LLNL-CONF-424743

LAWRENCE LIVERMORE NATIONAL LABORATORY

Laser Ray Tracing in a Parallel Arbitrary Lagrangian Eulerian Adaptive Mesh Refinement Hydrocode

N. D. Masters, T. B. Kaiser, R. W. Anderson, D. C. Eder, A. C. Fisher, A. E. Koniges

March 1, 2010

Inertial Fusion Sciences and Applications (IFSA)
San Francisco, CA, United States
September 6, 2009 through September 11, 2009

Laser Ray Tracing in a Parallel Arbitrary Lagrangian-Eulerian Adaptive Mesh Refinement Hydrocode

N D Masters[1], T B Kaiser[1], R W Anderson[1], D C Eder[1], A C Fisher[1], A E Koniges[2]

[1]Lawrence Livermore National Laboratory, Livermore, CA U.S.A.

[2]National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory, Berkeley, CA U.S.A

E-mail: masters6@llnl.gov

Abstract.

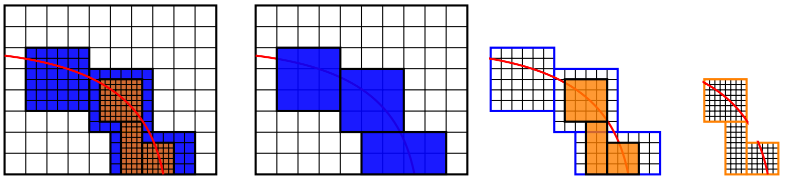
ALE-AMR is a new hydrocode that we are developing as a predictive modeling tool for debris and shrapnel formation in high-energy laser experiments. In this paper we present our approach to implementing laser ray tracing in ALE-AMR. We present the basic concepts of laser ray tracing and our approach to efficiently traverse the adaptive mesh hierarchy.

1. Introduction

Shrapnel and debris with the potential of damaging expensive optics, diagnostics, or fixturing, may be generated by the extreme conditions present in high energy laser facilities such as the National Ignition Facility (NIF), Laser Mégajoule (LMJ) and Omega. The ALE-AMR multi-physics hydrocode has been developed as a predictive modeling tool to identify and assess sources of shrapnel and debris and the hazards they pose so these may be mitigated [1–6]. Such simulations generally require large computational domains in which to track fragment formation and trajectories. ALE-AMR implements arbitrary Lagrangian-Eulerian (ALE) hydrodynamics within an adaptive mesh refinement (AMR) framework. This allows the mesh to be adapted in response to the current state: applying additional mesh resolution in regions with rapidly changing dynamics while a coarser mesh may be used in regions that are currently less interesting—enhancing computational efficiency without sacrificing accuracy or resolution. In evaluating debris and shrapnel formation we need to be concerned not only with the energy deposited by the 3 ω (UV) light—critical to target design—but also the unconverted 1 ω and 2 ω (red and green) that may irradiate support and diagnostic structures. Laser ray tracing is an efficient means for simulating laser propagation and energy deposition and provides a realistic energy source for hydrodynamic simulations.

2. ALE-AMR

The Structured Adaptive Mesh Refinement Application Infrastructure (SAMRAI) [7, 8] framework on which ALE-AMR is implemented provides a hierarchy of logically rectangular patches on which the problem physics are advanced. The simulation resolution can be improved in areas of interest, e.g., large gradients or fragmentation, by overlaying refined meshes that represent subsets of the domain



(a) AMR Mesh (b) Level 0 (c) Level 1 (d) Level 2

Figure 1. Rays traverse a virtual composite mesh. Rays are passed to the appropriate patch which may be at a higher, lower, or the same level of refinement

(see Figure 1). The hydrodynamics are evolved using a Lagrange-plus-remap algorithm. After problem initialization (mesh generation and initial conditions) the problem is advanced by repeating the following steps:

Lagrange Step The mesh is deformed by evaluating the forces (pressures and stresses) acting on the nodes and the resulting accelerations and displacements.

Remap Step Periodically the current Lagrange solution is remapped to an new mesh to avoid tangling, this may be either the original mesh (Eulerian) or some arbitrary mesh (ALE), usually a relaxed mesh intermediate to Lagrange and Eulerian configurations.

Adaptive Mesh Refinement Zones are tagged for coarsening or refinement by evaluating the current solution with respect to user defined criteria. A new mesh hierarchy is then created (regridding) onto which the current solution in mapped through coarsening or refining the solution variables.

Additional Physics If additional physics (heat conduction, radiation diffusion, or in this case laser ray tracing) are required they are performed at this point and the cumulative result is used for the next Lagrange step.

