Journal of Computational Physics 330 (2017) 924–942

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

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, 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

ARTICLE INFO

Article history:
Received 23 February 2016
Received in revised form 4 October 2016
Accepted 21 October 2016
Available online 26 October 2016

Keywords:
Radiative transfer
Hydrodynamics
Parallelization
Long characteristics
Adaptive mesh refinement

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.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Radiation-hydrodynamics (RHD) is a challenging numerical problem, but it is a crucial component in modeling several physical phenomena in the fields of astrophysics, laser physics, and plasma physics. Accurate solution of the radiative transfer (RT) equation, which governs the evolution of radiation interacting with matter, is difficult because of its high dimensionality. This equation depends on six independent variables: three spatial, two angles describing the direction of the propagation of photons, and one frequency dimension. For time-dependent RHD calculations, this solution must be obtained at every time step, and then coupled to the hydrodynamics. Even on parallel supercomputers direct solution of the RT equation at each time step of a time-dependent calculation is prohibitively expensive, because of this most numerical RHD codes use approximations to treat the evolution of the radiation field and its interaction with matter.

One common approach to solving the RHD equations is to reduce the dimensionality of the problem. This class of approximations are known as moment methods because they take the moments of the radiative transfer equation in direct analogy to the Chapman–Enskog procedure used to derive the hydrodynamic equations from the kinetic theory of gases [1,2]. This method averages over the angular dependence, and thus is a good approximation for smooth, diffuse radiation fields such as those present in optically thick media when the radiation is tightly coupled to the matter. The accuracy with

* Corresponding author.
E-mail address: alrosen@ucsc.edu (A.L. Rosen).

http://dx.doi.org/10.1016/j.jcp.2016.10.048
0021-9991/© 2016 Elsevier Inc. All rights reserved.

which moment methods recover the angular dependence of the true solution depends on the order at which the moments are closed, and on the closure relation adopted. Common approximations include flux-limited diffusion (FLD; closure at first moment) [3–5], the M1 method (closure at the 2nd moment using a minimum entropy closure) [6,7], and Variable Eddington Tensor (VET; closure at the 2nd moment using an approximate solution to the full transfer equation) [8–10]. Regardless of the order and closure relation, the computational cost of these methods usually scales as N or $N \log N$, where N is the number of cells, and the technique is highly parallelizable [1].

An alternative technique used to solve the RT equation numerically is characteristics-based ray tracing, which solves this equation directly along specific rays. With this method, the directionality of the radiative flux is highly accurate, but the accuracy depends on the sampling of rays. Two widely used schemes for ray tracing in grid-based codes are *long* and *hybrid* characteristics. Long characteristics traces rays on a cell by cell basis, and provides maximum possible accuracy. Hybrid characteristics is a combination of long characteristics within individual grids and short characteristics between grids (i.e., in which only neighboring grid cells are used to interpolate incoming intensities) [11,12]. The method of short characteristics is faster but more diffusive compared to long characteristics methods. The computational cost for both methods scales linearly with the number of sources, rays traced, and grid cells with which the rays interact, making these methods prohibitively expensive for treating diffuse radiation fields where every computational cell is a source. Instead, they are ideal for treating the radial radiation fields of point sources. Even for this application, however, one major drawback of ray tracing methods, especially long characteristics, is that they are difficult to parallelize in a code where the hydrodynamics is parallelized by domain decomposition. In such a configuration, each ray will usually cross multiple processor domains, creating significant communications overheads and serial bottlenecks.

In summary, moment methods are better at approximating the diffuse radiation field from a fluid but are poor at modeling the propagation of radiation from point sources where the direction of the field is important. Characteristics methods, in contrast, are good at approximating the direction-dependent radiation fields from point sources but are too computationally expensive for practical use in simulating a diffuse radiating fluid. When both point and diffuse radiation sources are present, therefore, a natural approach is to combine both techniques by using long characteristics to model the propagation of radiation from a point source and its subsequent interaction (e.g., absorption) with the fluid and then use a moment method to follow the subsequent diffuse re-emission.

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

The reason that many authors have resorted to special geometries or abandoned long characteristics is the difficulty in parallelizing long characteristics in a general geometry, particularly in the case of adaptive grids. The problem is difficult because it is unknown *a priori* how far rays will travel and what grids they will interact with in an adaptive grid framework. In a distributed memory paradigm where different grids may be stored in memory on different processors, this can easily result in a complex communication pattern with numerous serial bottlenecks. Indeed, all implementations of long characteristics on adaptive grids published to date use synchronous communication algorithms in which processors must wait for other processors to receive ray information [18], leading to exactly this problem.

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

2. Decomposition of the radiation-hydrodynamics problem

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$cG^0 = \int_0^\infty d\nu \int d\Omega \left[\kappa(\mathbf{n}, \nu) I_{\text{dir}}(\mathbf{n}, \nu) - \eta_{\text{dir}}(\mathbf{n}, \nu)\right] + \int_0^\infty d\nu \int d\Omega \left[\kappa(\mathbf{n}, \nu) I_{\text{diff}}(\mathbf{n}, \nu) - \eta_{\text{diff}}(\mathbf{n}, \nu)\right] \tag{9}$$

$$c\mathbf{G} = \int_0^\infty d\nu \int d\Omega \left[\kappa(\mathbf{n}, \nu) I_{\text{dir}}(\mathbf{n}, \nu) - \eta_{\text{dir}}(\mathbf{n}, \nu)\right] \mathbf{n} + \int_0^\infty d\nu \int d\Omega \left[\kappa(\mathbf{n}, \nu) I_{\text{diff}}(\mathbf{n}, \nu) - \eta_{\text{diff}}(\mathbf{n}, \nu)\right] \mathbf{n} \tag{10}$$

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

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

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
$dL_{\mathrm{ray},i}$	Absorbed luminosity from ray for the ith 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 ith 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 ith frequency bin
$L_{ray,i}$	Luminosity in ith frequency bin along ray
$\mathbf{n}$	Normal direction of ray
N_{CPU}	Number of processors
$N_{\mathrm{destroyed},k}$	Number of rays "destroyed" on processor k
$N_{\mathrm{destroyed}}$	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
$N_{\mathrm{pix}}(j)$	Number of HEALPix pixels on level j
$N_{\mathrm{ray},k}$	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

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

3. The *HARM*^{2} algorithm

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

3.1. Update cycle

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

Given this setup, our algorithm is as follows:

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

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

3.2. Direct radiation field: adaptive ray trace

We now describe the adaptive ray tracing procedure that forms step 1a(ii) of the algorithm above. Consider a single point source with a specific luminosity given by L_ν and luminosity given by $L_\star = \int_0^\infty L_\nu\,d\nu$. The generalization to multiple point sources is trivial. We discretize the point source spectrum in frequency into N_ν frequency bins, with the ith bin covering a 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 of the point source integrated over the ith frequency such that $\sum L_i = L_\star$. We generally expect that L_i will be the energy radiated per unit time in a given frequency bin, but the algorithm is identical if we instead take L_i to be a photon luminosity, measured in photons per unit time.

