EDGE: Development and Verification of a Large-scale Wave Propagation Software

Alex Breuer and Alex Heinecke (Intel)

ABSTRACT

The seismic modeling community strives to increase the frequency content of ground motion simulations. This effort imposes high demands on all components of the respective modeling and simulation environments. Further, stringent verification using realistic settings is mandatory to ensure the trustworthiness of production workflows. We present the developments resulting from a series of high-frequency ground motion simulations of the 2014 M5.1 La Habra, California earthquake using the Extreme-scale Discontinuous Galerkin Environment (EDGE).

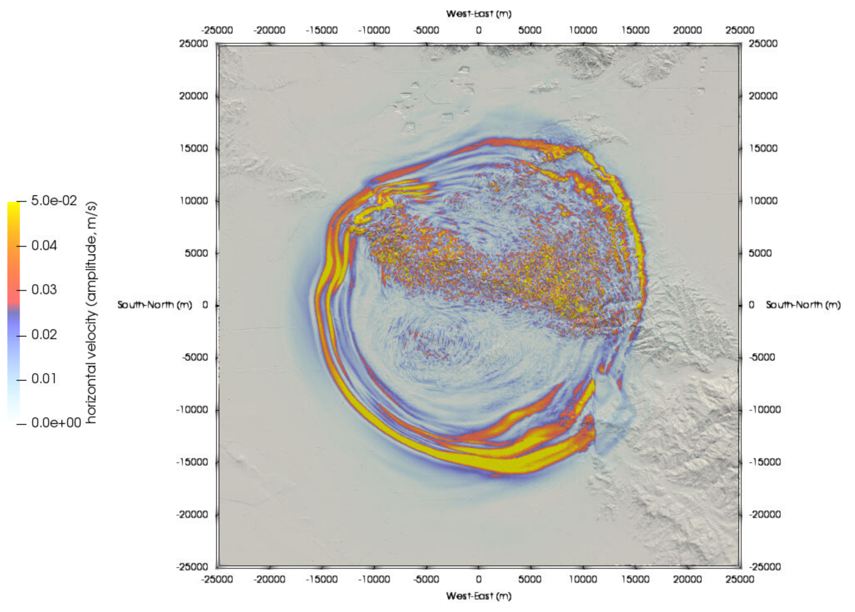

Figure 1
Visualization of the seismic wave field for a simulation of the 2014 Mw 5.1 La Habra Earthquake. Shown are the amplitudes of the horizontal particle velocities after seven seconds of simulated time.

EDGE uses the discontinuous Galerkin finite element method to solve hyperbolic partial differential equations on unstructured tetrahedral meshes. Our work includes the generation and pre-processing of large velocity-aware meshes with billions of elements, efficient data-interfaces to our core solver, as well as the development of an efficient and scalable local time stepping scheme supporting viscoelastic attenuation. We conclude our presentation by sharing lessons-learned on developing a production workflow while verifying it as part of a community exercise. The ability to conveniently and routinely conduct simulations on hundreds of compute-nodes together with a sole reliance on open-source software make our experiences appealing to the CSE-community.

EDGE

The Extreme-scale Discontinuous Galerkin Environment (EDGE) is end-to-end software for the simulation of wave phenomena:

- Focus: static meshes with high geometric complexity
- Novel scalable local time stepping scheme with a full integration in EDGE's preprocessing tools
- Unique support for fused simulations exploiting inter-simulation parallelism
- Rapid prototyping through support for: line elements, quads, triangles, hexes, tets
- Parallelization: assembly kernels for highest performance on many recent CPU architectures (AVX, AVX2, AVX512, aarch64); advanced OpenMP+MPI for hidden communication
- Extremely scalable with sustained petascale performance: 10.4 DP-PFLOPS on Cori II, 1.1 SP-PFLOPS in AWS
- Supporting tools for surface meshing, constrained velocity-aware volume meshing and partitioning
- Core solver and all tools are open source software (BSD-3), standard modeling and simulation pipeline relies exclusively on open source software

PRODUCTION WORKFLOW

We illustrate our production workflow using experiences gained from a series of large-scale ground motion simulations. The runs were conducted as part of a verification effort let by the Southern California Earthquake Center (see Fig. 3).