The code is designed to run on large parallel machines, therefore each processor is responsible for only a small subset of the problem, with communication limited to the information necessary to correctly synchronize adjacent or underlying patches through patch ghost cells or coarsened and refined solutions.

3. Laser Ray Tracing

Ray tracing is a powerful technique applied to high-energy laser simulations by Friedman [9] and Kaiser [10]. The key assumption is that the length scale of variations in the medium is larger than the laser wavelength over most of the computational domain. This allows the wave equations to be simplified as an equation of motion for rays and tracked through each zone. The electron density gradient is considered to be linear within each zone resulting in parabolic ray trajectories. Discontinuities at zone faces are treated via Snells law and energy deposition is computed as an inverse bremsstrahlung process. For a detailed description of the application of laser ray tracing to high energy laser simulations the reader is referred to the papers cited above.

Lagrangian or ALE modes result in deformed meshes consisting, in general, of quadrilateral elements in 2D and hexahedral elements in 3D (bounded by lines and doubly ruled surfaces, respectively). The intersections of the parabolic ray trajectories with these zones results in quadratic (2D) and quartic (3D) equations that must be solved for each face of the current zone. Although several solutions to these equations may exist valid solutions must fall within the boundaries of the current face. The exit face is identified as the one with the shortest exit time. Ambiguities may arise if a ray passes close to edges and/or corners. Incorrect selection of the exit face will result in a lost ray in the next zone as no valid exit points will be found. Such ambiguities are removed by reevaluating the ray traversal in the current zone after slightly perturbing the entry point within the entry face.

4. Logical Patches and AMR

We implemented laser ray tracing in ALE-AMR by considering the AMR hierarchy as a virtual composite mesh, i.e., rays are only propagated through the finest representation of the physical space as shown in Figure 1. The composite mesh is never explicitly constructed, rather at the end of each zone traversal an exit point and apparent destination zone are identified. The destination zone may be either: (a) on the same patch, but a finer representation exists , (b) on a different patch possibly at a different level (coarser or finer) , or (c) on the same level and patch . As a ray enters a new zone (either from a neighboring zone, from a different patch, or when introduced to the computational domain) we query the `is_finest` variable to determine if the zone is the finest representation of the physical spaced. The `is_finest` variable is updated each time the structure of the AMR grid is modified by, beginning with the coarsest level, filling pairs of coarse fine levels with ones and zeros, respectively, and then coarsening the fine level down to fill zeros in any coarse zone represented on the fine level. If the value of `is_finest` for a destination zone is zero then a finer representation exists to which the ray should be passed. In ALE-AMR, fine nodes in faces that lie on coarse-fine boundaries are constrained to be equally distributed within the parametric space of the coarse face. The fine destination index is found by determining the location of the ray entry point in the parametric space of the entry face (line in 2D, doubly ruled surface in 3D) and identifying the correct fine entry face (and associated fine zone) that covers that fraction of the parametric space. The destination patch is found by querying the SAMRAI `findOverlapIndices` function to identify the patch that overlaps the destination zone and then requesting the processor that holds this patch with the SAMRAI `getMappingForPatch` function (see SAMRAI reference documents supplied with the SAMRAI source [8]). Once identified (assuming that the destination processor is not the current processor) then the ray is queued to be passed to the processor when the current batch of rays have been processed.

For rays leaving the current patch (see (b)) the destination patch is found by again calling `findOverlapIndices` and `getMappingForPatch` for the apparent destination. If found, the rays are queued to be passed to their destinations (note: `is_finest` will be queried when they arrive—which may lead to subsequent refinement). If `findOverlapIndices` returns an empty overlap list then no patch at the current—or finer—level contains the destination zone and the destination must be coarsened. The index is coarsening by dividing the fine destination index by the refinement ratio and flooring the result, e.g., $\text{coarsen}\left((27, 19), \texttt{ratio} = (3, 3)\right) = (9, 6)$. `findOverlapIndices` is called again for the coarse index to identify the coarse destination patch. Coarsening and refinement is repeated until the finest representation of the physical destination is found.

Unless either of the above two events ((a) or (b)) happen, rays continue to propagate zone-to-zone on a given patch and deposit energy as detailed in the aforementioned references until the ray power is depleted. Energy deposited by all rays is accumulated in the zones traversed by rays and then coarsened down to underlying patches at the end of the ray tracing step. Coarsening of deposited energy is accomplished by taking a mass weighted average of the energy (or change in energy) in the fine zones, e.g., $e_c = (\sum \rho_i e_i V_i) / (\sum \rho_i V_i)$ where ρ_i, e_i, and V_i are the zonal densities, internal energies, and volumes of the fine zones associated with an underlying coarse zone (the coarse internal energy assigned the value e_c). The deposited energy may then be directly applied to the zonal internal energies or used as a source term in a subsequent diffusion step.

