001
tr2c(92) =  2.5256e+001
tr2c(93) =  2.5256e+001
tr2c(94) =  2.5241e+001
tr2c(95) =  2.5241e+001
tr2c(96) =  2.5241e+001
tr2c(97) =  2.5241e+001
tr2c(98) =  2.5241e+001
tr2c(99) =  2.5224e+001
tr2c(100) =  2.5224e+001
tr2c(101) =  2.5224e+001
tr2c(102) =  2.5224e+001
tr2c(103) =  2.5224e+001
tr2c(104) =  2.5224e+001
tr2c(105) =  2.5224e+001
tr2c(106) =  2.5206e+001
tr2c(107) =  2.5206e+001
tr2c(108) =  2.5206e+001
tr2c(109) =  2.5206e+001
tr2c(110) =  2.5206e+001
tr2c(111) =  2.5206e+001
tr2c(112) =  2.5206e+001
tr2c(113) =  2.5186e+001
tr2c(114) =  2.5186e+001
tr2c(115) =  2.5186e+001
tr2c(116) =  2.5186e+001
tr2c(117) =  2.5186e+001
tr2c(118) =  2.5186e+001
tr2c(119) =  2.5186e+001
tr2c(120) =  2.5164e+001
tr2c(121) =  2.5164e+001
tr2c(122) =  2.5164e+001
tr2c(123) =  2.5164e+001
tr2c(124) =  2.5164e+001
tr2c(125) =  2.5164e+001
tr2c(126) =  2.5164e+001
tr2c(127) =  2.5140e+001
tr2c(128) =  2.5140e+001
tr2c(129) =  2.5140e+001
tr2c(130) =  2.5140e+001
tr2c(131) =  2.5140e+001
tr2c(132) =  2.5140e+001
tr2c(133) =  2.5115e+001
tr2c(134) =  2.5115e+001
tr2c(135) =  2.5115e+001
tr2c(136) =  2.5115e+001
tr2c(137) =  2.5115e+001
tr2c(138) =  2.5115e+001
tr2c(139) =  2.5115e+001
tr2c(140) =  2.5088e+001
tr2c(141) =  2.5088e+001
tr2c(142) =  2.5088e+001
tr2c(143) =  2.5088e+001
tr2c(144) =  2.5088e+001
tr2c(145) =  2.5088e+001
tr2c(146) =  2.5059e+001
tr2c(147) =  2.5059e+001
tr2c(148) =  2.5059e+001
tr2c(149) =  2.5028e+001
tr2c(150) =  2.5028e+001
tr2c(151) =  2.5028e+001
tr2c(152) =  2.4996e+001
tr2c(153) =  2.4996e+001
tr2c(154) =  2.4962e+001
tr2c(155) =  2.4962e+001
tr2c(156) =  2.4927e+001
tr2c(157) =  2.4890e+001
tr2c(158) =  2.4851e+001
tr2c(159) =  2.4810e+001
tr2c(160) =  2.4768e+001
tr2c(161) =  2.4724e+001
tr2c(162) =  2.4679e+001
tr2c(163) =  2.4632e+001
tr2c(164) =  2.4532e+001
tr2c(165) =  2.4427e+001
tr2c(166) =  2.4315e+001
tr2c(167) =  2.4197e+001
tr2c(168) =  2.4073e+001
tr2c(169) =  2.3875e+001
tr2c(170) =  2.3665e+001
tr2c(171) =  2.3441e+001
tr2c(172) =  2.3124e+001
tr2c(173) =  2.2786e+001
tr2c(174) =  2.2335e+001
tr2c(175) =  2.1855e+001
tr2c(176) =  2.1348e+001
tr2c(177) =  2.0708e+001
tr2c(178) =  1.9922e+001
tr2c(179) =  1.8981e+001
tr2c(180) =  1.7880e+001
tr2c(181) =  1.6748e+001
tr2c(182) =  1.5344e+001
tr2c(183) =  1.3940e+001
tr2c(184) =  1.2311e+001
tr2c(185) =  1.0512e+001
tr2c(186) =  8.7298e+000
```