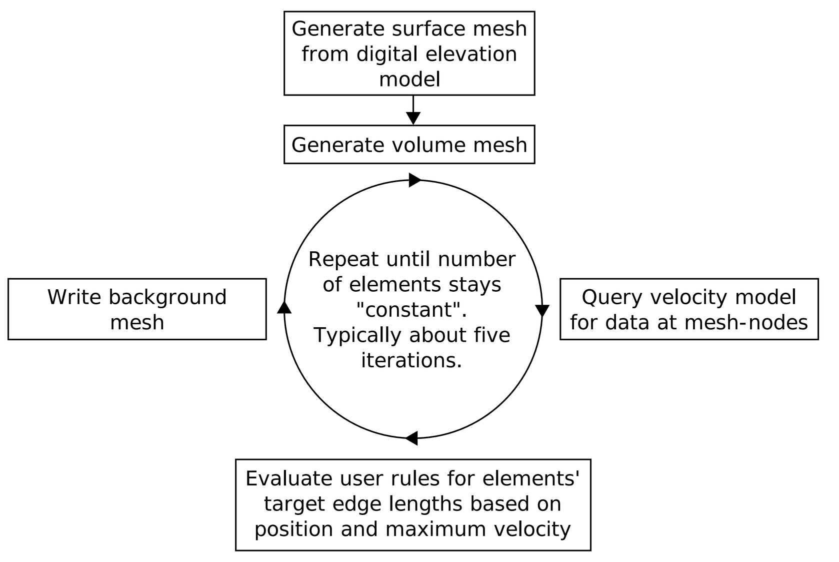

Figure 2
Illustration of our meshing workflow. We derive problem-aware tetrahedral meshes by iteratively feeding the mesher with a high-resolution surface mesh and information on the targeted edge-lengths in the computational domain.

Our workflow heavily relies on a rich pre-processing phase.This phase maximizes algorithmic and computational performance of the core solver. The first step of our pre-processing generates a problem-aware tetrahedral mesh (see Fig. 2). Most importantly we provide target edge-lengths based on the used seismic velocity model to the mesher. Here, we specify a certain number of elements per wave length, resulting in a highly refined mesh in low-velocity regions (see Fig. 4). The next pre-processing step derives the local time stepping time groups (see section below) and derives respective weights for the elements and faces representing computation and communication of the core solver. This information is then used to partition the mesh. Next, we reorder the mesh based on the elements' partitions, time groups, and finally by their role with respect communication in the distributed memory parallelization. The reordered mesh is written partition-wise to disk. Additionally, we assemble a second file per partition which contains data required by the core solver, e.g., the local communication structure or per-element seismic velocities.

At scale every process simply reads its respective mesh- and annotation-file without the need for time-consuming global mesh operations. Over the course of the simulation we periodically write the seismic wave field at the surface (see Fig. 1) and more frequently at stations to disk. The synthetic data is then used for verification (see Fig. 5).

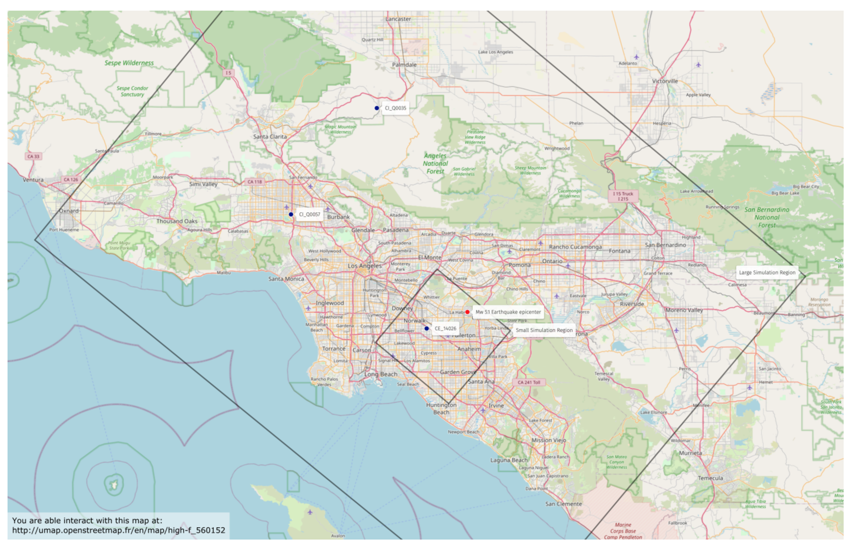

Figure 3
Study area of the Southern California Earthquake Center's High Frequency project simulating the 2014 M5.1 La Habra, California earthquake. Shown are the "small domain" through the inner box and partially the "large domain" through the outer box. Additionally the locations of the epicenter and three stations are given.

