ARTICLE IN PRESS
J. Parallel Distrib. Comput. ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

A survey of high level frameworks in block-structured adaptive mesh refinement packages

Anshu Dubey [a,*], Ann Almgren [a], John Bell [a], Martin Berzins [j], Steve Brandt [f,g], Greg Bryan [k], Phillip Colella [a,l], Daniel Graves [a], Michael Lijewski [a], Frank Löffler [f], Brian O'Shea [c,d,e], Erik Schnetter [h,i,f], Brian Van Straalen [a], Klaus Weide [b]

[a] Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[b] Flash Center for Computational Science, The University of Chicago, Chicago 60637, USA
[c] Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA
[d] Lyman Briggs College, Michigan State University, East Lansing, MI 48824, USA
[e] The Institute for Cyber-Enabled Research, Michigan State University, East Lansing, MI 48824, USA
[f] Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, USA
[g] Department of Computer Science, Louisiana State University, Baton Rouge, LA 70803, USA
[h] Perimeter Institute for Theoretical Physics, Waterloo, ON N2L 2Y5, Canada
[i] Department of Physics, University of Guelph, Guelph, ON N1G 2W1, Canada
[j] Mathematics and School of Computing, University of Utah, Salt Lake City, UT 84112, USA
[k] Department of Astronomy, Columbia University, New York, NY 10025, USA
[l] Computer Science Department, University of California, Berkeley, CA 94720, USA

HIGHLIGHTS

- A survey of mature openly available state-of-the-art structured AMR libraries and codes.
- Discussion of their frameworks, challenges and design trade-offs.
- Directions being pursued by the codes to prepare for the future many-core and heterogeneous platforms.

ARTICLE INFO

Article history:
Received 16 July 2013
Received in revised form
14 April 2014
Accepted 3 July 2014
Available online xxxx

Keywords:
SAMR
BoxLib
Chombo
FLASH
Cactus
Enzo
Uintah

ABSTRACT

Over the last decade block-structured adaptive mesh refinement (SAMR) has found increasing use in large, publicly available codes and frameworks. SAMR frameworks have evolved along different paths. Some have stayed focused on specific domain areas, others have pursued a more general functionality, providing the building blocks for a larger variety of applications. In this survey paper we examine a representative set of SAMR packages and SAMR-based codes that have been in existence for half a decade or more, have a reasonably sized and active user base outside of their home institutions, and are publicly available. The set consists of a mix of SAMR packages and application codes that cover a broad range of scientific domains. We look at their high-level frameworks, their design trade-offs and their approach to dealing with the advent of radical changes in hardware architecture. The codes included in this survey are BoxLib, Cactus, Chombo, Enzo, FLASH, and Uintah.

Published by Elsevier Inc.

1. Introduction

Block-structured adaptive mesh refinement (SAMR) [8,7] first appeared as a computational technique almost 30 years ago; since then it has been used in many individual research codes and, increasingly over the last decade, in large, publicly available code frameworks and application codes. The first uses of SAMR focused almost entirely on explicit methods for compressible hydrodynamics, and these types of problems motivated the building of many of the large code frameworks. SAMR frameworks have evolved along different paths. Some have stayed focused on specific domain areas, adding large amounts of functionality and problem-specific

* Correspondence to: One Cyclotron Road, Mailstop 50A1148, Berkeley, CA 94720, USA.
E-mail addresses: adubey@lbl.gov, adubey64@gmail.com (A. Dubey).

http://dx.doi.org/10.1016/j.jpdc.2014.07.001
0743-7315/Published by Elsevier Inc.

physics modules that are relevant to those applications. Examples of these include AstroBEAR [26], CRASH [91], Cactus [42,15], Enzo [14,38], FLASH [40,32], Overture [76], PLUTO [68,89], and Uintah [77,78]. Other frameworks have pursued a more general functionality, providing the building blocks for a larger variety of applications while enabling domain-specific codes to be built using that framework. As an example, while almost every SAMR framework can be used to solve systems of hyperbolic conservation laws explicitly, not all frameworks include the functionality to solve elliptic equations accurately on the entire hierarchy or a subset of levels. Examples of frameworks constructed specifically for solving hyperbolic conservation laws include AMROC [29] and AMRClaw [5], both based on the wave propagation algorithms of R. LeVeque. Extensions of AMRClaw include GeoClaw [41], the widely used tsunami simulation tool. BoxLib [13], Chombo [23], Jasmine [73] and SAMRAI [17,47] are more general in that they supply full functionality for solving equation sets containing hyperbolic, parabolic and elliptic equations, and facilitate the development of codes for simulating a wide variety of different applications. PARAMESH [39] supplies only the mesh management capability and as such is equation-independent. A more comprehensive list of codes that use SAMR, and other useful adaptive mesh refinement (AMR) resources can be found at [46].

