Report: 2016 SCEC Project

Increasing the Efficiency of Dynamic Rupture Simulations by Concurrently Executed Forward Runs

PI:
Alexander Breuer
San Diego Supercomputer Center
University of California, San Diego
La Jolla, California 92093-0505

Co-PI:
Yifeng Cui
San Diego Supercomputer Center
University of California, San Diego
La Jolla, California 92093-0505

External collaborator:
Alexander Heinecke
Intel Parallel Computing Lab
Intel Corporation
Santa Clara, California 95054-1549

Awarded Funds: $30,000 (100% A. Breuer, 0% Y. Cui)

Proposal Category: Integration and Theory

SCEC Science Objectives: 3e, 4e, 6b

1 Introduction: EDGE

Within this project, we extended the Extreme-Scale Discontinuous Galerkin Environment (EDGE) with basic dynamic rupture capabilities and performed respective code verification benchmarks. EDGE solves the elastic wave equations for an isotropic medium in velocity-stress formulation. We use the Discontinuous Galerkin Finite Element Method (DG-FEM) for spatial discretization of the computational domain. Time discretization is accomplished through the ADER-scheme, leading to arbitrary high, user-selectable convergence rates in both, space and time. Our software currently supports line elements, rectangular 4-node quadrilaterals, 3-node triangles, 8-node rectangular hexahedrons and 4-node tetrahedral elements.

EDGE was initiated through the Intel Parallel Parallel Computing Center (IPCC) *Accurate and Efficient Earthquake Simulations on Intel Xeon Phi*. In recent work [11] we presented EDGE's performance on the Cori Phase II and Theta supercomputers in strong and weak scaling studies. By utilizing 612,000 Intel Xeon Phi x200 cores of Cori Phase II, we were able to sustain 10.4 PFLOPS with double-precision arithmetic, the highest ever sustained performance for a seismic simulation. One of EDGE's most compelling features is the fusion of multiple simulations in one forward execution of the solver. While a traditional solver s uses fixed input i to obtain observations $o = s(i)$, our approach operates on a set of inputs. Assuming a set of m different inputs $I_m = (i_1, i_2, \ldots, i_m)$, a single execution of EDGE's solver S operates on the entire set of inputs in parallel to obtain the set of observations $O_m = (o_1, o_2, \ldots, o_m) = S(I_m)$. This approach has many advantages for increasing the efficiency of the simulations. Two examples are 1) constant data (e.g., the mesh or velocity model) is shared among the fused simulations, and 2) exploitation of inter-simulation parallelism. Using convergence rates 2-4, EDGE is able to outperform the simulation throughput of the software package SeisSol (version 201511) by a factor of 1.8 - 4.6 on the 68-core Xeon Phi 7250 processor [11]. Further outreach relevant for this project includes: [4, 8, 17, 2, 5, 6, 9, 7, 1, 10]

2 Internal Dynamic Rupture Boundaries

This project added support for internal boundaries to EDGE. For two-dimensional simulations, an internal boundary is given by a one-dimensional geometry. Analogue, in three-dimensional simulations, this leads to a two-dimensional geometry. The fault geometries described by the Community Fault Model (CFM) would, for example, result into two-dimensional internal boundaries. In the resulting mesh, the meshed geometry is given by the faces of our elements in the volume discretization, annotated with an identifier to describe the special handling of these faces. Our wave propagation solver heavily exploits linearity of the elastic wave equations by embedding the entire surface integration of the DG-FEM machinery into a series of matrix-matrix operations. However, nonlinear rupture physics require us to impose shear stress and fault-parallel

velocity perturbations according to the considered friction law. Numerically this might be handled by point-wise perturbations of the middle states of the linear Riemann solutions at the respective faces [13, 20, 19].

