Hybrid Adaptive Ray-Moment Method ($HARM^2$): A Highly Parallel Method for Radiation Hydrodynamics on Adaptive Grids

A.L. Rosen^a, M.R. Krumholz^b, J.S. Oishi^c, A.T. Lee^d, R.I. Klein^{d,e}

^a*Department of Astronomy & Astrophysics, University of California Santa Cruz, Santa Cruz, CA 95064, USA*

^b*Research School of Astronomy & Astrophysics, Australian National University, Canberra, ACT 2611, Australia*

^c*Department of Physics & Astronomy, Bates College, Lewiston, ME 04240, USA*

^d*Department of Astronomy, University of California, Berkeley, CA 94720, USA*

^e*Lawrence Livermore National Laboratory, PO Box L-23, Livermore, CA 94550, USA*

Abstract

We present a highly-parallel multi-frequency hybrid radiation hydrodynamics algorithm that combines a spatially-adaptive long characteristics method for the radiation field from point sources with a moment method that handles the diffuse radiation field produced by a volume-filling fluid. Our Hybrid Adaptive Ray-Moment Method ($HARM^2$) operates on patch-based adaptive grids, is compatible with asynchronous time stepping, and works with any moment method. In comparison to previous long characteristics methods, we have greatly improved the parallel performance of the adaptive long-characteristics method by developing a new completely asynchronous and non-blocking communication algorithm. As a result of this improvement, our implementation achieves near-perfect scaling up to $\mathcal{O}(10^3)$ processors on distributed memory machines. We present a series of tests to demonstrate the accuracy and performance of the method.

Keywords: radiative transfer, hydrodynamics, parallelization, long characteristics, adaptive mesh refinement

1. Introduction

Radiation-hydrodynamics (RHD) is a challenging numerical problem, but it is a crucial component in modeling several physical phenomena in the fields

18 of astrophysics, laser physics, and plasma physics. Accurate solution of the
 19 radiative transfer (RT) equation, which governs the evolution of radiation
 20 interacting with matter, is difficult because of its high dimensionality. This
 21 equation depends on six independent variables: three spatial, two angles
 22 describing the direction of the propagation of photons, and one frequency
 23 dimension. For time-dependent RHD calculations, this solution must be
 24 obtained at every time step, and then coupled to the hydrodynamics. Even
 25 on parallel supercomputers direct solution of the RT equation at each time
 26 step of a time-dependent calculation is prohibitively expensive, because of
 27 this most numerical RHD codes use approximations to treat the evolution of
 28 the radiation field and its interaction with matter.

29 One common approach to solving the RHD equations is to reduce the
 30 dimensionality of the problem. This class of approximations are known as
 31 moment methods because they take the moments of the radiative transfer
 32 equation in direct analogy to the Chapman-Enskog procedure used to derive
 33 the hydrodynamic equations from the kinetic theory of gases [1, 2]. This
 34 method averages over the angular dependence, and thus is a good approxi-
 35 mation for smooth, diffuse radiation fields such as those present in optically
 36 thick media when the radiation is tightly coupled to the matter. The ac-
 37 curacy with which moment methods recover the angular dependence of the
 38 true solution depends on the order at which the moments are closed, and on
 39 the closure relation adopted. Common approximations include flux-limited
 40 diffusion (FLD; closure at first moment) [3–5], the M1 method (closure at the
 41 2nd moment using a minimum entropy closure) [6, 7], and Variable Edding-
 42 ton Tensor (VET; closure at the 2nd moment using an approximate solution
 43 to the full transfer equation) [8–10]. Regardless of the order and closure rela-
 44 tion, the computational cost of these methods usually scales as N or $N \log N$,
 45 where N is the number of cells, and the technique is highly parallelizable [1].

46 An alternative technique used to solve the RT equation numerically is
 47 characteristics-based ray tracing, which solves this equation directly along
 48 specific rays. With this method, the directionality of the radiative flux is
 49 highly accurate, but the accuracy depends on the sampling of rays. Two
 50 widely used schemes for ray tracing in grid-based codes are *long* and *hybrid*
 51 characteristics. Long characteristics traces rays on a cell by cell basis, and
 52 provides maximum possible accuracy. Hybrid characteristics is a combina-
 53 tion of long characteristics within individual grids and short characteristics
 54 between grids (i.e., in which only neighboring grid cells are used to interpolate
 55 incoming intensities) [11, 12]. The method of short characteristics is faster

56 but more diffusive compared to long characteristics methods. The compu-
 57 tational cost for both methods scales linearly with the number of sources,
 58 rays traced, and grid cells with which the rays interact, making these meth-
 59 ods prohibitively expensive for treating diffuse radiation fields where every
 60 computational cell is a source. Instead, they are ideal for treating the ra-
 61 dial radiation fields of point sources. Even for this application, however,
 62 one major drawback of ray tracing methods, especially long characteristics,
 63 is that they are difficult to parallelize in a code where the hydrodynamics
 64 is parallelized by domain decomposition. In such a configuration, each ray
 65 will usually cross multiple processor domains, creating significant communi-
 66 cations overheads and serial bottlenecks.

67 In summary, moment methods are better at approximating the diffuse
 68 radiation field from a fluid but are poor at modeling the propagation of radi-
 69 ation from point sources where the direction of the field is important. Char-
 70 acteristics methods, in contrast, are good at approximating the direction-
 71 dependent radiation fields from point sources but are too computationally
 72 expensive for practical use in simulating a diffuse radiating fluid. When
 73 both point and diffuse radiation sources are present, therefore, a natural ap-
 74 proach is to combine both techniques by using long characteristics to model
 75 the propagation of radiation from a point source and its subsequent inter-
 76 action (e.g., absorption) with the fluid and then use a moment method to
 77 follow the subsequent diffuse re-emission.

78 This technique has been developed in several numerical codes in the past
 79 20 years, but these codes typically have been limited to cases where a ge-
 80 ometric symmetry simplifies the long characteristics solution. Wolfire and
 81 Cassinelli [13, 14] introduced a formal decomposition between the direct and
 82 dust-reprocessed radiation fields for a calculation in 1D spherical geometry.
 83 The first published 2D simulation using such a method is Murray et al. [15],
 84 who coupled long characteristics to FLD to model the direct (ray tracer) and
 85 scattered (FLD) radiation field in accretion disk coronae. Kuiper et al. [16]
 86 incorporated a similar hybrid approach in the 3D grid based code Pluto, but
 87 again limiting the problem to a special geometry: in this case a single point
 88 source at the origin of a spherical computational grid. Most recently, Klassen
 89 et al. [17] developed a hybrid scheme in the FLASH adaptive mesh refine-
 90 ment (AMR) code but uses FLD plus hybrid characteristics which, although
 91 faster, is less accurate than long characteristic methods.

92 The reason that many authors have resorted to special geometries or
 93 abandoned long characteristics is the difficulty in parallelizing long charac-

94 teristics in a general geometry, particularly in the case of adaptive grids.
 95 The problem is difficult because it is unknown *a priori* how far rays will
 96 travel and what grids they will interact with in an adaptive grid framework.
 97 In a distributed memory paradigm where different grids may be stored in
 98 memory on different processors, this can easily result in a complex communi-
 99 cation pattern with numerous serial bottlenecks. Indeed, all implementations
 100 of long characteristics on adaptive grids published to date use synchronous
 101 communication algorithms in which processors must wait for other processors
 102 to receive ray information [18], leading to exactly this problem.

103 In this paper we present our Hybrid Adaptive Ray-Moment Method
 104 ($HARM^2$) which uses long characteristics to treat radiation from point sources
 105 coupled to a moment method to handle the diffuse radiation field from the
 106 fluid. $HARM^2$ works on adaptive grids with asynchronous time stepping.
 107 We have greatly improved the parallelism of the long characteristics solve in
 108 a distributed memory framework through a new, completely asynchronous,
 109 non-blocking communication method. The rest of this paper is organized as
 110 follows. We begin with a formal derivation of our method for decomposing
 111 the radiation field into two components in section 2. Section 3 describes our
 112 numerical implementation of our hybrid radiation scheme in the astrophys-
 113 ical AMR code ORION. Next we confirm the robustness of our method by
 114 providing validation and performance tests in sections 4 and 5, respectively.
 115 Finally, we summarize our methods and results in section 6.

116 2. Decomposition of the Radiation-Hydrodynamics Problem

117 Here we describe a formal method to separate the radiation field into
 118 two components – (1) the diffuse radiation from the fluid and (2) the di-
 119 rect radiation field from point sources [19]. Formally, we consider a system
 120 consisting of a volume-filling radiating fluid plus point sources of radiation,
 121 and we wish to decompose the radiation fields produced by the fluid and the
 122 point sources. An example where such a decomposition is valuable is in the
 123 problem of simulating stars embedded in an optically thick, dusty medium
 124 such as is present during the early formation of a star cluster while the stars
 125 are actively accreting. The radiative flux from the stars will be absorbed
 126 by nearby dust and the dust will re-emit thermal radiation in the infrared.
 127 This radiation will be highly coupled to the interstellar medium and diffuse
 128 through the dense gas.