```
tr2c(187) =  6.9342e+000
tr2c(188) =  5.2289e+000
tr2c(189) =  3.7121e+000
tr2c(190) =  2.4584e+000
tr2c(191) =  1.4740e+000
tr2c(192) =  7.6965e-001
tr2c(193) =  3.5813e-001
tr2c(194) =  1.3879e-001
tr2c(195) =  46*1.0000e-001

isw(66) = 1
io(1)  = 5*1000

iobin  = 1000
io_netcdf = 100

dtpout(1) = 1.0e-12
tprbeg(1) = 0.0e-9
dtbout(1) = 0.5e-12
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.9 Time Dependent Diffusion in Spherical Coordinates (Section 4.4.2)

```
$input

nmax   = 10000
tmax   = 0.1e-9
ta     = 0.001e-9
dtb    = 1.e-16
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 2
isw(35) = -1
isw(36) = 0
isw(37) = 1
isw(38) = 0
iradbc = 0
irad = 2
nrtang = 5
nfg = 1
tbc=0.025

isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+03
radcon(1,2)=3.346e+03
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 3
isw(4) = 26
nvregn = 1
jmax   = 200

jmat(1)    = 200*1
jmn(1)     = 1
jmx(1)     = 200
jzn1(1)    = 99
jzn3(1)    = 99
zonfc1(1) = 0.15
zonfc3(1) = -0.09
regmas(1) = 869.6976
regms1(1) = 0.08119
regms3(1) = 869.604

isw(3) = 1

dn2b(1)   = 200*1.e21
do2b(1)   = 200*1.e21
atw2b(1)  = 200*1.0
atwo(1)   = 200*1.0
atn2b(1)  = 200*1.0
zo2b(1)   = 200*1.0
tn2c(1)   = 200*0.1
te2c(1)   = 200*0.1

tr2c(1) =  3.9682e+002
tr2c(2) =  3.9462e+002
tr2c(3) =  3.9328e+002
tr2c(4) =  3.9199e+002
tr2c(5) =  3.9068e+002
tr2c(6) =  3.8933e+002
tr2c(7) =  3.8792e+002
tr2c(8) =  3.8642e+002
tr2c(9) =  3.8482e+002
tr2c(10) =  3.8311e+002
tr2c(11) =  3.8127e+002
tr2c(12) =  3.7929e+002
tr2c(13) =  3.7716e+002
tr2c(14) =  3.7485e+002
tr2c(15) =  3.7236e+002
tr2c(16) =  3.6966e+002
tr2c(17) =  3.6673e+002
tr2c(18) =  3.6357e+002
tr2c(19) =  3.6014e+002
tr2c(20) =  3.5642e+002
tr2c(21) =  3.5241e+002
tr2c(22) =  3.4806e+002
tr2c(23) =  3.4336e+002
tr2c(24) =  3.3829e+002
tr2c(25) =  3.3282e+002
tr2c(26) =  3.2692e+002
tr2c(27) =  3.2058e+002
tr2c(28) =  3.1377e+002
tr2c(29) =  3.0646e+002
tr2c(30) =  2.9865e+002
tr2c(31) =  2.9031e+002
tr2c(32) =  2.8143e+002
tr2c(33) =  2.7199e+002
tr2c(34) =  2.6201e+002
```

```
tr2c(35) =  2.5148e+002
tr2c(36) =  2.4040e+002
tr2c(37) =  2.2881e+002
tr2c(38) =  2.1675e+002
tr2c(39) =  2.0423e+002
tr2c(40) =  1.9132e+002
tr2c(41) =  1.7809e+002
tr2c(42) =  1.6462e+002
tr2c(43) =  1.5100e+002
tr2c(44) =  1.3737e+002
tr2c(45) =  1.2380e+002
tr2c(46) =  1.1045e+002
tr2c(47) =  9.7445e+001
tr2c(48) =  8.4935e+001
tr2c(49) =  7.3041e+001
tr2c(50) =  6.1893e+001
tr2c(51) =  5.1607e+001
tr2c(52) =  4.2278e+001
tr2c(53) =  3.3962e+001
tr2c(54) =  2.6708e+001
tr2c(55) =  2.0515e+001
tr2c(56) =  1.5358e+001
tr2c(57) =  1.1175e+001
tr2c(58) =  7.8843e+000
tr2c(59) =  5.3760e+000
tr2c(60) =  3.5315e+000
tr2c(61) =  2.2266e+000
tr2c(62) =  1.3420e+000
tr2c(63) =  7.6974e-001
tr2c(64) =  4.1819e-001
tr2c(65) =  2.1402e-001
tr2c(66) =  1.0263e-001
tr2c(67) =  134*1.0000e-001

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000
io_netcdf = 100

dtpout(1) = 1.0e-12
tprbeg(1) = 0.0e-9
dtbout(1) = 0.5e-12
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.10 Time Dependent Diffusion in Cylindrical Coordinates (Section 4.4.3)

