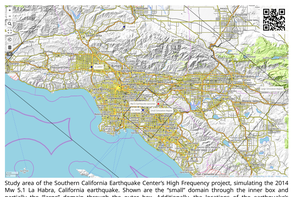
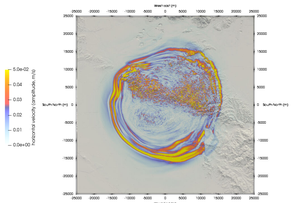
Tensor Processing Primitives in the Computational Sciences: Earthquake Simulations

A. Breuer, E. Georganas (Intel), A. Heinecke (Intel), A. Noack, K. Voronin (Intel)

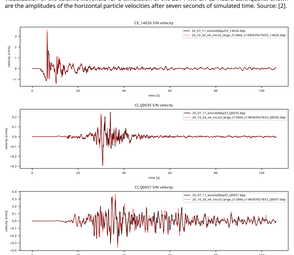
Earthquake Simulations



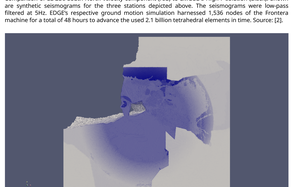
Study area of the Southern California Earthquake Center's High Frequency project, simulating the 2014 Mw 5.1 La Habra, California earthquake. Shown are the "small" domain through the inner box and partially the "large" domain through the outer box. Additionally, the locations of the earthquake's epicenter and three stations are given. Source: [2].



Visualization of the seismic wavefield 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. Source: [2].



Comparison of EDGE's South-North velocity component (red) to EMO3D's High-F solution (black). Shown are synthetic seismograms for the three stations depicted above. 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 elements in time. Source: [2].



Visualization of a velocity-aware tetrahedral mesh for the small La Habra simulation region. 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.

High-frequency Ground Motion Simulations: Grand challenge in computational seismology. Challenge is driven by two main factors which push solvers to their limit:

- Higher frequencies require extended numerical models, e.g., by considering anelasticity, or by including mountain topography.
- Even if models could be kept unchanged, simply doubling the frequency content of a seismic wave propagation solver requires a sixteen-fold increase in computational resources due to the used four-dimensional space-time domains.

2014 Mw 5.1 La Habra, California earthquake: High-Frequency (High-F) ground motion verification project of the Southern California Earthquake Center built the initial umbrella of conducted runs in [2]. Exemplary results of this work are given in the figures to the left.

Excellent agreement of synthetic seismograms when comparing EDGE and finite-difference solver EMO3D for the South-North velocity component. Map shows the location of the three stations; obtain an interactive version of the map by scanning the QR code.

Work presented in this poster builds on EDGE's extension of High-F's demanding verification setup. Additionally, our presented large-scale simulations consider two key model extensions which pose major obstacles for many earthquake packages: introduction of topography and the utilization of problem-aware meshes.

EDGE: Arbitrary high-order DERivatives (ADER) Discontinuous Galerkin (DG) finite element software EDGE solves the three-dimensional anelastic wave equations.

- Uses ADER-DG finite element method with tetrahedral elements.
- Focus: static meshes with high geometric complexity.
- Unique support for fused simulations exploiting inter-simulation parallelism.
- Parallelization: JITted kernels for highest performance on many recent CPU architectures (AVX2, AVX512, ASIMD, SVE); advanced OpenMP+MPI for hidden communication.
- Extremely scalable with sustained petascale performance: 10.4 FP64 PFLOPS on Cori II, 1.1 FP32 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), modeling and simulation pipeline relies exclusively on open source software (https://dial3343.org).

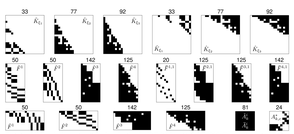
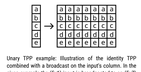


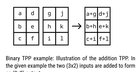
Illustration of all involved sparsity patterns for a fourth order ADER-DG discretization in EDGE. The numbers on top give the number of non-zero entries in the sparse matrices.

