DRAFT VERSION SEPTEMBER 27, 2013
Preprint typeset using LaTeX style emulateapj v. 12/16/11

TESTING FULLY DYNAMICAL ADAPTIVE MESH REFINEMENT IN THE EINSTEIN TOOLKIT

CUTTER A. CORYELL, ROLAND HAAS[1], AND CHRISTIAN D. OTT[1,2,*]

Draft version September 27, 2013

ABSTRACT

The explosion mechanism of core-collapse supernovae is not yet fully understood. Two candidate mechanisms exist: magnetically-driven explosions and neutrino-driven explosions. Initial simulations with spherical (1-dimensional) and axial (2-dimensional) symmetry did not yield robust explosions, despite including neutrino-heating in their models. Instead these simulations saw the post-bounce shock stall, never making it to the surface of the star to cause explosions that would be visible from Earth. Current 3-dimensional general relativistic simulations, such as the one featured in Ott et al. (2013), study the explosion mechanism in greater detail to draw conclusions about the supernova from the observed signal. To do so, the shock must be simulated with high-resolution, but increasing the resolution of the entire simulation is too computationally costly. We develop a routine in the Einstein Toolkit that finds the location of the shock at any time and applies adaptive mesh refinement to resolve the shock surface without resolving unimportant regions.

Keywords: adaptive mesh refinement – Einstein Toolkit – hydrodynamics – Stars: supernovae: general

1. INTRODUCTION

Supernovae are cosmic explosions so bright that they can briefly outshine the galaxies in which they occur. The observational signatures of supernovae are quite diverse, but are generally separated into two categories: Type I supernovae and Type II supernovae. These are distinguished by the strong presence of hydrogen emission lines in Type II supernova spectra and the lack of these lines in Type I supernovae. About 80% of Type I supernovae have nearly identical light curves, or evolutions of their brightnesses over time (Bethe 1990). These supernovae are thought to be the thermonuclear explosions of white dwarves, a process we will not discuss further in this report. The rest of Type I supernovae, as well as all Type II supernovae, are thought to be the result of core-collapse in massive ($\gtrsim 8\,M_\odot$) stars (Woosley & Janka 2005).

Core-collapse supernovae occur when the iron core of massive stars, which is held up against gravity by electron degeneracy pressure, reaches the Chandrasekhar mass through accretion of newly-fused iron. Above this mass, the core cannot be supported against gravity by electron degeneracy pressure and consequently collapses. It is the gravitational energy released by this collapse that powers the supernova explosion, though the mechanism of the transfer of energy from gravitational energy to kinetic energy, electromagnetic energy, and neutrino energy that result in the observational signature of supernovae is detailed. Essential to the explosion is a shock wave that starts as the inner core reaches densities exceeding nuclear density ($\rho > 4-5\times10^{14}$ g cm^{-3}). At these densities, the strong nuclear force becomes repulsive, resulting in a stiffened equation of state. The inner core consequently overshoots its equilibrium and rebounds (Woosley & Janka 2005), launching a shock wave that propagates outwards and eventually blows off the outer parts of the star. However, current simulations do not get that far; the wave quickly stalls as it emits neutrinos and disassociates iron nuclei in the still-infalling outer core. These stars consequently fail to explode, indicating that our current models are insufficient. A possible mechanism to resolve this problem may be related to the prodigious neutrino generation of the collapsed inner core, which is now a proto-neutron star. This neutrino generation is a process that releases the majority of the supernova's energy. If a small fraction of the neutrinos are absorbed by the matter near the shock, enough energy may be deposited for the shock to be revived. The deposition of neutrino energy in the shock would lead to convective instability, which would in turn help conserve the shock energy from neutrino losses and lead to non-radial deformation that facilitates further energy deposition from neutrinos (Woosley & Janka 2005). It is unknown as to whether this proposed mechanism for reviving the shock is sufficient to cause the observed explosions, and the frontier of supernova simulation seeks to answer this question.

Early core-collapse supernova simulations, such as that of Colgate & White (1966), were one-dimensional and spherically-symmetric. Later two-dimensional, axisymmetric models allowed for more complex fluid motion such as convection. However, these simulations did not yield robust explosions. The current frontier is three-dimensional, fully-general-relativistic simulations such as that of Ott et al. (2013). These simulations typically have zones of different simulation resolutions to allocate computing resources to fixed spatial locations based on the desired accuracy at those locations. For example, Ott et al. (2013) use a set of nested boxes, centered on the proto-neutron star, in which the inner boxes have higher resolutions than the outer boxes. In this way, the dense regions in the interior are resolved more finely than the infalling, lower-density material.

However, this method of nested resolution zones has disadvantages. In the current implementation, the zones are cubical, whereas spherical zones would more closely match the roughly spherically-symmetric density distribution in the star. Interior must always have higher resolution than the boxes

[1] TAPIR, Mailcode 350-17, California Institute of Technology, Pasadena, CA 91125, USA, cott@tapir.caltech.edu
[2] Kavli Institute for the Physics and Mathematics of the Universe (Kavli IPMU), The University of Tokyo, Kashiwa, Japan
[*] Alfred P. Sloan Research Fellow

that contain them, which means that features at larger radii can never have higher resolution than features at smaller radii. Because the shock wave is an important feature, one of the higher-resolution nested boxes is dynamically expanded to always contain the shock. Consequently, the region between the shock and the proto-neutron star receives the same or greater resolution than the shock, even though it is relatively unimportant in the evolution of the supernova.