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

$$\frac{\partial L_{\mathrm{ray},i}}{\partial r} = -\kappa_i L_{\mathrm{ray},i}, \tag{11}$$

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

$$dL_{\mathrm{ray},i} = L_{\mathrm{ray},i}\left(1 - e^{-\tau_i}\right). \tag{12}$$

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

$$\frac{dE}{dt} = \sum_{i=1}^{N_\nu} dL_{\mathrm{ray},i} \tag{13}$$

$$\frac{d\mathbf{p}}{dt} = \sum_{i=1}^{N_\nu} \frac{dL_{\mathrm{ray},i}}{c}\mathbf{n}. \tag{14}$$

The total absorption rate for each cell is simply the sum of dE/dt and $d\mathbf{p}/dt$ over all rays from all point sources that pass through it. When computing the line segments dl, we only consider grids that are not masked by any finer grid. That is, when solving Equation (11), we only ever consider the most highly spatially resolved data at any given position.

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

$$\frac{\Omega_{\rm cell}}{\Omega_{\rm ray}} = \frac{N_{\rm pix}(j)}{4\pi}\left(\frac{\Delta x}{r}\right)^2 < \Phi_{\rm c}, \tag{15}$$

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

3.3. Parallelization

Thus far the algorithm we have described is substantially identical to that of Wise and Abel [18]. However, we adopt a very different, and much more efficient strategy to parallelize this procedure. The primary challenge to parallelizing this algorithm is avoiding serial bottlenecks. The grids through which the rays must be traced may be distributed across any number of processors, and solution of Equation (11) is an intrinsically serial process because the rate of change of the energy and momentum in any cell due to radiation arriving along a particular ray depends upon the properties of all cells that lie between the point source and the cell in question. Since the numbers and positions of point sources and computational grids in the AMR structure, and their distribution in memory, are not known *a priori*, minimizing bottlenecks requires an opportunistic approach: rays should be able to be processed in arbitrary order, with each processor performing ray tracing given the data available to it, and waiting for other processors only when no useful work can be done. To this end, communication must be non-blocking and asynchronous. At the same time, each process must be able to determine when the entire ray trace has been completed, so that it can proceed to the remainder of the update cycle (the moment method, hydrodynamics, etc.), and this determination must be robust against race conditions.

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

$$N_{\rm max} = N_{\rm src}\times 12\times 4^{j_{\rm max}} \tag{16}$$

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

$$N_{{\rm destroyed},k} = \sum_{N_{{\rm ray},k}} 4^{j_{\rm max}-j} \tag{17}$$

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

$$N_{\rm destroyed} = \sum_{N_{\rm CPU}} N_{{\rm destroyed},k}, \tag{18}$$

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

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

For each processor, our algorithm is as follows:

1. Inject rays from point sources to grids.
2. Compute $N_{\rm max}$ (Equation (16)).

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

```
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;
```

3. Set ALLDONE to **false**, set LOOPITER = 0, and enter outer **while**(not ALLDONE) loop.
   (a) Set WORKREMAINS to **true** if rays exist on grids that belong to this processor, otherwise set WORKREMAINS to **false**.
   (b) If WORKREMAINS is **true** enter **while**(WORKREMAINS) loop.
      i. Loop over all grids that belong to this processor and **for each** grid advance all rays that belong to that grid until they (i) leave the grid and need to be moved to another grid, (ii) become extinct, or (iii) leave the domain. Every time (ii) or (iii) occurs for a ray we delete the ray and increase the value of $N_{\text{destroyed},k}$ on this processor per Equation (17). If (i) occurs we determine the new grid and processor for that ray.
      ii. Set WORKREMAINS to **false**.
      iii. Loop over rays that must be transferred to other grids and check if they belong to a grid on this processor or another processor. If the former is true place the ray on the correct grid and set WORKREMAINS to **true**, otherwise the ray must be transferred to another processor so we place the ray in a linked list, **outgoing_ray_list**, to be communicated to the other processors.
      iv. **Repeat** until all rays have been either destroyed and/or placed into **outgoing_ray_list**.
   (c) Loop over rays in **outgoing_ray_list**. Place rays from **outgoing_ray_list** into a contiguous array for MPI communication (one array per receiving processor) and perform a non-blocking **MPI_Isend** for each array to send it to the appropriate processor.
   (d) Set DATARECV to **true** and if (LOOPITER **modulo** $N_{\text{CPU}} == 0$)[2] then enter the **while**(DATARECV) loop to begin receiving rays from other processors.
      i. Probe other processors with the non-blocking **MPI_ Iprobe** function to see if rays need to be received, if this is true set DATARECV to **true** otherwise set it to **false**.
      ii. If DATARECV is **true** then receive rays on this processor with **MPI_Recv**. Place the incoming rays onto the new grids they belong to.
      iii. **Repeat** until **MPI_Iprobe** returns false.
   (e) If this processor sent rays to another processor then perform non-blocking **MPI_Testsome** to test for some given ray send requests to complete.
   (f) If WORKREMAINS is false then perform non-blocking **MPI_Testsome** to test for some of the $N_{\text{destroyed, k}}$ send requests to complete.

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

(g) Set COUNTRECV to **true** and if (LOOPITER **modulo** $N_{\rm CPU} == 0$) then enter the **while**(COUNTRECV) loop to begin receiving rays from other processors.
   i. Probe other processors with the non-blocking **MPI_ Iprobe** function to see if the value of $N_{{\rm destroyed},k}$ on those processors needs to be received. If this is true set COUNTRECV to **true**, otherwise set it to **false**.
   ii. Receive $N_{{\rm destroyed},k}$ from processor k on this processor with **MPI_Recv** if COUNTRECV is **true** and assign to the kth element of an array **rayDestProc** containing $N_{\rm CPU}$ elements.
   iii. **Repeat** until **MPI_Iprobe** returns false.