129 We begin with the equations of radiation-hydrodynamics (RHD) written
 130 in the lab-frame [20, 21]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (1)$$

$$\frac{\partial}{\partial t} (\rho \mathbf{v}) + \nabla \cdot (\rho \mathbf{v} \mathbf{v}) = -\nabla P + \mathbf{G} \quad (2)$$

$$\frac{\partial}{\partial t} (\rho E) + \nabla \cdot [(\rho E + P \mathbf{v})] = cG^0 \quad (3)$$

131 where ρ , $\mathbf{v}$, E , and P are the density, velocity, specific energy (thermal plus
 132 kinetic), and thermal pressure of the fluid, respectively; and $(G^0, \mathbf{G})$ is the
 133 radiation four-force density which is the negative of the radiation energy
 134 stress tensor and is given by

$$cG^0 = \int_0^\infty d\nu \int d\Omega [\kappa(\mathbf{n}, \nu) I(\mathbf{n}, \nu) - \eta(\mathbf{n}, \nu)] \quad (4)$$

$$c\mathbf{G} = \int_0^\infty d\nu \int d\Omega [\kappa(\mathbf{n}, \nu) I(\mathbf{n}, \nu) - \eta(\mathbf{n}, \nu)] \mathbf{n} \quad (5)$$

135 where $I(\mathbf{n}, \nu)$ is the intensity of the radiation field at frequency ν in di-
 136 rection $\mathbf{n}$. We note that the physical quantities given in Equations (1)-(3)
 137 depend on spatial position and time. The time-like and space-like compo-
 138 nents of $(G^0, \mathbf{G})$ represent the rate of energy and momentum transfer from
 139 the radiation to the fluid, respectively. The intensity is governed by the
 140 time-independent radiative transfer equation

$$\mathbf{n} \nabla I(\nu, \mathbf{n}) = -\kappa(\mathbf{n}, \nu) I(\mathbf{n}, \nu) + \eta(\mathbf{n}, \nu) \quad (6)$$

141 where $\kappa(\mathbf{n}, \nu)$ and $\eta(\mathbf{n}, \nu)$ are the direction and frequency dependent absorp-
 142 tion and emission coefficients in the lab-frame, respectively. For simplicity,
 143 we have neglected the effects of scattering because we expect to solve the
 144 equations of RHD in astrophysical problems where absorption is the dom-
 145 inant transfer mechanism. However, it would be straightforward to extend
 146 the method to include scattering in the diffuse component, as we point out
 147 below. We also ignore the time-dependence of the radiative transfer equation
 148 because our primary target application is systems where the light travel time
 149 is orders of magnitude smaller than the system dynamical time, and thus the
 150 radiation intensity is always in instantaneous equilibrium.

151 We now separate $I(\mathbf{n}, \nu)$ into two components

$$I(\mathbf{n}, \nu) = I_{\text{dir}}(\mathbf{n}, \nu) + I_{\text{diff}}(\mathbf{n}, \nu) \quad (7)$$

152 to describe the direct radiation fields from point sources (I_{dir}) and the diffuse
 153 radiation field (I_{diff}) emitted by the fluid. Since the sources that contribute
 154 to the direct radiation field are point sources we can represent their intensity
 155 as a sum of δ -functions

$$I_{\text{dir}}(\mathbf{n}, \nu) = \sum_{i=1}^N I_{\text{src},i}(\nu) \delta(\mathbf{n} - \mathbf{n}_{\text{src},i}), \quad (8)$$

156 where $\mathbf{n}_{\text{src},i} = (\mathbf{x} - \mathbf{x}_i)/|\mathbf{x} - \mathbf{x}_i|$ for any position $\mathbf{x}$ in the computational
 157 domain, $\mathbf{x}_i$ and $I_{\text{src},i}$ are the position and intensity of the i th point source,
 158 and we assume that the sources are isotropic emitters, so $I_{\text{src},i}$ is independent
 159 of $\mathbf{n}$.¹ With this formulation I_{dir} is non-zero only at special values of $\mathbf{n}$, i.e.,
 160 those that point from the position of a source $\mathbf{x}_i$ to the position in question
 161 $\mathbf{x}$, and zero for all others; while I_{diff} will be non-zero everywhere. However,
 162 because the four-force vector ($G^0, \mathbf{G}$) depends on integrals over $\mathbf{n}$, the δ -
 163 function contributions from I_{dir} may dominate at some positions, while the
 164 contribution from I_{diff} dominates elsewhere. This makes solution with a pure
 165 moment method difficult, and motivates us to treat the radiation fields of
 166 the point sources and fluid separately so that we can properly take into
 167 account the direction of the radiation fields from point sources. With this
 168 decomposition Equations (4)-(5) become

$$\begin{aligned} cG^0 &= \int_0^\infty d\nu \int d\Omega [\kappa(\mathbf{n}, \nu) I_{\text{dir}}(\mathbf{n}, \nu) - \eta_{\text{dir}}(\mathbf{n}, \nu)] \\ &\quad + \int_0^\infty d\nu \int d\Omega [\kappa(\mathbf{n}, \nu) I_{\text{diff}}(\mathbf{n}, \nu) - \eta_{\text{diff}}(\mathbf{n}, \nu)] \end{aligned} \quad (9)$$

$$\begin{aligned} c\mathbf{G} &= \int_0^\infty d\nu \int d\Omega [\kappa(\mathbf{n}, \nu) I_{\text{dir}}(\mathbf{n}, \nu) - \eta_{\text{dir}}(\mathbf{n}, \nu)] \mathbf{n} \\ &\quad + \int_0^\infty d\nu \int d\Omega [\kappa(\mathbf{n}, \nu) I_{\text{diff}}(\mathbf{n}, \nu) - \eta_{\text{diff}}(\mathbf{n}, \nu)] \mathbf{n} \end{aligned} \quad (10)$$

169 where $\eta_{\text{dir}}(\mathbf{n}, \nu)$ and $\eta_{\text{diff}}(\mathbf{n}, \nu)$ describes the emission due to point sources
 170 and the fluid, respectively.

¹Note that this limits our method to non-relativistic problems, where we can neglect the effects of relativistic beaming of the source radiation fields.

171 This decomposition allows the following general approach to a hybrid
172 scheme: (1) use a long characteristics method to solve for I_{dir} , (2) use a
173 moment method to solve for I_{diff} , (3) add the two components to obtain the
174 radiation four-force density ($G^0, \mathbf{G}$), (4) update the hydrodynamic state using
175 the radiation four-force density. As a further benefit to this approach, we note
176 that there is no requirement that steps (1) and (2) use the same frequency
177 resolution, since $(G^0, \mathbf{G})$ depends only on an integral over frequency. It
178 is relatively straightforward to bin the intensity from the point sources by
179 frequency with a ray tracer since each ray can be approximated by an array
180 of intensities, while using a lower frequency resolution in the (generally more
181 expensive) moment method. This is ideal for point sources such as stars
182 which have color temperatures much higher than the absorbing medium.

183 3. The *HARM*² Algorithm

184 In this section we describe the *HARM*² algorithm. We have implemented
185 this algorithm in the ORION astrophysical adaptive mesh refinement (AMR)
186 code [4, 22–24] and we use this implementation for all the algorithm tests
187 described below. ORION uses grid-based adaptivity [25, 26] with individual
188 time steps for each level, and the *HARM*² algorithm can be applied to any
189 AMR code following this design. Variable definitions from this section are
190 defined in table 1. ORION uses the FLD approximation for its moment
191 method [4], and we will use this for all tests below, but *HARM*² is equally
192 compatible with any other moment method.

193 3.1. Update Cycle

194 Consider an adaptive mesh covering some computational domain of in-
195 terest. The mesh consists of levels with different cells sizes, with $l = 0$
196 denoting the coarsest level and $l = l_{\text{max}}$ the finest. Each level, in turn, is
197 made up of a union of rectangular grids, each with the same cell size. In
198 a distributed-memory parallel computation, different grids may be stored in
199 memory on different processors or nodes. The grids on a given level need not
200 be contiguous, but they are required to be non-overlapping, and the grids
201 are properly nested such that a cell of a level l grid may have as its neighbor
202 the domain edge, another level l cell, or a cell of level $l - 1$ or $l + 1$, but not
203 a cell of any other level. Point sources are only placed at locations covered
204 by a grid of level l_{max} . Each level advances on a time step dt_l , ordered such
205 that $dt_l \geq dt_{l+1}$, and so that, after some number of time steps on level $l + 1$,

the simulation time t_{l+1} on that level will be equal to the time t_l on the next coarsest level. That is, we require that, a level $l + 1$ syncs up in time with the next coarsest level l . In all the tests we perform with ORION the time steps obey $dt_{l+1} = dt_l/2$, and synchronization occurs every 2 fine time steps, but this is not required by *HARM*².

211

212 Given this setup, our algorithm is as follows:

- 213 1. Operator split the direct and diffuse components of the radiation field:
 - 214 (a) if l equals 0 **or** $t_{\text{start},l}$ is greater than $t_{\text{start},l-1}$, where $t_{\text{start},i}$ is the
 215 current time on level i , then
 - 216 i. Loop over point sources and inject rays onto grids that belong
 217 to level l_{max} where they are located.
 - 218 ii. Advance rays across grids on level l_{max} and all coarser grids
 219 that the rays interact with, store the rates at which radiative
 220 energy and momentum, dE/dt and $d\mathbf{p}/dt$, are absorbed by
 221 the gas (Section 3.2).
 - 222 iii. Restrict dE/dt and $d\mathbf{p}/dt$ from finer level grids down to level
 223 l .
 - 224 (b) Add $(dE/dt) dt_l$ and $(d\mathbf{p}/dt) dt_l$ to the gas energy and momenta,
 225 respectively.
 - 226 (c) Update the diffuse radiation field with a moment method.
- 227 2. Apply hydrodynamics update to all cells on level l .
- 228 3. Update point sources if $l = l_{\text{max}}$.

229 For the pattern of time steps used by ORION, whereby there are 2 fine time
 230 steps per coarse time step, this method results in $2^{l_{\text{max}}}$ ray trace updates per
 231 update on the coarsest level. Note that, because we only perform a ray trace
 232 if $t_{\text{start},l} > t_{\text{start},l-1}$, we do not perform any redundant ray tracing steps. In
 233 other words, we perform the ray trace at a given time only if we have not
 234 already performed it at that time.

235 3.2. Direct Radiation Field: Adaptive Ray Trace

236 We now describe the adaptive ray tracing procedure that forms step 1a(ii)
 237 of the algorithm above. Consider a single point source with a specific lumi-
 238 nosity given by L_ν and luminosity given by $L_\star = \int_0^\infty L_\nu d\nu$. The general-
 239 ization to multiple point sources is trivial. We discretize the point source
 240 spectrum in frequency into N_ν frequency bins, with the i th bin covering a

241 range in frequency $(\nu_{i-1/2}, \nu_{i+1/2})$. Let $L_i = \int_{\nu_{i-1/2}}^{\nu_{i+1/2}} L_\nu d\nu$ be the luminosity
 242 of the point source integrated over the i th frequency such that $\sum L_i = L_\star$.
 243 We generally expect that L_i will be the energy radiated per unit time in a
 244 given frequency bin, but the algorithm is identical if we instead take L_i to
 245 be a photon luminosity, measured in photons per unit time.