A technique called adaptive mesh refinement (AMR), pioneered by Berger & Oliger (1984), allows for dynamically-varying resolution at any location in a Cartesian-gridded simulation. In this technique, Cartesian grid patches of higher resolution are laid down on top of a lower-resolution base-grid, or coarse grid. Multiple of these patches can be stitched together together to form refined regions of any shape. Most importantly, these patches can be created and destroyed on the fly according to any desired criteria. After the coarse grid is evolved, the refined patches receive their boundary information from the coarse grid and are themselves evolved, after which values from the refined patches are interpolated back into the coarse grid.

The Einstein Toolkit is a computational platform designed for use in numerical relativity and relativistic astrophysics (Löffler et al. 2012). A module for the Einstein Toolkit called `CarpetRegrid2` (`CR2`) allows for adaptive mesh refinement as well as the nested refinement boxes mentioned earlier. In this project, we attempted to apply the AMR functionality of `CR2` to refine the shock wave of two models: a simple, spherically-symmetric explosion, and a snapshot of a post-bounce core-collapse supernova.

In Section 2 of this report, we present the routine implemented to refine the shock wave, as well as descriptions of the two explosion models to which we apply it. In Section 3, we discuss the qualitative and quantitative performance of our application to the two models. In Section 4, we make closing remarks and present ideas for further work.

Our report and simulation use units where the universal gravitational constant G and the speed of light c have both been set to one. The basic unit for length and time are both one solar mass (1 $M_{\odot}$, referred to as one simulation unit), which is equivalent to 1.48 km or 4.93 μs.

We will refer to "resolution" and "linear point density" interchangeably, meaning the number of grid points per length unit (1.48 km) along a line parallel to one of the coordinate axes. The resolution could be different along different coordinate directions, but in our simulations the resolution is the same along each direction.

2. METHODS

2.1. *The Shock-Tracking Routine*

The goal of our application of AMR is to resolve the shock front, but to do so the shock must be localized. A shock is a sudden spatial or temporal change in the properties of a substance: in our case, the matter in the supernova. We choose to focus on the spatial definition of the shock, therefore the quantity of interest is the spatial gradient of the properties of the stellar matter. The shock manifests itself as an abrupt change in any of the simulation quantities, but *a priori* we did not know which quantity would reveal the best-defined shock surface. Possibilities include density, pressure, entropy, internal energy, and fluid velocity.

For quantity ϕ, we define the shock strength χ_{ϕ}:

$$\chi_{\phi} = \left| \frac{\nabla \phi}{\phi} \right| \tag{1}$$

Because the shock strength is proportional to the gradient, we define the shock to be at points where the shock strength is large, i.e., larger than an arbitrary threshold. But unlike the gradient, the shock strength is independent of the direction of the spatial change in the simulation quantity, and because we normalize by the quantity's value the shock strength's units[4] are independent of the chosen quantity.

One might argue that the direction information of the gradient is useful; for example, perhaps we only wish to apply AMR to radial shock fronts, where the shock surface wraps spherically around the proto-neutron star. In that case, a better shock strength metric would be $\xi_{\phi} = |\hat{\mathbf{r}} \cdot \nabla \phi / \phi|$, which picks out only radial change. While this would certainly focus AMR on spherical shock fronts, it would be less sensitive to deformed or off-center fronts. Additionally, it would neglect interesting features such as convection zones that may possess strong angular rates of change. Because convection as well as oscillating deformations of the shock front may be essential to the rejuvenation of the shock, we decided to use absolute, direction-unbiased rate of change as our criterion for locating the shock and other important features.

Other more sophisticated definitions of the shock strength that include further derivatives or combine functions of multiple simulation quantities may lead to more accurate or better-defined shock fronts. In this project we develop shock tracking with the simplest model possible as a first attempt at applying AMR in a supernova simulation, but due to the modularity of our design, the shock-tracking component of our routine can be refined at any point in the future.

Simulations in the Einstein Toolkit have a set of grid points, divided between the coarse level and the refined levels, that permeate the simulation space. These are the points at which the simulation is evolved. Variables called grid functions have values at each grid point and at each time-step. Density, pressure, internal energy, entropy, and temperature are all represented as grid functions in our simulation, as are their gradients and shock strengths.

The first component of our routine, which runs at every time-step, calculates the shock strength χ_{ϕ} at each point for a given simulation quantity ϕ. The shock front can then be considered to be the set of points at which the χ_{ϕ} exceeds a certain threshold value. This shock-tracking component of the routine can be used in its own right for purposes other than AMR; for example, the shock can be visualized or the shock shape and size can be calculated. The second part of our routine applies AMR to the now-localized shock front. To do this, it calls upon the AMR capability of `CarpetRegrid2`, a module of the Einstein Toolkit.

2.2. *Interfacing with* `CarpetRegrid2`

[4] The unit for shock strength in our simulation is $1/1.48$ km; see Section 1.

Whenever `CarpetRegrid2` is called, it examines a grid function called the level mask, which can be set by other routines. The level mask is a non-negative integer[5] which specifies the desired level of refinement. If the level mask is n at a point, then after `CR2` runs there will be a block centered there with resolution 2^n times the coarse resolution. The linear size of the block, given in units of the spacing between points on the coarse grid, is a parameter that can be set called the adaptive block size. Because the level mask can be set at each grid point, refined regions of arbitrary shape can be created, ideally creating a region that follows the shock surface.