Therefore, we implemented a special handling of the surface integration in the presence of internal boundaries. Our internal boundary solver consist of five steps solving the generalized Riemann problems [22] through numerical quadrature. First, we rotate the modal Degrees Of Freedom (DOFs) in both adjacent elements to a face-aligned coordinate system to obtain shear and normal stresses relative to the fault. Second, our solver evaluates the modal basis at face-local quadrature points. This leads to point-wise Riemann problems since no continuity between elements is enforced for DG-FEM. In the third step, we compute the middle states of the Riemann solution by either jumping over the P- and S-waves from the left or from the right side of the face. The fourth step calls the solver for the considered friction law. If the fault is sliding, the middle states are perturbed, resulting in a minus- and a plus-side middle state. However, if the fault is locked, no perturbations are performed and the middle states of the linear Riemann solution are used. The last step of our internal boundary solver performs the spatio-temporal quadrature of the point-wise values, multiplies them with our test functions and computes the net-contributions of the numerical fluxes to the time step for both elements adjacent to a face.

Software Design In this project, we limited the embedded friction solver to linear slip weakening in two and three dimensions. However, our internal boundary solver uses C++-template parameters for the friction law and associated data, allowing for easy extensibility with new friction laws at high performance.

In terms of EDGE's fusion capabilities, all parameters of the friction law (static and dynamic friction coefficients, as well as the critical slip distance) can be set independently for all fused simulations. Further, the normal and shear background stresses at the fault can be set completely independent for all fused simulations. The result is a high degree of independence from one fused simulation to another, leading to flexible software for parameter studies. One could, for example, study the influence of nucleation at different spatial positions within the fused simulations.

Further, the internal boundary solver is completely integrated into EDGE's control flow for maximum performance. In comparison to earlier work [12, 18], EDGE only ensures correctness by enforcing dependencies of work packages, but does not enforce artificial, time-consuming synchronization of the worker threads in between the integration steps. This allows the cores of the targeted manycore architectures to continue independently with the next step, once they finished all computations of the current step.

3 Benchmarks

We tested EDGE's dynamic rupture physics in a series of benchmarks. Due to space limitations of this report, we only describe our convergence setup and limit

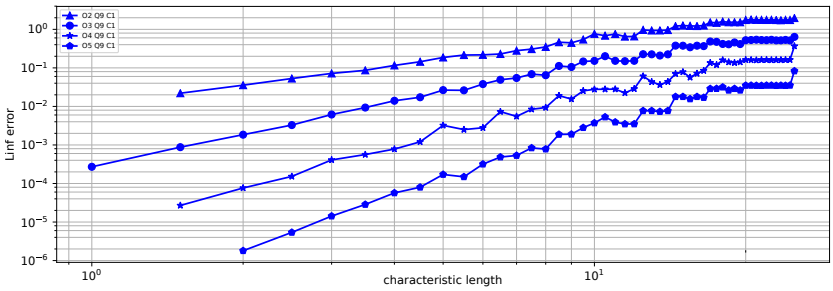
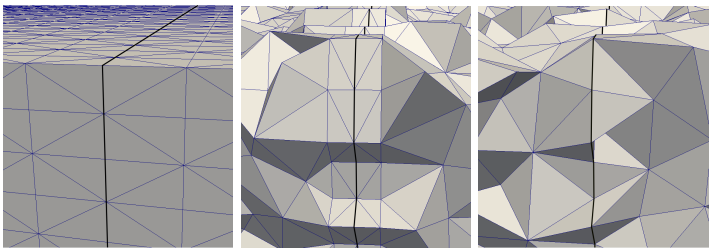


Figure 1: Convergence behavior of EDGE using the newly developed internal boundary solver and gmsh's unstructured Delaunay meshing algorithm. Shown is the L^∞-error for orders $\mathcal{O}2 - \mathcal{O}5$ ($\mathcal{P}1 - \mathcal{P}4$ elements) of the particle velocity in z-direction in dependency of the characteristic length as specified in gmsh.

the SCEC/USGS benchmarks to TPV3 and TPV5. We obtained all unstructured meshes through the open-source mesher gmsh [15]. Further, all presented simulations were conducted on the Stampede-KNL supercomputer.