Tensor Processing Primitives

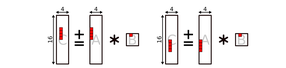
Tensor Processing Primitives (TPPs): Small set of about 50 versatile low-dimensional tensor operators which allow to formulate complex high-dimensional tensor operations [1].



Unary TPP example: Illustration of the identity TPP combined with a broadcast on the input's column. In the given example the (5x1) input is broadcasted to an (5x7) output.



Binary TPP example: Illustration of the addition TPP. In the given example the two (3x2) inputs are added to form an (3x2) output.



Ternary TPP example: Illustration of the GEMM TPP. Shown are a respective ASIMD kernel for M=16, N=4 and K=4 as implemented in LIBXSMM. The fifth and sixth ASIMD FMLA instructions are highlighted on the left and right respectively.

Sparse TPPs: The (sparse matrix) x (3D tensor) TPP and (3D tensor) x (sparse matrix) TPP are crucial for the application workload of this poster. As illustrated, the implemented TPPs exploit sparsity while vectorizing over the third dimension.

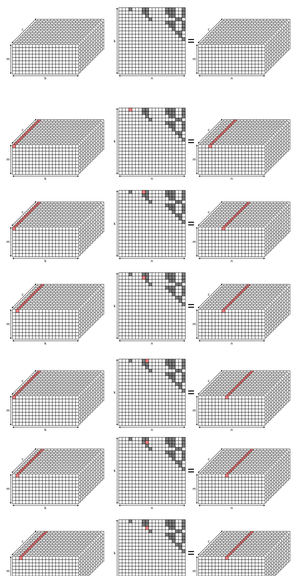
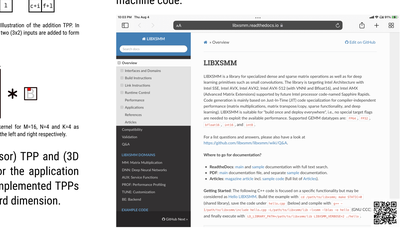


Illustration of the (3D tensor) x (sparse matrix) TPP. In the given example M=9, N=20, K=20 and C=16. Each of the input tensor's sixteen MxK matrices is multiplied with a sparse KxN matrix. Respectively, resulting matrices of the output tensor are obtained. The illustrated LIBXSMM implementation of the TPP vectorizes over the third dimension C. Shown are the first six FMA SIMD operations. The discussed seismic solver uses the TPP extensively when fusing sixteen simulations in single precision floating for AVX512 and 512-bit SVE.

LIBXSMM: Often times machine learning or computational sciences workloads require few TPP instantiations, i.e., they use the same tensor shape over and over and over again. Our approach in LIBXSMM hardwires kernel- and hardware-specific optimizations through Just-In-Time (JIT) generation of machine code.



Screenshot of LIBXSMM's entry page. The library builds the computational backbone of many applications in the computational sciences and in machine learning. LIBXSMM is available under the permissive BSD3-clause license.

Previous State of the Art:

- Full support of tensor processing primitives for x86 vector extensions (AVX2 and AVX512) and AArch64's ASIMD in LIBXSMM.
- LIBXSMM's implementation for Scalable Vector Extension was limited, i.e., only supported dense matrix-matrix multiplications.
- Solver EDGE only used LIBXSMM-JITted small GEMMS and (sparse matrix) x (3D tensor) and (3D tensor) x (sparse matrix) kernels. Remainder parts compiler-optimized.

