John von Neumann Institute for Computing

Development Strategies for Modern Predictive Simulation Codes

Alice E. Koniges, Brian T.N. Gunney, Robert W. Anderson,
Aaron C. Fisher, Nathan D. Masters

published in

Parallel Computing: Architectures, Algorithms and Applications ,
C. Bischof, M. Bücker, P. Gibbon, G.R. Joubert, T. Lippert, B. Mohr,
F. Peters (Eds.),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **38**, ISBN 978-3-9810843-4-4, pp. 697-704, 2007.
Reprinted in: Advances in Parallel Computing, Volume **15**,
ISSN 0927-5452, ISBN 978-1-58603-796-3 (IOS Press), 2008.

http://www.fz-juelich.de/nic-series/volume38

Development Strategies for Modern Predictive Simulation Codes

Alice. E. Koniges, Brian T.N. Gunney, Robert W. Anderson, Aaron C. Fisher, Nathan D. Masters

Lawrence Livermore National Laboratory
Livermore, CA 94550, U.S.A.
E-mail: koniges@llnl.gov

Modern simulation codes often use a combination of languages and libraries for a variety of reasons including reducing time to solution, automatic parallelism when possible, portability, and modularity. We describe the process of designing a new multiscale simulation code, which takes advantage of these principles. One application of our code is high-powered laser systems, where hundreds of laser beams are concentrated on centimeter-sized targets to enable the demonstration of controlled fusion and exciting new experiments in astrophysics and high-energy-density science. Each target must be carefully analyzed so that debris and shrapnel from the target will be acceptable to optics and diagnostics, creating new simulation regimes. These simulations rely on a predictive capability for determining the debris and shrapnel effects. Our new three-dimensional parallel code uses adaptive mesh refinement (AMR) combined with more standard methods based on Arbitrary Lagrangian Eulerian (ALE) hydrodynamics to perform advanced modelling of each different target design. The AMR method provides a true multiscale simulation that allows for different physical models on different scales.

We discuss our code development strategies. The code is built on top of the SAMRAI library (structured adaptive mesh refinement application interface) that provides scalable automatic parallelism. During the development phase of this code we have instituted testing procedures, code writing styles, and team coordination applicable to a rapidly changing source code, several dependent libraries, and a relatively small team including university collaborators with their own source and platforms. We use modern coding techniques and open source tools when possible for code management and testing including CppUnit, Subversion (and previously GNU ARCH), TiddlyWiki and group chat rooms. Additionally, we are conducting experiments aimed at providing a data set for validation of the fragmentation models.

1 Introduction

The high-performance computing environment continues to provide new opportunities for simulations of physical phenomena. Processes that a few years ago were too complex for accurate computer models are now amenable to computer solutions. Yet the simulation environment contains new challenges because of the diversity of tools and methodology that accompany these new simulations. The code developers must know not only the physics, engineering, and mathematics of their simulation, but also some expertise in a variety of programming languages, parallel libraries or languages, and tools for code development is required. We describe our experiences in the design of a new code based on an ALE-AMR method. The code is currently being used to model target fragmentation properties for high-powered laser systems. We start this paper with a discussion of the application environment and the new challenges for this particular area of simulation including the ALE-AMR approach. The next section, Section 3, overviews some of the basic principles of our code development approach including team development methods, verification and other issues. Automatic parallelism is obtained by calling parallel libraries and building the code on a parallel framework. We discuss this framework in Section 4. Finally, in Section 5 we give examples of some simulations.