Our routine sets the level mask to a given value at each point where the shock strength exceeds a given threshold. Optionally, the level mask can be set to this value throughout a cube of specified size surrounding each point where the shock strength exceeds the threshold. The motivation for this option was to make it possible to set the edges of a refined region far from a discontinuity such as a shock front if desired.

2.3. *Test Cases*

For this project we applied our shock-tracking/AMR routine to two test cases: a simple, spherically-symmetric explosion called the `sphereshock` simulation and a snapshot of a post-bounce core-collapse supernova with octant symmetry called the `octant` simulation.

2.3.1. `sphereshock` *Simulation*

The `sphereshock` simulation takes place inside a cubical simulation domain with a side-length of 50 simulation units (74 km). This domain is filled with a fluid. A large amount of energy is deposited in a spherically-symmetric region in the center of the domain, leading to a spherically-symmetric explosion with outwardly-propagating shock wave. While the system itself is symmetric, there is no artificial symmetry imposed by the simulation – the entire domain is evolved. The equation of state used is simple, where internal energy is proportional to pressure, and entropy is not defined. This leaves density and pressure as the most simple quantities with which to locate the shock.

Figures 1 and 2 depict the density ρ and the pressure p, respectively, in a cross-section through the origin of `sphereshock` after it has been evolved for 80 time-steps. Figures 3 and 4 show the corresponding shock strengths, χ_ρ and χ_p, respectively. Figure 4 shows a much better-defined shock surface than Figure 3, which is diffuse. For this reason, we chose to use pressure as the quantity with which to track the shock for `sphereshock`.

We chose a shock threshold of 1000 inverse simulation units. Figure 4 shows what is left when we remove the regions that do not exceed this threshold. The remaining (three-dimensional) annular surface is defined to be the shock surface. When AMR is applied to this simulation, `CR2`'s level mask will be set to 1 on this surface and 0 elsewhere, resulting in refinement only near the shock.

In order to characterize the efficiency of AMR versus alternatives, we performed experiments whereby `sphereshock` was evolved with identical initial conditions for 80 time-steps.

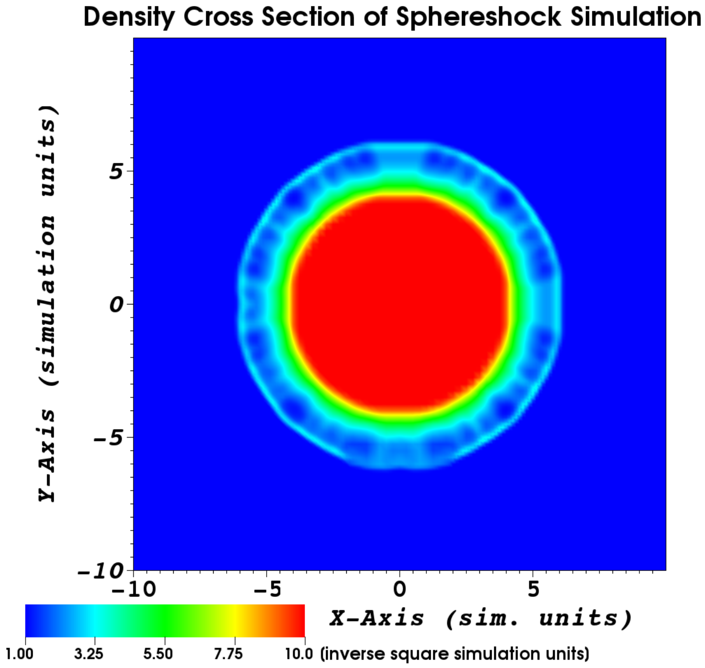

Figure 1. The $z = 0$ cross section through the `sphereshock` simulation after 80 time-steps, depicting the density ρ. Density, having units of mass per length cubed, has units of inverse square simulation units because mass and length are both measured in simulation units. One inverse square simulation unit is $6.18 \times 10^{17} g\ \mathrm{cm}^{-3}$.

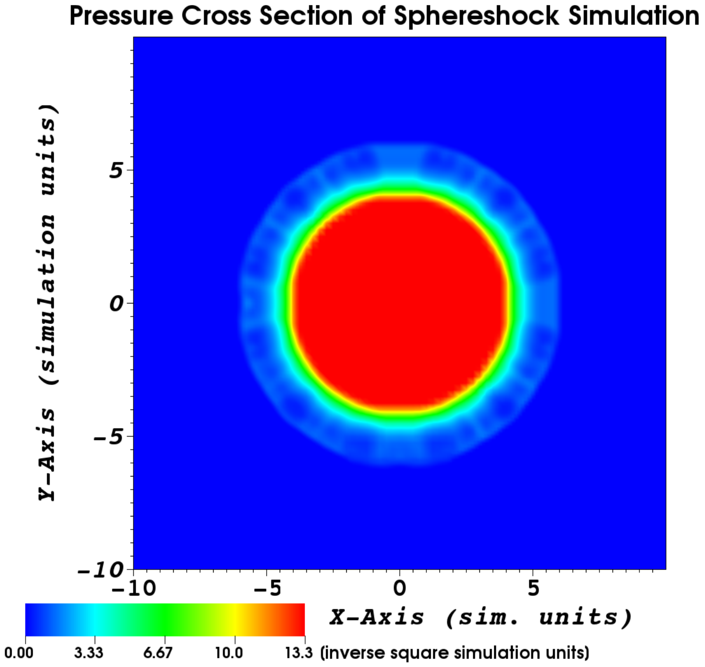

Figure 2. The $z = 0$ cross section through the `sphereshock` simulation after 80 time-steps, depicting the pressure p. Pressure, having units of mass per length per time squared, has units of inverse square simulation units because mass, length, and time are all measured in simulation units. One inverse square simulation unit is 5.55×10^{37} Pa.