```
$input

nmax   = 10000
tmax   = 0.1e-9
ta     = 0.001e-9
dtb    = 1.e-16
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-13
dtmax  = 1.e-13

isw(9) = 2
isw(35) = -1
isw(36) = 0
isw(37) = 1
isw(38) = 0
iradbc = 0
irad = 2
nrtang = 5
nfg = 1
tbc=0.025

isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
filses(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=3.346e+03
radcon(1,2)=3.346e+03
radcon(1,3)=1.e-30
isw(16) = 1
isw(96) = -9

idelta = 2
isw(4) = 26
nvregn = 1
jmax   = 200

jmat(1)    = 200*1
jmn(1)     = 1
jmx(1)     = 200
jzn1(1)    = 99
jzn3(1)    = 99
zonfc1(1) = 0.1
zonfc3(1) = -0.088
regmas(1) = 13.0455
regms1(1) = 1.3045-3
regms3(1) = 13.0439

isw(3) = 1

dn2b(1)   = 200*1.e21
do2b(1)   = 200*1.e21
atw2b(1)  = 200*1.0
atwo(1)   = 200*1.0
atn2b(1)  = 200*1.0
zo2b(1)   = 200*1.0
tn2c(1)   = 200*0.1
te2c(1)   = 200*0.1
tr2c(1) =  2.4740e+002
tr2c(2) =  2.4738e+002
tr2c(3) =  2.4736e+002
tr2c(4) =  2.4733e+002
tr2c(5) =  2.4731e+002
tr2c(6) =  2.4728e+002
tr2c(7) =  2.4726e+002
tr2c(8) =  2.4722e+002
tr2c(9) =  2.4719e+002
tr2c(10) =  2.4715e+002
tr2c(11) =  2.4711e+002
tr2c(12) =  2.4706e+002
tr2c(13) =  2.4701e+002
tr2c(14) =  2.4695e+002
tr2c(15) =  2.4689e+002
tr2c(16) =  2.4682e+002
tr2c(17) =  2.4675e+002
tr2c(18) =  2.4666e+002
tr2c(19) =  2.4657e+002
tr2c(20) =  2.4647e+002
tr2c(21) =  2.4636e+002
tr2c(22) =  2.4624e+002
tr2c(23) =  2.4611e+002
tr2c(24) =  2.4597e+002
tr2c(25) =  2.4581e+002
tr2c(26) =  2.4563e+002
tr2c(27) =  2.4544e+002
tr2c(28) =  2.4522e+002
tr2c(29) =  2.4499e+002
tr2c(30) =  2.4473e+002
tr2c(31) =  2.4445e+002
tr2c(32) =  2.4414e+002
tr2c(33) =  2.4380e+002
tr2c(34) =  2.4342e+002
tr2c(35) =  2.4301e+002
```

```
tr2c(36) =  2.4256e+002
tr2c(37) =  2.4206e+002
tr2c(38) =  2.4152e+002
tr2c(39) =  2.4092e+002
tr2c(40) =  2.4027e+002
tr2c(41) =  2.3955e+002
tr2c(42) =  2.3876e+002
tr2c(43) =  2.3790e+002
tr2c(44) =  2.3695e+002
tr2c(45) =  2.3591e+002
tr2c(46) =  2.3478e+002
tr2c(47) =  2.3353e+002
tr2c(48) =  2.3217e+002
tr2c(49) =  2.3069e+002
tr2c(50) =  2.2906e+002
tr2c(51) =  2.2729e+002
tr2c(52) =  2.2536e+002
tr2c(53) =  2.2325e+002
tr2c(54) =  2.2095e+002
tr2c(55) =  2.1845e+002
tr2c(56) =  2.1573e+002
tr2c(57) =  2.1278e+002
tr2c(58) =  2.0959e+002
tr2c(59) =  2.0612e+002
tr2c(60) =  2.0238e+002
tr2c(61) =  1.9834e+002
tr2c(62) =  1.9399e+002
tr2c(63) =  1.8932e+002
tr2c(64) =  1.8431e+002
tr2c(65) =  1.7895e+002
tr2c(66) =  1.7324e+002
tr2c(67) =  1.6716e+002
tr2c(68) =  1.6072e+002
tr2c(69) =  1.5393e+002
tr2c(70) =  1.4678e+002
tr2c(71) =  1.3931e+002
tr2c(72) =  1.3152e+002
tr2c(73) =  1.2346e+002
tr2c(74) =  1.1516e+002
tr2c(75) =  1.0667e+002
tr2c(76) =  9.8055e+001
tr2c(77) =  8.9382e+001
tr2c(78) =  8.0733e+001
tr2c(79) =  7.2171e+001
tr2c(80) =  6.3796e+001
tr2c(81) =  5.5708e+001
tr2c(82) =  4.7990e+001
tr2c(83) =  4.0726e+001
tr2c(84) =  3.4003e+001
tr2c(85) =  2.7879e+001
tr2c(86) =  2.2412e+001
tr2c(87) =  1.7627e+001
tr2c(88) =  1.3533e+001
tr2c(89) =  1.0119e+001
tr2c(90) =  7.3505e+000
tr2c(91) =  5.1707e+000
tr2c(92) =  3.5122e+000
tr2c(93) =  2.2949e+000
tr2c(94) =  1.4367e+000
tr2c(95) =  8.5867e-001
tr2c(96) =  4.8725e-001
tr2c(97) =  2.6136e-001
tr2c(98) =  1.3167e-001
tr2c(99) =  104*0.1000e+000