246 We wish to solve the transfer equation along rays that end at this source.
 247 Along a ray characterized by a direction $\mathbf{n}$ and a solid angle Ω_{ray} that it sub-
 248 tends, the propagation of the radiation is described by the time-independent
 249 transfer equation (i.e., Equation (6)), with the emission term η set to zero
 250 because we are taking the direct radiation field to have zero emissivity except
 251 at the point sources. Multiplying both sides of this equation by $4\pi r^2/\Omega_{\text{ray}}$,
 252 we obtain an integrated form of the transfer equation

$$\frac{\partial L_{\text{ray},i}}{\partial r} = -\kappa_i L_{\text{ray},i}, \quad (11)$$

253 where $L_{\text{ray},i}(r)$ is the luminosity along the ray at a distance r from the point
 254 source and κ_i is the total absorption opacity for the i th frequency bin in units
 255 of cm^{-1} . This equation is subject to the boundary condition $L_{\text{ray},i}(0) =$
 256 L_i/N_{pix} , where $N_{\text{pix}} = 4\pi/\Omega_{\text{ray}}$. We solve this equation by discretizing it
 257 along the line segments defined by the intersection of the ray with the cells
 258 of the computational mesh. Specifically, when a ray with luminosity $L_{\text{ray},i}$
 259 passes through a cell along a segment of length dl , the optical depth of the
 260 segment is $\tau_i = \kappa_i dl$, and the luminosity of the ray decreases by an amount

$$dL_{\text{ray},i} = L_{\text{ray},i} (1 - e^{-\tau_i}). \quad (12)$$

261 In the process, the cell absorbs an amount of energy and momentum at a
 262 rate

$$\frac{dE}{dt} = \sum_{i=1}^{N_\nu} dL_{\text{ray},i} \quad (13)$$

$$\frac{d\mathbf{p}}{dt} = \sum_{i=1}^{N_\nu} \frac{dL_{\text{ray},i}}{c} \mathbf{n}. \quad (14)$$

263 The total absorption rate for each cell is simply the sum of dE/dt and $d\mathbf{p}/dt$
 264 over all rays from all point sources that pass through it. When computing
 265 the line segments dl , we only consider grids that are not masked by any finer

266 grid. That is, when solving Equation (11), we only ever consider the most
 267 highly spatially resolved data at any given position.

268 We choose the directions $\mathbf{n}$ and solid angles Ω_{ray} using the angular dis-
 269 cretization introduced by Abel and Wandelt [27] and Wise and Abel [18].
 270 In this approach, $\mathbf{n}$ and Ω_{ray} are chosen using the Hierarchical Equal Area
 271 isoLatitude Pixelization of the sphere (HEALPix) scheme [28], which divides
 272 the surface area of a sphere into equal area pixels that can be further sub-
 273 divided into four equal-area sub-pixels. There are $N_{\text{pix}}(0) = 12$ pixels at the
 274 coarsest HEALPix level, and there are $N_{\text{pix}}(j) = 12 \times 4^j$ pixels on HEALPix
 275 level j ; note that the HEALPix level j and the AMR grid level l are distinct
 276 and in general are not the same. The scheme is adaptive in that, as we trace
 277 rays away from point sources, we subdivide them as needed to ensure that
 278 cells are adequately resolved. Specifically, we divide a ray into 4 sub-rays if
 279 it satisfies the condition

$$\frac{\Omega_{\text{cell}}}{\Omega_{\text{ray}}} = \frac{N_{\text{pix}}(j)}{4\pi} \left(\frac{\Delta x}{r} \right)^2 < \Phi_c, \quad (15)$$

280 where $\Omega_{\text{cell}} = (\Delta x/r)^2$ is the solid angle subtended by a cell of linear size
 281 Δx at a distance r from the point source. The quantity Φ_c is the minimum
 282 number of rays required to go through each cell, which we usually set to
 283 4 following the resolution tests of Krumholz et al. [29] and Wise and Abel
 284 [18] but, in general, the exact value for Φ_c is problem-dependent. **The initial**
 285 **luminosity per ray for frequency bin i is $L_{\text{ray},i,j_0} = L_i/N_{\text{pix}}(j_0)$ where j_0 is the**
 286 **initial healpix level.** When a ray splits, we solve the transfer equation along
 287 the sub-rays using the boundary condition $L_{\text{ray},i,j+1}(R) = L_{\text{ray},i,j}(R)/4$, where
 288 $L_{\text{ray},i,j}(R)$ is the luminosity of the ray at frequency bin i on HEALPix level
 289 j . As proposed by Krumholz et al. [29], we randomly rotate the orientation
 290 of the rays every time they are cast to minimize errors due to discretization
 291 in angle. Finally, we terminate the ray trace when either a ray exits the
 292 computational domain, or when **$L_{\text{ray},j}(r) < 0.001 L_{\text{ray},j}(0)$ where $L_{\text{ray},j}(0) =$**
 293 **$\sum_i L_i / (12 \times 4^{j-j_0})$, i.e., when 99.9% of the energy originally assigned to that**
 294 **ray on ray level j has been absorbed.**

295 3.3. Parallelization

296 Thus far the algorithm we have described is substantially identical to
 297 that of Wise and Abel [18]. However, we adopt a very different, and much
 298 more efficient strategy to parallelize this procedure. The primary challenge to

299 parallelizing this algorithm is avoiding serial bottlenecks. The grids through
 300 which the rays must be traced may be distributed across any number of
 301 processors, and solution of Equation (11) is an intrinsically serial process
 302 because the rate of change of the energy and momentum in any cell due
 303 to radiation arriving along a particular ray depends upon the properties of
 304 all cells that lie between the point source and the cell in question. Since
 305 the numbers and positions of point sources and computational grids in the
 306 AMR structure, and their distribution in memory, are not known *a priori*,
 307 minimizing bottlenecks requires an opportunistic approach: rays should be
 308 able to be processed in arbitrary order, with each processor performing ray
 309 tracing given the data available to it, and waiting for other processors only
 310 when no useful work can be done. To this end, communication must be
 311 non-blocking and asynchronous. At the same time, each process must be
 312 able to determine when the entire ray trace has been completed, so that
 313 it can proceed to the remainder of the update cycle (the moment method,
 314 hydrodynamics, etc.), and this determination must be robust against race
 315 conditions.

316 Recall that we consider the tracing of a particular ray done when it ei-
 317 ther exits the computational domain or when 99.9% of its energy has been
 318 absorbed. To handle the problem of determining when the algorithm should
 319 terminate without relying on blocking communication, we *pretend* we know
 320 how many rays *could* be created by computing a maximum number of rays
 321 to be used as a counter:

$$N_{\max} = N_{\text{src}} \times 12 \times 4^{j_{\max}} \quad (16)$$

322 where N_{src} is the number of sources and $j_{\max}$ is the maximum HEALPix
 323 level we allow; we set this to 20 in all of our tests. Our algorithm involves
 324 accounting for “all rays” that are destroyed on each processor by computing

$$N_{\text{destroyed},k} = \sum_{N_{\text{ray},k}} 4^{j_{\max}-j} \quad (17)$$

325 where k is the processor number, j is the HEALPix level of the ray that is
 326 deleted due to absorption or leaving the computational domain, and $N_{\text{ray},k}$
 327 is the total number of rays that have been deleted on processor k . This infor-
 328 mation is communicated to all other processors. Once the total destroyed,

$$N_{\text{destroyed}} = \sum_{N_{\text{CPU}}} N_{\text{destroyed},k}, \quad (18)$$

329 on each processor equals $N_{\max}$ the ray trace is complete.

330 With this bookkeeping method understood, we present our message pass-
331 ing scheme as algorithm 1. Note that this algorithm requires version 3.0 or
332 later of the Message Passing Interface (MPI) standard, because only that
333 version supports **some of** the non-blocking communications we require (e.g.,
334 **MPI_Iprobe** see Algorithm 1). A detailed description of our parallelization
335 strategy follows. Every processor has 4 flags that can change: (i) ALLDONE
336 which is initially set to **false** and will be set to **true** once all rays have
337 been destroyed, (ii) WORKREMAINS which is set to **true** if rays exist on this
338 processor and **false** otherwise, (iii) DATARECV which is set to **true** when
339 the processor can receive rays from other processor(s), and (iv) COUNTRECV
340 which is set to **true** when the processor can receive $N_{\text{destroyed},k}$, the number of
341 rays destroyed from other processor(s). Every processor also has a counter,
342 LOOPITER, which tracks the number of times the parallelization algorithm is
343 iterated over.

344 For each processor, our algorithm is as follows:

- 345 1. Inject rays from point sources to grids.
- 346 2. Compute $N_{\max}$ (Equation (16)).
- 347 3. Set ALLDONE to **false**, set LOOPITER = 0, and enter outer **while**(not
348 ALLDONE) loop.
 - 349 (a) Set WORKREMAINS to **true** if rays exist on grids that belong to
350 this processor, otherwise set WORKREMAINS to **false**.
 - 351 (b) If WORKREMAINS is **true** enter **while**(WORKREMAINS) loop.
 - 352 i. Loop over all grids that belong to this processor and **for each**
353 grid advance all rays that belong to that grid until they (i)
354 leave the grid and need to be moved to another grid, (ii)
355 become extinct, or (iii) leave the domain. Every time (ii) or
356 (iii) occurs for a ray we delete the ray and increase the value
357 of $N_{\text{destroyed},k}$ on this processor per Equation (17). If (i) occurs
358 we determine the new grid and processor for that ray.
 - 359 ii. Set WORKREMAINS to **false**.
 - 360 iii. Loop over rays that must be transferred to other grids and
361 check if they belong to a grid on this processor or another
362 processor. If the former is true place the ray on the correct
363 grid and set WORKREMAINS to **true**, otherwise the ray must
364 be transferred to another processor so we place the ray in a

365 linked list, **outgoing_ray_list**, to be communicated to the
366 other processors.

367 iv. **Repeat** until all rays have been either destroyed and/or placed
368 into **outgoing_ray_list**.

369 (c) Loop over rays in **outgoing_ray_list**. Place rays from **outgo-**
370 **ing_ray_list** into a contiguous array for MPI communication (one
371 array per receiving processor) and perform a non-blocking **MPI_Isend**
372 for each array to send it to the appropriate processor.

373 (d) Set **DATARECV** to **true** and if (**LOOPITER modulo** $N_{\text{CPU}} == 0$)²
374 then enter the **while**(**DATARECV**) loop to begin receiving rays from
375 other processors.