We compared AMR to a control case with no refinement and to a case with one refined, nested box. We refer to the AMR case as `amr`, the nested box case as `nested`, and the control case with no refinement as `control`. The resolution of the refined box in `nested` and of the refined region in `amr`

[5] Technically, `CR2`'s implementation of the level mask is a non-negative real number, which is truncated to an integer before being used.

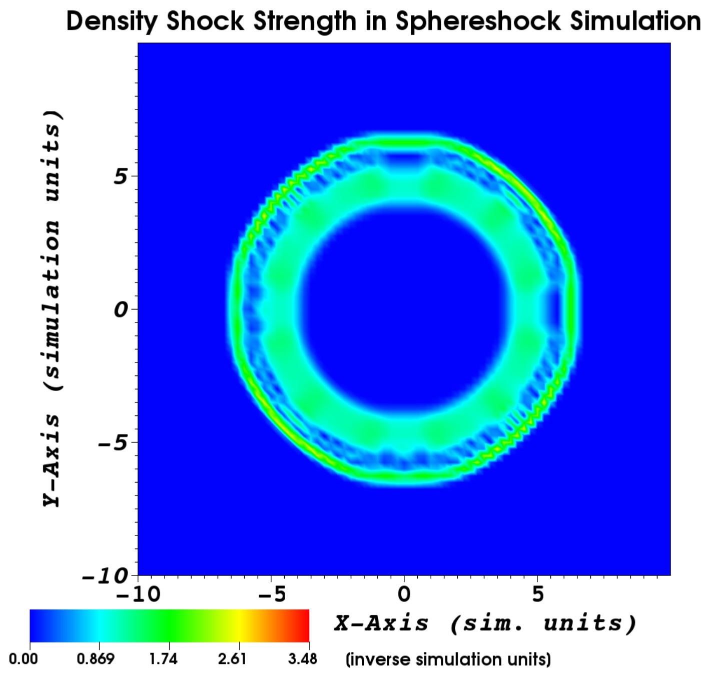

Figure 3. The $z = 0$ cross section through the `sphereshock` simulation after 80 time-steps, depicting the density shock strength χ_ρ as defined in Equation 1. The units of shock strength are inverse simulation units. One inverse simulation unit is 6.76×10^{-4} m^{-1}.

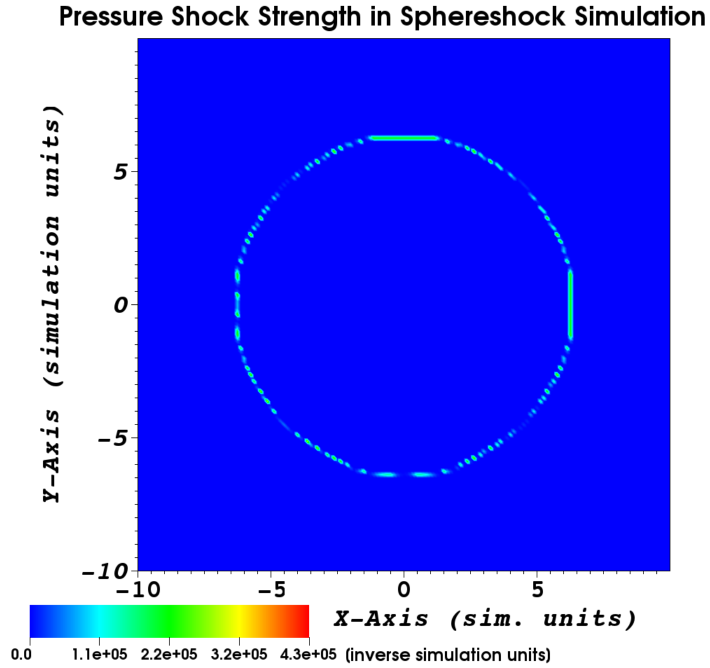

Figure 4. The $z = 0$ cross section through the `sphereshock` simulation after 80 time-steps, depicting the pressure shock strength χ_ρ as defined in Equation 1. The units of shock strength are inverse simulation units. One inverse simulation unit is 6.76×10^{-4} m^{-1}.

are the same: twice the resolution of the coarse region. Three experiments were performed for these three refinement methods: one where the resolution of the coarse level was varied, one where the the adaptive block size of `amr` was varied, and one where the radius of the level mask cubes (described in Section 2.2) of `amr` was varied. The runtime of each simulation was measured as well as the total number of grid points in the simulation domain, a proxy for memory usage.