2 Application Environment

The environment of a high-powered laser chamber contains a good deal of open space from the vacuum that surrounds the small fusion target, often a hohlraum, located at the chamber centre. One goal of our simulation code is to model how the target dismantles after being hit by either laser beams or by x-rays that result from the lasers interacting with other target components. For the purposes of our simulation the "target" includes not only the hohlraum where the laser is focused, but also ancillary target elements such as cooling rings, shields, or appendages that improve diagnostic capabilities. Pieces of the target that are closest to target centre where the laser is focused will be vaporized and thus are relatively benign. However target components that are further from the main laser focus point are subject to lower levels of energy and therefore may be fragmented. It is important to determine the size of these fragmented pieces and their velocity vectors after the laser shot so that optics and diagnostics that line the chamber will be protected from damage. Dedicated experiments as well as experience from recent high-powered laser shots provide more information on this environment and the usefulness of mitigation procedures to direct fragments in benign directions.[1,2]

This environment creates a natural basis for the combination of an ALE method with AMR. An Arbitrary Lagrangian Eulerian (ALE) Method is a standard technique whereby the evolution of the continuum equations is performed by an initial Lagrangian step which allows the mesh to deform according to the fluid motion. After this step, the mesh quality is evaluated and an optional remesh step is performed to recover either the original mesh (fully Eulerian) or more likely an intermediate or "arbitrary" mesh that is optimized for the simulation. Combining this method with Adaptive Mesh Refinement (AMR) allows one to not only gain advantages from the ALE technique but also to concentrate the calculation on areas of the domain where the fragments are being formed.

The ALE-AMR method was pioneered by Anderson and Pember in a series of works for hydrodynamics resulting in the basic code that we are using and updating for solution of problems in solid mechanics.[3] Features of our ALE-AMR code that are relevant to the solid mechanics implementation include multimaterial interface reconstruction[4] and material model specific features.[5]

3 Team Development and Tools

There are a variety of free or open source tools that make code development much easier than previously. We briefly discuss a few of these that our team has found useful. We also discuss our verification ideas.

Subversion[6] is a version control system designed for the open source community, released under an Apache/BSD-style opens source license. Our group found the version control system particularly easy to use, and given that some of our developers had previous experience with systems like CVS, using this system was quite straight forward. On

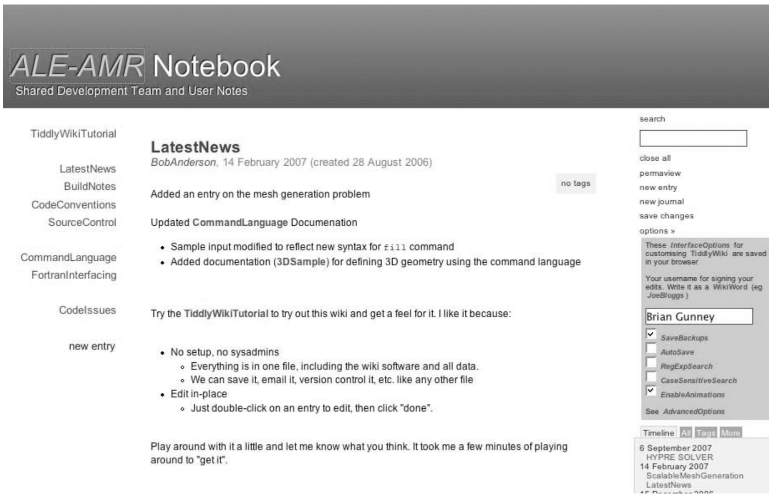


Figure 1. The ALE-AMR notebook demonstrates the use of a wiki for fast code documentation.

top of the subversion system we have scripts that control nightly builds and other checks as part of code verification and maintenance.

The term Wiki, as used in computer science is defined as a collaborative website, in which participants can directly edit the text of the "living" document. Perhaps the most famous implementation of a computer science Wiki is Wikipedia, the online free encyclopedia that is written collaboratively by people throughout the world. Wiki in computer science is derived from the Hawaiian word *wiki* that means to hurry. The notion of "hurry" or fast, is particularly important for us, since it encourages people to enter their documentation directly and quickly without being bound by a particular format. While the wiki itself is not a replacement for a traditional "user's manual," it serves as a convenient place to store daily or weekly information that is easily and quickly accessible. The particular wiki format we have chosen is TiddlyWiki. [7] A picture of our TiddlyWiki is given in Fig. 1. For reference, this figure shows the help message from the basic Wiki available online.