376 i. Probe other processors with the non-blocking **MPI_Iprobe**
377 function to see if rays need to be received, if this is true set
378 **DATARECV** to **true** otherwise set it to **false**.

379 ii. If **DATARECV** is **true** then receive rays on this processor with
380 **MPI_Recv**. Place the incoming rays onto the new grids they
381 belong to.

382 iii. **Repeat** until **MPI_Iprobe** returns false.

383 (e) If this processor sent rays to another processor then perform non-
384 blocking **MPI_Testsome** to test for some given ray send requests
385 to complete.

386 (f) If **WORKREMAINS** is false then perform non-blocking **MPI_Testsome**
387 to test for some of the $N_{\text{destroyed},k}$ send requests to complete.

388 (g) Set **COUNTRECV** to **true** and if (**LOOPITER modulo** $N_{\text{CPU}} == 0$)
389 then enter the **while**(**COUNTRECV**) loop to begin receiving rays
390 from other processors.

391 i. Probe other processors with the non-blocking **MPI_Iprobe**
392 function to see if the value of $N_{\text{destroyed},k}$ on those processors
393 needs to be received. If this is true set **COUNTRECV** to **true**,
394 otherwise set it to **false**.

395 ii. Receive $N_{\text{destroyed},k}$ from processor k on this processor with
396 **MPI_Recv** if **COUNTRECV** is **true** and assign to the k th
397 element of an array **rayDestProc** containing N_{CPU} elements.

398 iii. **Repeat** until **MPI_Iprobe** returns false.

²This requirement reduces the time spent on performing MPI_Iprobes which becomes more expensive as N_{procs} increases.

- 399 (h) If $N_{\text{destroyed},k}$ on this processor has increased in the outer loop
400 iteration then send this number with a non-blocking **MPI_Isend**
401 to all other processors.
- 402 (i) Compute sum of **rayDestProc**. If this value equals N_{max} termi-
403 nate the outer loop, else repeat outer loop and increment LOOP-
404 ITER by 1.

405 4. Validation Tests

406 In this section we demonstrate the accuracy of our adaptive ray tracing
407 algorithm, the absorption of the direct radiation field from point sources sup-
408 plied by our adaptive ray tracing scheme, and our hybrid radiative transfer
409 algorithm by performing four tests. In the first test, we turn off absorption
410 of the radiation field to trace the radiative flux from a point source located
411 at the center to demonstrate that our method recovers the correct r^{-2} fall-off
412 of the flux. The second test focuses on the coupling of the hydrodynamics
413 with the adaptive ray trace. We also demonstrate that *HARM*² can handle
414 multiple sources with the third and fourth tests. Our third test demon-
415 strates that our hybrid radiative transfer method can properly cast shadows
416 and transfer radiative energy to the gas by illuminating a dense clump in a
417 low-density medium with two point sources when the hydrodynamics is ne-
418 glected. Finally, our fourth test follows a similar setup as the third test but
419 demonstrates how the hydrodynamics are coupled to *HARM*². We set $\Phi_c = 4$
420 for all validation tests in this section, and we use only a single frequency bin.
421 For the first two tests we disable the moment method part of *HARM*², so
422 that we can focus on the ray-tracing part of the algorithm. Further valida-
423 tion tests for the moment method have been presented in Krumholz et al.
424 [4].

425 4.1. Flux Test

426 To demonstrate the accuracy of our adaptive ray trace and its ability to
427 maintain spherical symmetry, we place a point source of luminosity $L_\star$ at
428 the origin of a cubical domain extending from -1 to $+1$ pc in all directions.
429 We set the opacity in all cells to zero and take $L_\star = 10^6 L_\odot$ where $L_\odot =$
430 $3.84 \times 10^{33} \text{erg s}^{-1}$ is the luminosity of the Sun. We use a base resolution of
431 128^3 and two levels of refinement. We refine cells that are located within 16
432 cells from the star.

Data: Rays and destroyed ray counts

```
compute maxRays;
all_done = False;
do
  do
    foreach grid that belongs to this processor do
      advance all rays, return number "destroyed" and add to
      destroyedCount;
    end
    check grids to see if work_remains;
  while work_remains;
  Non-blocking MPI_Isend rays to other processors;
do
  Non-blocking MPI_Iprobe other processors for rays;
  if MPI_Iprobe returns true then
    Blocking MPI_Recv(rays);
  end
  while MPI_Iprobe for rays returns true;
  Non-blocking MPI_Testsome rays MPI_Isend requests;
  Non-blocking MPI_Testsome destroyed counts MPI_Isend requests;
do
  Non-blocking MPI_Iprobe other processors for destroyed
  counts;
  if MPI_Iprobe returns true then
    Blocking MPI_Recv(processor destroyed counts);
  end
  while MPI_Iprobe for destroyed counts returns true;
  if destroyedCount greater than previous destroyedCount then
    Non-blocking MPI_Isend(destroyedCount to all processors);
  end
  if no work_Remains and Ray Send Requests == 0 and Destroyed
  Count Send Requests == 0 and sum(destroyedCount) ==
  maxRays then
    all_done = True;
  end
while not all_done;
```

Algorithm 1: Asynchronous parallelization algorithm developed for the communication of rays to other processors.

Table 1: Variable definitions used in Section 3.

Variable	Description
dl	Path length of ray across cell
dE/dt	Energy absorbed by fluid in cell from direct radiation
$d\mathbf{p}/dt$	Momenta absorbed by fluid in cell from direct radiation
$dL_{\text{ray},i}$	Absorbed luminosity from ray for the i th frequency bin
dt_i	Time step on level i
j	HEALPix ray level
j_0	Initial HEALPix ray level
j_{max}	“Maximum” ray level for adaptive ray trace
κ_i	total absorption opacity for the i th frequency bin
l	AMR Level
l_{max}	Maximum AMR level
L_ν	Specific luminosity of point source
L_i	Luminosity of point source integrated over i th frequency bin
$L_{\text{ray},i}$	Luminosity in i th frequency bin along ray
$\mathbf{n}$	Normal direction of ray
N_{CPU}	Number of processors
$N_{\text{destroyed},k}$	Number of rays “destroyed” on processor k
$N_{\text{destroyed}}$	Number of rays “destroyed” on all processors
N_{max}	Maximum number of rays used as a counter
N_ν	Number of frequency bins used for adaptive ray trace
$N_{\text{pix}}(j)$	Number of HEALPix pixels on level j
$N_{\text{ray},k}$	Number of rays deleted on processor k
N_{src}	Number of point sources
Δx	linear size of cell
Φ_c	Angular resolution of ray trace [rays/cell]
Ω_{cell}	Solid Angle subtended by a cell
Ω_{ray}	Solid angle associated with ray
τ	Optical depth of cell

433 In this setup, the total energy contained in the region that is a distance
 434 $< r$ from the origin should be exactly $E_{\text{exact}}(< r) = L_\star r/c$, where the quantity
 435 r/c is simply the light-crossing time of the distance r . We can compare this
 436 to the total energy in this region returned by our code, which we can compute
 437 by noting that the radiation energy density of a given cell that is traversed

438 by a series of rays is

$$U_{\text{rad}} = \sum_i \frac{L_{\text{ray},i}}{dV} \frac{dl_i}{c} \quad (19)$$

439 where $L_{\text{ray},i}$ is the luminosity for ray i , dV is the cell volume, and dl_i is the
 440 path length of the ray through the cell. The total energy contained within
 441 cells whose distance r_j from the origin is $< r$ is then

$$E_{\text{num}}(< r) = \sum_{\text{cells}, r_j < r} U_{\text{rad},j} dV \quad (20)$$

442 where $U_{\text{rad},j}$ is the total radiation energy density summed over all cells with
 443 distance $< r$ from the origin. Perfect agreement would consist of $E_{\text{num}}(< r) = E_{\text{exact}}(< r)$.
 444

445 Our results are shown in Figure 1. The left panel shows the line-of-sight
 446 projected radiation energy density of the point source radiation field (i.e.,
 447 Equation (19)) integrated over the line of sight, which drops off as r^{-2} as
 448 expected. The two right panels compare $E_{\text{num}}(< r)$ and $E_{\text{exact}}(< r)$. We find
 449 that the difference between the numerical and exact results is always $< 5\%$,
 450 with the maximum error occurring close to the source where the resolution is
 451 poor. This error is expected because of the fact that we are using Cartesian
 452 rather than spherical grids.

453 4.2. Radiation-Pressure-Dominated HII Region

454 Next we perform a test to illustrate the coupling of the radiative transfer
 455 from the adaptive ray trace with the hydrodynamics, based on a similarity
 456 solution obtained by Krumholz and Matzner [30]. We consider an initially-
 457 uniform, cold gas with an isothermal equation of state. At time $t = 0$ a
 458 point source of radiation with luminosity L_* turns on and begins depositing
 459 momentum in the gas. We consider material that has a very high opacity
 460 to photons coming directly from the point source, but a very low opacity to
 461 any re-emitted photons. A real-world example of this would be interstellar
 462 dust absorbing ultraviolet photons from a star, and then re-emitting them
 463 as infrared light, to which the dust is essentially transparent.

464 Because the opacity is high, all of the radiation from the point source is
 465 absorbed in an extremely thin layer, but then escapes immediately. Thus the
 466 point source deposits radial momentum into the gas at a rate $dp/dt = L_*/c$.
 467 After a short time the material around the point source will have been swept

Figure 1: Performance test for the adaptive ray trace. Left panel: Projection plot of the stellar radiation density for a source with luminosity $10^6 L_\odot$. The source flux falls off as $F(r) \propto r^{-2}$ as expected. Right panels: Comparison of the numerical and analytical results of the energy enclosed within radius r (i.e, $E_{\text{num}}(<r)$) for the same source. Top panel: The pink dashed line shows the exact analytical solution and the teal solid line is the numerical result from the adaptive ray trace. The bottom panel shows the residuals from the exact and numerical solutions.