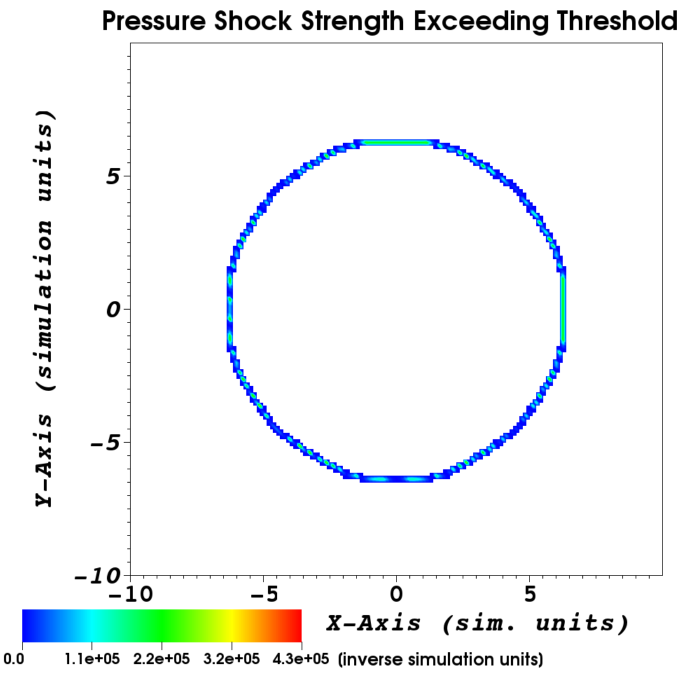

Figure 5. A cut-out of Figure 4 for where the pressure shock strength exceeds a threshold of 1000 inverse simulation units. What remains is considered the shock surface and is the output of the shock-tracking routine. One inverse simulation unit is 6.76×10^{-4} m^{-1}.

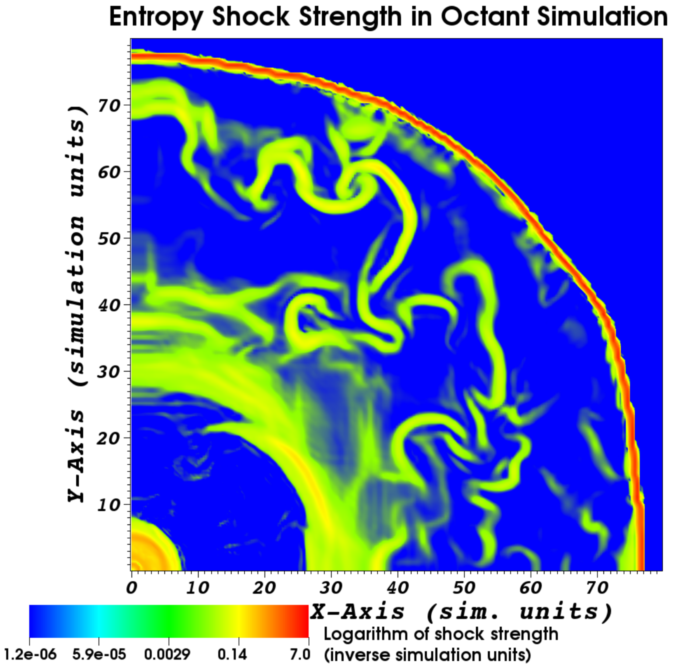

Figure 6. A cross-section of the `octant` simulation depicting the entropy shock strength.

The experiments where the adaptive block size was varied and the experiments where the level mask cube radius was varied all occurred at a resolution of 4 per simulation unit. The experiments where resolution was varied and the experiments where the adaptive block size was varied all occurred at a level mask cube radius of 0 simulation units. The adaptive block

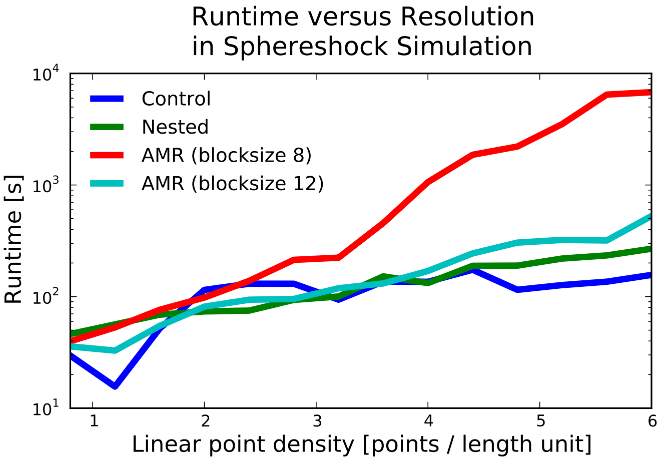

Figure 7. The time required to complete 80 time-steps of `control`, `nested`, and two variants of `amr`, one with blocksize of 8 coarse-grid spacings and one with blocksize of 12 coarse-grid spacings, as a function of coarse-grid resolution in the `sphereshock` simulation. Both variants of `amr` had a level mask cube radius of 0, meaning that the only points that received nonzero level-mask were those where the shock strength exceeded the set threshold. At resolutions higher than 4 points per simulation unit, the runtime `amr` with blocksize 8 is more than an order of magnitude larger than that of the other simulations, which are all within an order of magnitude of each other.