isw(66) = 1
io(1)  = 5*1000
iobin  = 1000
io_netcdf = 100

dtpout(1) = 1.0e-12
tprbeg(1) = 0.0e-9
dtbout(1) = 0.5e-12
tpbbeg(1) = 0.0e-9

nfdout = 100000

$end
```

A.11 Su and Olson Marshak Wave Problem (Section 5.1)

```
$input

nmax   = 500000
tmax   = 3.0e-10
ta     = 0.0
dtb    = 1.e-15
tscte  = 0.05
tsctn  = 0.05
tsctr  = 0.10
tscv   = 0.05
tscc   = 0.10
dtmin  = 1.e-15
dtmax  = 1.e-15

isw(9) = 1
isw(35) = -1
isw(36) = 0
isw(37) = 0
iradbc = 1
irad = 2
nrtang = 2
nfg = 1
tbc=0.025
filerh(1)='SuOlson_radbc'

isw(6) = 1

ideos(1)  =  3
idopac(1) =  3
fileos(1) = 'eos.dat.uw.benchmark'
radcon(1,1)=11.547
radcon(1,2)=11.547
radcon(1,3)=11.547
ibench(2) = 1
con(60) = 5.48792e-4
isw(16) = 1
isw(96) = -9

idelta = 1
isw(4) = 26
nvregn = 1
jmax   = 200

jmat(1)   = 200*1
jmn(1)    = 1
jmx(1)    = 200
jzn1(1)   = 99
jzn3(1)   = 99
zonfc1(1) = 0.05
zonfc3(1) = -0.05
regmas(1) = 1.0000e+1
regms1(1) = 5.0000e-2
regms3(1) = 9.9444e00

isw(3) = 1
con(1) = 1.e-30
con(2) = 1.e-30
mxtiter = 1

dn2b(1)   = 200*6.02e23
do2b(1)   = 200*6.02e23
atw2b(1)  = 200*1.0000
atwo(1)   = 200*1.0000
atn2b(1)  = 200*1.0000
zo2b(1)   = 200*1.0000
tn2c(1)   = 202*1.
te2c(1)   = 202*1.
tr2c(1)   = 202*1.

isw(66) = 1
io(1)  = 5*1000

dtpout(1) = 1.0e-11
tprbeg(1) = 0.e-11
dtbout(1) = 0.001e-12
tpbbeg(1) = 2.880e-12
dtbout(2) = 1.e-12
tpbbeg(2) = 2.890e-12
dtbout(3) = 0.001e-11
tpbbeg(3) = 2.880e-11
dtbout(4) = 1.e-11
tpbbeg(4) = 2.890e-11
dtbout(5) = 0.001e-10
tpbbeg(5) = 2.880e-10
dtbout(6) = 1.e-10
tpbbeg(6) = 2.890e-10

nfdout = 100000

$end
```

B BUCKY Resource Files

B.1 eos.dat.uw.benchmark

```
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
ppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppp
*****                                                                                 *****
*****                              EOS AND OPACITY TABLE                              *****
*****                                       for                                       *****
*****                                    Benchmark                                    *****
*****                                                                                 *****
*****                                                                                 *****
*****                                                                                 *****
*****                                                                                 *****
*****                                                                                 *****
*****                     (The calculation is done on MM-DD-YY)                       *****
*****                                                                                 *****
wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww
ppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppppp


