Los Alamos
NATIONAL LABORATORY
EST.1943

LA-UR-16-25578
Approved for public release; distribution is unlimited.

Title: An Adaptive Mesh Algorithm: Mesh Structure and Generation

Author(s): Scannapieco, Anthony J.

Intended for: LA-UR

Issued: 2016-07-28 (rev.1)

Disclaimer:
Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

An Adaptive Mesh Algorithm: Mesh Structure and Generation

A. J. Scannapieco

June 21, 2016

1 Adaptive Mesh Refinement

The purpose of Adaptive Mesh Refinement is to minimize spatial errors over the computational space *not* to minimize the number of computational elements. The additional result of the technique is that it *may* reduce the number of computational elements needed to retain a given level of spatial accuracy. Adaptive mesh refinement is a computational technique used to dynamically select, over a region of space, a set of computational elements designed to minimize spatial error in the computational model of a physical process. The fundamental idea is to increase the mesh resolution in regions where the physical variables are represented by a broad spectrum of modes in k-space, hence increasing the effective global spectral coverage of those physical variables. In addition, the selection of the spatially distributed elements is done dynamically by cyclically adjusting the mesh to follow the spectral evolution of the system.

Over the years three types of AMR schemes have evolved; block, patch and locally refined AMR. In block and patch AMR logical blocks of various grid sizes are overlaid to span the physical space of interest, whereas in locally refined AMR no logical blocks are employed but locally nested mesh levels are used to span the physical space. The distinction between block and patch AMR is that in block AMR the original blocks refine and coarsen entirely in time, whereas in patch AMR the patches change location and zone size with time.

1.1 Computational Errors

There are two types of computational errors, *temporal* and *spatial*, which are defined in terms of a power of either time step or minimum zone width that represent the

order of the error terms in the discrete solution of the physical equations. Temporal errors are proportional to a power of the time step and are intrinsic to the temporal discretization in the solution method.

$$E_\tau \propto (\delta\tau)^o$$

where o is the order of the temporal error. Spatial errors are set by the size and distribution of the computational elements spanning the physical system and are intrinsic to the spatial discretization and solution method.

$$E_s \propto (\delta x)^o$$

where o is the order of the spatial error. **Therefore decreasing the zone size decreases the spatial error.**

Anywhere in space where the bandwidth in k-space of a spatial structure is broad the structure must be resolved at a finer resolution in physical space to enhance the number of modes in k-space that represent the structure. Where there is an *impedance* jump in a physical variable, the range of spectral wavelengths necessary to represent the physical variable becomes large. These impedance jumps occur at material boundaries where there is a change in density, sound speed, opacity, compressibility, shear modulus, etc. Impedance jumps also occur at shock fronts where there is an abrupt change in density, internal energy and momentum, as well as at radiation fronts where there is an abrupt change in ion, electron and radiation temperatures (e.g. Marshak waves).

1.1.1 Justification for Local Refinement

A justification for local refinement as a tool to decrease spatial errors can be found in three facts. The first fact is that a function $f(x)$ can be Fourier decomposed over a *finite* set of modes defined by a discrete nonuniform set of sampling points in space, and the Fourier coefficients can be found exactly. **The second fact is that the distance between sampling points in space is equivalent to the bandwidth of the modes in k-space. The denser the points in space the larger the bandwidth in k-space and the smaller the error in the Fourier representation of the function.** The third fact is that if the function $f(x)$ is convolved with a sampling function such as a top hat function, that has a value of one over the sampling points where the function $f(x)$ is rapidly varying in space and a value of zero elsewhere, the coefficients of the Fourier transform of the function remain virtually unchanged, thus retaining the spectral content of the function over the sampled modes.

1.2 Fourier analysis

If a (reasonably well-behaved) function is periodic, then it can be written as a discrete sum of trigonometric or exponential functions at discrete frequencies. A general function that isn't necessarily periodic (but that is still reasonably well-behaved) can be written as a continuous integral of trigonometric or exponential functions with a continuum of possible frequencies. Both types of Fourier decompositions are illustrated in the next four sections.

1.2.1 Discrete Fourier trigonometric series

In one dimension, $2N$ equally spaced points are chosen to span a length L. The points in space at which the function $f(x_p)$ are defined are

$$x_p = p\frac{L}{(2N-1)} \tag{1}$$

and the n^{th} wave vector is

$$k_n = \frac{2\pi n}{L} \tag{2}$$

Increasing the number of points, $2N$ that spans the length, L increases the bandwidth of the decomposition and decreases the spatial error.

The discrete Fourier decomposition of $f(x_p)$ is

$$f(x_p) = \sum_{n=0}^{2N-1} \left(a_n \cos(k_n x_p) + b_n \sin(k_n x_p)\right) \tag{3}$$

$$a_n = \frac{1}{2N}\sum_{p=0}^{2N-1} f(x_p)\cos(k_n x_p) \qquad n = 0, 1, ..., 2N-1 \tag{4}$$

$$b_n = \frac{1}{2N}\sum_{p=0}^{2N-1} f(x_p)\sin(k_n x_p) \qquad n = 0, 1, ..., 2N-1 \tag{5}$$

where a_n and b_n represent the Fourier transform functions for the *cosine* and *sine* terms.

1.2.2 Fourier trigonometric series

On the interval $0 \leq x \leq L$

$$f(x) = (a_0/2) + \sum_{n=1}^{\infty}(a_n \cos(2\pi nx/L) + b_n \sin(2\pi nx/L)) \tag{6}$$

$$a_n = \frac{2}{L}\int_0^L f(x)\cos(2\pi nx/L)dx \tag{7}$$

$$b_n = \frac{2}{L}\int_0^L f(x)\sin(2\pi nx/L)dx \tag{8}$$

$$\int_0^L \sin(\frac{2\pi nx}{L})\cos(\frac{2\pi mx}{L})dx = 0$$

$$\int_0^L \cos(\frac{2\pi nx}{L})\cos(\frac{2\pi mx}{L})dx = \frac{L}{2}\delta_{nm}$$

$$\int_0^L \sin(\frac{2\pi nx}{L})\sin(\frac{2\pi mx}{L})dx = \frac{L}{2}\delta_{nm}$$

where a_n and b_n represent the Fourier transform functions for the *cos* and *sin* terms.