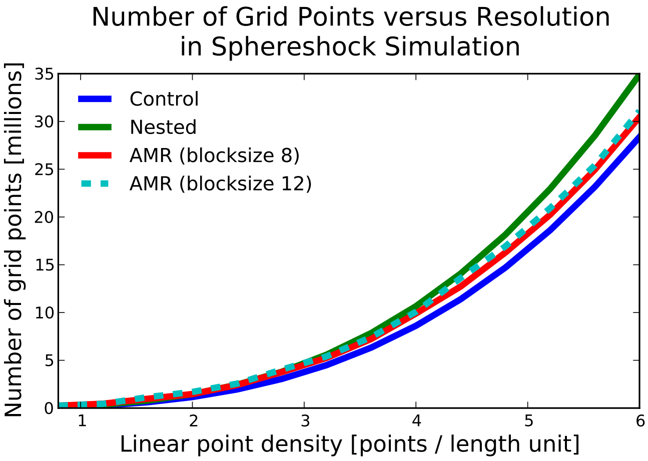

Figure 8. The number of simulation grid points used after 80 time-steps of `control`, `nested`, and two variants of `amr`, one with blocksize of 8 coarse-grid spacings and one with blocksize of 12 coarse-grid spacings, as a function of coarse-grid resolution in the `sphereshock` simulation. Both variants of `amr` had a level mask cube radius of 0, and both variants of `amr` used slightly fewer grid points than `nested`, but slightly more than `control`.

size of `amr` was 8 coarse-level spacings in the experiments where level mask cube radius was varied and was either 8 or 16 coarse-level spacings in the experiments where resolution was varied (see Figures 7 and 8).

2.3.2. *`octant` Simulation*

In the `octant` simulation, the 27 solar mass core-collapse supernova model of Woosley et al. (2002) was evolved post-bounce so that the shock could be seen (Figure 6). In this experiment, we selected a shock threshold of 3.0 inverse simulation units and entropy for our shock-tracking quantity. We attempted to apply adaptive mesh refinement to the shock surface and to evolve the simulation.

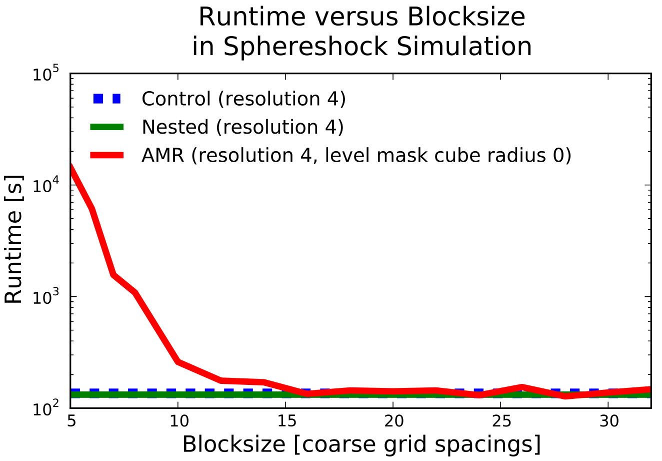

Figure 9. The time required to complete 80 time-steps of `control`, `nested`, and `amr`, each at a resolution of 4 points per simulation unit, versus the blocksize of `amr`. Because `control` and `nested` are independent of adaptive block size, they are constant and plotted here for comparison with `amr`. The level mask cube radius of `amr` was 0.

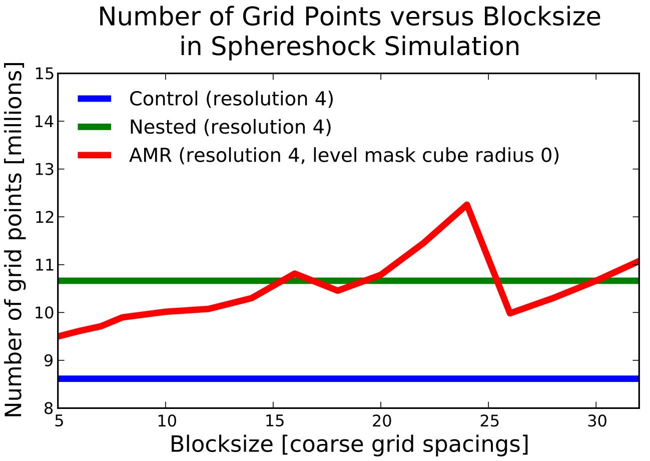

Figure 10. The number of simulation grid points used after 80 time-steps of `control`, `nested`, and `amr`, each at a resolution of 4 points per simulation unit, versus the blocksize of `amr`. Because `control` and `nested` are independent of adaptive block size, they are constant and plotted here for comparison with `amr`. The level mask cube radius of `amr` was 0.