(h) If $N_{{\rm destroyed},k}$ on this processor has increased in the outer loop iteration then send this number with a non-blocking **MPI_Isend** to all other processors.

(i) Compute sum of **rayDestProc**. If this value equals $N_{\rm max}$ terminate the outer loop, else repeat outer loop and increment LOOPITER by 1.

4. Validation tests

In this section we demonstrate the accuracy of our adaptive ray tracing algorithm, the absorption of the direct radiation field from point sources supplied by our adaptive ray tracing scheme, and our hybrid radiative transfer algorithm by performing four tests. In the first test, we turn off absorption of the radiation field to trace the radiative flux from a point source located at the center to demonstrate that our method recovers the correct r^{-2} fall-off of the flux. The second test focuses on the coupling of the hydrodynamics with the adaptive ray trace. We also demonstrate that *HARM*^{2} can handle multiple sources with the third and fourth tests. Our third test demonstrates that our hybrid radiative transfer method can properly cast shadows and transfer radiative energy to the gas by illuminating a dense clump in a low-density medium with two point sources when the hydrodynamics is neglected. Finally, our fourth test follows a similar setup as the third test but demonstrates how the hydrodynamics are coupled to *HARM*^{2}. We set $\Phi_{\rm c} = 4$ for all validation tests in this section, and we use only a single frequency bin. For the first two tests we disable the moment method part of *HARM*^{2}, so that we can focus on the ray-tracing part of the algorithm. Further validation tests for the moment method have been presented in Krumholz et al. [4].

4.1. Flux test

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

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

$$U_{\rm rad} = \sum_i \frac{L_{{\rm ray},i}}{dV}\frac{dl_i}{c} \tag{19}$$

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

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

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

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

4.2. Radiation-pressure-dominated HII region

Next we perform a test to illustrate the coupling of the radiative transfer from the adaptive ray trace with the hydrodynamics, based on a similarity solution obtained by Krumholz and Matzner [30]. We consider an initially-uniform, cold gas with an isothermal equation of state. At time $t = 0$ a point source of radiation with luminosity $L_\star$ turns on and begins depositing momentum in the gas. We consider material that has a very high opacity to photons coming directly from the point

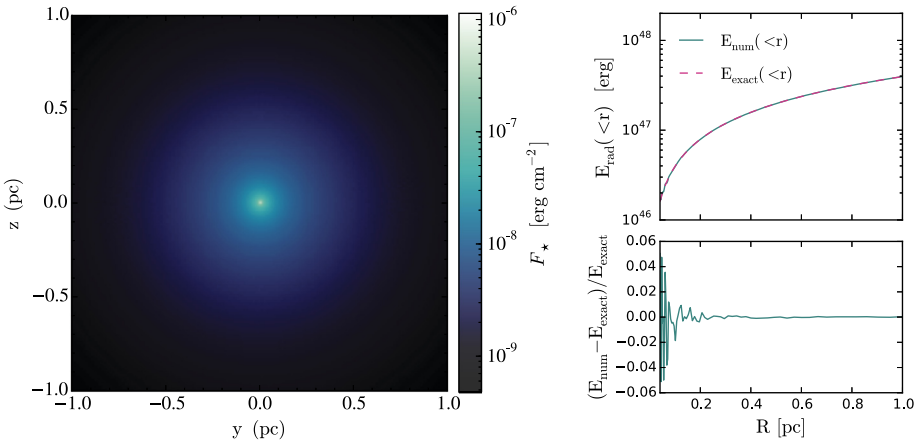

Fig. 1. Performance test for the adaptive ray trace. Left panel: Projection plot of the stellar radiation density for a source with luminosity $10^6\,L_\odot$. The source flux falls off as $F(r) \propto r^{-2}$ as expected. Right panels: Comparison of the numerical and analytical results of the energy enclosed within radius r (i.e, $E_{\rm num}(<r)$) 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. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

source, but a very low opacity to any re-emitted photons. A real-world example of this would be interstellar dust absorbing ultraviolet photons from a star, and then re-emitting them as infrared light, to which the dust is essentially transparent.

Because the opacity is high, all of the radiation from the point source is absorbed in an extremely thin layer, but then escapes immediately. Thus the point source deposits radial momentum into the gas at a rate $dp/dt = L_\star/c$. After a short time the material around the point source will have been swept into a thin shell of radius $r_{\rm sh}$ and mass $M_{\rm sh} = 4\pi r_{\rm sh}^3 \rho_0/3$, where ρ_0 is the initial density. The shell obeys an equation of motion

$$\frac{d}{dt}\left(M_{\rm sh}\dot{r}_{\rm sh}\right) = \frac{L_\star}{c}. \tag{21}$$

This equation admits a similarly solution given by

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

where $n_0 = \rho_0/\mu m_{\rm p}$ is the number density, μ is the mean molecular weight which we set to 2.33 for molecular hydrogen and helium mixed in the usual cosmic ratio, and $m_{\rm p}$ is the proton mass.

To test the ability of our code to reproduce this solution, we consider a domain with a width of 1 pc, a uniform number density of $n_0 = 10^5\ {\rm cm}^{-3}$ ($\rho_0 = 3.89\times10^{-19}\ {\rm g\,cm}^{-3}$), and a point source of luminosity $L_\star = 10^6 L_\odot$ at the origin. We set the specific opacity to $\kappa/\rho = 10^6\ {\rm cm}^2\,{\rm g}^{-1}$. We perform 3 tests on non-adaptive grids with varying resolution (64^3, 128^3, and 256^3) to explore how the accuracy of the adaptive ray trace depends on resolution.