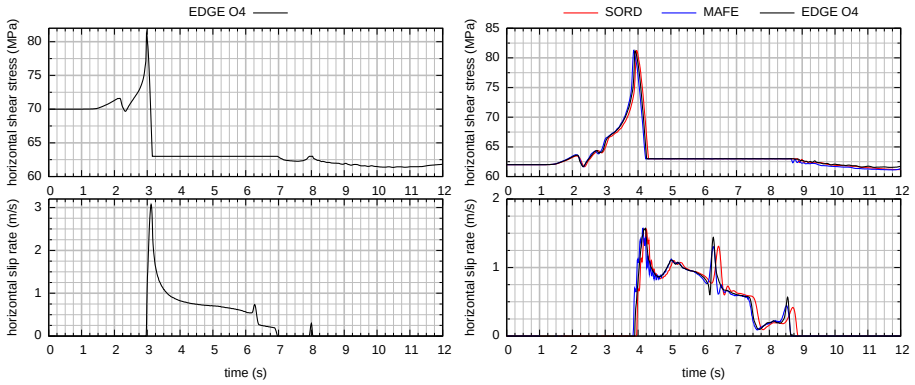
Convergence Our first benchmark explores EDGE's high order wave propagation solver in the presence of internal boundaries. We used a domain of size $[0, 100]^3$, periodic boundaries, and initial DOFs following the eigenvector decomposition of the linear system of PDEs [14]. We generated a series of setups by using unstructured, tetrahedral meshes (see Fig. 2c) with decreasing edge lengths. We prescribed three locked faults at $y = 25$, $y = 50$, and $y = 75$, covering $[25, 75]^2$ of the xz-plane. Each of the faults has an artificial high fault-normal background stress $\sigma_{xx}^{\mathrm{BG}} = -120 \cdot 10^6$ and zero-valued shear stresses $\sigma_{xy}^{\mathrm{BG}} = \sigma_{yz}^{\mathrm{BG}} = 0$ to avoid any rupture. Fig. 1 shows the observed L^∞ errors of the particle velocity in z-direction for second to fifth order configurations.

TPV3 & TPV5 In contrast to the conducted convergence analysis, we used a different meshing strategy for TPV3 and TPV5 (see Sec. 4 for details). Here, we extruded the surface mesh of the fault by one layer in both fault-normal directions and used gmsh's frontal meshing algorithm for the remaining volume. As illustrated in Fig. 2b, the result of this approach is a symmetric mesh at the internal fault boundary. We used a computational domain of $[-25\mathrm{km}, 25\mathrm{km}]^2 \times [-15\mathrm{km}, 15\mathrm{km}]$ for TPV3 with the fault placed in the center. Similar, we used a domain of $[-25\mathrm{km}, 25\mathrm{km}]^2 \times [-25\mathrm{km}, 0\mathrm{km}]$ for TPV5, where $z = 0$ is the free-surface boundary. We exploited mesh refinement capabilities by using a characteristic length of 200m for the fault-boundary and a length of 1000m for the free-surface and outflow boundaries. The volume mesh in between was coarsened gradually from 200m to 1000m. Simulation results of the two benchmarks, for an exemplary fault-receiver are presented in Fig. 3.



(a) regular (b) extruded, frontal (c) Delaunay

Figure 2: Different tetrahedral mesh types used throughout this project. The planar fault in the center is highlighted in black.



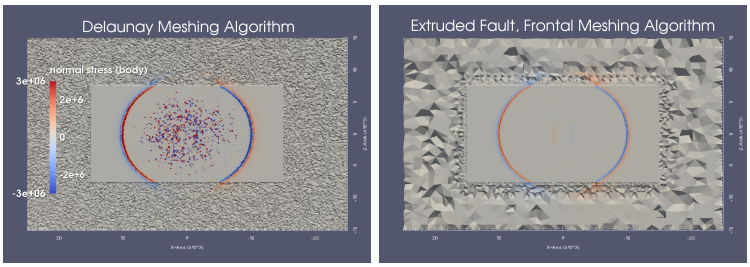
(a) TPV3 (b) TPV5

Figure 3: Simulation results of the TPV3 and TPV5 benchmarks at fault receiver faultst075dp075 (strike 7.5 km, dip 7.5 km). For both benchmarks we used a fourth order wave propagation scheme in EDGE and a characteristic length of 200m at the fault. The TPV5 results are compared with results of the two codes SORD and MAFE, obtained from [16].