3. RESULTS AND DISCUSSION

3.1. *`sphereshock` Results*

Figure 7 depicts the runtimes of the coarse-grid-only case `control`, the refined-box case `nested`, and the adaptive case `amr`, versus resolution of the coarse grid. Initially `amr` had only been run with a blocksize of 8 coarse-level spacings, which resulted in runtimes more than an order of magnitude longer than those of `control` and `nested`, which were basically equivalent. However, the results of Figure 9 motivated us to test `amr` with a blocksize of 12 coarse-level spacings, the result of which can be seen in Figure 7. Figure 9 shows that, at a resolution of 4 points per simulation units, `amr` with a blocksize of 12 performs no worse than twice as slow as `control` and `nested`, whereas `amr` with a blocksize of 8 performs a factor of 10 slower. However, the danger of increasing the block size is losing the advantage of AMR, which is minimizing the volume of simulation space that is refined. Thus, the number of grid points, a proxy for

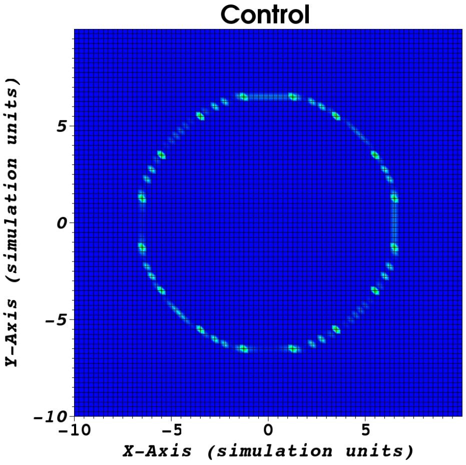

Figure 11. The coarse-grid-only structure of `control` at a resolution of 4 points per simulation unit.

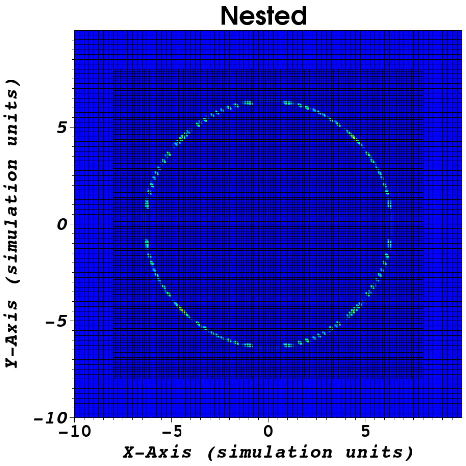

Figure 12. The coarse-grid and refined nested-box of `nested` at a resolution of 4 points per simulation unit. The refined region is shaded.

the memory usage of the simulation, goes up. However, Figure 10 demonstrates that the number of grid points in `amr` only weakly increases with blocksize, and is comparable to the number of points used in `control` and `nested`. In fact, Figure 8 shows that the number of grid points used by the two variants of `amr` is less that that of `nested` for every tested resolution, though any refinement uses more grid points than the coarse-level-only control, as expected. Figure 14 shows that even with this larger block size, the interior region is left unrefined.

The cube radius of the level mask-setting routine is a parameter whose implementation was motivated by problems experienced with `octant`. The goal was to move the edges of the refined region far enough from the shock front so that no abrupt changes crossed the boundary between the refined region and the unrefined region. Figures 15 and 16 demonstrate that the runtime of `amr` goes down while the number of grid points used goes up as the level mask cube radius is increased.

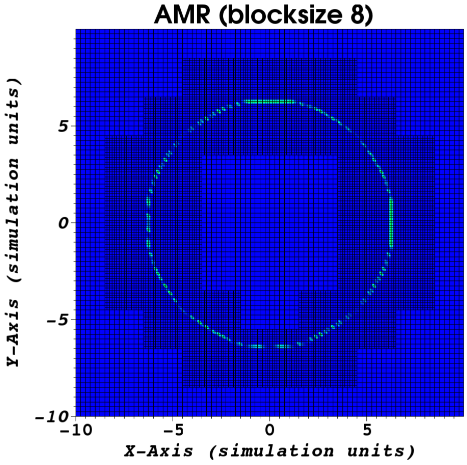

Figure 13. The coarse-grid and adaptively-refined regions of `amr` with blocksize of 8 coarse-grid spacings. The interior region is visible unrefined. The refined region is shaded.

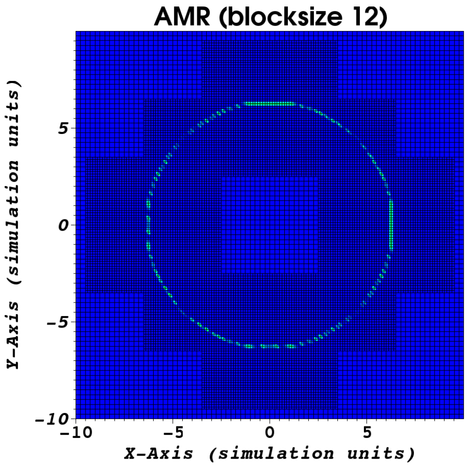

Figure 14. The coarse-grid and adaptively-refined regions of `amr` with blocksize of 12 coarse-grid spacings. The interior region is visible unrefined. The refined region is shaded.

3.2. `octant` *Results*

Our preliminary attempts to apply AMR to the `octant` simulation failed because negative temperatures appeared when the simulation was evolved. Negative temperatures are invalid for the equation of state used in the supernova, so the simulation was aborted. The reason for their appearance may be that interpolation between the coarse layer and the refined layer near the rapidly-changing shock lead to unphysical results. This phenomenon may have been hidden in `sphereshock` because of the very simple equation of state used there; `octant` uses a tabulated equation of state that may be more prone to this problem.

To attempt to solve this problem, we varied both the adaptive block size and the level mask cube radius to change the geometry of the refined region. However, the problem with negative temperatures only disappeared when the block size was so large that the refined region was basically a nested box, defeating the purpose of the adaptive mesh refinement.