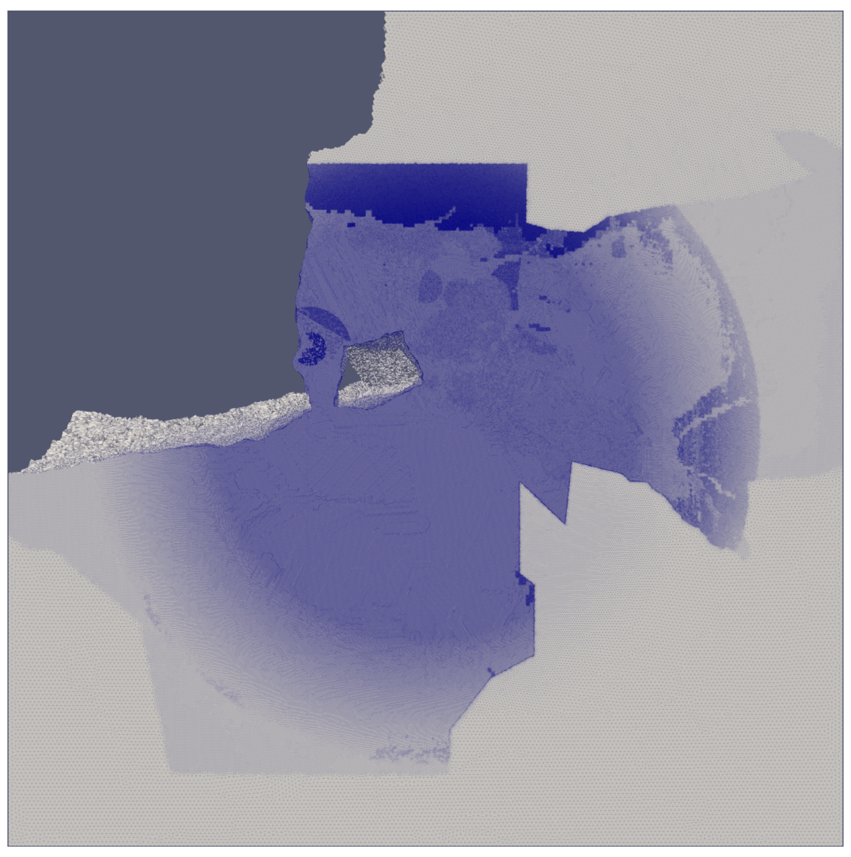

Figure 4
Visualization of a velocity-aware tetrahedral mesh for the small simulation region shown in Fig. 3. The example shows a highly increased mesh resolution in the near-surface parts of the Los Angeles basin. Some partitions in the North-West part of the computational domain are excluded. The epicenter is located in the center of the computational domain. The user also defined a cylindrical high-resolution region around the epicenter in addition to the velocity-derived target edge-lengths.