Contributions:

- Scalable Vector Extension (SVE) support for unary and binary tensor processing primitives (256-bit and 512-bit vectors) in LIBXSMM.
- SVE support for (sparse matrix) x (3D tensor) and (3D tensor) x (sparse matrix) tensor processing primitives.
- Integration of unary and binary tensor processing primitives into Extreme-scale Discontinuous Galerkin Environment (EDGE) for demanding seismic setups.
- Support for TPP-equation-based integration kernels (time, volume) in EDGE.
- Thorough benchmarking, analyses and discussion of key standalone primitives on eight recent processors: 2nd Generation Intel(R) Xeon(R) Scalable Processor formerly code-named Cascade Lake (abbreviation used here: CLX), 3rd Generation Intel(R) Xeon(R) Scalable Processor formerly code-named Icelake (abbreviation used here: ICX), 4th Generation Intel(R) Xeon(R) Scalable Processor formerly code-named Sapphire Rapids (abbreviation used here: SPR), AMD EPYC 7742/7J13, Fujitsu A64FX and Amazon Graviton2/Graviton3.
- Full application benchmarking on eight recent processors. Discussion of the unstructured (time and space) solver EDGE's results when running a single seismic forward simulation and when fusing simulations.
- Two large-scale simulations of the 2014 Mw5.1 La Habra, California earthquake on up to 2,048 nodes of the Frontera supercomputer.

Performance Portability

Supported Vector Extensions: Performance portability of TPP-based software depends on an efficient backend. LIBXSMM backend supports the following vector extensions:

Extension	Example Microarchitectures
AVX2	Zen 3
AVX512	CLX, ICX
Neon/ASIMD	Neoverse N1
SVE256 (this work)	Neoverse V1
SVE512 (this work)	A64FX

Testbeds: A large variety of used systems showcases the performance portability of the TPP-enhanced solver EDGE. The microbenchmarked FP32 performance is provided below. Benchmark hot-loops over independent FMA operations.

Processor	Microarch.	#Cores	TFLOPS
Intel Xeon Platinum 8280	CLX	28	4.28
Intel Xeon Platinum 8380	ICX	40	6.36
4th Gen. Xeon "SPR"[a]	SPR	hcc	≥8.5
AMD EPYC 7742	Zen 3	64	5.29
AMD EPYC 7J13	Zen 3	64	≥5.9
Amazon Graviton2	Neoverse N1	64	2.56
Fujitsu A64FX	A64FX	48+4	5.45
Amazon Graviton3	Neoverse V1	64	3.42

[a] Pre-Production.

Small GEMM Performance: The ternary GEMM TPP is a key kernel for single seismic forward simulations using EDGE. Exact configuration of the matrix shapes depends on the setup. Examples for a fifth order instantiation of the solver with viscoelastic attenuation are presented below. All kernels were benchmarked out of a hot L1 cache on a single core.

Processor	M, N, K	Dense %Peak	Matrix Id	Sparse %Peak	Speedup: Sparse/Dense
Intel Xeon Platinum 8280	56, 9, 35	58.2	tet4_5_stiffV0	29.7	3.40
	35, 9, 20	37.1	tet4_4_stiffV0	32.3	5.64
	56, 9, 9	54.9	tet4_starMatrix	13.8	0.85
	35, 9, 9	30.5	tet4_starMatrix	15.8	1.75
Intel Xeon Platinum 8380	56, 9, 35	71.5	tet4_5_stiffV0	32.9	3.07
	35, 9, 20	48.8	tet4_4_stiffV0	25.9	3.44
	56, 9, 9	56.9	tet4_starMatrix	19.9	1.18
	35, 9, 9	40.4	tet4_starMatrix	27.1	2.26
4th Gen. Xeon "SPR"	56, 9, 35	82.0	tet4_5_stiffV0	65.3	5.31
	35, 9, 20	63.9	tet4_4_stiffV0	53.1	5.38
	56, 9, 9	77.6	tet4_starMatrix	30.5	1.33
	35, 9, 9	65.9	tet4_starMatrix	51.1	2.62
AMD EPYC 7742/7J13	56, 9, 35	92.9	tet4_5_stiffV0	61.8	4.43
	35, 9, 20	51.2	tet4_4_stiffV0	56.4	7.14
	56, 9, 9	93.6	tet4_starMatrix	62.1	2.24
	35, 9, 9	42.1	tet4_starMatrix	62.3	5.00
Fujitsu A64FX	56, 9, 35	60.8	tet4_5_stiffV0	24.2	2.65
	35, 9, 20	42.0	tet4_4_stiffV0	22.5	3.47
	56, 9, 9	47.3	tet4_starMatrix	22.3	1.59
	35, 9, 9	34.3	tet4_starMatrix	24.3	2.39
Amazon Graviton2	56, 9, 35	93.5	tet4_5_stiffV0	45.0	3.21
	35, 9, 20	77.4	tet4_4_stiffV0	39.0	3.26
	56, 9, 9	79.6	tet4_starMatrix	36.0	1.53
	35, 9, 9	68.4	tet4_starMatrix	38.2	1.89
Amazon Graviton3	56, 9, 35	94.3	tet4_5_stiffV_0	55.5	3.92
	35, 9, 20	69.6	tet4_4_stiffV_0	45.6	4.25
	56, 9, 9	81.2	tet4_starMatrix	43.6	1.81
	35, 9, 9	68.6	tet4_starMatrix	41.4	2.04