1.2.3 Fourier exponential series

$$f(x) = \sum_{n=-\infty}^{\infty} C_n e^{i2\pi nx/L} \tag{9}$$

$$C_n = \frac{1}{L}\int_0^{\infty} f(x)e^{-i2\pi nx/L}dx \tag{10}$$

$$\int_0^{\infty} e^{i2\pi nx/L}e^{-i2\pi mx/L}dx = L\delta_{nm}$$

where δ_{nm} is the Kronecker delta, which is defined as

$$\begin{aligned} \delta_{nm} &= 0 \qquad & n \neq m \\ \delta_{nm} &= 1 \qquad & n = m \end{aligned}$$

1.2.4 Fourier transforms

$$f(x) = \int_{-\infty}^{\infty} C(k) e^{ikx} dk \tag{11}$$

$$C(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(x) e^{-ikx} dx \tag{12}$$

$$P(k) = C(k) C^*(k) \tag{13}$$

where $C(k)$ is the Fourier transform and $P(k)$ is the power spectrum in k-space.

1.3 Power spectra of spatial structures

The power spectrum of several spatial structures is calculated in one dimension. The decay rate of the power spectrum is a measure of the bandwidth of the structure in k-space. The bandwidth of each structure is used as a justification for the rules chosen in the algorithm presented in this paper. The decay rate of the power spectrum is important in deciding where to concentrate the mesh. The slower the decay the finer the structure must be resolved in space, so as to maximize its bandwidth in k-space. For each spatial variation chosen below, both the Fourier transform $C(k)$ and the power spectrum $P(k)$ are calculated.

1.3.1 Constant

$$f(x) = A \tag{14}$$

$$C(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} A e^{-ikx} dx \tag{15}$$

$$C(k) = \delta(k) \tag{16}$$

$$P(k) = \delta^2(k) \tag{17}$$

1.3.2 Gradient

$$f(x) = Ax \tag{18}$$

$$C(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Axe^{-ikx} dx \tag{19}$$

$$C(k) = iA\frac{d\delta(k)}{dk} \tag{20}$$

$$P(k) = A^2(\frac{d\delta(k)}{dk})^2 \tag{21}$$

1.3.3 Gaussian

$$f(x) = Ae^{-\alpha x^2} \tag{22}$$

$$C(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Ae^{-\alpha x^2} e^{-ikx} dx \tag{23}$$

$$C(k) = \frac{Ae^{-k^2/4\alpha}}{2\sqrt{(}\pi\alpha)} \tag{24}$$

$$P(k) = \frac{A^2 e^{-k^2/2\alpha}}{4\pi\alpha} \tag{25}$$

1.3.4 Exponential

$$f(x) = Ae^{-b|x|} \tag{26}$$

$$C(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Ae^{-b|x|} e^{-ikx} dx \tag{27}$$

$$C(k) = \frac{Ab}{\pi(b^2 + k^2)} \tag{28}$$

$$P(k) = \frac{A^2 b^2}{\pi^2 (b^2 + k^2)^2} \tag{29}$$

1.3.5 Unit step function

$$\begin{aligned} f(x) &= 0 \qquad & x < 0 \\ f(x) &= 1 \qquad & x > 0 \end{aligned} \tag{30}$$

$$C(k) = \frac{1}{2\pi} \int_0^\infty e^{-ikx} dx \tag{31}$$

$$C(k) = \frac{1}{2}\delta(k) - \frac{i}{2\pi k} \tag{32}$$

$$P(k) = \frac{1}{4}\delta^2(k) + \frac{1}{4\pi^2 k^2} \tag{33}$$

1.3.6 Square wave

$$\begin{aligned} f(x) &= A \qquad & -a \le x \le a \\ f(x) &= 0 \qquad & x < -a, \;\; x > a \end{aligned} \tag{34}$$

$$C(k) = \frac{1}{2\pi} \int_{-a}^{a} A e^{-ikx} dx \tag{35}$$

$$C(k) = \frac{A \sin(ka)}{\pi k} \tag{36}$$

$$P(k) = \frac{A^2 \sin^2(ka)}{\pi^2 k^2} \tag{37}$$

Both a constant and a uniform gradient are peeked only at a single wave vector $k = 0$ thus no refinement is necessary for these spatial variations in the physical variables. A power spectrum of a Gaussian spatial variation decays as a Gaussian in k-space. The decay rate is dependent on the value of α in the Gaussian. The variable α determines the amount of spatial resolution needed to resolve the Gaussian in k-space. The exponential spatial variation decays as the fourth power of the wave vector. The exponential variation requires a moderate amount of resolution to resolve the bandwidth. However, it is clear that spatial discontinuities decay slowly: inversely with the square of the wave vector. Therefore, discontinuities have a broad bandwidth of modes, which necessitates resolving these structures at the finest mesh zone size in an AMR calculation.

1.4 Mesh Structure

1.4.1 AMR Data Structure and Layout

The type of AMR described herein is a locally refined AMR. This technique minimizes the number of computational elements necessary to span the physical space. Although, locally refined AMR obviates the question of flux continuity across block or patch boundaries, flux continuity must be maintained across cell boundaries.

The structure and generation of the mesh is the first consideration in any adaptive mesh algorithm. In the algorithm described here, at any point in physical space only *one* zone exists at whatever level of mesh that is appropriate for that physical location. Parent zones, or zones that are more coarse then the local zone, are not retained. This is the description of a so-called **flat mesh**.