468 into a thin shell of radius r_{sh} and mass $M_{\text{sh}} = 4\pi r_{\text{sh}}^3 \rho_0 / 3$, where ρ_0 is the
 469 initial density. The shell obeys an equation of motion

$$\frac{d}{dt} (M_{\text{sh}} \dot{r}_{\text{sh}}) = \frac{L_\star}{c}. \quad (21)$$

470 This equation admits a similarly solution given by

$$r_{\text{sh}}(t) = 1.15 \left(\frac{n_0}{10^5 \text{ cm}^{-3}} \right)^{-1/4} \left(\frac{L_\star}{10^6 L_\odot} \right)^{1/4} \left(\frac{t}{\text{Myr}} \right)^{1/2} \text{ pc} \quad (22)$$

471 where $n_0 = \rho_0 / \mu m_p$ is the number density, μ is the mean molecular weight
 472 which we set to 2.33 for molecular hydrogen and helium mixed in the usual
 473 cosmic ratio, and m_p is the proton mass.

474 To test the ability of our code to reproduce this solution, we consider a
 475 domain with a width of 1 pc, a uniform number density of $n_0 = 10^5 \text{ cm}^{-3}$
 476 ($\rho_0 = 3.89 \times 10^{-19} \text{ g cm}^{-3}$), and a point source of luminosity $L_\star = 10^6 L_\odot$ at
 477 the origin. We set the specific opacity to $\kappa/\rho = 10^6 \text{ cm}^2 \text{ g}^{-1}$. We perform 3
 478 tests on non-adaptive grids with varying resolution (64^3 , 128^3 , and 256^3) to
 479 explore how the accuracy of the adaptive ray trace depends on resolution.

480 Figure 2 shows a snapshot of the simulation results at $t = 0.1 \text{ Myr}$. The
 481 top panels show the density slices and the bottom panels show the deposition
 482 rate of the stellar radiation energy density. We run the simulations to $t = 0.35$
 483 Myr, but at later times we develop the carbuncle instability which distorts
 484 the shape of shock waves that move along grid directions [31, 32]. One can
 485 eliminate this instability by implementing extra dissipation in grid-aligned
 486 flows [32], but since real applications are never perfectly grid-aligned, we
 487 have not done so here.

488 In Figure 3 we show how the radius of the shell in our simulation compares
 489 to the analytic similarity solution. We define the radius of the shell to be the
 490 density weighted average distance from the origin for cells where the density
 491 exceeds $1.5 \rho_0$:

$$R_{\text{sh}} = \frac{\sum_{\text{cells}, \rho_j > 1.5 \rho_0} \rho_j r_j}{\sum_{\text{cells}, \rho_j > 1.5 \rho_0} \rho_j}. \quad (23)$$

492 The top panel shows the shell radius as a function of time for each resolution
 493 and for the analytic solution, while the bottom panel shows the residuals. As
 494 in Figure 1, the residuals are largest at early times when the shell is poorly
 495 resolved, but the agreement becomes excellent at later times. The accuracy
 496 of the solution also improves with increasing resolution, as expected.

497 4.3. *Hybrid Radiation Tests with Multiple Sources*

498 To demonstrate that our hybrid radiation algorithm can properly cast
 499 shadows we perform two validation tests in which we have two point sources
 500 irradiate a dense clump of material. These tests are similar to the tests
 501 presented in Rijkhorst et al. [11], Jiang et al. [9], and Klassen et al. [17]. In
 502 our setup, we have a dense clump of material with radius 267 AU at the center
 503 of a $(2000 \text{ AU})^3$ computational domain. We take the clump density to be
 504 $\rho_c = 3.89 \times 10^{-17} \text{ g cm}^{-3}$ with uniform temperature of 20 K. It is surrounded

Figure 2: Results from the radiation-pressure-dominated H II region test to demonstrate the performance of the adaptive ray trace coupled to the hydrodynamics. Top (bottom) panels show slice plots of the gas density (rate of absorbed radiation energy density per unit time) for our radiation dominated sphere test at three different uni-grid resolutions (64^3 , 128^3 , and 256^3) taken at $t = 0.1$ Myr. As the bottom panels show, the direct radiation is absorbed only by the dense shell due to the high specific opacity, $\kappa = 10^6$ cm²/g, used.

by a low density medium with density $\rho_a = 3.89 \times 10^{-20}$ g cm⁻³. The clump is irradiated by two point sources, each with luminosity $L = 1 L_\odot$. The two point sources are both located 368 AU from the edge of the spherical clump, and they are placed 90° apart from one another, so that they irradiate the clump from two angles.

We perform two tests to show that our new hybrid radiative transfer method can properly cast shadows and affect the hydrodynamics. The first test, presented in Section 4.3.1, assumes that the ambient medium is at the

Figure 3: Shell position (top panel) and residuals (bottom panel) as compared to the analytical solution (i.e., eqn 22) for our radiation dominated sphere test at three different uni-grid resolutions (64^3 , 128^3 , and 256^3). The largest deviations from the analytical solution occur at early times when the shell is located close to the source but the numerical result follows the analytical solution better as the shell expands.

513 same temperature as the clump (20 K) so that we can follow the radiative
514 heating of the gas. For this test we disable the hydrodynamics. In the second
515 test, we set the temperature of the ambient medium to be 20×10^3 K so that
516 the clump is initially in pressure balance with the low-density medium. In
517 this test we enable the hydrodynamics. For both tests we use one frequency
518 bin for the absorption of the direct radiation field and assume that the gas
519 opacity is $\kappa = 64 \text{ cm}^2/\text{g}$ following the frequency dependent dust opacities in
520 Weingartner and Draine [33] (their $R_v = 5.5$ extinction curve) corresponding
521 to a frequency of $\nu = kT_{\text{eff}}/h = 1.21 \times 10^{14} \text{ Hz}$, where we have assumed
522 $T_{\text{eff}} = 5800 \text{ K}$ (i.e, the peak color temperature of the Sun). Our moment
523 method uses the gray opacities from Semenov et al. [34].

524 4.3.1. Irradiation of a Dense Clump by two Point Sources with HARM²

525 We perform our first hybrid radiation test on a uniform $(256)^3$ grid, dis-

ablating the hydrodynamics but allowing for absorption of the radiative energy by the gas. We run this test for $t = 2 \times 10^6$ s for two cases. In the first case, we only include absorption by the direct radiation field, i.e., we do not use the moment method, and in the second case we use the *HARM*² algorithm. Figure 4 shows the irradiation from the direct radiation field from point sources (left panel), the irradiation from the dust-reprocessed radiation field (middle panel), and the combined irradiation from the point sources and dusty gas (right panel). The white region for the source irradiation demonstrates that our adaptive ray tracing method can properly cast shadows. The dusty gas absorbs energy from the direct radiation field, heats up, and re-emits as shown in the middle panel of Figure 4.

Figure 5 compares the temperature distribution for our run using the full *HARM*² algorithm (right panel) to a case where we use only the ray trace, and omit the moment method (left panel). The temperature is highest near the point sources due to the larger radiative flux, thus leading to a larger absorption rate by the low-density gas. As the irradiation decreases farther away from the source, the gas attains lower temperatures due to the decreased heating rate of the absorbing medium. The right panel of Figure 5 shows that the re-emission by dust leads to lower temperatures and smoothes the radiation field. Note the almost discontinuous change in the gas temperature in the vicinity of the point sources in the full *HARM*² case. This is a result of a sharp drop in the opacity used for the moment method calculation, which causes a corresponding drop in the emissivity of the gas and thus a rise in its temperature. The physical origin of this discontinuity is sublimation of dust grains at temperatures above ~ 1000 K (e.g., see Figure 4 of Semenov et al. [34]), but from a numerical standpoint this is less important than the capability that this demonstrates. Our code can properly capture situations where the opacity that should be used for the direct radiation is very different than the opacity that is relevant for the reprocessed radiation field, for example because the direct and reprocessed radiation fields have very different color temperatures.

4.3.2. Irradiation of a Dense Clump by two Point Sources with *HARM*² and Hydrodynamics

Our fourth validation test demonstrates our *HARM*² algorithm operating in the context of a full radiation-hydrodynamic problem. We set the initial conditions to be exactly the same as the previous test except for the ambient gas temperature, which we set to 20×10^3 K so that the clump is initially

Figure 4: Irradiation of a dense clump by two point sources in an homogenous medium. The left panel shows the irradiation by the point sources’ direct radiation fields while the middle panel shows the dust-reprocessed irradiation. The right panel shows the total irradiation by the stellar sources and dusty gas. The white region in the left panel, which shows where the direct irradiation is zero, demonstrates that our method can properly cast shadows.

Figure 5: Same as the left and right panels of Figure 4, except showing the gas temperature where we only consider heating by the direct radiation field by including the adaptive ray trace (left panel) and where we consider heating by the direct radiation field and cooling by the dusty gas using *HARM*² (right panel).

563 in pressure balance with the ambient medium. Additionally, we enable the
 564 hydrodynamics and we also set the ambient medium to be transparent to
 565 the direct and diffuse radiation fields. Our base grid is $(128)^3$ and we refine
 566 the entire clump to one greater level of refinement (i.e., the clump has $(256)^3$
 567 refinement).

568 Figure 6 shows the evolution of the density, temperature, and clump ve-
 569 locity of the simulation. This Figure demonstrates that the absorption of

Figure 6: Snapshots of the radiation-hydrodynamical time evolution of a dense clump in a low-density transparent medium that is irradiated by two point sources using our *HARM*² radiative transfer method. Different columns show different times, and different rows show different quantities: density (top), temperature (middle), and velocity magnitude (bottom). The heating and momentum deposition by the point sources' direct radiation fields cause the edges of the clump to expand. The gray stars denote the location of the radiating point sources.

570 the point sources' direct radiation field heats up the sides of the clump that
571 are closest to the sources, causing the edges to expand. The momentum
572 imparted by the direct radiation field to the clump also causes it to expand
573 non-uniformly. We note that as the cloud becomes over-pressurized it ex-
574 pands into the ambient medium yielding large velocities along the edges of
575 the cloud. As the simulation evolves the cloud is slowly disrupted leading to
576 low density material expanding towards the point sources. The expansion of
577 this low density material leads to instabilities as the clump material mixes
578 with the low density gas. At late times the expansion of the low-density
579 clump material closest to the point sources slows as the direct radiation field
580 imparts momentum to the gas.