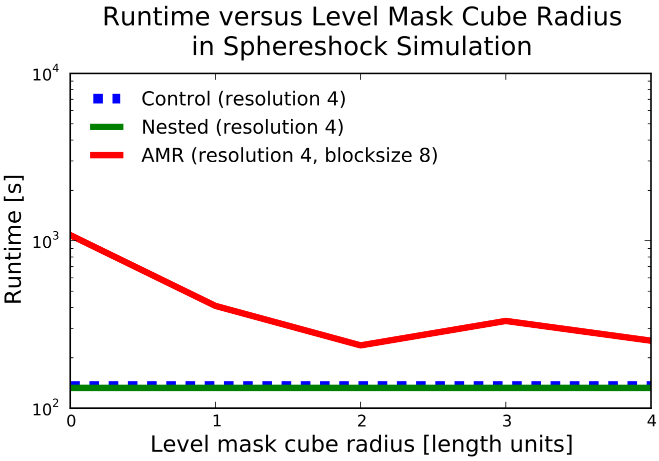

Figure 15. The time required to complete 80 time-steps of `control`, `nested`, and `amr`, each at a resolution of 4 points per simulation unit, versus the level mask cube radius of `amr`. Because `control` and `nested` are independent of level mask cube radius, they are constant and plotted here for comparison with `amr`. The adaptive block size of `amr` was 8.

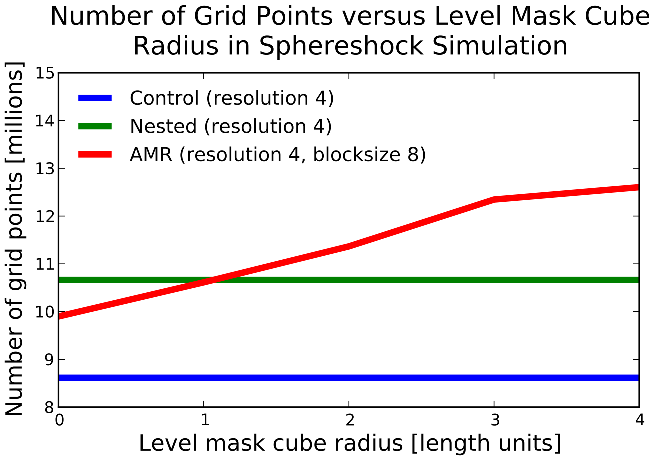

Figure 16. The number of simulation grid points used after 80 time-steps of `control`, `nested`, and `amr`, each at a resolution of 4 points per simulation unit, versus the level mask cube radius of `amr`. Because `control` and `nested` are independent of level mask cube radius, they are constant and plotted here for comparison with `amr`. The adaptive block size of `amr` was 8.

4. CONCLUSION

In this project we successfully implemented a shock-tracking routine which locates shock fronts by outputting a "shock strength" grid function, the shock surface being where the shock strength is large. This routine can be used in conjunction with any other routine or analytical tool that requires localization of the shock wave in real-time. The shock-tracker can localize multiple shocks in one simulation, and the shocks can have any shape.

We successfully coupled adaptive mesh refinement to the shock tracker in the simple, spherically-symmetric explosion simulation `sphereshock`. In doing so, we showed that an adaptive block size of 12 coarse-level spacings results in simulation runtimes that are comparable to runtimes of simulations with nested-box refinement and simulations with no refinement at all. The number of grid points used by these adaptive runs was less what was used by the nested-box runs, which is important because current supernova simulations are largely limited by available memory.

In a full supernova simulation such as `octant`, where the evolution steps are much more costly than in our simple `sphereshock` simulation, the runtime with AMR may be faster than the runtime with nested boxes because the amount of time spent doing evolution is proportional to the number of points that need to be evolved. Unfortunately, we were unable to test this possibility because of the interpolation problems mentioned in Section 3.2.

Adaptive mesh refinement is a promising technique that can reduce memory usage in supernovae simulations. If the adaptive block size is set correctly, the runtimes of simulations using AMR are comparable to those using nested boxes instead. In the future, we plan to investigate ways to successfully apply AMR to the `octant` simulation so that we can characterize the benefits of AMR in supernova simulations. We may also look to other supernova models to see if the problem persists. Additionally, we plan to apply AMR to other parts of the supernova, such as the proto-neutron star, to resolve them more effectively.

ACKNOWLEDGEMENT

The primary author wishes to thank Dr. Ernazar Abdikamalov, Dr. Sean M. Couch, Dr. Philipp Mösta, and Dr. Erik Schnetter for their invaluable contributions and advice, as well as the Laser Interferometer Gravitational-Wave Observatory for funding this work. Simulation cross-sectional plots were made with `VisIt`. All other plots were made with `matplotlib`.

REFERENCES

Berger, M. J., & Oliger, J. 1984, J. Comput. Phys., 53, 484
Bethe, H. A. 1990, Rev. Mod. Phys., 62, 801
Colgate, S. A., & White, R. H. 1966, ApJ, 143, 626
Löffler, F., Faber, J., Bentivegna, E., et al. 2012, Class. Quantum Grav., 29, 115001
Ott, C. D., Abdikamalov, E., Mösta, P., et al. 2013, ApJ, 768, 115
Woosley, S. E., Heger, A., & Weaver, T. A. 2002, Rev. Mod. Phys., 74, 1015
Woosley, S. E., & Janka, H.-T. 2005, Nature Physics, 1, 147