Although there are two mesh structures, the thermodynamic mesh and the duel kinematic mesh; once the *new* thermodynamic mesh is defined the *new* duel kinematic mesh is reestablished using the *new* thermodynamic mesh. This procedure is mirrored in the map of the kinematic variables. The map of the kinematic variables is done by calculating the contributions of kinematic variables on the *old* thermodynamic mesh, mapping the kinematic variable contributions onto the *new* thermodynamic mesh and then synthesizing the mapped kinematic variables on the *new* kinematic mesh.

Neighbor index information is carried in two arrays $N_{low}(nz, idir)$ and $N_{hgh}(nz, idir)$, which are functions of zone number nz and direction $idir$. A neighbor index array of a zone points to the neighbor with the lower logical index in the transverse direction, which is illustrated in Figures 1 and 2 in two and three dimensions. All other neighbors of a zone nz are found by using the N_{low} and N_{hgh} arrays of the neighbor zones of the original zone nz. Neighbor zones at the same and finer levels have been chosen to illustrate this point. The neighbor zones of a zone at a finer level than its neighbors are illustrated in two and three dimensions in Figures 3 and 4. It is unnecessary to describe the one dimensional adaptive mesh because it is a simple subset of the two dimensional mesh. Additional variables and arrays necessary to define the structure of the mesh are displayed in Table 1.