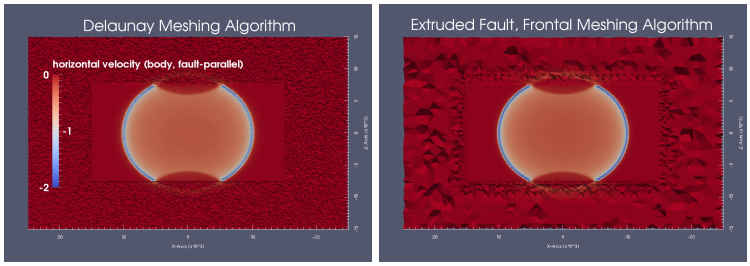
4 Discussion and Outlook

As shown in Fig. 4, we observed unphysical behavior of the scheme, briefly described in Sec. 2 and in greater detail in [13, 20, 19], when using meshes generated via gmsh's unstructured Delaunay meshing algorithm (see Fig. 4c). The observed artifacts, visible in Fig. 4a, seem to be dominant in the fault-normal stress component. In contrast, the fault-parallel, horizontal velocity remains smooth, even if unstructured Delaunay meshing is used.

In addition, our simulations on rectangular hexahedral meshes and structured tetrahedral meshes (see Fig. 2a) solve the TPV3 test case as expected. This led to our temporary workaround of using an extruded fault geometry for unstructured meshes (see Fig. 2b). Our comparisons to the software package SeisSol (version c9d8fc56d11cd), show comparable behavior of both codes. SeisSol uses a similar numerical scheme, but a completely independent implementa-



(a) Normal stress, Delaunay (b) Normal stress, extruded



(c) Horizontal velocity, Delaunay (d) Horizontal velocity, extruded

Figure 4: Simulation results of the TPV3 benchmark with a characteristic length of 200m at the fault. The computational domain is cut along the xz-plane (y-direction is fault normal). Shown are the normal stress and horizontal velocity component in the volume. Note, that these are not the middle states of the intermediate Riemann solutions. Compared are unphysical results obtained through completely unstructured meshes (Delaunay, see Fig. 2c) and results of an extruded fault geometry having a symmetric layer on each side of the fault (see Fig. 2b).

tion, additionally indicating a troubled numerical scheme rather than pointing to a bug in the actual implementation.

We are currently investigating the observations made throughout this SCEC project. Current and future steps include 1) isolation of the solver's troubled behavior in specific two-dimensional and three-dimensional test cases, 2) comparison of our high-order ADER-DG solver to EDGE's separated, first order Finite Volume solver to determine the need for a DG-limiter, and 3) comparison to other approaches, e.g., [3, 21]. Here, EDGE's generic support for different convergence rates, element types, regular and unstructured meshes, as well as different numbers of dimensions proofs to be of high value in our investigations. Further, our results show the risk of "overfitting" benchmark problems by using fault-symmetric meshes, e.g., regular, rectangular meshes. Such setups are only suited for a limited problem range and the underlying numerical scheme might fail, if applied to fault geometries with curvature or branches, where symmetry of fault-adjacent elements can not be enforced. We are planning to share the test cases under development, using asymmetric meshes for planar faults, with the community.

References

[1] A. Breuer, A. Heinecke, M. B. High Performance Algorithms and Software for Seismic Simulations. Poster at 17th SIAM Conference on Parallel Processing for Scientific Computing, April 2016.

[2] A. Breuer, A. Heinecke, Y. C. Recent Advances of the ADER-DG Finite Element Method for Seismic Simulations. Poster at SCEC Annual Meeting 2016, September 2016.

[3] Benjemaa, M., Glinsky-Olivier, N., Cruz-Atienza, V. M., and Virieux, J. 3-d dynamic rupture simulations by a finite volume method. *Geophysical Journal International 178*, 1 (2009), 541–560.

[4] Breuer, A. High Performance Earthquake Simulations. Presentation at Kolloquium zum GI Dissertationspreis 2015, May 2016.