Direct MPI calls are used to first communicate which processors will send rays and the intended recipients, then the queued rays are sent and received and added to the list of rays that have arrived from other processors or are being introduced to that processor at domain boundaries. Laser ray tracing ends when all rays on all processors have terminated.

In the first timestep of ray tracing ray entry points are found by testing for ray crossings of each face touching a domain boundary on the coarsest level. In subsequent timesteps we search in the neighborhood of the last valid entry zone. This local search may fail if the domain boundaries are allowed to move, in which case the search region may be expanded or an exhaustive search again applied. Searching the coarsest level reduces the number of face crossing evaluations and avoids needing to determine which patches touch domain boundaries prior to ray insertion as only the coarsest level is guaranteed to exist at all points in the simulation domain. If the insertion zone is not the finest representation we repeat the face crossing search to accurately determine the entry point on the fine zones that replace the coarse zone.

One additional difficulty arises when rays insertion passes close to patch boundaries during start up. If a ray passes to close to an edge (in 3D, or a corner in 2D) we remove any ambiguity by applying a small random perturbation to the ray trajectory. At startup, such rays will receive independent random perturbations on different processors—possibly leading to multiple copies of the same ray. We avoid this by identifying potential duplicates (any that enter zones along patch boundaries) and compare this to the list of suspects on all other processors and remove the duplicates (preferring those from lower rank processors). Once an initial entry point (and associated search neighborhood) has been found, duplicate rays become very rare.

Current load balancing involves processing rays in batches that cascade to neighboring patches/processors. This evens the load on the processors that will participate in ray tracing. However, some processors may hold patches or levels that are traversed by few if any rays and may remain idle during a given laser ray tracing step. Future work will seek to address load balancing issues.

To illustrate the AMR laser ray tracing algorithm we will follow a ray through the mesh in Figure 1(a). The ray is introduced on the left side of the mesh into a zone on Level 0. After traversing one zone a finer representation of the destination is found Figure 1(b). The ray is passed to the appropriate Level 1 patch and ray propagation continues until a patch boundary is encountered (see Figure 1(c)). The ray is then passed to the neighboring Level 1 patch. After one zone traversal a finer representation is again indicated and the ray passed to Level 2. After traversing this Level 2 patch the ray leaves and no patch exists at the current level to receive it (see Figure 1(d)), the destination is coarsened and the process continued until the ray leaves the domain (or is fully depleted)

5. Conclusions and Future Work

We have implemented laser ray tracing in the ALE-AMR hydrocode to provides an efficient means for modeling laser energy deposition. This provides the means to deposit energy in a realistic manner as we seek to mitigate potential damage through predictive modeling of debris and shrapnel. Future work will explore load balancing and present applications of the ALE-AMR laser ray tracing package in simulating debris formation in high energy laser experiments.

Acknowledgments

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344

References

[1] Masters N, Anderson R, Elliott N, Fisher A, Gunney B and Koniges A 2008 *Journal of Physics: Conference Series* **112** 022017

[2] Fisher A C, Masters N D, Dixit P, Benson D J, Koniges A E, Anderson R W, Gunney B T N, Wang P and Becker R 2008 *Journal of Physics: Conference Series* **112** 022027

[3] Eder D, Koniges A, Landen O, Masters N, Fisher A, Jones O, Suratwala T and Suter L 2008 *Journal of Physics: Conference Series* **112** 032023

[4] Koniges A E, Anderson R W, Wang P, Gunney B T N, Becker R, Eder D C, MacGowan B J and Schneider M B 2006 *Journal de Physique IV* **133** 587–593

[5] Anderson R W, Pember R B and Elliott N S 2001 *40th American Institute for Aeronautics and Astronautics Aerospace Science Meeting and Exhibit* (Reno, Nevada) AIAA 2002-0738

[6] Anderson R W, Elliott N S and Pember R B 2004 *Journal of Computational Physics* **199** 598–617

[7] Wissink A M, Hornung R D, Kohn S R, Smith S S and Elliott N S 2001 *SC01 Proceedings* (Denver, CO)

[8] Samrai website `https://computation.llnl.gov/casc/SAMRAI/`

[9] Friedman A 1983 *Bulletin of the American Physical Society* **28** 1124

[10] Kaiser T B 2000 *Physical Review E* **61**(1) 895–905