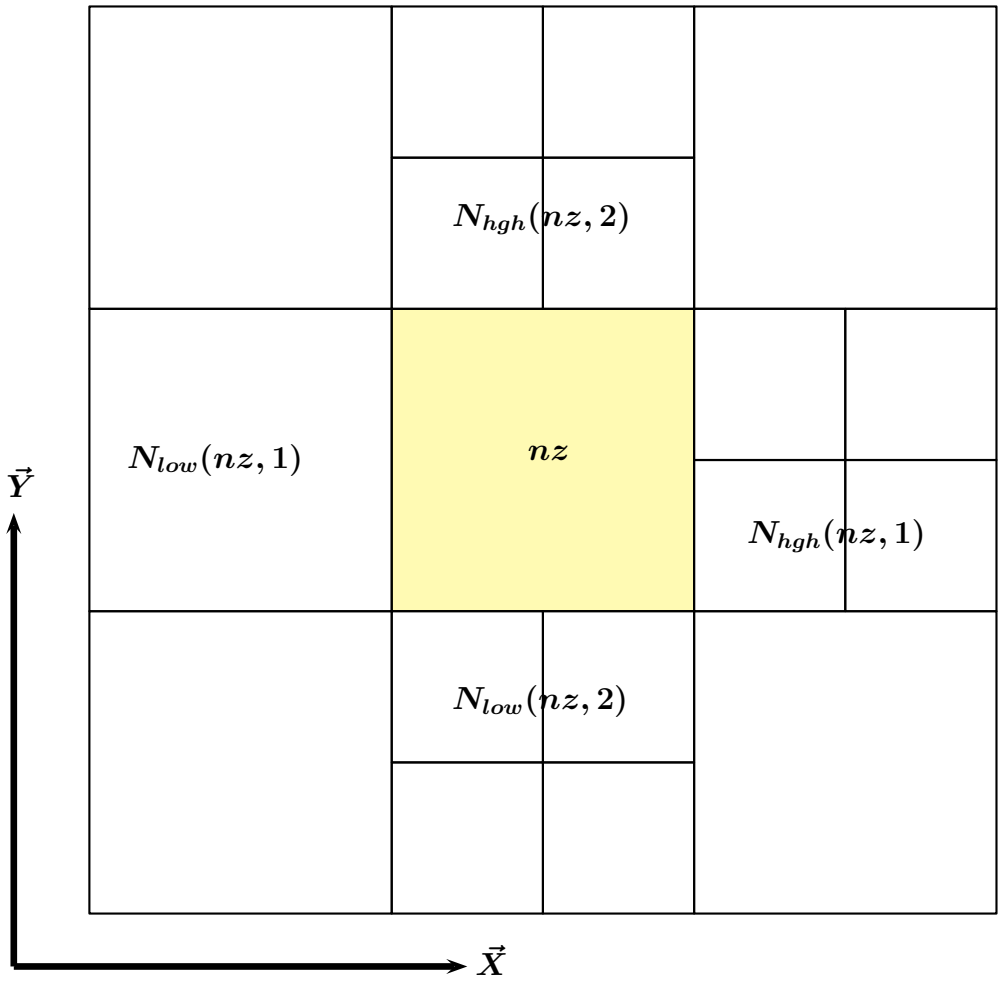

Figure 1: Two dimensional AMR data structure and neighbor index layout

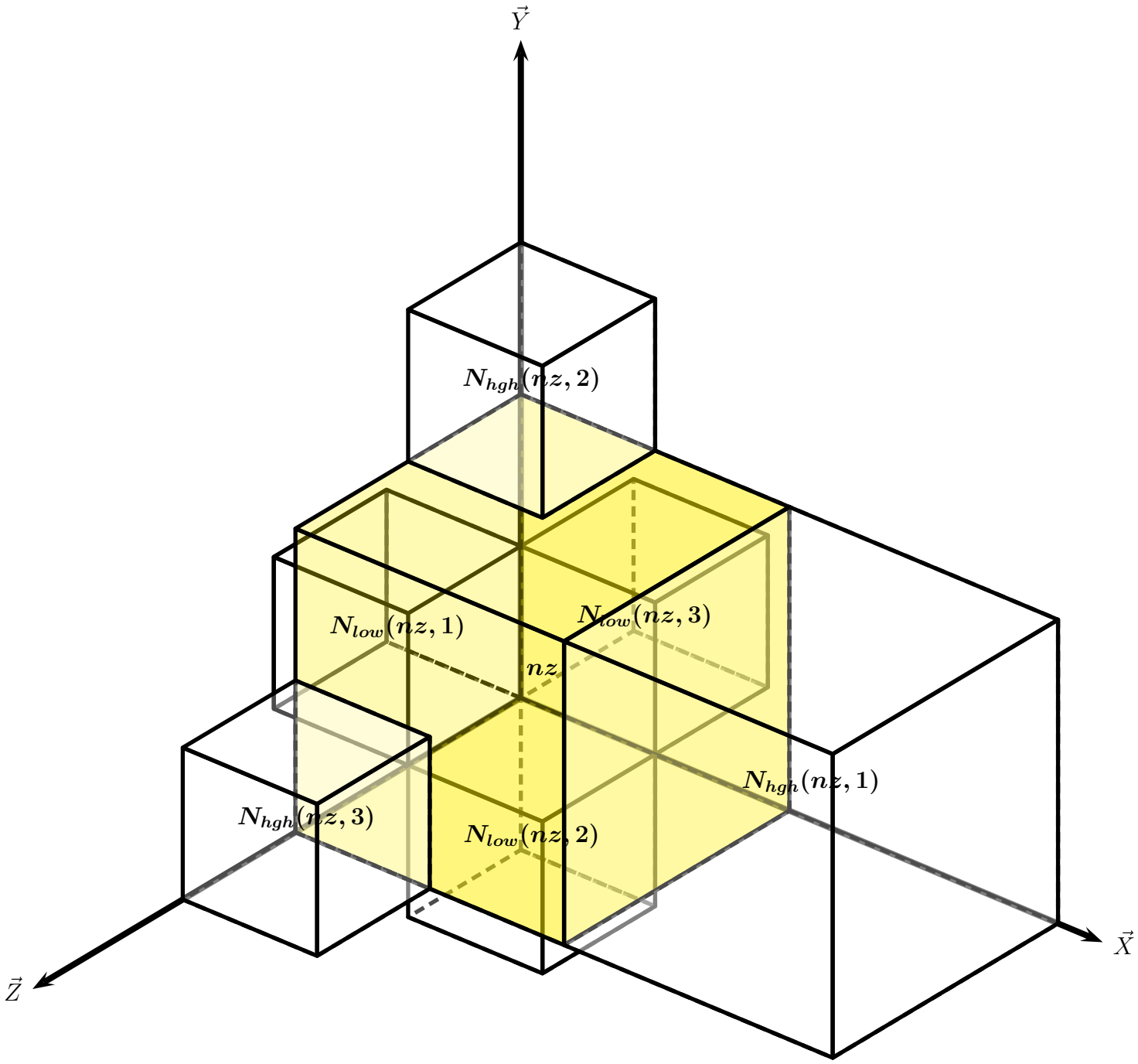

Figure 2: Three dimensional AMR data structure and neighbor index layout

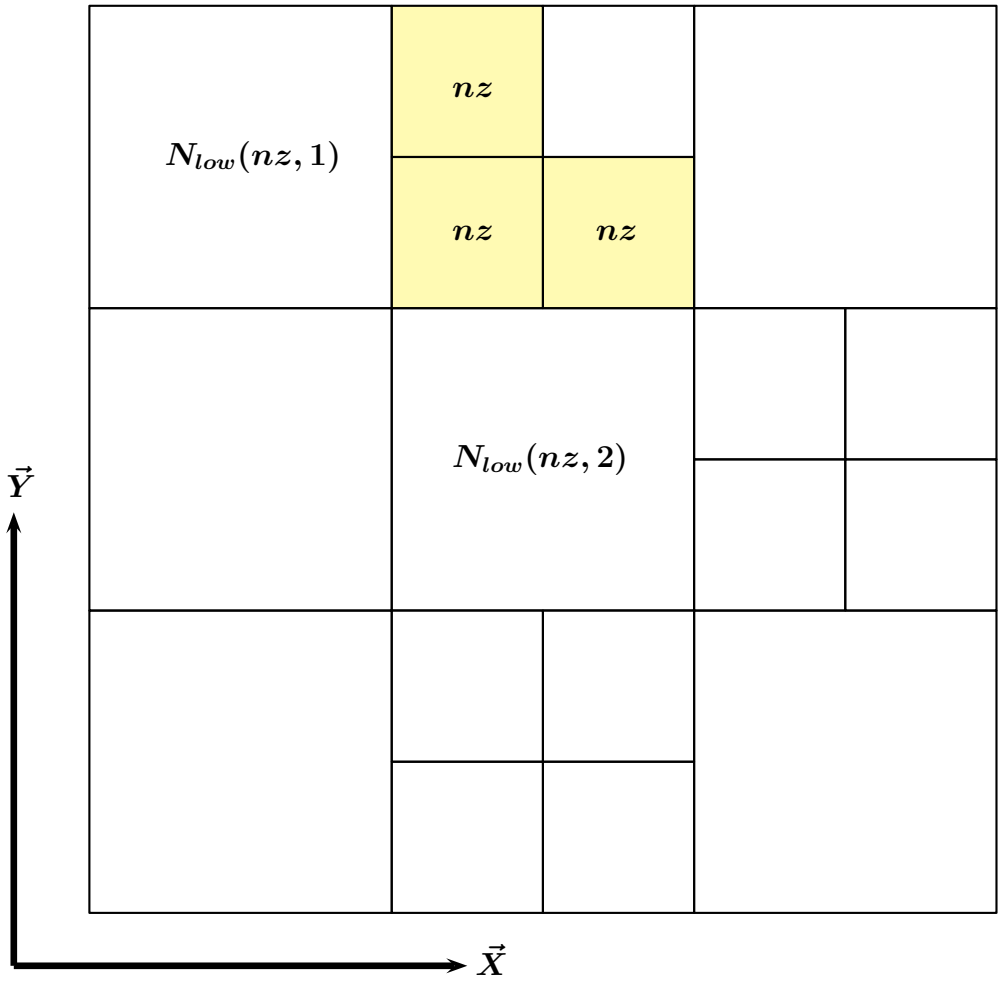

Figure 3: Two dimensional neighbor zones at a coarser level than zone nz. The coarser neighbor zones are neighbors to multiple nz zones