[5] Breuer, A. Petascale Local Time Stepping for the ADER-DG Finite Element Method. Presentation at ICOSAHOM 2016, June 2016.

[6] Breuer, A. Petascale Local Time Stepping for the ADER-DG Finite Element Method. Presentation at IPDPS 2016, May 2016.

[7] Breuer, A. Seismic Simulations with Local Time Stepping on Xeon and Xeon Phi Processors. Presentation at IXPUG 2016 Annual US Meeting, September 2016.

[8] Breuer, A. Towards 100+ PFLOPS: A Roadmap for AWP-ODC and UQ with DG-FEM on next-generation Intel Xeon Phi Processor. Presentation at SCEC Cybershake and High-F Workshop, June 2016.

[9] Breuer, A., Heinecke, A., and Bader, M. Petascale local time stepping for the ader-dg finite element method. In *Parallel and Distributed Processing Symposium, 2016 IEEE International* (2016), IEEE, pp. 854–863.

[10] Breuer, A., Heinecke, A., and Cui, Y. EDGE: Extreme-Scale Discontinuous Galerkin Environment. Presentation at SSA Annula Meeting 2017 (upcoming), April 2017.

[11] Breuer, A., Heinecke, A., and Cui, Y. EDGE: Extreme Scale Fused Seismic Simulations with the Discontinuous Galerkin Method. In *ISC High Performance (accepted for publication)* (2017).

[12] Breuer, A., Heinecke, A., Rettenberger, S., Bader, M., Gabriel, A.-A., and Pelties, C. Sustained petascale performance of seismic simulations with seissol on supermuc. In *International Supercomputing Conference* (2014), Springer, pp. 1–18.

[13] de la Puente, J., Ampuero, J.-P., and Käser, M. Dynamic rupture modeling on unstructured meshes using a discontinuous galerkin method. *Journal of Geophysical Research: Solid Earth 114*, B10 (2009).

[14] Dumbser, M., and Käser, M. An arbitrary high-order discontinuous galerkin method for elastic waves on unstructured meshes—ii. the three-dimensional isotropic case. *Geophysical Journal International 167*, 1 (2006), 319–336.

[15] Geuzaine, C., and Remacle, J.-F. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International Journal for Numerical Methods in Engineering 79*, 11 (2009), 1309–1331.

[16] Harris, R., Barall, M., Archuleta, R., Dunham, E., Aagaard, B., Ampuero, J., Bhat, H., Cruz-Atienza, V., Dalguer, L., Dawson, P., et al. The scec/usgs dynamic earthquake rupture code verification exercise. *Seismological Research Letters 80*, 1 (2009), 119–126.

[17] Heinecke, A., Breuer, A., Bader, M., and Dubey, P. High order seismic simulations on the intel xeon phi processor (knights landing). In *International Conference on High Performance Computing* (2016), Springer, pp. 343–362.

[18] Heinecke, A., Breuer, A., Rettenberger, S., Bader, M., Gabriel, A.-A., Pelties, C., Bode, A., Barth, W., Liao, X.-K., Vaidyanathan, K., et al. Petascale high order dynamic rupture earthquake simulations on heterogeneous supercomputers. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (2014), IEEE Press, pp. 3–14.

[19] Pelties, C., Gabriel, A., and Ampuero, J. Verification of an ader-dg method for complex dynamic rupture problems. *Geoscientific Model Development Discussion 6* (2013), 5981–6034.

[20] Pelties, C., Puente, J., Ampuero, J.-P., Brietzke, G. B., and Käser, M. Three-dimensional dynamic rupture simulation with a high-order discontinuous galerkin method on unstructured tetrahedral meshes. *Journal of Geophysical Research: Solid Earth 117*, B2 (2012).

[21] Tago, J., Cruz-Atienza, V. M., Virieux, J., Etienne, V., and Sánchez-Sesma, F. J. A 3d hp-adaptive discontinuous galerkin method for modeling earthquake dynamics. *Journal of Geophysical Research: Solid Earth 117*, B9 (2012).

[22] Toro, E. F. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.