Another resource that our code team has used to great advantage is a "chat room." Use of the chat room allows developers who reside in physically different offices to discuss code changes, machine changes, strategies, etc. For the purposes of chat room security, the following policies are implemented: 1) Off-site chat does not work unless you set up an SSH tunnel to the server, 2) Not all rooms are "public"; a user must be invited before they can successfully join a room, 3) There may be more rooms in existence than what you see: rooms may be defined as invisible to outsiders. Our team has found that this strategy eliminates slow email question and answer periods and also encourages answers from developers who might not be queried on a particular issue. We have noticed a marked increase in the productivity of new team members when their chat room configuration was implemented.

A large portion of our code is based on libraries. The major library/framework that encompasses all of the parallelism is described in the next section. It is a patch-based Adaptive Mesh Refinement (AMR) library/framework. We also use the visualization package

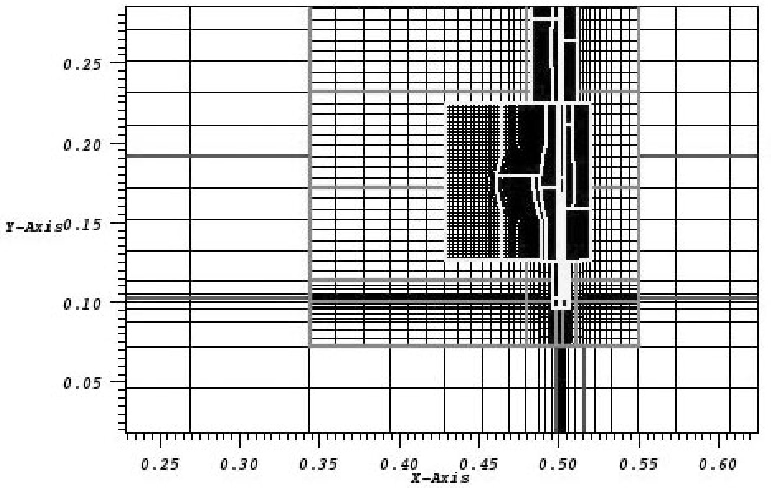


Figure 2. The figure shows the mesh as well as the patch boundaries. VisIt allows one to display mesh only, patch boundaries only, or both. Note that grid lines are intentionally not equally spaced, as is allowed in our ALE-AMR method. This yields better shock-capturing physics in the simulation.

called VisIt .[8] VisIt, which can be downloaded from the web, is particularly well suited to moving hydrodynamic simulations. Enhanced capabilities for parallelism and AMR grids are also included. Figure 2 shows example output from VisIt. Controls in VisIt allow one to selectively look at patches and/or mesh pieces for a better understanding of the simulation. Here we show the combination of both mesh and patches. In some areas concentration of mesh fills in as black. Zoom features allow one to study these regions. Automatic movie making commands are also included. VisIt also automatically handles groups of parallel files that are generated by the parallel code. Finally the visualization engine may be run remotely and/or in parallel. This visualization toolkit also helps in finding code errors and depicting how the automatic refinement works in the AMR scheme. This capability is useful in both finding bugs and also in determining the appropriate trigger for mesh refinement and de-refinement (coarsening).

4 The SAMRAI Framework

SAMRAI is an object-oriented structured AMR (SAMR) library supporting multiscale multiphysics applications and serving as a platform for SAMR algorithms research at LLNL.

SAMRAI factors out most of the common components and tasks in SAMR applications, allowing the application writers to focus on the specifics of their application. Among the most prominent tasks are mesh generation, mesh adaptation, data management, data transfer, high-level math operations, plotting interfaces, restart capabilities and parallelism. Less prominent time-saving services include computing mesh dependencies and mesh boundaries.