 atomic #s of gases:          1
 relative fractions:   1.00E+00
********************     mesh parameters for EoS            ********************
******************** temperature(eV), density(cm-3) and group ********************
******************** nT, (Temp(i),i=1,nT)                   ********************
******************** nD, (deni(i),i=1,nD)                   ********************
******************** rho0 (g/cm3)                           ********************
   2
    1.0000E-01  1.0000E+05
   2
    1.0000E+18  1.0000E+23
0.0000E+00
********************       mesh  parameters for opacity      ********************
******************** nT, (Temp(i),i=1,nT)                   ********************
******************** nD, (deni(i),i=1,nD)                   ********************
******************** Ngroup                                 ********************
   2
    1.0000E-01  1.0000E+05
   2
    1.0000E+18  1.0000E+23
   1
********************              group structure (eV)       ********************
 1.000E-02  1.000E+06
********************                     Zbar                ********************
  1.0000000000E-05   1.0000000000E-05   1.0000000000E-05   1.0000000000E-05
********************                 Eint (J/g)              ********************
  2.0000000000E-30   2.0000000000E-30   2.0000000000E-30   2.0000000000E-30
********************                 dE/dT (J/g/eV)          ********************
  2.0000000000E-30   2.0000000000E-30   2.0000000000E-30   2.0000000000E-30
********************                 dE/dN                   ********************
  1.0000000000E-30   1.0000000000E-30   1.0000000000E-30   1.0000000000E-30
********************                 Eion (J/g)              ********************
  1.0000000000E-30   1.0000000000E-30   1.0000000000E-30   1.0000000000E-30
********************                 Eele (J/g)              ********************
  1.0000000000E-30   1.0000000000E-30   1.0000000000E-30   1.0000000000E-30
********************               dEi/dT (J/g/eV)           ********************
  1.0000000000E-30   1.0000000000E-30   1.0000000000E-30   1.0000000000E-30
********************               dEe/dT (J/g/eV)           ********************
  1.0000000000E-30   1.0000000000E-30   1.0000000000E-30   1.0000000000E-30
********************               Pion (dyne/cm**2)         ********************
  1.0000000000E-30   1.0000000000E-30   1.0000000000E-30   1.0000000000E-30
********************               Pele (dyne/cm**2)         ********************
  1.0000000000E-30   1.0000000000E-30   1.0000000000E-30   1.0000000000E-30
********************           dPi/dT (dyne/cm**2/eV)        ********************
  1.0000000000E-30   1.0000000000E-30   1.0000000000E-30   1.0000000000E-30
```

```
********************            dPe/dT (dyne/cm**2/eV)        ********************
  1.0000000000E-30   1.0000000000E-30   1.0000000000E-30   1.0000000000E-30
********************     Rosseland Mean Opacity (cm**2/g)     ********************
 1.000E+00  1.000E+00  1.000E+00  1.000E+00
********************  emission Planck Mean Opacity (cm**2/g) ********************
 1.000E+00  1.000E+00  1.000E+00  1.000E+00
******************** absorption Planck Mean Opacity (cm**2/g) ********************
 1.000E+00  1.000E+00  1.000E+00  1.000E+00
```

B.2 benchmark_radbc

```
#                                                        #
#      Radiation drive for rad transport benchmarks      #
#             ----------------------------               #
#                                                        #
#                                                        #
#                                                        #
#  time (psec)      T_rad (eV)    T_bright_bc            #
#  -----------      ----------    ------------           #
    0.                100.0          100.0
    100000.           100.0          100.0
   -1.               -1.            -1.
```

B.3 Uniform_extsource

```
#                                                        #
#      External source for rad transport benchmarks      #
#             ----------------------------               #
#                                                        #
#                                                        #
#                                                        #
#  time (psec)      T_rad (eV)    T_bright_bc            #
#  -----------      ----------    ------------           #
    0.                100.0          100.0
    100000.           100.0          100.0
   -1.               -1.            -1.
```

B.4 SuOlson_radbc

```
#                                                        #
#      Radiation drive for Su Olson marshak save prblm   #
#             ----------------------------               #
#                                                        #
#                                                        #
#                                                        #
#  time (psec)      T_rad (eV)    T_bright_bc            #
#  -----------      ----------    ------------           #
    0.                0.1           0.1
    0.01000           1000.0        1000.0
    100000.           1000.0        1000.0
   -1.                -1.           -1.
```

B.5 SuOlson_extsource

```
#                                                        #
#   External source for Su Olson non-equilibrium prblm   #
#             ----------------------------               #
#                                                        #
#                                                        #
#                                                        #
#  time (psec)      T_rad (eV)    T_bright_bc            #
#  -----------      ----------    ------------           #
    0.                0.1            0.1
    0.001             100.0          100.0
    16.678            100.0          100.0
    16.679            0.0            0.0
   -1.               -1.            -1.
```

C Descriptions of BUCKY Namelist Variables