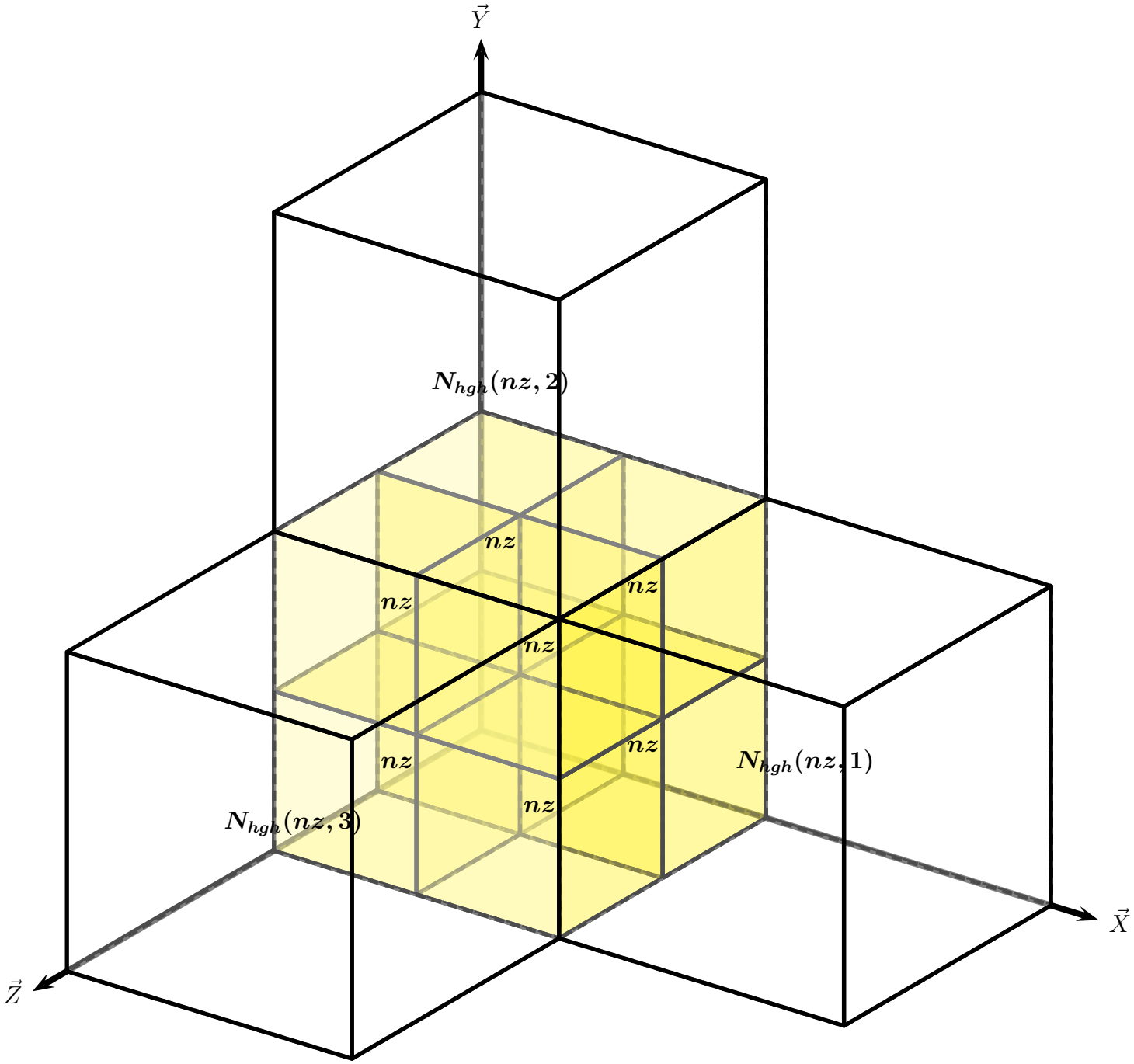

Figure 4: Three dimensional neighbor zones at a coarser level than zone nz. The coarser neighbor zones are neighbors to multiple nz zones

$numdim$	integer number of directions
$geop$	integer index of geometry $geop = 1$, rectilinear $geop = 2$, cylindrical $geop = 3$, spherical $numdim = 1$, $geop = 1, 2, 3$ $numdim = 2$, $geop = 1, 2$ $numdim = 3$, $geop = 1$
$levmx$	integer index of the finest level mesh
$nbe(lev) = 2^{lev}$	integer number of boundary zones. Only one boundary zone at each mesh level is retained.
$imax(lev, idir)$	integer number of zones at level lev in direction $idir$, which includes the nbe zones at the lower and upper edges of the mesh in the $idir$ direction.
$limax(lev, idir)$	integer number of the first upper boundary zone at level lev in direction $idir$. $limax(lev, idir) = imax(lev, idir) - nbe(lev) + 1$
$dxs(lev, idir)$	zone width of the level lev in direction $idir$.
$level(nz)$	mesh level of zone nz. $level(nz) = 0$, coarsest level mesh. $level(nz) = levmx$, finest level mesh.
$iv(nz, idir)$	integer index of zone nz in direction $idir$.
$indx(nz)$	generalized index of zone nz, in 1, 2 and 3 dimensions $indx(nz) = iv(nz, 1)$ $indx(nz) = iv(nz, 1)+$ $\quad imax(level(nz), 1) * (iv(nz, 2) - 1)$ $indx(nz) = iv(nz, 1)+$ $\quad imax(level(nz), 1) * (iv(nz, 2) - 1)+$ $\quad imax(level(nz), 1) *$ $\quad imax(level(nz), 2) * (iv(nz, 3) - 1)$
$dxe(nz, idir)$	zone width of zone nz in direction $idir$.
$xe(nz, idir)$	geometrical center of zone nz in direction $idir$.