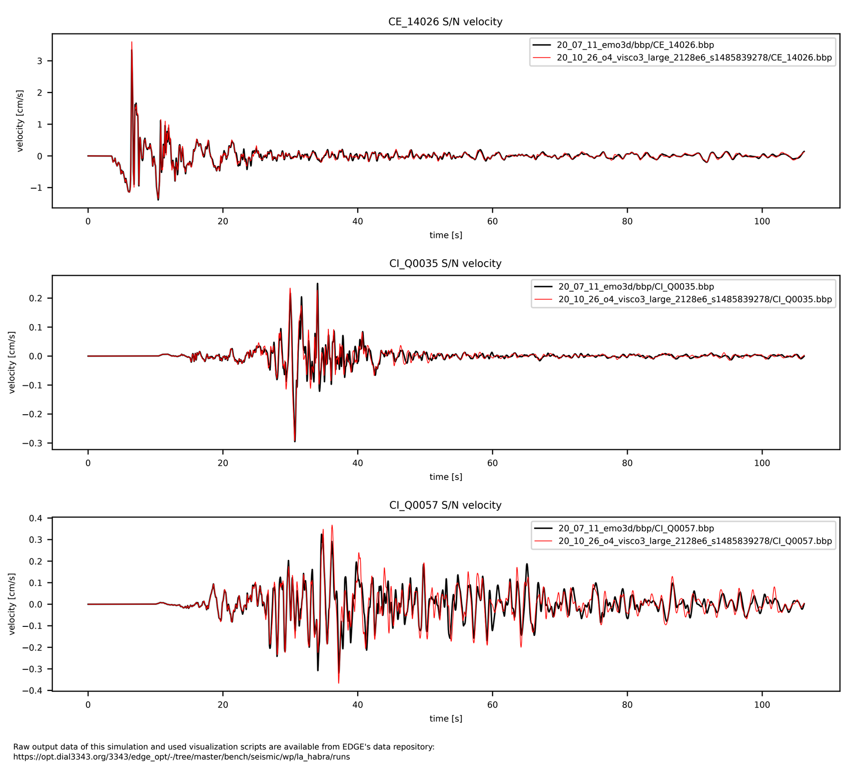

Figure 5
Comparison of EDGE's South-North velocity component (red) to another solver of the High Frequency project (black). Shown are synthetic seismograms for the three stations depicted in Fig. 3. The seismograms were low-pass filtered at 5Hz. EDGE's respective ground motion simulation harnessed 1,536 nodes of the Frontera machine for a total of 48 hours to advance the used 2.1 billion tetrahedral elemenmesh in time.

LOCAL TIME STEPPING

We implemented a completely overhauled version of the grouped Local Time stepping (LTS) scheme presented in [1]. Our new scheme supports viscoelastic attenuation efficiently through a refactored data storage. This change limits us to rate-2 LTS. However, we introduced an optimized coverage of the time step range through the time groups as shown in Fig. 6.

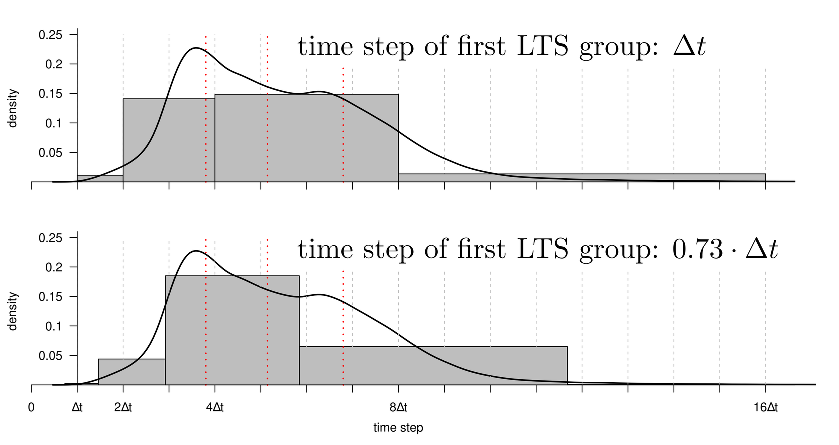

Figure 6
Examplary illustration of EDGE's enhanced rate-2 LTS scheme which optimizes the first group's time step. The solid line shows the element density w.r.t. the time step, the boxes those of the grouped approach. We obtain a 3.16x (top) and 3.33x (bottom) theoretical speedup over global time stepping.

CONCLUSIONS

We presented our most important experiences and approaches when conducting production simulations with the Extreme Scale Discontinuous Galerkin Environment. We learned the following key lessons:

- Efficient and automated pre-processing is key
- The core solver running at scale should be kept as simple as possible. Complex runtime operations should be offloaded to pre-processing. These operations, e.g., mesh queries or the assembly of communication structures, limit scalability and result in bottlenecks
- The end-to-end use of open source software in the entire production workflow allows us to share our software, setups and results easily

[1]: Breuer, Alexander, Alexander Heinecke, and Michael Bader. "Petascale local time stepping for the ADER-DG finite element method." 2016 IEEE international parallel and distributed processing symposium (IPDPS). IEEE, 2016.

Interested in this work? Reach out!

alex.breuer@uni-jena.de
alexander.heinecke@intel.com
https://dial3343.org
https://scalable.uni-jena.de

FRIEDRICH-SCHILLER-UNIVERSITÄT JENA