SAMRAI's interface design[9] is flexible and expandable while supporting powerful features. Typical mesh data (cell-centred, node-centred and face-centred) are provided. One can seamlessly integrate arbitrary user-defined data without having to modify SAMRAI. For example, arbitrary data residing in sparse cells has been used to represent surfaces cutting through the mesh. Application writers implement a small set of operations whose interfaces are defined by SAMRAI, and SAMRAI integrates the new data as if it were natively supported data. Data interpolation and copying is similarly integrated. SAMRAI natively supports Cartesian meshes, though its interface allows user-defined meshes such as deforming meshes such as that used by our ALE-AMR code.

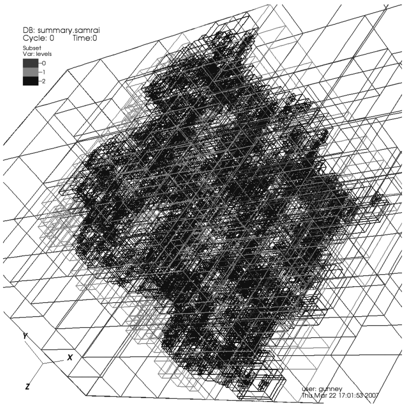
Parallelism is largely hidden, allowing application developers to focus on serial code development and still build a parallel application. All mesh data communications occur outside the interface seen by the application developers. User codes required for supporting user-defined data are all serial. Techniques such as combining messages are employed to enhance communication efficiency.

SAMRAI also serves as a research platform in new SAMR algorithms, resulting in significant recent advances such as box search algorithms,[10] clustering algorithms,[11] load balancing algorithms[12] and a new approach for managing SAMR meshes on distributed memory computers.[12] Figure 3 shows the differences between traditional and new approaches and their associated weak scaling characteristics. The baseline results, from current widely used SAMR algorithms show that although the serial physics engine scales well, other costs increase rapidly and quickly overwhelm the simulation. The new results give much more ideal scaling behaviour.

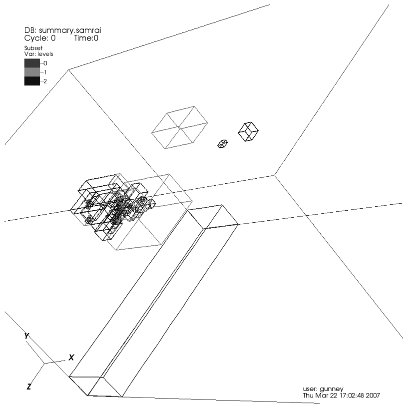
5 Code Applications

The code is being used to model the process of fragmentation and dismantling of high-powered laser facility targets. Figure 4 shows spall off the back of a thin target element that has energy deposition at the front surface of the thin plate in a localized region. This simulation is similar to Fig. 2 that shows the ALE-AMR grid and VisIt options. Here, the energy deposition first causes a plasma blow-off at the front surface of the plate. The top half of the figure is colour-coded to show density contours and the bottom half of the figure is colour-coded to show temperature contours. The plasma blow-off boundary is reconstructed via the multimaterial scheme described in Masters, et al.[4] The interface between the metal and the gas is shown in black. AMR patch boundaries are given in light grey. If this shield were part of a target configuration, we would use the code to determine the direction and quantity of plasma blow-off to the left, and the size and velocity of particle spall off the back. Spall off the back comes from the pressure wave that propagates horizontally through the material following the energy deposition at the front face.

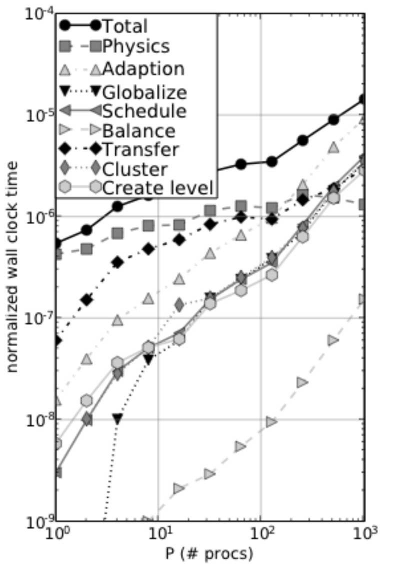
Another simulation of interest to high-powered laser facilities is what happens to material that surrounds a can-shaped hohlraum target. For instance, for the target to be cryogenically cooled, it is often surrounded by metal washer-shaped cooling rings. Figure 5(a) shows a simulation of the failure and fragmentation of a typical Al cooling ring subject to an impulsive loading. Annular shaped spall planes break off from the main cooling ring. In the right view, Fig. 5(b) shows a modification of a cooling ring simulation as described also in Koniges, et al.[2] Here the ring was not a complete annulus, as in full ring, but instead was notched for experimental purposes. Simulations of this configuration showed how



