High performance earthquake simulations in viscoelastic media using SeisSol

Carsten Uphoff[1], Michael Bader[1], Sebastian Rettenberger[1], Alice-Agnes Gabriel[2]
(1) Department of Informatics, Technical University of Munich
(2) Department of Earth and Environmental Sciences, LMU Munich

Introduction

SeisSol solves the seismic wave equations in velocity-stress formulation in elastic [1], viscoelastic [2], and viscoplastic media (see poster of Stephanie Wollherr) on unstructured tetrahedral meshes.

Standard rupture format

SeisSol supports the standard rupture format version 1.0 and 2.0 (see [3]) for moment tensor (point) sources.

- Enables large number of subfaults with distint source time functions
- Supports rake rotation
- Mapping to unstructured mesh via converter. Supports geocentric coordinates or any projection available in the library proj.4.

Dynamic rupture

- Simulation of rupture dynamics coupled to wave propagation
- Tetrahedral meshes allow complicated fault geometry
- Flux-based implementation remains free of spurious oscillations [4]

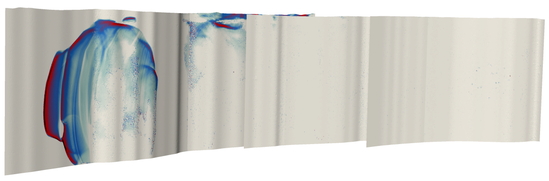

Figure 1: Absolute slip rate on Landers fault system.

3D velocity models

- SeisSol can use ASAGI, “a pArallel Server for Adaptive GeoInformation”, which allows fast loading of 3D datasets.
- Example: Lamé parameters on volume $187x210x103$ km^3 with 200 m resolution, i.e. 6.6 GB data, took 6.4s on 512 nodes.

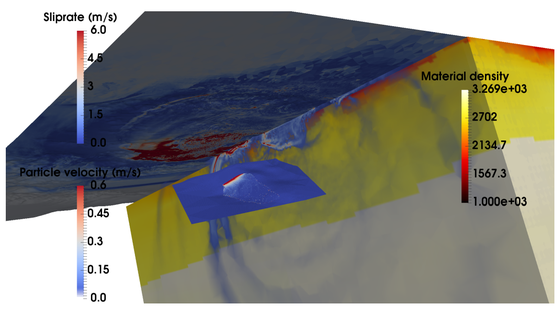

Figure 2: 1994 Northridge earthquake with 3D velocity model [5].

Code generation

The ADER-DG scheme of SeisSol boils down to small matrix matrix multiplications, where the sizes of the matrices depend on the order of the scheme.
In order to achieve high performance, we generate customized code for each matrix multiplication using libxsmm as a back-end. This approach leads to a high hardware utilisation, reaching up to 8.6 PFLOPS on Tianhe-2 [6].

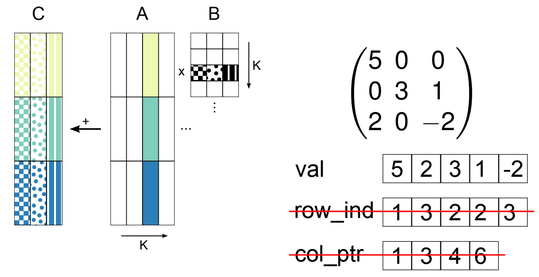

Figure 3: We generate optimized microkernels for dense x dense (left) and unroll sparse x dense or dense x sparse multiplications (right).

Recently, we extended our code generator in order to incorporate viscoelastic attenuation and simplify future model extensions [7]:

- Automatic detection of irrelevant matrix entries in matrix chain products

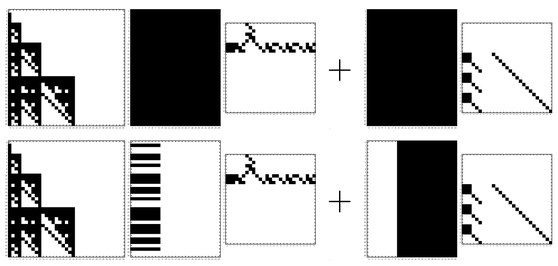

Figure 4: Partial kernel before (top) and after (bottom) removing irrelevant entries in matrix chain products.