581 5. Performance Tests

582 It is important for our code to scale well with number of processors,
 583 especially for large simulations. Scaling tests demonstrate the efficiency of a
 584 parallel application when increasing the number of processors. In this section
 585 we present both weak and strong scaling tests to demonstrate the parallel
 586 performance of our adaptive ray trace algorithm. We also perform a strong
 587 scaling test for our hybrid radiation algorithm in section 5.2.3 in an AMR
 588 simulation to demonstrate the scaling capability of *HARM*² for a demanding,
 589 research application. For all tests, we set $\Phi_c = 4$ and also have the initial
 590 ray level set to 4 so that 3072 rays are initialized at the beginning of each
 591 ray trace step. All of the following performance tests were run on the Sandy
 592 Bridge nodes on the NASA NAS machine Pleiades.

593 5.1. Weak Scaling

594 Weak scaling tests demonstrate how well a parallel code scales with the
 595 number of processors while the workload assigned to each processor remains
 596 the same. For this purpose, we perform a weak scaling test on non-adaptive
 597 grids where each processor has one 32^3 grid and one radiating sink particle.
 598 Each 32^3 grid represents a $(1 \text{ pc})^3$ domain with constant gas density, $n =$
 599 10^4 cm^{-3} , with a radiating source placed at the center. The physics modules
 600 we include are the adaptive ray trace and hydrodynamics. We set the opacity
 601 in all cells to zero so that no absorption of the radiation field occurs. To
 602 ensure that each processor performs the same amount of work with the ray
 603 trace, including the propagation of rays and the subsequent communication
 604 of rays to other processors, we terminate rays once they have traveled 0.6
 605 pc from their originating source. This allows for rays to propagate to their
 606 neighboring grids and also enforces that all grids, except the grids along the
 607 domain edges, communicate the same number of rays to their neighbors. In
 608 short, the rays interact with the cells they cross but do not add energy or
 609 momenta to the fluid.

610 Our weak scaling tests were run on $N_{\text{CPU}} = n^3$ processors, where $n=[1,2,\dots,9,10]$,
 611 for 50 time steps per weak scaling test. The weak scaling test results are pre-
 612 sented in Figure 7 and show the total time spent per time step (black solid
 613 line) and the timing of the adaptive ray trace components: ray communi-
 614 cation (gray dotted line), ray trace across cells (pink dashed line), adaptive
 615 ray trace overhead (i.e., locating ray grids, ray splitting, etc. – purple dot-
 616 dashed line), total adaptive ray trace (ray trace and associated overhead

617 – blue dashed line), and the full adaptive ray trace which includes the ray
 618 communication (dot-dashed teal line). We note that a horizontal line denotes
 619 perfect weak scaling.

620 Our timing results show that the tracing of rays across cells has near-
 621 perfect weak scaling for all processor counts and that the adaptive ray trace
 622 overhead exhibits near perfect weak scaling until ~ 216 processors. We also
 623 find that the costs associated with the overhead are more expensive than
 624 ray tracing alone. Finally, our ray communication algorithm is cheaper than
 625 the ray tracing up to ~ 343 processors and cheaper than the costs associated
 626 with the adaptive ray trace overhead up to ~ 729 processors. The ray com-
 627 munication only becomes as expensive as the adaptive ray trace at ~ 1000
 628 processors. This is because our asynchronous communication algorithm, de-
 629 scribed in section 3.3, has a $N_{\text{CPU}}^{0.67}$ dependence. These results confirm that
 630 our communication algorithm is much more scalable and efficient when com-
 631 pared to previous methods. For example, the ray communication timing in
 632 Wise and Abel [18] followed a $N_{\text{CPU}}^{1.5}$ dependence and became the dominant
 633 cost of the ray trace at only ~ 200 CPUs, despite the fact that their weak
 634 scaling test uses 64^3 rather than 32^3 blocks, and thus is significantly less
 635 stringent than ours.

636 5.2. Strong Scaling

637 Strong scaling demonstrates how well the code performs as the number
 638 of processors for a given problem increases while the total workload remains
 639 the same. We perform three tests to demonstrate the strong scalability of
 640 the adaptive ray trace and the $HARM^2$ algorithm in a demanding, research
 641 application. The first two tests are performed on non-adaptive grids. The
 642 first test measures the strong scalability of the adaptive ray trace based on
 643 the number of cells that interact with rays (see section 5.2.1) while the second
 644 test focuses on the strong scalability of the adaptive ray trace as a function
 645 of the number of frequency bins used (see section 5.2.2). The third test
 646 shows the parallel performance of $HARM^2$ in a demanding, research AMR
 647 simulation. In addition to $HARM^2$, this test also includes other physics
 648 modules in ORION such as hydrodynamics, self-gravity, and sub-grid star
 649 particles. Each test and their results are described below.

650 5.2.1. Uni-grid Ray Trace Test with Varying Termination Lengths

651 We first use a setup similar to a single instance of our weak scaling test: a
 652 $(1 \text{ pc})^3$ domain with a single point source placed at its center. The resolution

Figure 7: Weak scaling test with one 32^3 block per process. Each block is $(1 \text{ pc})^3$ and contains one radiating source at its center. Rays are terminated after they have traveled 0.6 pc from the source to ensure communication of rays to neighboring grids. Weak scaling results are shown for the ray communication (gray dotted line), ray tracing across cells (pink dashed line), overhead associated with the adaptive ray trace (purple dot-dashed line), adaptive ray trace excluding the communication of rays (blue line), the total cost of the adaptive ray trace including parallel communication of rays (teal dot-dashed line), and the total time spent on the hydrodynamics and adaptive ray tracing (solid black line). Ray communication is cheaper than the adaptive ray trace (ray trace and associated overhead) for $N_{\text{CPU}} \lesssim 1000$ processors. The communication shows a $N_{\text{CPU}}^{0.67}$ dependence.

653 of the computational domain is 256^3 cells and each grid consists of 16^3 cells
654 yielding a total of 4096 grids, with no adaptivity. We perform three sub-
655 tests with this setup in which the rays are destroyed after traveling 0.2 pc
656 from their source, 0.4 pc from their source, or allowed to traverse the entire
657 domain, respectively. These calculations were performed on $N_{\text{CPU}} = 2^n$
658 processors with $n=[2, \dots, 9, 10]$ for 5 time steps per test.

659 Our strong scaling results are presented in Figure 8, which shows the
660 total CPU time per time step, t_{CPU} , for the adaptive ray trace. (Note that,

Table 2: Fitted scaling results from our uni-grid strong scaling tests presented in Figure 8, together with the fraction of the computational volume over which the ray trace is performed. A value of $a = 0$ would imply perfect strong scaling.

D_{Ray}	Ray-interaction Volume	$a (N_{\text{CPU}}^a)$
0.2 pc	0.0335	0.52
0.4 pc	0.2681	0.27
Whole	1	0.084

whereas in Figure 7 we plotted the time per processor, here we plot the total time summed over all processors, so that perfect scaling would again appear as a flat horizontal line.) To better quantify the results, we perform a χ^2 fit of our measured results to the functional dependence $t_{\text{CPU}} \propto N_{\text{CPU}}^a$; perfect strong scaling would be $a = 0$. We report these results in Table 2.

When we allow rays to traverse the entire computational domain, we find near-perfect strong scaling out to 1024 processors: $t_{\text{CPU}} \propto N_{\text{CPU}}^{0.084}$. As we lower the distance that rays propagate, the scaling deteriorates, for the obvious reason that processors which are assigned computational domains that rays do not reach are simply idle because they do not contribute to the ray trace computation. Indeed, we also report the fraction of the computational volume over which rays propagate in Table 2, and it is clear that the scaling is worse when this value is small.

5.2.2. Timing with Varying Frequency Bins

Our adaptive ray trace algorithm allows for an arbitrary number of frequency bins N_ν . Each ray has two arrays that contain N_ν doubles (size 8 bytes) that hold the ray’s initial and current frequency-dependent luminosities, respectively. The choice of N_ν impacts the cost of the computation in two ways: (1) the ray trace operations must loop over all frequency bins when creating rays, advancing them across cells, and checking if they become extinct due to absorption by the fluid; and (2) MPI communication operations depend on the size of the message that is being sent and/or received. Therefore, increasing the number of frequency bins for the adaptive ray trace will lead to an increased cost in the overhead associated with the advancement and communication of rays.

To test the scaling efficiency of the adaptive ray trace as a function of N_ν we ran a series of tests where we vary the number of frequency bins.

Figure 8: Strong scaling test with a 256^3 uni-grid calculation with a radiating point source at the center. We performed two tests where the rays are terminated after they travel 0.2 or 0.4 pc from their source and a third test where the rays transverse the entire domain. Perfect strong scaling would yield a flat line for each test. Our results show that the strong scaling performance improves as the volume that the rays interact with increases and that near-perfect strong scaling is attained when the rays transverse the entire domain.

Our initial setup of our test problem is the same as the strong scaling test discussed in section 5.2.1 in which a radiating source is at the center of a $(1 \text{ pc})^3$ box. We terminate the rays after they have travelled 0.5 pc and perform tests for $N_\nu = (1, 2, 8, 16, 20, 32, 48, 64)$ frequency bins. Our base grid is 256^3 and we ran our scaling tests on 128 processors for 50 time steps per test. Perfect strong scaling on this test would be a computational cost proportional to N_ν , since the number of ray-cell interactions is linear in N_ν .

Our strong scaling results with varying N_ν are shown in Figure 9. We find that the wall clock time spent per ray trace increases with N_ν as expected, but that this increase is highly-sublinear, particularly at small N_ν . We find $t_\nu \propto N_\nu^{0.14}$ for 1-8 frequency bins and $t_\nu \propto N_\nu^{0.65}$ for 16-64 frequency bins. While these results might at first seem surprising, they make sense when

Figure 9: Frequency bin scaling test where we have varied the number of frequency bins N_ν . There is one source at the center of a $(128)^3$ domain and we truncate rays once they have traveled 0.5 pc. The cost of the ray trace rises with N_ν , as expected, but this effect is small for low N_ν .

700 we recall that the overhead associated with the ray tracing – solving the
 701 geometric problem of finding the paths of rays through cells and grids, the
 702 probing and handshaking parts of the communication steps – does not scale
 703 with N_ν . As we increase N_ν , this overhead is “amortized” over a larger
 704 number of frequency bins, and thus we obtain what appears to be better-
 705 than-perfect strong scaling. As the number of frequency bins increases, this
 706 effect becomes less important, and the parts of the computation that do scale
 707 with N_ν – computing the opacities of cells and updating fluxes, transferring
 708 flux data between processors – begin to dominate. For sufficiently large N_ν
 709 we do begin to approach the expected $N_\nu^{1.0}$ scaling, but our results thus
 710 far demonstrate that we can use up to ~ 10 frequency bins at near-zero
 711 additional cost, and several tens at only modest cost, compared to the single-
 712 frequency case.