SAMR codes all rely on the same fundamental concept, viz. that the solution can be computed in different regions of the domain with different spatial resolutions, where each region at a particular resolution has a logically rectangular structure. In some SAMR codes the data is organized by level, so that the description of the hierarchy is fundamentally defined by the union of blocks at each level; while others organize their data with unique parent–child relationships. Along with the spatial decomposition, different codes solving time-dependent equations make different assumptions about the time stepping, i.e., whether blocks at all levels advance at the same time step, or blocks advance with a time step unique to their level. Finally, even when frameworks are used to solve exactly the same equations with exactly the same algorithm, the performance can vary due to the fact that different frameworks are written in different languages, with different choices of data layout, and also differ in other implementation details. However, despite their differences in infrastructure and target domain applications, the codes have many aspects that are similar, and many of these codes follow a set of very similar software engineering practices.

In this survey paper we examine a representative set of SAMR packages and SAMR-based codes that: (1) have been in existence for half a decade or more, (2) have a reasonably sized and active user base outside of their home institutions, and most importantly, (3) are publicly available to any interested user. In selecting the codes we have taken care to include variations in spatial and temporal refinement practices, load distribution and meta-information management. Therefore, we have octree and patch based SAMR, no subcycling and subcycling done in different ways, load distribution on a level by level or all levels at once, and globally replicated meta-data or local view. Additionally, the set covers a broad range of scientific domains that use SAMR technology in different ways. We look at their high-level frameworks, consider the trade-offs between various approaches, and the challenges posed by the advent of radical changes in hardware architecture. The codes studied in detail in this survey are BoxLib, Cactus, Chombo, Enzo, FLASH, and Uintah. The application domains covered by a union of these codes include astrophysics, cosmology, general relativity, combustion, climate science, subsurface flow, turbulence, fluid–structure interactions, plasma physics, and particle accelerators.

2. Overview of the codes

BoxLib is primarily a framework for building massively parallel SAMR applications. The goals of the BoxLib framework are twofold: first, to support the rapid development, implementation and testing of new algorithms in a massively parallel SAMR framework; and second, to provide the basis for large-scale domain-specific simulation codes to be used for numerical investigations of phenomena in fields such as astrophysics, cosmology, subsurface flow, turbulent combustion, and any other field which can be fundamentally described by time-dependent PDE's (with additional source terms, constraints, etc.). In accordance with both goals, the core remains relatively agile and agnostic, and is not tied to a particular time-stepping or spatial discretization strategy, or to a particular set of physics packages.

That said, while BoxLib itself supplies a very general capability for solving time-dependent PDE's on an adaptive mesh hierarchy, there are a number of large, domain-specific BoxLib-based application codes in scientific use today. The most widely used include CASTRO [3,93,94] for fully compressible radiation-hydrodynamics; MAESTRO [74], for low Mach number astrophysical flows; Nyx [4] for cosmological applications; LMC [28] for low Mach number combustion; and the structured grid component of the Amanzi code for modeling subsurface flow [79].

Chombo is an offshoot of the BoxLib framework, having branched off from BoxLib in 1998. As a result Chombo shares many features with BoxLib, including the hybrid C++/Fortran approach, and the separation of concerns between the parts that are best handled in C++ (abstractions, memory management, I/O, flow control) and Fortran dialects (loop parallelism, stencil computations). Chombo articulates its APIs explicitly for easier plugin by the client application code. It has also diverged from BoxLib in the design of its data containers. Chombo currently supports applications in a range of disciplines, including the following: MHD for tokamaks using all-speed projection methods, and large eddy simulations of wind turbines at KAUST; compressible CFD + collision-less particle cosmology simulations (CHARM code, [70,71]); physics of the solar wind and its interaction with the interstellar medium using compressible hyperbolic CFD + electromagnetic and kinetic effects (MS-FLUKSS code [61,81,80]); general astrophysics modeling (PLUTO code [69]); astrophysical MHD turbulence (ORION code [56]); SF Bay and Delta hydrology modeling – shallow water (Realm code [6]); plasma-wakefield accelerators – compressible viscous flow at LBNL; blood flow in cerebral arteries – fluid/solid coupling (UCB ParLab project [30]); pore-scale subsurface reacting flow (Chombo-Crunch code, [72]); conjugate heat transfer in nuclear reactors [25]); 4D gyrokinetic models of tokamak edge plasmas (COGENT code, [31,18]); land ice model for climate simulation(BISICLES code, [24]); and atmospheric models for climate simulation—low-Mach number CFD at University of Michigan.

Cactus [42,15] was designed as a general-purpose software framework for high-performance computing with AMR as one of its features. The first set of applications that used the framework was astrophysical simulations of compact objects involving general relativity (GR) such as black holes and neutron stars. These scenarios require high resolution inside and near the compact objects, and at the same time need to track gravitational waves propagating to large distances of several hundred times the radii of the compact objects. This leads very naturally to the use of AMR. In GR, gravitational effects are described via hyperbolic (wave-type) equations, propagating at the speed of light. This removes the need for an elliptic solver that is necessary in Newtonian gravity to calculate the gravitational potential. While the Cactus framework is generic, its most prominent user today is the Einstein Toolkit [55,95,37], a large set of physics modules for relativistic astrophysics simulations. The Einstein Toolkit includes modules for solving the Einstein equations and relativistic magneto-hydrodynamics, as well as modules for initial conditions, analysis, and so on.

Enzo is a standalone application code [14,38] that was originally designed to simulate the formation of large-scale cosmological structure, such as clusters of galaxies and the intergalactic medium. Since the formation of structures in the Universe is a