- Automatic determination of zero-paddings for aligned loads and stores
- Enabling block decompositions for structured sparse matrices

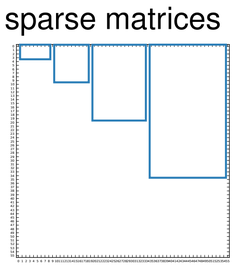

Figure 5: Block decomposition of stiffness matrix.

Furthermore, we exploit the tensor product structure of the discretisation in order to reduce the computational overhead of many relaxation mechanisms:

$$K\left(Q_{n,e}\ Q_{n,v}\right)\begin{pmatrix} A_1 & \omega \otimes A_2 \\ 0 & 0 \end{pmatrix} = \left(KQ_{n,e}A_1\ \ \omega \otimes KQ_{n,e}A_2\right)$$

Validation

Formal convergence test: Plane wave in homogeneous medium. Depending on the number of basis functions used we obtain up to 7th order convergence.

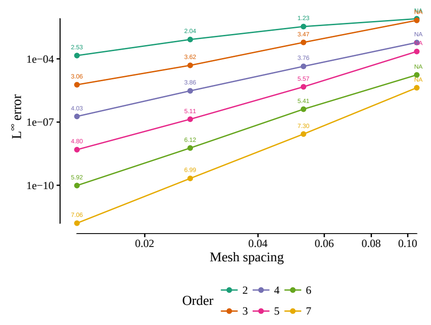

Figure 6: Double logarithmic plot of mesh spacing vs. maximum error.

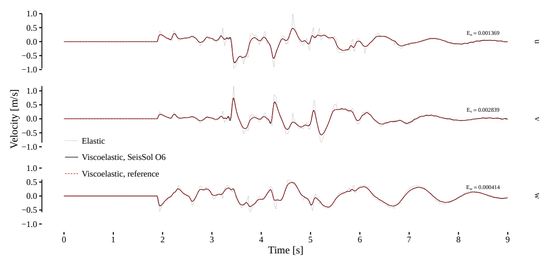

Figure 7: Layer over halfspace benchmark (LOH.3) as proposed in [8].

Performance

Evaluated on SuperMUC Phase 2. Each node has a dual-socket Haswell processor with 28 cores.

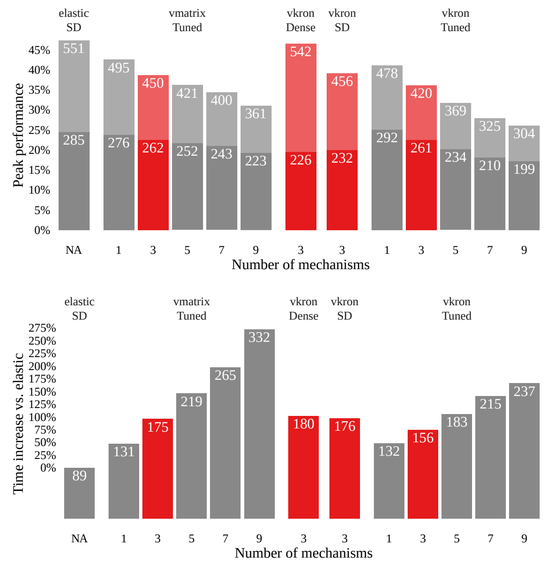

Figure 8: Single node performance, 6th order. Overhead of viscoelasticity about 75 %.

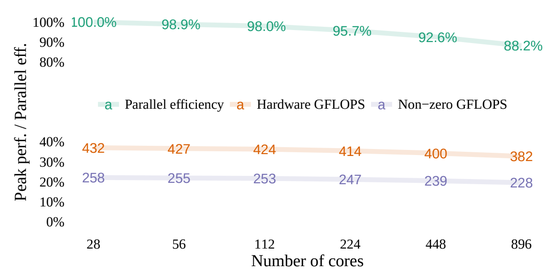

Figure 9: Strong scaling of layer over halfspace benchmark with 1.1 million elements.