```
c ...    TIME CONTROL PARAMETERS
c        *******************************************************
c ......... Parameter Descriptions ..............{} for default
c
c ......... nmax  => maximum # of run cycles .............................................{0}
c ......... tmax  => maximum simulation time .............................................{0.}
c ......... ta    => beginning simulation time ...........................................{0.}
c ......... dtb   => beginning simulation time step ......................................{1.e-12}
c ......... tscte => time step control - (delta Te)/Te ...................................{0.05}
c ......... tsctn => time step control - (delta Ti)/Ti ...................................{0.05}
c ......... tsctr => time step control - (delta Er)/Er ...................................{0.1}
c ......... tscv  => time step control - (delta V)/V .....................................{0.05}
c ......... tscc  => time step control - courant .........................................{0.05}
c ......... dtmin => minimum delta t .....................................................{0.1*dtb}
c ......... dtmax => maximum delta t .....................................................{0.01*tmax}


c ...    RADIATION TRANSPORT PARAMETERS AND BOUNDARY CONDITIONS
c        ********************************************************
c ......... Parameter Descriptions ..............{} for default
c
c ......... isw(9)  => radiation boundary condition type .................................{0}
c .........                    j=1         j=jmaxp1
c .........         = 1;       trans.      trans.
c .........         = 2;       refl.       trans.
c .........         = 3;       trans.      refl.
c .........   not 1, 2, or 3   ****not allowed****
c !!!!!!!!!! isw(35) => diffusion flux-limiter control ....................................{0}
c .........         = 0; SUM flux-limiter
c .........         = 1; MAX flux-limiter
c .........         = 2; Larsen flux-limiter
c .........         = 3; Simplified Approximate Levermore-Pomraning flux-limiter
c .........         < 0; No flux-limiter (classical diffusion)
c !!!!!!!!!! isw(36) => time-dependent diffusion control ..................................{0}
c .........         = 0; time-dependent diffusion (alpha is evaluated)
c .........         = 1; time-independent diffusion (alpha = 0)
c !!!!!!!!!! isw(37) => Boundary source type control ......................................{0}
c .........         = 0; Boundary source specified for radiation flowing into sample
c .........         = 1; Dirichlet boundary source (E is fixed on boundary)
c !!!!!!!!!! isw(38) => External radiation source control .................................{0}
c .........         = 0; no external radiation source
c .........         = 1; external radiation input at j = jext(*) in format:
c .........                   timrbc, tradbc
c .........         = 2; external radiation input at j = jext(*) in format:
c .........                   timrbc, tradbc, t_bright_bc
c !!!!!!!!!! srccon(j) => multiplier on external source term for zone j ...................{0.}
c !!!!!!!!!! ss2b(j)   => effective emission opacity for external sources
c .........               defined as a temperature source {cm^-1} ........................{1.}
c !!!!!!!!!! ibench(3) => Control for benchmark problems .................................{0}
c .........           = 0; no effect on the diffusion boundary conditions
c .........           = 1; force Dirichlet BC at j=jmax with a value of B(T')
c .........                for T'=con(74)*T(radbc)
c ......... iradbc  => radiation source term .............................................{0}
c .........         = 0; no radiation source
c .........         = 1; radiation incident at j = 1 in format:
c .........                   timrbc, tradbc
c .........         = 2; radiation incident at j = 1 in format:
c .........                   timrbc, tradbc, t_bright_bc
c .........         = 3; time and frequency dependent radiation bc at j = 1
c .........         *note:  any value < 0 puts radiation bc at j = jmax +1
c ......... irad    => radiation transport model ........................................{2}
c .........         = 2; multi-group diffusion
c .........         = 3; multi-group/multi-angle short characteristics
c ......... nrtang  => # of angles in MG MA rt model (2 or 5) ............................{2}
c ......... nfg     => # of frequency groups .............................................{0}
```

```
c .......... tbc     => initial temperature boundary condition ............................{0.}
c .......... filerh(1) => file containing time-dependant radiation BC specs
c .......... con(4)  => avoid divide by zero in flux-limiter ..............................{1.e-10}
c .......... con(23) => emission coefficient ..............................................{6.33e04}
c !!!!!!!!!! con(72) => n factor in the Larsen flux limiter ...............................{2.}
c !!!!!!!!!! con(73) => reflectivity (albedo) for reflective BC ...........................{1.}
c .......... con(74) => multiplier for flux/group input in freq-dep BC ....................{1.}
c .......... con(75) => multiplier for radiation temperature BC ...........................{1.}
c >>>>>>>>>> regroup_visrad_inc_flux_data => Boolean flag to indicate a
c ..........                                        regrouping of the f-dep bc............{.false.}