713 5.2.3. AMR Simulation: Application to High-Mass Star Formation

714 Our final strong scaling test is an AMR simulation that includes hy-
 715 drodynamics, self-gravity, radiative transfer, and radiating sink particles to
 716 demonstrate how our new *HARM*² algorithm scales in a demanding research
 717 application. Here we perform strong scaling tests for two different outputs
 718 from an ORION AMR simulation of the formation of a high-mass stellar sys-
 719 tem. The results of this simulation will be presented in Rosen et al. [35] but
 720 we briefly summarize our problem setup here.

721 Our initial condition is a rotating, laminar $150 M_{\odot}$ molecular core with
 722 radius 0.1 pc. The core follows a $\rho(r) \propto r^{-3/2}$ density profile. We use
 723 a domain size of 0.4 pc on each side, a base resolution of 128^3 and five
 724 levels of refinement which yields a maximum resolution of 20 AU on the
 725 finest level. To properly model the absorption of the direct radiation field
 726 from stars we use the frequency dependent stellar atmosphere profiles from
 727 Lejeune et al. [36] to model the stellar spectra. Our choice of the opacities
 728 depend on whether the primary absorber is dust or molecular gas. Dust
 729 is the primary absorber for gas temperatures below $T_{\text{sub}} = 1500$ K (i.e.,
 730 the temperature at which dust sublimates) [34] while molecular hydrogen is
 731 the primary absorber for gas temperatures within $T_{\text{sub}} \leq T < T_{\text{H II}}$ where
 732 $T_{\text{H II}} \approx 10^4$ K is the temperature at which we expect hydrogen to become
 733 fully ionized, and thus to have the usual Thompson opacity for electron
 734 scattering. If the primary absorber is dust we use the frequency dependent
 735 dust opacities from Weingartner and Draine [33] (their $R_v = 5.5$ extinction
 736 curve), if it is molecular hydrogen we set the molecular gas opacity to 0.01
 737 $\text{cm}^2 \text{g}^{-1}$, and if $T \geq T_{\text{H II}}$ we set the opacity to zero. The last of these
 738 is a numerical convenience, because we have not implemented scattering or
 739 ionization chemistry, and because the regions in our computation with $T >$
 740 $T_{\text{H II}}$ generally contain so little mass they will be optically thin anyway. We
 741 assume a dust-to-gas ratio of 0.01 and choose $N_{\nu} = 10$.

742 At $t = 0$ the molecular core begins to gravitationally collapse. As the
 743 core collapses a star forms at the center and continues to grow in mass via
 744 accretion. An accretion disk forms around the star due to conservation of an-
 745 gular momentum of the infalling material. Gravitational instabilities develop
 746 in the disk causing it to fragment into companion stars. The absorption of
 747 energy and momenta from the direct stellar radiation field and the diffuse
 748 dust-reprocessed radiation field from the fluid results in low-density, radia-
 749 tion pressure dominated bubbles near the poles of the most massive star that

750 expand with time. Figure 10 shows slices parallel to the x-direction of the gas
 751 density (top panels) and absorbed direct radiation energy density (bottom
 752 panels) by the dust and gas for two different snapshots of this simulation at
 753 $t = 15.22$ kyrs and $t = 23.67$ kyrs. We only show the central $(8000 \text{ AU})^2$ re-
 754 gion of the computational domain because the majority of the domain is not
 755 affected by the direct radiation field. The most massive stars in these snap-
 756 shots are $16.59 M_{\odot}$ and $33.57 M_{\odot}$, respectively. The early snapshot contains
 757 one star while the later snapshot contains eight stars where the companions
 758 range from $0.01 - 1.48 M_{\odot}$ in stellar mass. These snapshots represent typical
 759 “early” and “late” stages of the computation, with the latter being much
 760 more computationally expensive due to the larger number of sources and the
 761 greater distances that the direct radiation can propagate before being ab-
 762 sorbed. We note that both the early and late stages are strong tests of the
 763 scalability, because the radiating sources are confined to a small portion of
 764 the computational volume, rather than being scattered throughout (c.f., the
 765 test presented in Wise and Abel [18], which used a cosmological simulation
 766 where point sources were distributed nearly-isotropically.)

767 Our strong scaling results are shown in Figure 11, where we measure the
 768 time spent on the hydrodynamics, gravity, FLD, adaptive ray trace, and the
 769 total radiation module (adaptive ray trace and FLD). A horizontal line would
 770 correspond to perfect strong scaling. We ran each timing test for five time
 771 steps on $N_{\text{CPU}} = 16 \times n$ processors where $n = [1, 2, \dots, 8]$. The early snapshot
 772 contains 448 grids and the later snapshot contains 1137 grids at the beginning
 773 of each test. The top panel shows the timing results for the early snapshot
 774 and the bottom panel shows the results for the later snapshot. Comparison
 775 of the two panels show that the scalability for all modules in ORION become
 776 better at later times, especially for the adaptive ray trace when more grids are
 777 processing rays. This is due to the increase in number of grids per processor
 778 which reduces the MPI communication costs. A general rule of thumb for
 779 patch-based AMR methods such as ORION is that the code is efficient at
 780 ~ 4 grids per CPU or more, and our tests are consistent with this. We find
 781 that our timing results for the adaptive ray trace follows $t_{\text{WC,ART}} \propto N_{\text{CPU}}^{0.97}$
 782 for the early snapshot and $t_{\text{WC,ART}} \propto N_{\text{CPU}}^{0.56}$ for the later snapshot. These
 783 results agree with our strong scaling results from section 5.2.1 which showed
 784 that our parallelization procedure for the adaptive ray trace becomes more
 785 efficient as the number of grids that interact with rays increases. We find
 786 that the moment method, FLD in our case, is the most expensive module
 787 while gravity is the cheapest, and that the adaptive ray trace can be cheaper

Figure 10: Example AMR simulation that uses our *HARM*² algorithm. Here we show slice plots along the x-direction of the mass density (top) and absorbed direct radiation energy density (bottom) for two snapshots of a simulation of the formation a high mass star system. Gray stars denote the location of the stars, with the most massive star being largest. The left (right) panels show the snapshot when the simulation has progressed to 15.22 kyrs (23.67 kyrs) where the most massive star is $16.59 M_{\odot}$ ($33.57 M_{\odot}$).

and/or about the same expense as the hydrodynamics.

6. Summary

In this paper, we have presented our implementation of *HARM*² – a new highly-parallel multi-frequency hybrid radiation hydrodynamics module that combines an adaptive long characteristics method for the (direct) radial radiation field from point sources with a moment method that handles the (thermal) diffuse radiation field produced by a volume-filling fluid. Our new method is designed to be used with adaptive grids and is not limited to specific geometries. We have coupled *HARM*² to the hydrodynamics in the astrophysical AMR code ORION which includes flux limited diffusion, but our method can be applied to any AMR hydrodynamics code that has asynchronous time stepping and can incorporate any moment method. Although our implementation is not the first hybrid radiation scheme implemented in

Figure 11: Strong scaling test with a 128^3 AMR simulation with 5 levels of refinement of the formation of a massive star system shown at two different simulation outputs from 15.22 kyrs (top) and 23.67 kyrs (bottom). The early (late) snapshot has 448 (1137) grids. The bottom panel shows that the scalability of the adaptive ray trace increases as the simulation progresses because rays interact with a larger volume of the computational domain (e.g., see Figure 10).

801 an AMR code, it is more accurate than previous methods because it uses long
 802 rather than hybrid characteristics. Furthermore, our new algorithm can be
 803 used in a variety of radiation hydrodynamics problems in which the radiation
 804 from point sources and diffuse radiation field from the fluid should be mod-

805 elled. Such examples are the study of the formation of isolated high-mass
806 stars and clustered star formation in the dusty interstellar medium.

807 One of the major difficulties with incorporating a long characteristics
808 method in an AMR code that allows for a general geometry, where the hy-
809 drodynamics is parallelized by domain decomposition, is the parallel com-
810 munication of rays. This is because ray tracing is a highly serial process and
811 each ray will usually cross multiple processor domains. In order to avoid
812 significant communication overheads and serial bottlenecks that often occur
813 with long characteristics methods we have implemented a new completely
814 asynchronous and non-blocking communication algorithm for ray commu-
815 nication. We performed a variety of weak and strong scaling tests of this
816 method, and found that its performance is dramatically improved compared
817 to previous long characteristics methods. In idealized tests without adap-
818 tive grids we obtain near-perfect weak scaling out to > 1000 cores, and, in
819 problems where the characteristic trace covers the entire computational do-
820 main, near-perfect strong scaling as well. Previous implementations became
821 communications-bound at processor counts a factor of ~ 4 smaller than this.
822 In a realistic, demanding research application with a complex, adaptive grid
823 geometry, and using 10 frequency bins for the characteristic trace, we find
824 excellent scaling as long as there are at least $\sim 3 - 4$ grids per CPU, and we
825 find that the cost of adaptive ray tracing is smaller than or comparable to
826 hydrodynamics, and significantly cheaper than flux limited diffusion.

827 Since *HARM*² works for adaptive grids in a general geometry, it can be
828 used in a variety of high-resolution simulations that require radiative transfer.
829 Our implementation in ORION will be made public in an upcoming release of
830 the ORION code, and the *HARM*² source code will be made available imme-
831 diately upon request to any developers who are interested in implementing
832 *HARM*² in their own AMR codes.