Sparse Matrix x 3D Tensor and 3D Tensor x Sparse Matrix: EDGE may run in two modes:

- **Single forward simulation**: Solver simulates a single earthquake, i.e., it reads single kinematic seismic source and computes respective seismic wave propagation. Uses small GEMMs. Standalone GEMM performance for shapes in volume kernel given as "Dense %Peak" in table below.
- **Fused forward simulations**: Solver simulates multiple earthquakes in a single run, i.e., it reads multiple kinematic seismic sources and computes respective seismic wave propagation for all of them. Vector-parallelism exploited over the sources (right-hand side). Uses (sparse matrix) x (3D tensor) or (3D tensor) x (sparse matrix) kernels.

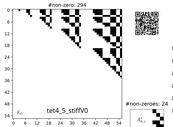


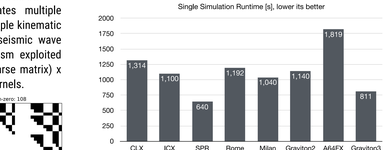
Illustration of the sparsity patterns of two stiffness matrices for a sixth order method with a degree five tetrahedral basis (tet4_5_stiffV0) and a fifth order method with a degree four basis (tet4_4_stiffV0) used in EDGE's volume integration routine. Futher, the sparsity pattern of matrix tet4_starMatrix which is also used in the volume integrator is given. EDGE's single simulations only exploit zero blocks and compute all matrix ops dense. EDGE's fused simulations fully exploit sparsity and vectorize over the right-hand side.

In fused mode, EDGE's performance heavily depends on sparse TPPs. Approach enables exploitation of zeros in appearing matrix structures, e.g., the stiffness matrices. Example matrix patterns are given above. Scan QR code for all matrices. Respective standalone performance is given as "Sparse %Peak" in table below.

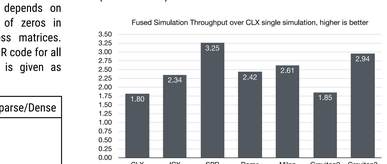
Application Performance and Scalability

Seismic Setup: Layer over halfspace 3 benchmark using order five in space and time. Three relaxation mechanisms (viscoelastisty) and local time stepping. Problem-adapated unstructured mesh with 743,066 tetrahedrons.

End-to-end Performance: We present the same simulation scenario (LOH.3) on all platforms.



We observe that the simulation time correlates mostly with the peak performance numbers discussed earlier, validating the performance portability of our approach. In this case performance portability means that we achieve the same fraction of theoretical peak performance when only executing FMAs out of registers. Only major exception is A64FX: already visible on for the kernel-only efficiency only plots as it has performance is comparibly low with masking/predication (small GEMMs).