Performance (cont’d)

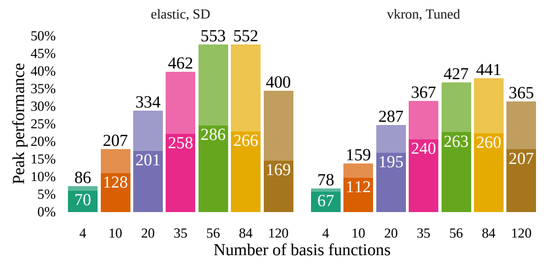

Figure 10: High order leads to compute bound code.

Application

We simulated the 1994 Northridge earthquake using a mesh with 75 million tetrahedrons and frequency-dependent Q as in [9]:

$$Q(f) = \begin{cases} Q_0 & 0 < f < f_T, \\ Q_0 \cdot (f/f_T)^\gamma & f_T \leq f, \end{cases}$$

with $Q_0 = 0.1 v_s, \gamma = 0.6, f_T = 1$Hz, and $Q_P = 2Q_S$. We use 3 relaxation mechanims and logarithmic frequency spacing [10] for fitting the Q model.

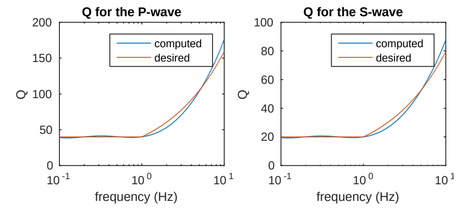

Figure 11: Q model using 3 relaxation mechanisms and $Q_0 = 20$.

On 512 nodes (14336 cores) we achieve a sustained performance of 187 TFLOPS, which corresponds to about 30% of peak performance.

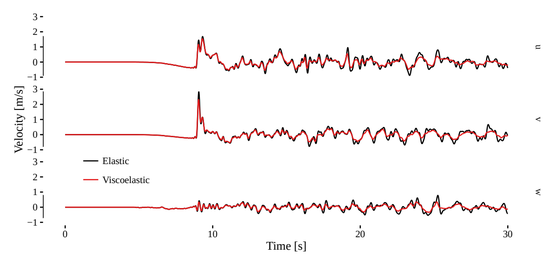

Figure 12: EW, NS, and vertical velocity component at Sylmar converter station east (34.312°N, 118.481°W).

References

[1] M. Dumbser and M. Käser, Geophysical Journal International 167, 319 (2006).

[2] M. Käser, M. Dumbser, J. de la Puente, and H. Igel, Geophysical Journal International 168, 224 (2007).

[3] R. Graves, Standard rupture format 2.0, `http://hypocenter.usc.edu/research/SRF/srf-v2.0_rev1.pdf`, 2014, Accessed: 2016-08-08.

[4] C. Pelties, J. de la Puente, J.-P. Ampuero, G. B. Brietzke, and M. Käser, Journal of Geophysical Research 117 (2012).

[5] S. Rettenberger, O. Meister, M. Bader, and A.-A. Gabriel, ASAGI – A parallel server for adaptive geoinformation, in *2016 Exascale Applications & Software Conference (EASC2016)*, Stockholm, 2016.

[6] A. Heinecke et al., Petascale high order dynamic rupture earthquake simulations on heterogeneous supercomputers, in *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 3–14, 2014.

[7] C. Uphoff and M. Bader, Generating high performance matrix kernels for earthquake simulations with viscoelastic attenuation, in *Proceedings of HPCS 2016*, 2016.

[8] S. M. Day et al., Tests of 3d elastodynamics codes: Final report for lifelines program task 1a02, Pacific Earthquake Engineering Research Center, 2003.

[9] K. B. Withers, K. B. Olsen, and S. M. Day, Bulletin of the Seismological Society of America 105, 3129 (2015).

[10] H. Emmerich and M. Korn, Geophysics 52, 1252 (1987).

uphoff@in.tum.de

github.com/SeisSol/SeisSol