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

In Fig. 3 we show how the radius of the shell in our simulation compares to the analytic similarity solution. We define the radius of the shell to be the density weighted average distance from the origin for cells where the density exceeds 1.5 ρ_0:

$$R_{\rm sh} = \frac{\sum_{\rm cells,\ \rho_j > 1.5\rho_0} \rho_j r_j}{\sum_{\rm cells,\ \rho_j > 1.5\rho_0} \rho_j}. \tag{23}$$

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

4.3. Hybrid radiation tests with multiple sources

To demonstrate that our hybrid radiation algorithm can properly cast shadows we perform two validation tests in which we have two point sources irradiate a dense clump of material. These tests are similar to the tests presented in Rijkhorst

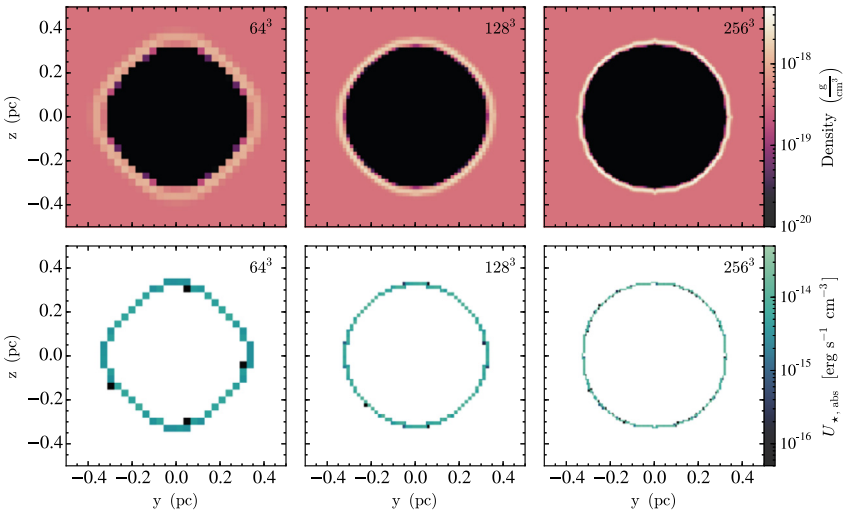

Fig. 2. Results from the radiation-pressure-dominated H II region test to demonstrate the performance of the adaptive ray trace coupled to the hydrodynamics. Top (bottom) panels show slice plots of the gas density (rate of absorbed radiation energy density per unit time) for our radiation dominated sphere 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, $\kappa = 10^6$ cm^2/g, used.

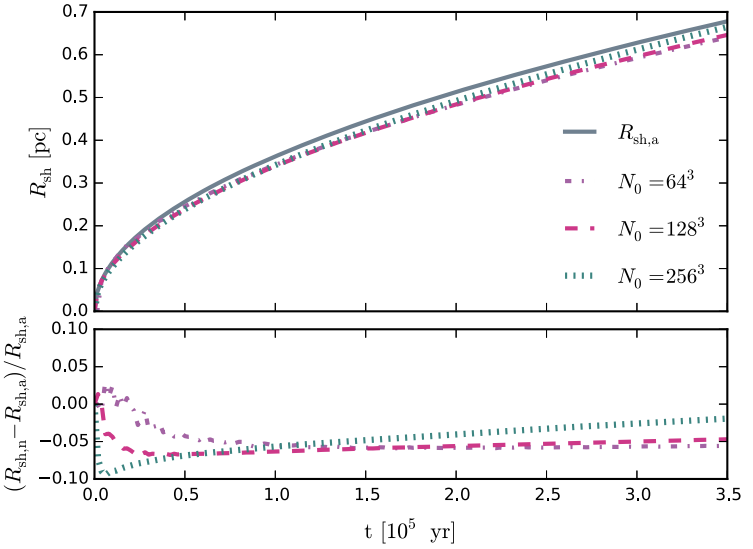

Fig. 3. Shell position (top panel) and residuals (bottom panel) as compared to the analytical solution (i.e., eqn. (22)) for our radiation dominated sphere test at three different uni-grid resolutions (64^3, 128^3, and 256^3). The largest deviations from the analytical solution occur at early times when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

et al. [11], Jiang et al. [9], and Klassen et al. [17]. In our setup, we have a dense clump of material with radius 267 AU at the center of a (2000 AU)^3 computational domain. We take the clump density to be $\rho_c = 3.89 \times 10^{-17}$ g cm^{-3} with uniform temperature of 20 K. It is surrounded 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 same temperature as the clump (20 K) so that we can follow the radiative heating of the gas. For this test we disable the hydrodynamics. In the second test, we set the temperature of the ambient medium to be 20×10^3 K so that the clump is initially in pressure balance with the low-density medium. In this test we enable the hydrodynamics. For both tests we use one frequency bin for the absorption of the direct radiation field and assume that the gas opacity is $\kappa = 64$ cm^2/g following the frequency dependent dust opacities in Weingartner and Draine [33] (their $R_V = 5.5$ extinction curve) corresponding to a frequency

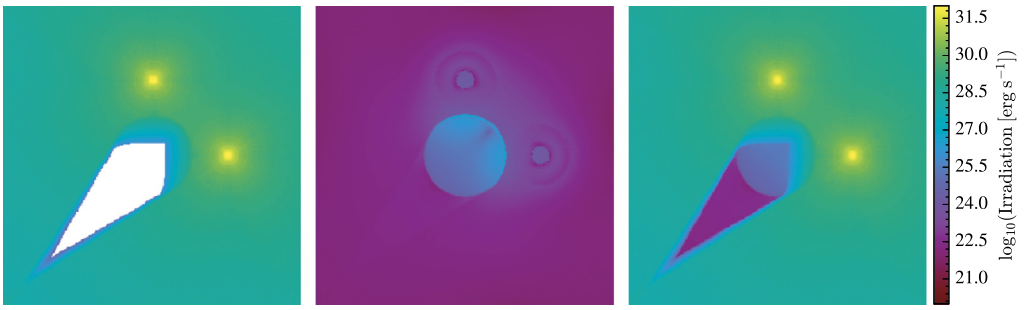

Fig. 4. Irradiation of a dense clump by two point sources in a homogenous medium. The left panel shows the irradiation by the point sources' direct radiation fields while the middle panel shows the dust-reprocessed irradiation. The right panel shows the total irradiation by the stellar sources and dusty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