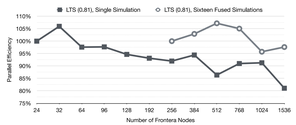


When running fused simulations we observe an additional speed-up in simulation throughput of about 1.6x - 2.2x. Again, this is seen on all test platforms with various architectures and microarchitectures. This further underlines our sustained performance portability w.r.t. relative peak performance even for small sparse linear algebra tensor operations in the HPC world.

Not shown: we also tested the codes without element-wise TPP on various X86 platforms with various compilers (ICC, GCC, clang) and measured performance variations of up to 10%. With our TPP-based approach those vanished while being within 1 - 1.5% of each fastest compiler choice.

Strong Scalability:

- Setup for the 2014 Mw 5.1 La Habra earthquake.
- Model enhancements over High-F version used for verification:
  - Incorporated topography information.
  - Reduced cutoff for minimal shear wave velocity from 500m/s to 250m/s.
- 237,861,634 tetrahedral elements, fith order ADER-DG.
- Sustained Local Time Stepping (LTS) performance on Frontera:
  - 2.25 FP32 PFLOPS for single simulation (includes zero ops).
  - 1.91 FP32 PFLOPS for fused simulations (sparse kernels, no zero ops).



Strong scaling studies of the solver EDGE for a ground motion simulation setup of the 2014 Mw 5.1 La Habra earthquake. The used problem-aware tetrahedral mesh has 237,861,634 elements and includes mountain topography. Shown is the parallel efficiency of EDGE's next-generation local time stepping scheme on the Frontera supercomputer. We used the solver by running a single forward simulation and by fusing an ensemble of sixteen simulations. Source [2].

Large-scale Simulations: The strong scaling runs were followed by large-scale simulations. Runs used 16 fused kinematic sources (descriptions of different realistic earthquake ruptures).

Min vs	#Elements	Theoretical LTS Speedup	#Frontera Nodes	Runtime
250m/s	242,595,220	9.92x	128	12.23h
100m/s	421,290,625	6.53x	2,048	14.86h

Conclusions

- Tensor Processing Primitives (TPPs) enable highest performance in the computational sciences inpendent of the used instruction set architecture or processor.
- Demonstrated performance portability of LIBXSMM's TPP backend for key kernels of solver EDGE on recent microarchitectures: CLX, ICX, SPR, Zen 3, A64FX, Neoverse N1 and Neoverse V1.
- Demonstrated feasibility of TPP-based approach in computational sciences through demanding earthquake simulations reaching high relative performance on all studied processors.

References

The optimized tensor processing primitives are available through LIBXSMM: https://github.com/libxsmm/libxsmm.

The Extreme-scale Discontinuous Galerkin Environment (EDGE) is available from: https://dial3343.org.

For workloads and configurations visit https://short.dial3343.org/sc22. Results may vary.

[1]: E. Georganas, D. Kalamkar, S. Avancha, M. Adelman, D. Aggarwal, C. Anderson, A. Breuer, J. Bruestle, N. Chaudhary, A. Kundu, D. Kutnick, F. Laub, V. Md, S. Misra, R. Mohanty, H. Pabst, B. Retford, B. Ziv, A. Heinecke. Tensor Processing Primitives: A Programming Abstraction for Efficiency and Portability in Deep Learning & HPC Workloads. Frontiers in Applied Mathematics and Statistics, section Mathematics of Computation and Data Science. Extended version of SC21 research paper arXiv: 2104.05755 (2022).

[2]: A. Breuer and A. Heinecke. Next-Generation Local Time Stepping for the ADER-DG Finite Element Method. IPDPS22. arXiv: 2202.10313 (2022).

Interested in this work? Reach out!

alex.breuer@uni-jena.de
https://dial3343.org
https://scalable.uni-jena.de

FRIEDRICH-SCHILLER-UNIVERSITÄT JENA