Table 1: Mesh variables and arrays

1.4.2 Mesh Generation and Evolution

The dynamic creation of a locally refined computational mesh is made practical by a judicious selection of four mesh rules.

1. A zone refines completely in all directions.

2. Coarsening must produce a *parent* zone, in which all daughter zones that comprise the parent zone must be scheduled to be coarsened.

3. For any mesh level, neighbor zones come from the same level, the next finer level, or the next coarser level.

4. Signal propagation over a computational cycle is anticipated by ensuring that all captured physics is embedded at least one zone inside its capture level.

With these rules the mesh is evolved via a *mesh potential* designed to concentrate the finest mesh in regions where the physics is modally dense, and coarsen zones in regions where the physics is modally sparse. A mesh potential $\Phi(nz)$, is a zonal integer array that indicates whether a zone should be coarsened, refined, or left unchanged, depending upon whether the mesh potential is negative, positive or zero. A zone can be coarsened only if all the zones that make up the parent zone have a negative mesh potential and they are all at the same mesh level.

A sequential process is executed once at the end of each hydro cycle. This process is characterized by three distinct steps; *release*, *capture*, and *nesting*.

1. Release (i.e. mesh coarsening, $\Phi(nz) < 0$)

   (a) At the end of each hydro cycle all zones are scheduled for release, $\Phi(nz) = -1000000$.

   (b) If the release of a zone causes a variation in the zone that exceeds a predetermined fraction of the extensive variables of the daughter zones, the zone is retained, $\Phi(nz) = 0$.

   (c) If all the daughter zones of a parent zone cannot be released, all are retained. $\Phi(nz)$ is set to zero for each daughter zone.

   (d) Finally, if a zone is scheduled to be coarsened and a neighbor zone at the same level is scheduled to be unchanged the zone is also scheduled to be unchanged, $\Phi(nz) = 0$.

2. Capture (i.e. mesh refinement, $\Phi(nz) > 0$)

(a) Material interfaces or contact discontinuities are embedded in the finest level mesh.

(b) Shocks are captured via the artificial viscosity and are embedded at the *shock capture* level. The *shock capture* level is usually the finest level.

(c) In addition to the above capture criteria the zonal variation in the interpolated kinetic, electron, ion, and radiation energy densities are checked over the mesh to ensure that the variation is maintained within set bounds. This is accomplished by the use of an energy error estimator define as

$$e = 2(E_{max} - E_{min})/(E_{max} + E_{min})$$

where E_{min} and E_{max} are the minimum and maximum zonal values of each of the energy densities obtained by interpolation from the zonal edge values in each direction. A zone is scheduled to be refined if $e > 2.0$, $\Phi(nz) > 0$. If however, $0.5 \leq e \leq 2.0$ the zone is retained, $\Phi(nz) = 0$, while the zone is scheduled to be coarsened if $e < 0.5$, $\Phi(nz) < 0$.

3. Nesting (i.e. mesh level transition)

The mesh potential generated by the capture and release algorithms is checked for continuity across different mesh level boundaries. If there is a discontinuity in the mesh of more than one level the mesh potential is adjusted in the coarser level zone to ensure that the one level jump criteria is maintained. This process ensures that refinement will take place in the coarser level, guaranteeing that nesting will occur. The algorithm proceeds sequentially in three steps, each of which proceeds from the finest to the coarsest mesh level.

(a) First, ensure that a neighbor zone nb at the same level as zone nz will not have a mesh potential $\Phi(nb)$ that is less than the mesh potential $\Phi(nz)-1$ of zone nz.

(b) Second, if a neighbor zone nb is at a finer level then zone nz and the mesh potential of the neighbor zone $\Phi(nb) > -2$ set the mesh potential $\Phi(nz)$ of zone nz to

$$\Phi(nz) = \max(1 + (\Phi(nb) + 1)/2, \Phi(nz))$$

Also calculate the maximum value of the mesh potential Φ_{max} over the mesh.

(c) Third, proceed from a mesh potential value $\Phi = \Phi_{max}$, to a mesh potential value of $\Phi = 0$. If the mesh potential of zone nz is equal to Φ, then set the neighbor zones nb at the same mesh level as nz to

$$\Phi(nb) = \max(\Phi - 1, \Phi(nb))$$

The mesh potential $\Phi(nz)$ generated by the above algorithm is then used to create the new mesh, and map the physical variables. At the end of the hydro cycle, after the mesh potential is created, if no mesh refinement nor mesh coarsening are needed the reestablishment of the mesh and the mapping algorithms are skipped for that cycle.

1.4.3 Demonstration of the Algorithm

To illustrate the evolution of an AMR mesh using the algorithm described, a two dimensional blast problem was run, which had four materials and four AMR mesh levels. The mesh and the density are shown at three problem times in Figures 5, 6, and 7. The problem evolution illustrates how the algorithm captures the material boundaries, shocks and change in the kinetic and internal energy densities while nesting the various mesh levels to ensure that the four mesh criteria used in the algorithm are satisfied.