c ...    HYDRODYNAMICS PARAMETERS AND BOUNDARY CONDITIONS
c        ******************************************************
c .......... Parameter Descriptions ..............{} for default
c
c .......... isw(6)  => hydro switch ......................................................{0}
c ..........         = 0; hydro motion is computed
c ..........         = 1; no hydro motion
c .......... isw(7)  => hydro boundary conditions .........................................{0}
c ..........         = 0; both boundaries fixed
c ..........         = 1; allow free expansion of both boundaries
c ..........         = 2; allow free expansion of outer boundary
c .......... isw(13) => quiet start option ................................................{0}
c ..........         = 0; quiet start off
c ..........         = 1; quiet start on, zones can only move if
c ..........               T > con(19) for it and the surrounding zones.
c !!!!!!!!!!         = 2; quiet start on, inner zone boundary can only
c ..........               move if T > con(19) in that zone.
c .......... con(19) => quiet start temperature (eV) .......................................{0.15}

c ...    EQUATION OF STATE AND OPACITY PARAMETERS
c        ******************************************************
c .......... Parameter Descriptions ..............{} for default
c .......... isw(12)   => ideal gas option ................................................{0}
c ..........           = 0; do not use ideal gas
c ..........           = 1; use ideal gas (Z_bar = 0)
c ..........           = 2; use ideal gas (Z_bar = 1)
c !!!!!!!!!!           = 3; use ideal gas {Z_bar = EOSOPA table look-up}
c .......... con(5)    => constant value for log(lambda) ..................................{0.}
c ..........           = 1 for ideal gas
c .......... ideos(1)  => EOS file type ...................................................{-1}
c ..........           = 0; UW/WP file format
c ..........           = 1; UW/IONMIX file format
c ..........           = 2; SESAME file format
c ..........           = 3; UW/EOSOPA new file format
c ..........           < 0; ideal gas
c .......... idopac(1) => Opacity file type ...............................................{-1}
c ..........           = 0; UW/WP file format
c ..........           = 1; UW/IONMIX file format
c ..........           = 2; UW/EOSOPA old file format
c ..........           = 3; UW/EOSOPA new file format
c ..........           < 0; ideal gas
c !!!!!!!!!! ibench(2) => Control for benchmark problems to set Cv = aT^3 .................{0}
c ..........           = 0; read heat capacity from table
c ..........           = 1; set heat capacity to Cv=aT^3
c !!!!!!!!!! con(60)   => Value for a in Cv = aT^3 ........................................{1}
c .......... fileos(1) => file(s) containing the EOSOPA data...............................{none}
c .......... filses(1) => file(s) containing the SESAME data...............................{none}
c .......... radcon(i,1) => opacity multipier for the rosseland absorption
c ..........                in the i'th material.........................................{1}
c .......... radcon(i,2) => opacity multipier for the planck absorption
c ..........                in the i'th material.........................................{1}
c .......... radcon(i,3) => opacity multipier for the planck emission
c ..........                in the i'th material.........................................{1}
c .......... isw(16)   => negative temperature control.....................................{0}
c ..........           = 0; if T < 0, then stop calculation
```

```
c .........           = 1; if T < 0, then fix it and go on, but notify user
c ......... isw(96)   => control for eos grid boundary extrapolation .....................{0}
c .........           = -9; do not stop if off UW EOS grid, do extrapolation
c >>>>>>>>>> regroup_eosopa_mg_opacs => Boolean flag for indicating a regroup..............{.false.}
c >>>>>>>>>> num_rad_group_sections =>  number of different regroup sections...............{1}
c >>>>>>>>>> num_groups_in_section(1)=> number of rad groups in each section...............{nfg}
c .........    *note:  total number specified overides the specified nfg!!!
c >>>>>>>>>> rad_group_section_energy(1)=> the group bounds for each section...............{min,max}
c .........    *note:  you must specify 1 more boundary than number of sections
c >>>>>>>>>> t_cutoff_ideal_gas => cut-off temperature before which to use ideal gas EOS....{0.3}