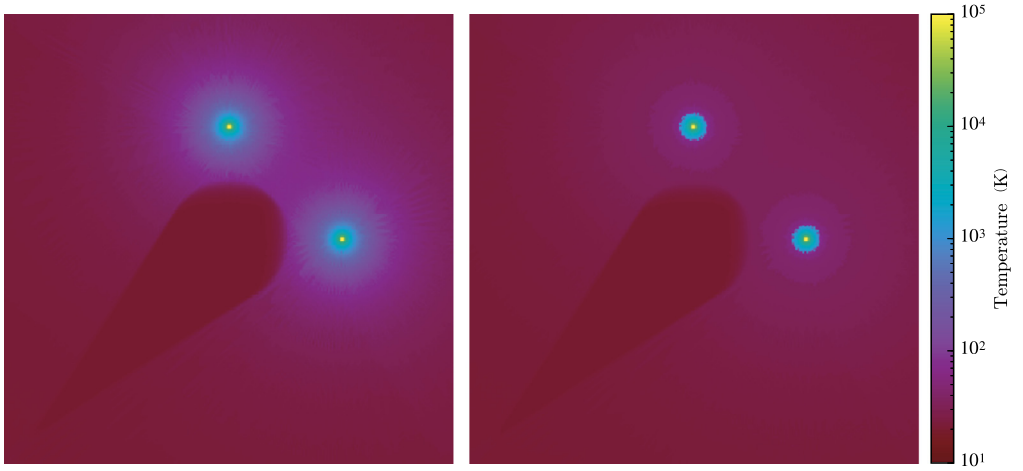

Fig. 5. Same as the left and right panels of Fig. 4, except showing the gas temperature where we only consider heating by the direct radiation field by including the adaptive ray trace (left panel) and where we consider heating by the direct radiation field and cooling by the dusty gas using *HARM*^{2}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

of $\nu = kT_{\rm eff}/h = 1.21 \times 10^{14}$ Hz, where we have assumed $T_{\rm eff} = 5800$ K (i.e, the peak color temperature of the Sun). Our moment method uses the gray opacities from Semenov et al. [34].

4.3.1. Irradiation of a dense clump by two point sources with *HARM*^{2}

We perform our first hybrid radiation test on a uniform $(256)^3$ grid, disabling 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. Fig. 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 Fig. 4.

Fig. 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 Fig. 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 Fig. 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.

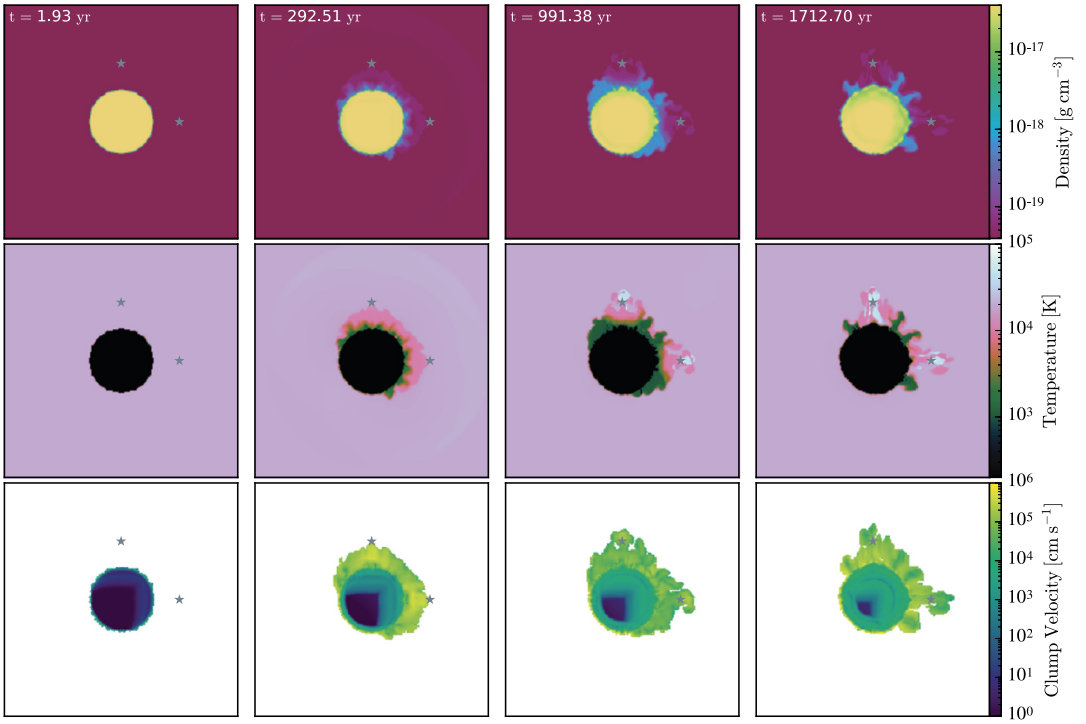

Fig. 6. Snapshots of the radiation-hydrodynamical time evolution of a dense clump in a low-density transparent medium that is irradiated by two point sources using our *HARM*^{2} radiative transfer method. Different columns show different times, and different rows show different quantities: density (top), 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.

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

Fig. 6 shows the evolution of the density, temperature, and clump velocity of the simulation. This Figure demonstrates that the absorption of the point sources' direct radiation field heats up the sides of the clump that are closest to the sources, causing the edges to expand. The momentum imparted by the direct radiation field to the clump also causes it to expand non-uniformly. We note that as the cloud becomes over-pressurized it expands into the ambient medium yielding large velocities along the edges of the cloud. As the simulation evolves the cloud is slowly disrupted leading to low density material expanding towards the point sources. The expansion of this low density material leads to instabilities as the clump material mixes with the low density gas. At late times the expansion of the low-density clump material closest to the point sources slows as the direct radiation field imparts momentum to the gas.

5. Performance tests

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

5.1. Weak scaling

Weak scaling tests demonstrate how well a parallel code scales with the number of processors while the workload assigned to each processor remains the same. For this purpose, we perform a weak scaling test on non-adaptive grids