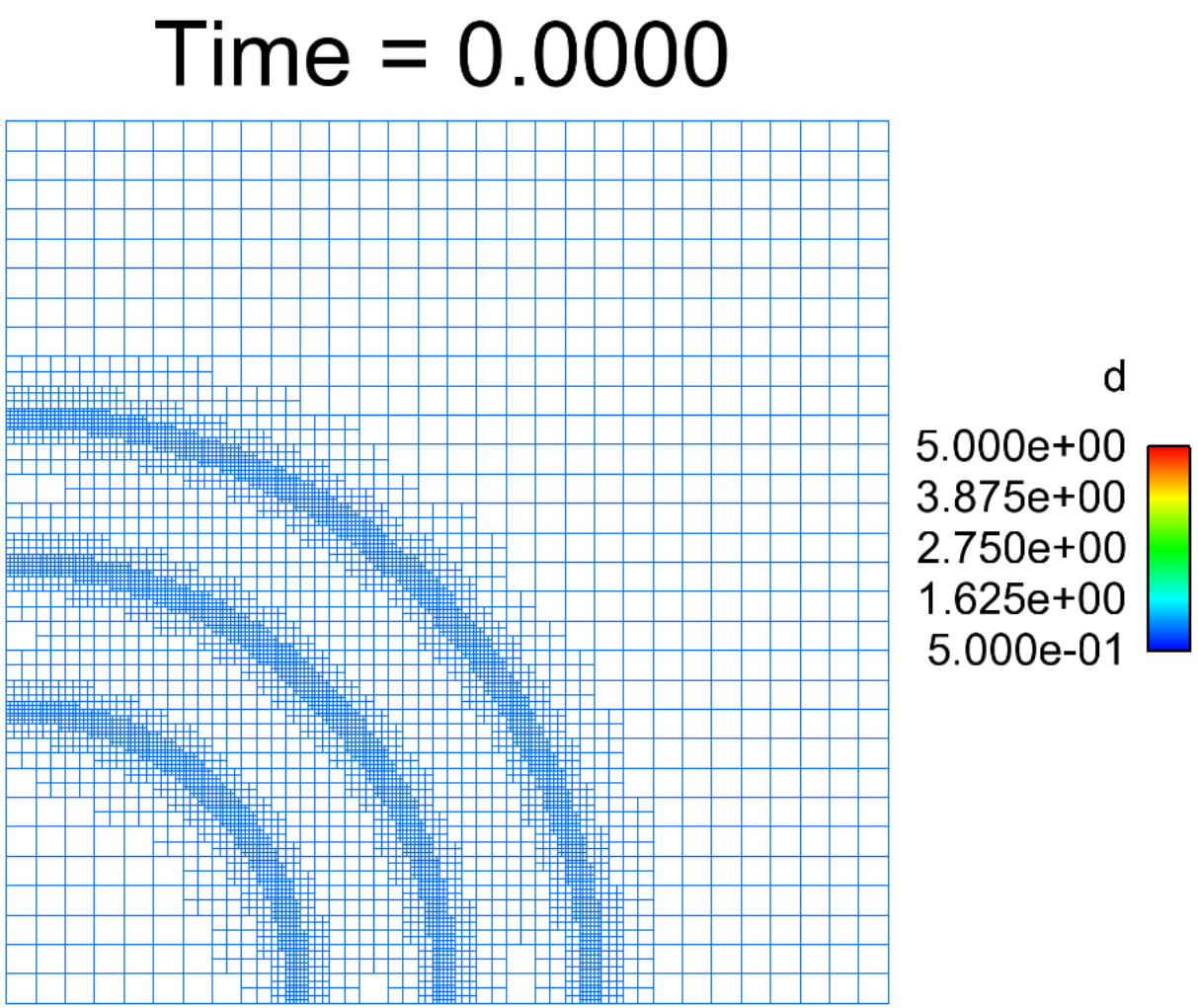

Figure 5: Blast mesh and density at $0.0000\mu s$

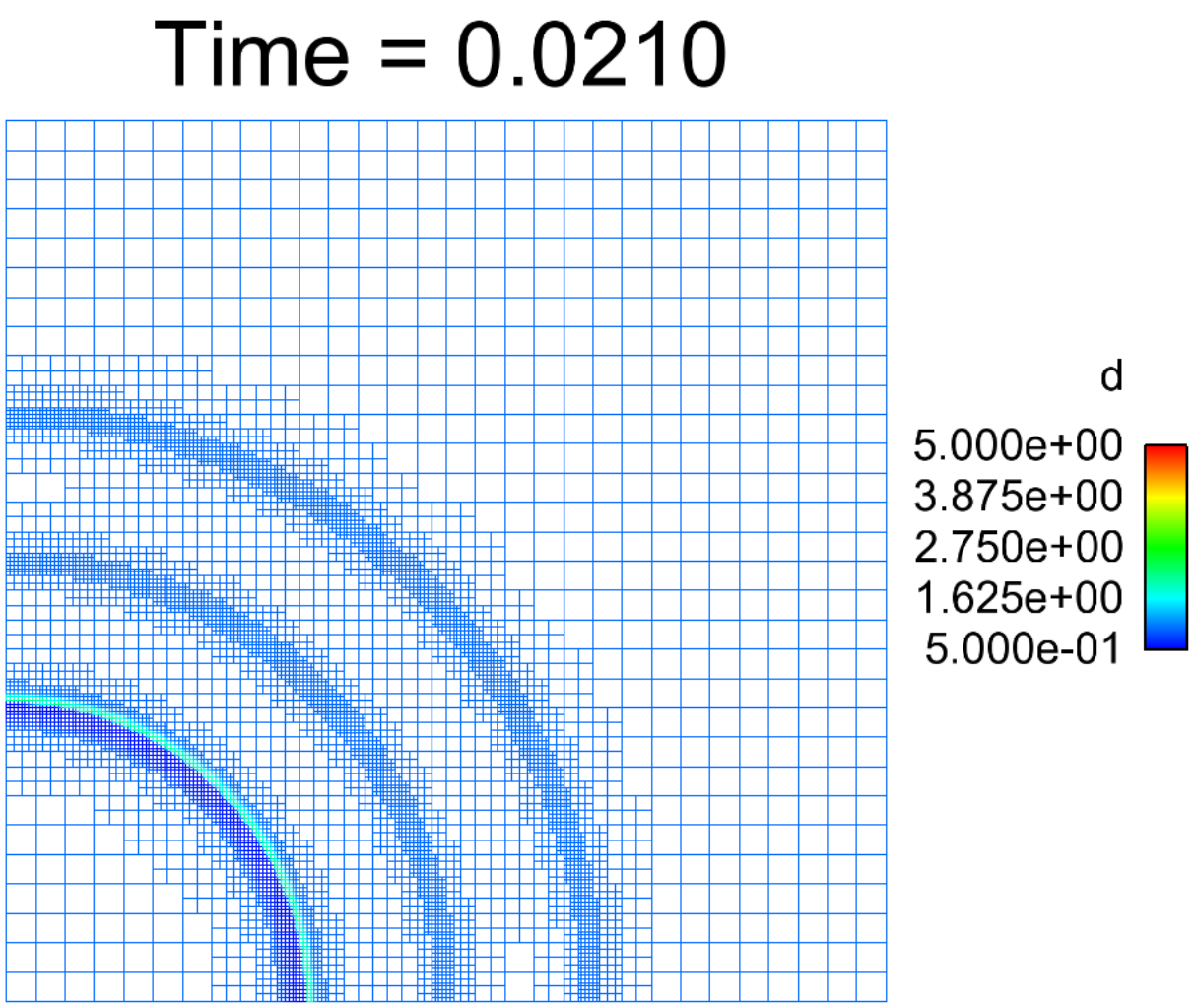

Figure 6: Blast mesh and density at $0.0210\mu s$

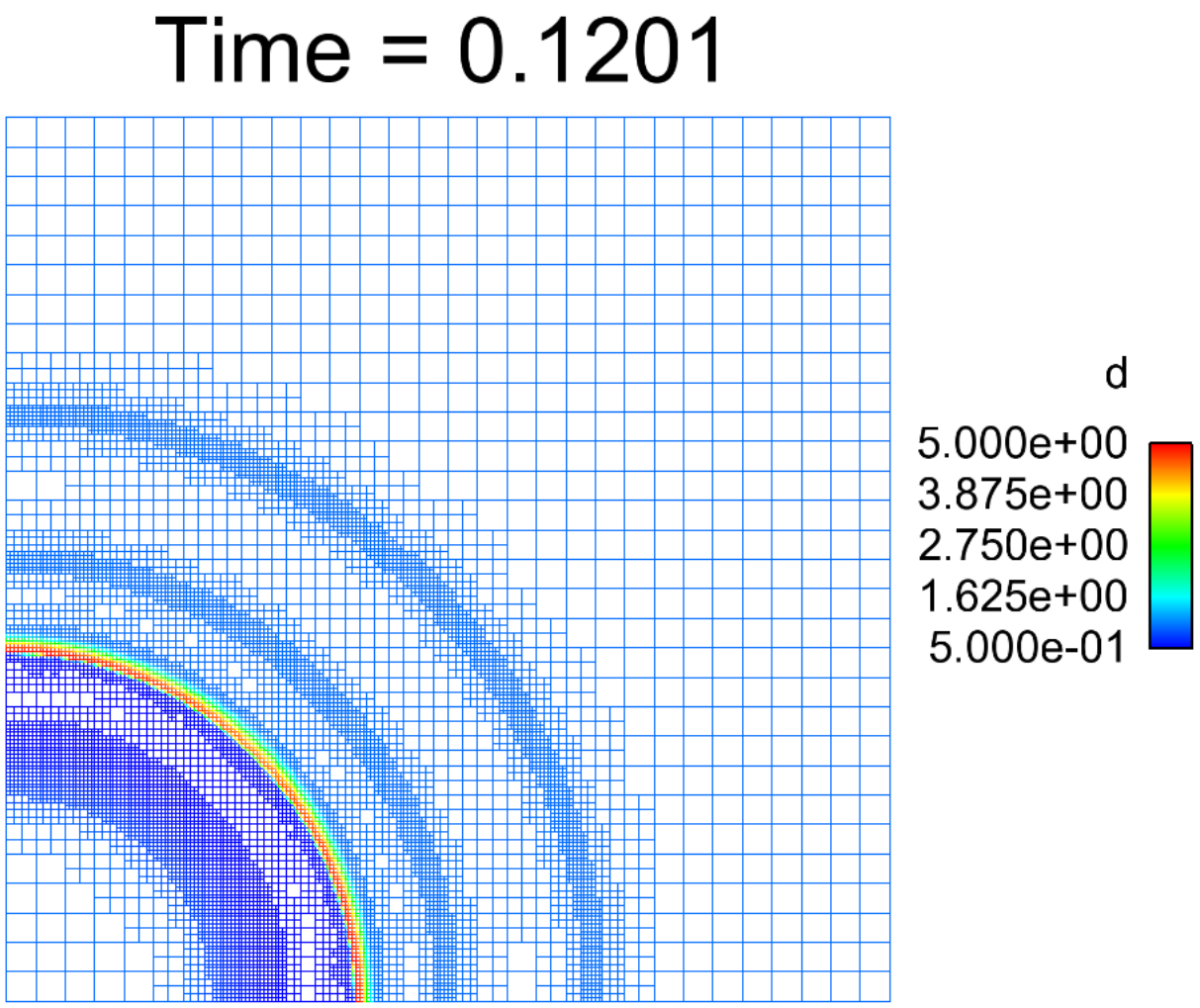

Figure 7: Blast mesh and density at $0.1201\mu s$

1.4.4 Conclusion

Once the structure of the *new* mesh is defined, it then remains to map the *old* mesh data onto the *new* mesh. This process is the subject of the next report "An Adaptive Mesh Algorithm: Mapping the Mesh Variables".