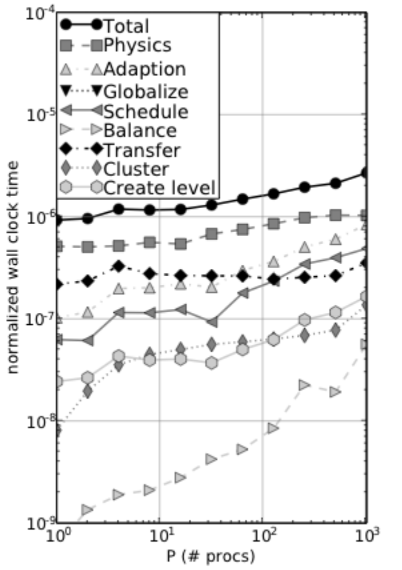
Baseline mesh view



New mesh view



Baseline performance



New performance

Figure 3. The top figures show the mesh as seen by the baseline approach and the new approach. The baseline approach stores and manages the global mesh. The new approach distributes the mesh information to reduce the amount of work per processor. The bottom figures shows the weak scaling characteristics for common tasks in an SAMR applications. For ideal weak scaling, the time for a given task does not vary with the number of processors. The new approach achieves more ideal scaling.

potentially damaging pieces could form when the stress in the ring were not distributed uniformly in the annulus. Such simulations and experiments aid in the validation process.