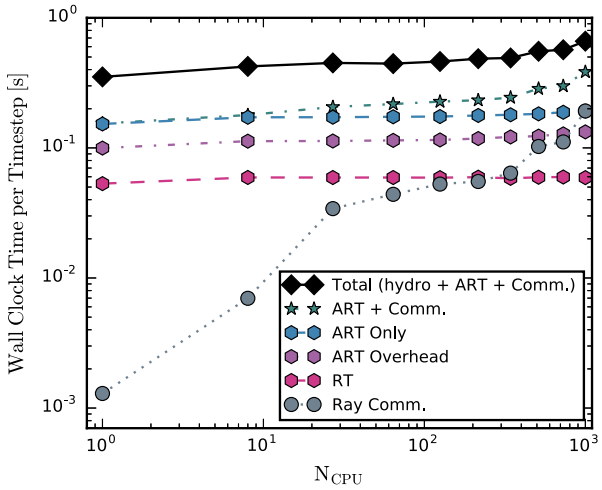

Fig. 7. Weak scaling test with one 32^3 block per process. Each block is $(1\ \text{pc})^3$ and contains one radiating source at its center. Rays are terminated after they have traveled 0.6 pc from the source to ensure communication of rays to neighboring grids. Weak scaling results are shown for the 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. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where each processor has one 32^3 grid and one radiating sink particle. Each 32^3 grid represents a $(1\ \text{pc})^3$ domain with constant gas density, $n = 10^4\ \text{cm}^{-3}$, with a radiating source placed at the center. The physics modules we include are the adaptive ray trace and hydrodynamics. We set the opacity in all cells to zero so that no absorption of the radiation field occurs. To ensure that each processor performs the same amount of work with the ray trace, including the propagation of rays and the subsequent communication of rays to other processors, we terminate rays once they have traveled 0.6 pc from their originating source. This allows for rays to propagate to their neighboring grids and also enforces that all grids, except the grids along the domain edges, communicate the same number of rays to their neighbors. In short, the rays interact with the cells they cross but do not add energy or momenta to the fluid.

Our weak scaling tests were run on $N_{\text{CPU}} = n^3$ processors, where $n = [1, 2, ..., 9, 10]$, for 50 time steps per weak scaling test. The weak scaling test results are presented in Fig. 7 and show the total time spent per time step (black solid line) and the timing of the adaptive ray trace components: ray communication (gray dotted line), ray trace across cells (pink dashed line), adaptive ray trace overhead (i.e., locating ray grids, ray splitting, etc. – purple dot-dashed line), total adaptive ray trace (ray trace and associated overhead – blue dashed line), and the full adaptive ray trace which includes the ray communication (dot-dashed teal line). We note that a horizontal line denotes perfect weak scaling.

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

5.2. Strong scaling

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

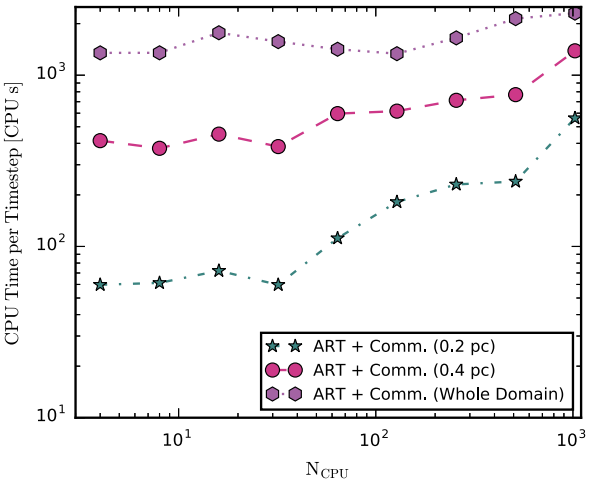

Fig. 8. Strong scaling test with a 256^3 uni-grid calculation with a radiating point source at the center. We performed two tests where the rays are terminated after they travel 0.2 or 0.4 pc from their source and a third test where the rays transverse the entire domain. Perfect strong scaling would 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.

Table 2
Fitted scaling results from our uni-grid strong scaling tests presented in Fig. 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^a_{CPU})
0.2 pc	0.0335	0.52
0.4 pc	0.2681	0.27
Whole	1	0.084

5.2.1. Uni-grid ray trace test with varying termination lengths

We first use a setup similar to a single instance of our weak scaling test: a $(1\ \text{pc})^3$ domain with a single point source placed at its center. The resolution of the computational domain is 256^3 cells and each grid consists of 16^3 cells yielding a total of 4096 grids, with no adaptivity. We perform three sub-tests with this setup in which the rays are destroyed after traveling 0.2 pc from their source, 0.4 pc from their source, or allowed to transverse the entire domain, respectively. These calculations were performed on $N_{CPU} = 2^n$ processors with $n = [2, ..., 9, 10]$ for 5 time steps per test.

Our strong scaling results are presented in Fig. 8, which shows the total CPU time per time step, t_{CPU}, for the adaptive ray trace. (Note that, whereas in Fig. 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 $t_{CPU} \propto N^a_{CPU}$; 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: $t_{CPU} \propto N^{0.084}_{CPU}$. 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. 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 traveled 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

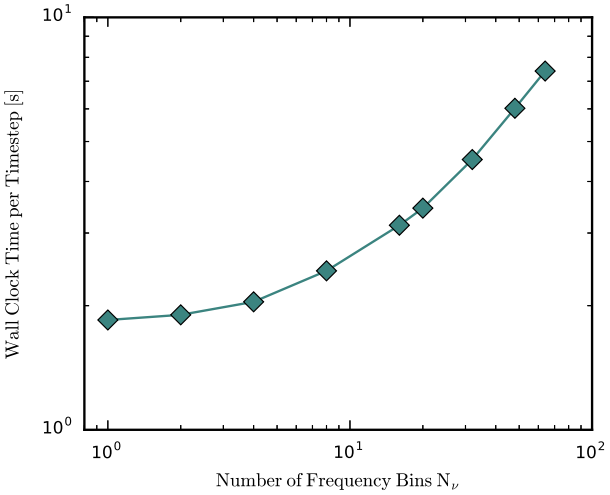

Fig. 9. Frequency bin scaling test where we have varied the number of frequency bins N_ν. There is one source at the center of a $(128)^3$ domain and we truncate rays once they have traveled 0.5 pc. The cost of the ray trace rises with N_ν, as expected, but this effect is small for low N_ν.

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

5.2.3. AMR simulation: application to high-mass star formation

Our final strong scaling test is an AMR simulation that includes hydrodynamics, self-gravity, radiative transfer, and radiating sink particles to demonstrate how our new *HARM*^{2} algorithm scales in a demanding research application. Here we perform strong scaling tests for two different outputs from an Orion AMR simulation of the formation of a high-mass stellar system. The results of this simulation will be presented in Rosen et al. [35] but we briefly summarize our problem setup here.