833 Acknowledgements

834 We thank the referees for their comments and suggestions, which improved
835 the quality of this work. We thank Nathan Roth, Andrew Myers, and Pak
836 Shing Li for their assistance in integrating *HARM*² into ORION, Chris Mc-
837 Kee for helpful comments throughout the development process, and John
838 Wise and Christian Baczynski for comments and advice. ALR and MRK
839 acknowledge support from the National Aeronautics and Space Administra-
840 tion (NASA) through Hubble Archival Research grant HST-AR-13265.02-A

841 issued by the Space Telescope Science Institute, which is operated by the
 842 Association of Universities for Research in Astronomy, Inc., under NASA
 843 contract NAS 5-26555 and Chandra Theory Grant Award Number TM5-
 844 16007X issued by the Chandra X-ray Observatory Center, which is operated
 845 by the Smithsonian Astrophysical Observatory for and on behalf of NASA
 846 under contract NAS8-03060. ALR and ATL acknowledge support from the
 847 NSF Graduate Research Fellowship Program. RIK acknowledges support
 848 from NASA through ATP grant NNX13AB84G, the NSF through grant
 849 AST-1211729 and the US Department of Energy at the Lawrence Livermore
 850 National Laboratory under contract DE-AC52-07NA27344. MRK and RIK
 851 acknowledge support from NASA TCAN grant NNX-14AB52G. MRK ac-
 852 knowledges support from Australian Research Council grant DP160100695.

- 853 [1] M. R. Krumholz, Star Formation with Adaptive Mesh Refinement Ra-
 854 diation Hydrodynamics, in: J. Alves, B. G. Elmegreen, J. M. Girart,
 855 V. Trimble (Eds.), *Computational Star Formation*, vol. 270 of *IAU Sym-*
 856 *posium*, 187–194, doi:10.1017/S1743921311000354, 2011.
- 857 [2] R. Teyssier, Grid-Based Hydrodynamics in Astrophysical Fluid
 858 Flows, *Annu. Rev. Astron. & Astrophys.* 53 (2015) 325–364, doi:
 859 10.1146/annurev-astro-082214-122309.
- 860 [3] C. D. Levermore, G. C. Pomraning, A flux-limited diffusion theory,
 861 *Astrophys. J.* 248 (1981) 321–334, doi:10.1086/159157.
- 862 [4] M. R. Krumholz, R. I. Klein, C. F. McKee, J. Bolstad, Equations and
 863 Algorithms for Mixed-frame Flux-limited Diffusion Radiation Hydrody-
 864 namics, *Astrophys. J.* 667 (2007) 626–643, doi:10.1086/520791.
- 865 [5] B. Commerçon, R. Teyssier, E. Audit, P. Hennebelle, G. Chabrier, Ra-
 866 diation hydrodynamics with adaptive mesh refinement and application
 867 to prestellar core collapse. I. Methods, *Astron. & Astrophys.* 529 (2011)
 868 A35+, doi:10.1051/0004-6361/201015880.
- 869 [6] M. González, E. Audit, Numerical Treatment of Radiative Transfer,
 870 *Astrophys. & Space Sci.* 298 (2005) 357–362, doi:10.1007/s10509-005-
 871 3969-2.
- 872 [7] J. Rosdahl, R. Teyssier, A scheme for radiation pressure and photon dif-
 873 fusion with the M1 closure in RAMSES-RT, *Mon. Not. Roy. Astron. Soc.*
 874 449 (2015) 4380–4403, doi:10.1093/mnras/stv567.

- 875 [8] P. G. Dykema, R. I. Klein, J. I. Castor, A New Scheme for Multidi-
876 mensional Line Transfer. III. A Two-dimensional Lagrangian Variable
877 Tensor Method with Discontinuous Finite-Element SN Transport, *Astrophys. J.* 457 (1996) 892, doi:10.1086/176784.
878
- 879 [9] Y.-F. Jiang, J. M. Stone, S. W. Davis, A Godunov Method for
880 Multidimensional Radiation Magnetohydrodynamics Based on a Vari-
881 able Eddington Tensor, *Astrophys. J. Supp.* 199 14, doi:10.1088/0067-
882 0049/199/1/14.
- 883 [10] S. W. Davis, J. M. Stone, Y.-F. Jiang, A Radiation Transfer Solver
884 for Athena Using Short Characteristics, *Astrophys. J. Supp.* 199 9, doi:
885 10.1088/0067-0049/199/1/9.
- 886 [11] E.-J. Rijkhorst, T. Plewa, A. Dubey, G. Mellema, Hybrid characteris-
887 tics: 3D radiative transfer for parallel adaptive mesh refinement hydro-
888 dynamics, *Astron. & Astrophys.* 452 (2006) 907–920, doi:10.1051/0004-
889 6361:20053401.
- 890 [12] L. Bunttemeyer, R. Banerjee, T. Peters, M. Klassen, R. E. Pudritz, Ra-
891 diation hydrodynamics using characteristics on adaptive decomposed
892 domains for massively parallel star formation simulations, *New Astron.*
893 43 (2016) 49–69, doi:10.1016/j.newast.2015.07.002.
- 894 [13] M. G. Wolfire, J. P. Cassinelli, The temperature structure in accretion
895 flows onto massive protostars, *Astrophys. J.* 310 (1986) 207–221, doi:
896 10.1086/164676.
- 897 [14] M. G. Wolfire, J. P. Cassinelli, Conditions for the formation of massive
898 stars, *Astrophys. J.* 319 (1987) 850–867.
- 899 [15] S. D. Murray, J. I. Castor, R. I. Klein, C. F. McKee, Accretion disk
900 corone in high-luminosity systems, *Astrophys. J.* 435 (1994) 631–646,
901 doi:10.1086/174842.
- 902 [16] R. Kuiper, H. Klahr, C. Dullemond, W. Kley, T. Henning, Fast and ac-
903 curate frequency-dependent radiation transport for hydrodynamics sim-
904 ulations in massive star formation, *Astron. & Astrophys.* 511 A81, doi:
905 10.1051/0004-6361/200912355.

- 906 [17] M. Klassen, R. Kuiper, R. E. Pudritz, T. Peters, R. Banerjee, L. Bunte-
 907 meyer, A General Hybrid Radiation Transport Scheme for Star For-
 908 mation Simulations on an Adaptive Grid, *Astrophys. J.* 797 4, doi:
 909 10.1088/0004-637X/797/1/4.
- 910 [18] J. H. Wise, T. Abel, ENZO+MORAY: radiation hydrodynamics
 911 adaptive mesh refinement simulations with adaptive ray tracing,
 912 *Mon. Not. Roy. Astron. Soc.* 414 (2011) 3458–3491, doi:10.1111/j.1365-
 913 2966.2011.18646.x.
- 914 [19] M. L. Norman, P. Paschos, T. Abel, Simulating inhomogeneous reion-
 915 ization, *Mem. Soc. Astron. Ital.* 69 (1998) 455.
- 916 [20] D. Mihalas, R. I. Klein, On the solution of the time-dependent inertial-
 917 frame equation of radiative transfer in moving media to $O(v/c)$, *Jour-
 918 nal of Computational Physics* 46 (1982) 97–137, doi:10.1016/0021-
 919 9991(82)90007-9.
- 920 [21] D. Mihalas, L. H. Auer, On laboratory-frame radiation hydrodynamics,
 921 *Quant. Spectrosc. Radiat. Transfer* 71 (2001) 61–97, doi:10.1016/S0022-
 922 4073(01)00013-9.
- 923 [22] R. I. Klein, Star formation with 3-D adaptive mesh refinement: the col-
 924 lapse and fragmentation of molecular clouds., *Journal of Computational
 925 and Applied Mathematics* 109 (1999) 123–152.
- 926 [23] R. T. Fisher, Single and Multiple Star Formation in Turbulent Molecular
 927 Cloud Cores, Ph.D. thesis, University of California, Berkeley, 2002.
- 928 [24] P. S. Li, D. F. Martin, R. I. Klein, C. F. McKee, A Stable, Accurate
 929 Methodology for High Mach Number, Strong Magnetic Field MHD Tur-
 930 bulence with Adaptive Mesh Refinement: Resolution and Refinement
 931 Studies, *Astrophys. J.* 745 (2012) 139–151.
- 932 [25] M. J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial
 933 differential equations, *J. Comp. Phys.* 53 (1984) 484–512.
- 934 [26] M. J. Berger, P. Colella, Local adaptive mesh refinement for shock hy-
 935 drodynamics, *J. Comp. Phys.* 82 (1989) 64–84.

- [27] T. Abel, B. D. Wandelt, Adaptive ray tracing for radiative transfer around point sources, *Mon. Not. Roy. Astron. Soc.* 330 (2002) L53–L56, doi:10.1046/j.1365-8711.2002.05206.x.
- [28] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, M. Bartelmann, HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere, *Astrophys. J.* 622 (2005) 759–771, doi:10.1086/427976.
- [29] M. R. Krumholz, J. M. Stone, T. A. Gardiner, Magnetohydrodynamic Evolution of H II Regions in Molecular Clouds: Simulation Methodology, Tests, and Uniform Media, *Astrophys. J.* 671 (2007) 518–535, doi:10.1086/522665.
- [30] M. R. Krumholz, C. D. Matzner, The Dynamics of Radiation-pressure-dominated H II Regions, *Astrophys. J.* 703 (2009) 1352–1362, doi:10.1088/0004-637X/703/2/1352.
- [31] M. Pandolfi, D. D’Ambrosio, Numerical Instabilities in Upwind Methods: Analysis and Cures for the ”Carbuncle” Phenomenon, *Journal of Computational Physics* 166 (271) (2001) 31.
- [32] J. M. Stone, T. A. Gardiner, P. Teuben, J. F. Hawley, J. B. Simon, Athena: A New Code for Astrophysical MHD, *Astrophys. J. Supp.* 178 (2008) 137–177, doi:10.1086/588755.
- [33] J. C. Weingartner, B. T. Draine, Dust Grain-Size Distributions and Extinction in the Milky Way, Large Magellanic Cloud, and Small Magellanic Cloud, *Astrophys. J.* 548 (2001) 296–309, doi:10.1086/318651.
- [34] D. Semenov, T. Henning, C. Helling, M. Ilgner, E. Sedlmayr, Rosseland and Planck mean opacities for protoplanetary discs, *Astron. & Astrophys.* 410 (2003) 611–621, doi:10.1051/0004-6361:20031279.
- [35] A. L. Rosen, M. R. Krumholz, C. F. McKee, R. I. Klein, An Unstable Truth: How Massive Stars get their Mass, *ArXiv e-prints* .
- [36] T. Lejeune, F. Cuisinier, R. Buser, Standard stellar library for evolutionary synthesis. I. Calibration of theoretical spectra, *Astron. & Astrophys. Supp.* 125 (1997) 229–246, doi:10.1051/aas:1997373.