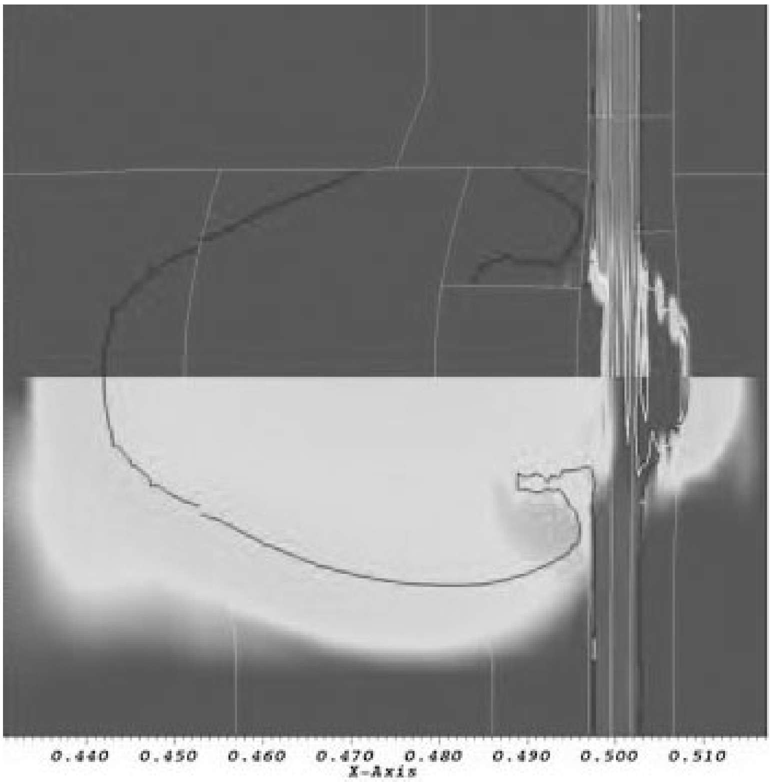


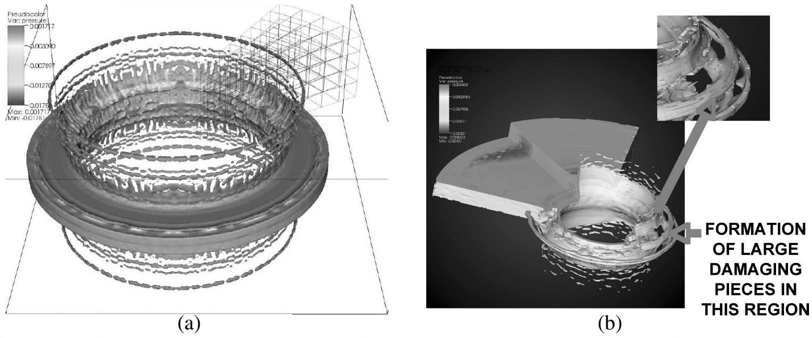
Figure 4. Formation of spall planes off the back of a thin target is shown. Additionally, contours of density (top) and temperature (bottom) are given. Interface reconstruction between the thin target metal and the gas surrounding the target is shown in black. AMR patch boundaries are given in light grey.

6 Summary

In this paper we have described the development of a new simulation code by a team of roughly 4 - 6 people. The code uses a variety of languages and some of the modern interaction techniques such as a wiki and a chat room are described. The code is scalable and parallel, based on the use of an advanced library framework known as SAMRAI. The use of SAMRAI isolates most of the parallel code from the physics code leading to faster development and better debugging opportunity. Additionally, parallel scalability improvements to the SAMRAI library can be seamlessly incorporated into the main code. The code is providing new results for an upcoming area of physics/engineering, namely the protection of high-powered laser facility target chambers.

Acknowledgements

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



(a) (b)

Figure 5. A ring of metal is impulsively loaded from the centre to demonstrate the fragmentation pattern are target cooling rings. In the left view, the ring is symmetric and loaded from the interior. In the right view, an alternative design has a notched ring. This design shows the formation of potentially damaging pieces during its fragmentation process.

References

1. A. E. Koniges, R. W. Anderson, P. Wang, B. T. N. Gunney and R. Becker, *Modeling NIF experimental designs with adaptive mesh refinement and Lagrangian hydrodynamics* J. Physics IV, **133**, 587–593, (2006)
2. A. E. Koniges, et al., *Experiments for the validation of debris and shrapnel calculations*, J. Physics, (submitted 2007).
3. R. W. Anderson, N. S. Elliott and R. B. Pember, *An arbitrary Lagrangian–Eulerian method with adaptive mesh refinement for the solution of the Euler equations*, J. Comp. Phys., **199**, 598–617, (2004).
4. N. Masters, et al., *Interface reconstruction in two- and three-dimensional arbitrary Lagrangian-Eulerian adaptive mesh refinement simulations* J. Physics, (submitted 2007).
5. A. Fisher, et al., *Hierarchical material models for fragmentation modeling in NIF-ALE-AMR*, J. Physics, (submitted 2007).
6. `http://subversion.tigris.org/`
7. `http://www.tiddlywiki.org/`
8. `http://www.llnl.gov/visit/doc.html`
9. R. D. Hornung and S. R. Kohn, *Managing application complexity in the SAMRAI object-oriented framework*, Concurrency and Computation: Practice and Experience, **14**, 347–368, (2002).
10. A. M. Wissink, D. Hysom and R. D. Hornung, *Enhancing scalability of parallel structured AMR calculations*, in: Proc. 17th ACM International Conference on Supercomputing (ICS03), pp. 336–347, San Francisco, (2003).
11. B. T. N. Gunney, A. M. Wissink and D. A. Hysom. *Parallel clustering algorithms for structured AMR*. Journal of Parallel and Distributed Computing, **66**, 1419–1430, (2006).
12. B. T. N. Gunney. *Scalable metadata and algorithms for structured AMR*, SIAM Computational Science and Engineering Conference (2007), UCRL-PRES-228081.