c ...    LAGRANGIAN ZONING PARAMETERS
c       ************************************************************
c ......... Parameter Descriptions ..............{} for default
c
c ......... idelta   => geometry specification ...........................................{0}
c .........           = 1; 1-D planar
c .........           = 2; 1-D cylindrical
c .........           = 3; 1-D spherical
c ......... isw(4)   => automatic zoning model ...........................................{0}
c .........           = 0; manual zoning
c .........           = 1; ZONERP
c .........           = 2-9; ZONER2
c .........           = 10-15; ZONERC
c .........           = 20-25; ZONER3
c .........           = 26-30; ZONER4
c ......... nvregn   => number of regions in the spatial mesh ............................{0}
c ......... jmax     => total number of zones in the spatial mesh ........................{0}
c ......... jmat(i)  => material assignment to each zone "#zones*mat#" in region i ........{1}
c ......... jmn(i)   => zone number of first zone in region i ............................{0}
c ......... jmx(i)   => zone number of last zone in region i .............................{0}
c ......... jzn1(i)  => in ZONER4 -> number of zones in first sub-region in region i.......{0}
c ......... jzn3(i)  => in ZONER4 -> number of zones in third sub-region in region i.......{0}
c ......... zonfc1(i)=> in ZONER4 -> mass multiplier in first sub-redion in region i.......{0}
c ......... zonfc3(i)=> in ZONER4 -> mass multiplier in third sub-redion in region i.......{0}
c ......... regmas(i)=> total mass in region i (in g/cm^2 for planar, ....................{0}
c .........                 g/cm for cylindrical, and g for spherical)
c ......... regms1(i)=> in ZONER4 -> total mass in first sub-region in region i ..........{0}
c ......... regms3(i)=> in ZONER4 -> total mass in third sub-region in region i ..........{0}

c ...    PLASMA/TARGET PARAMETERS
c       ***********************************************************
c ......... Parameter Descriptions ..............{} for default
c
c ......... isw(3)   => Plasma Model..................................................{1}
c .........           = 1; 1-T model (Te=Ti)
c .........           = 2; 2-T model (Te .ne. Ti)
c .........           = 3; simultaneous solution of TR and pl.E (1-T)
c !!!!!!!!!! mxtiter  => number of iterations in the plasma temperature solution ..........{1}
c !!!!!!!!!! con(84)  => convergence criteria for ion (plasma) temperature solution ........{1.e-3)
c !!!!!!!!!! con(85)  => convergence criteria for electron temperature solution ............{1.e-3)
c ......... dn2b(i)  => ion number density of zones in region i "#zones*density" ..........{0}
c ......... do2b(i)  => number density of non-DT species in region i "#zones*density"......{0}
c ......... dd2b(i)  => number density of D species in region i "#zones*density" ..........{0}
c ......... dt2b(i)  => number density of T species in region i "#zones*density" ..........{0}
c ......... atwb(i)  => atomic weight of species in region i "#zones*A" ...................{0}
c ......... atob(i)  => atomic weight of non-DT species in region i "#zones*A" ............{0}
c ......... atnb(i)  => atomic number of species in region i "#zones*Z" ...................{0}
c ......... zo2b(i)  => atomic number of non-DT species in region i "#zones*Z" ............{0}


c ...    OUTPUT PARAMETERS
c       ************************************************************
c ......... Parameter Descriptions ..............{} for default
c
c ......... isw(66)  => output control ..................................................{0}
```

```
c ..........           = 0; based on number of hydro cycles
c ..........           = 1; based on simulation time
c .......... io(i)     => output controller for text file; ..................................{-1}
c ..........         i < 0; none
c ..........         i = 1; hydro quantities
c ..........         i = 2; energy conservation
c ..........         i = 3; number densities
c ..........         i = 4; short edit
c ..........         i = 5; multi-frequency radiation
c ..........         i = 6; fusion burn
c ..........         i = 9; CRE post-processing
c .......... iobin     => binary output frequency (cycles/dump) ............................{-1}
c .......... nfdout    => number of binary outputs per freq-dep binary output ..............{1}
c .......... dtpout(i)=> time between each print to output file ...........................{-1.}
c .......... tprbeg(i)=> time to begin i'th dtpout ........................................{0.}
c .......... dtbout(i)=> time between each write to binary file ...........................{-1.}
c .......... tpbbeg(i)=> time to begin i'th dtbout ........................................{0.}
c !!!!!!!!!! tpfbeg(i)=> time to begin i'th freq-dep binary outputs ........................{0.}
c !!!!!!!!!! tpfend(i)=> time to end i'th freq-dep binary outputs ..........................{0.}
c ..........             **** freq-dep dumps are made at the same frequency
c ..........             **** as the binary write (dtbout(j)) where tpfbeg(i)
c ..........             **** is > tpbbeg(j), but tpfend(i) is < tpbbeg(j+1)
c .......... tpnbeg(i)=> time to begin i'th dtnout ........................................{0.}
c .......... dtnout(i)=> time between each write to netcdf file ...........................{-1.}
c .......... isw(5)    => frequency of tabulation of overpressure and heat flux .............{20}
c ..........             at the outer boundary
c .......... io_netcdf => netcdf (exodus) format output frequency (cycles/dump).............{0}
```