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

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

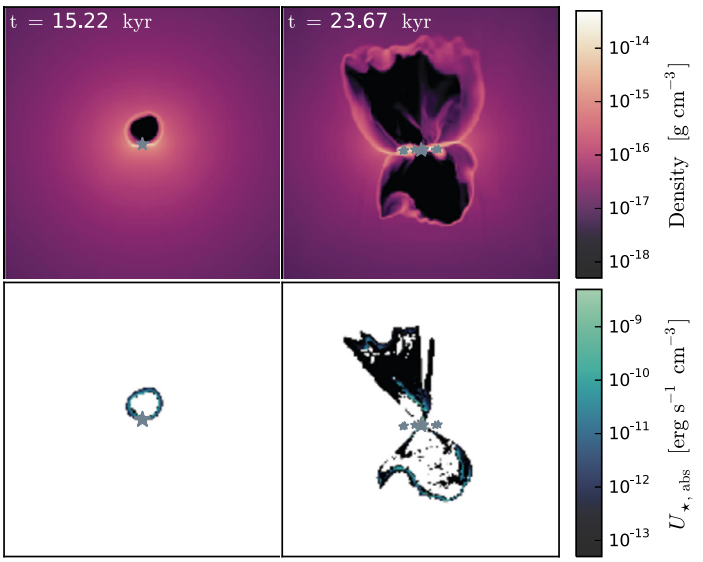

Fig. 10. Example AMR simulation that uses our *HARM*^{2} algorithm. Here we show slice plots along the x-direction of the mass density (top) and absorbed direct radiation energy density (bottom) for two snapshots of a simulation of the formation a high mass star system. Gray stars denote the location 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 $M_\odot$).

expand with time. Fig. 10 shows slices parallel to the x-direction of the gas density (top panels) and absorbed direct radiation energy density (bottom panels) by the dust and gas for two different snapshots of this simulation at $t = 15.22$ kyrs and $t = 23.67$ kyrs. We only show the central (8000 AU)^2 region of the computational domain because the majority of the domain is not affected by the direct radiation field. The most massive stars in these snapshots are 16.59 $M_\odot$ and 33.57 $M_\odot$, respectively. The early snapshot contains one star while the later snapshot contains eight stars where the companions range from 0.01–1.48 $M_\odot$ in stellar mass. These snapshots represent typical "early" and "late" stages of the computation, with the latter being much more computationally expensive due to the larger number of sources and the greater distances that the direct radiation can propagate before being absorbed. We note that both the early and late stages are strong tests of the scalability, because the radiating sources are confined to a small portion of the computational volume, rather than being scattered throughout (cf., the test presented in Wise and Abel [18], which used a cosmological simulation where point sources were distributed nearly-isotropically.)

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

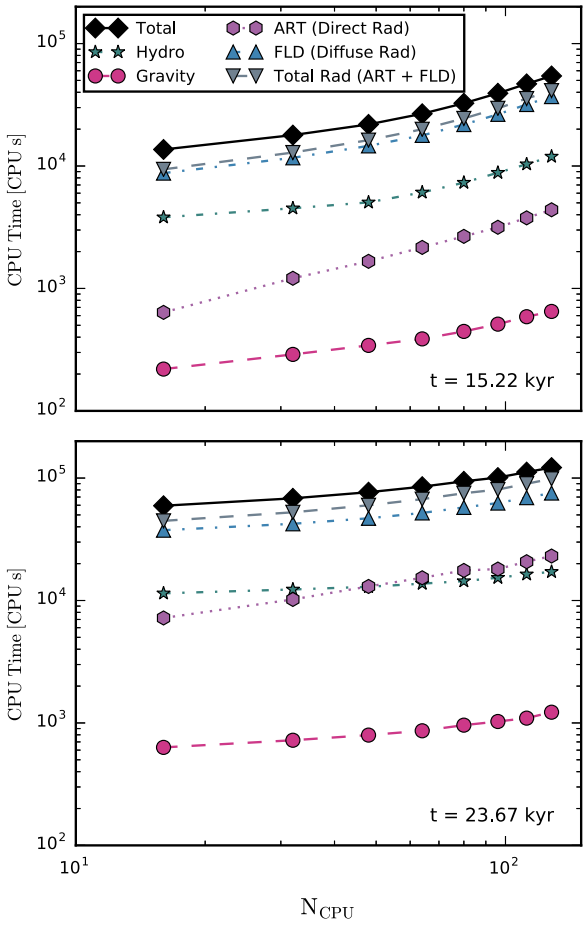

Fig. 11. Strong scaling test with a 128^3 AMR simulation with 5 levels of refinement of the formation of a massive star system shown at two different simulation outputs from 15.22 kyrs (top) and 23.67 kyrs (bottom). The early (late) snapshot has 448 (1137) grids. The bottom panel shows that the scalability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

method. Although our implementation is not the first hybrid radiation scheme implemented in an AMR code, it is more accurate than previous methods because it uses long rather than hybrid characteristics. Furthermore, our new algorithm can be used in a variety of radiation hydrodynamics problems in which the radiation from point sources and diffuse radiation field from the fluid should be modeled. Such examples are the study of the formation of isolated high-mass stars and clustered star formation in the dusty interstellar medium.

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

Since *HARM*^{2} works for adaptive grids in a general geometry, it can be used in a variety of high-resolution simulations that require radiative transfer. Our implementation in Orion will be made public in an upcoming release of the Orion code, and the *HARM*^{2} source code will be made available immediately upon request to any developers who are interested in implementing *HARM*^{2} in their own AMR codes.

Acknowledgements

We thank the referees for their comments and suggestions, which improved the quality of this work. We thank Nathan Roth, Andrew Myers, and Pak Shing Li for their assistance in integrating *HARM*^{2} into Orion, Chris McKee for helpful comments throughout the development process, and John Wise and Christian Baczynski for comments and advice. ALR and MRK acknowledge support from the National Aeronautics and Space Administration (NASA) through Hubble Archival Research grant HST-AR-13265.02-A issued by the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 5-26555 and Chandra Theory Grant Award Number TM5-16007X issued by the Chandra X-ray Observatory Center, which is operated by the Smithsonian Astrophysical Observatory for and on behalf of NASA under contract NAS8-03060. ALR and ATL acknowledge support from the NSF Graduate Research Fellowship Program (number DGE-0809125). RIK acknowledges support from NASA through ATP grant NNX13AB84G, the NSF through grant AST-1211729 and the US Department of Energy at the Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. MRK and RIK acknowledge support from NASA TCAN grant NNX-14AB52G. MRK acknowledges support from Australian Research Council grant DP160100695.

References

[1] M.R. Krumholz, Star formation with adaptive mesh refinement radiation hydrodynamics, in: J. Alves, B.G. Elmegreen, J.M. Girart, V. Trimble (Eds.), Computational Star Formation, in: IAU Symposium, vol. 270, 2011, pp. 187–194.
[2] R. Teyssier, Grid-based hydrodynamics in astrophysical fluid flows, Annu. Rev. Astron. Astrophys. 53 (2015) 325–364, http://dx.doi.org/10.1146/annurev-astro-082214-122309.
[3] C.D. Levermore, G.C. Pomraning, A flux-limited diffusion theory, Astrophys. J. 248 (1981) 321–334, http://dx.doi.org/10.1086/159157.
[4] M.R. Krumholz, R.I. Klein, C.F. McKee, J. Bolstad, Equations and algorithms for mixed-frame flux-limited diffusion radiation hydrodynamics, Astrophys. J. 667 (2007) 626–643, http://dx.doi.org/10.1086/520791.
[5] B. Commerçon, R. Teyssier, E. Audit, P. Hennebelle, G. Chabrier, Radiation hydrodynamics with adaptive mesh refinement and application to prestellar core collapse. I. Methods, Astron. Astrophys. 529 (2011) A35+, http://dx.doi.org/10.1051/0004-6361/201015880.
[6] M. González, E. Audit, Numerical treatment of radiative transfer, Astrophys. Space Sci. 298 (2005) 357–362, http://dx.doi.org/10.1007/s10509-005-3969-2.
[7] J. Rosdahl, R. Teyssier, A scheme for radiation pressure and photon diffusion with the M1 closure in RAMSES-RT, Mon. Not. R. Astron. Soc. 449 (2015) 4380–4403, http://dx.doi.org/10.1093/mnras/stv567.
[8] P.G. Dykema, R.I. Klein, J.I. Castor, A new scheme for multidimensional line transfer, III: a two-dimensional Lagrangian variable tensor method with discontinuous finite-element SN transport, Astrophys. J. 457 (1996) 892, http://dx.doi.org/10.1086/176784.
[9] Y.-F. Jiang, J.M. Stone, S.W. Davis, A Godunov method for multidimensional radiation magnetohydrodynamics based on a variable Eddington tensor, Astrophys. J. Supp. 199 (2012) 14, http://dx.doi.org/10.1088/0067-0049/199/1/14.
[10] S.W. Davis, J.M. Stone, Y.-F. Jiang, A radiation transfer solver for athena using short characteristics, Astrophys. J. Supp. 199 (2012) 9, http://dx.doi.org/10.1088/0067-0049/199/1/9.
[11] E.-J. Rijkhorst, T. Plewa, A. Dubey, G. Mellema, Hybrid characteristics: 3D radiative transfer for parallel adaptive mesh refinement hydrodynamics, Astron. Astrophys. 452 (2006) 907–920, http://dx.doi.org/10.1051/0004-6361:20053401.
[12] L. Buntemeyer, R. Banerjee, T. Peters, M. Klassen, R.E. Pudritz, Radiation hydrodynamics using characteristics on adaptive decomposed domains for massively parallel star formation simulations, New Astron. 43 (2016) 49–69, http://dx.doi.org/10.1016/j.newast.2015.07.002.
[13] M.G. Wolfire, J.P. Cassinelli, The temperature structure in accretion flows onto massive protostars, Astrophys. J. 310 (1986) 207–221, http://dx.doi.org/10.1086/164676.
[14] M.G. Wolfire, J.P. Cassinelli, Conditions for the formation of massive stars, Astrophys. J. 319 (1987) 850–867.
[15] S.D. Murray, J.I. Castor, R.I. Klein, C.F. McKee, Accretion disk coronae in high-luminosity systems, Astrophys. J. 435 (1994) 631–646, http://dx.doi.org/10.1086/174842.
[16] R. Kuiper, H. Klahr, C. Dullemond, W. Kley, T. Henning, Fast and accurate frequency-dependent radiation transport for hydrodynamics simulations in massive star formation, Astron. & Astrophys. 511 (2010) A81, http://dx.doi.org/10.1051/0004-6361/200912355.
[17] M. Klassen, R. Kuiper, R.E. Pudritz, T. Peters, R. Banerjee, L. Buntemeyer, A general hybrid radiation transport scheme for star formation simulations on an adaptive grid, Astrophys. J. 797 (2014) 4, http://dx.doi.org/10.1088/0004-637X/797/1/4.
[18] J.H. Wise, T. Abel, ENZO+MORAY: radiation hydrodynamics adaptive mesh refinement simulations with adaptive ray tracing, Mon. Not. R. Astron. Soc. 414 (2011) 3458–3491, http://dx.doi.org/10.1111/j.1365-2966.2011.18646.x.
[19] M.L. Norman, P. Paschos, T. Abel, Simulating inhomogeneous reionization, Mem. Soc. Astron. Ital. 69 (1998) 455.
[20] D. Mihalas, R.I. Klein, On the solution of the time-dependent inertial-frame equation of radiative transfer in moving media to $O(v/c)$, J. Comput. Phys. 46 (1982) 97–137, http://dx.doi.org/10.1016/0021-9991(82)90007-9.
[21] D. Mihalas, L.H. Auer, On laboratory-frame radiation hydrodynamics, J. Quant. Spectrosc. Radiat. Transf. 71 (2001) 61–97, http://dx.doi.org/10.1016/S0022-4073(01)00013-9.
[22] R.I. Klein, Star formation with 3-D adaptive mesh refinement: the collapse and fragmentation of molecular clouds, J. Comput. Appl. Math. 109 (1999) 123–152.
[23] R.T. Fisher, Single and Multiple Star Formation in Turbulent Molecular Cloud Cores, Ph.D. thesis, University of California, Berkeley, 2002.
[24] P.S. Li, D.F. Martin, R.I. Klein, C.F. McKee, A stable, accurate methodology for high Mach number, strong magnetic field MHD turbulence with adaptive mesh refinement: resolution and refinement studies, Astrophys. J. 745 (2012) 139–151.
[25] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484–512.
[26] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64–84.
[27] T. Abel, B.D. Wandelt, Adaptive ray tracing for radiative transfer around point sources, Mon. Not. R. Astron. Soc. 330 (2002) L53–L56, http://dx.doi.org/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, http://dx.doi.org/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, http://dx.doi.org/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, http://dx.doi.org/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, J. Comput. Phys. 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. Suppl. Ser. 178 (2008) 137–177, http://dx.doi.org/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, http://dx.doi.org/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, http://dx.doi.org/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. Suppl. Ser. 125 (1997) 229–246, http://dx.doi.org/10.1051/aas:1997373.