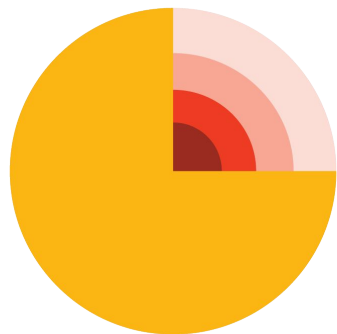




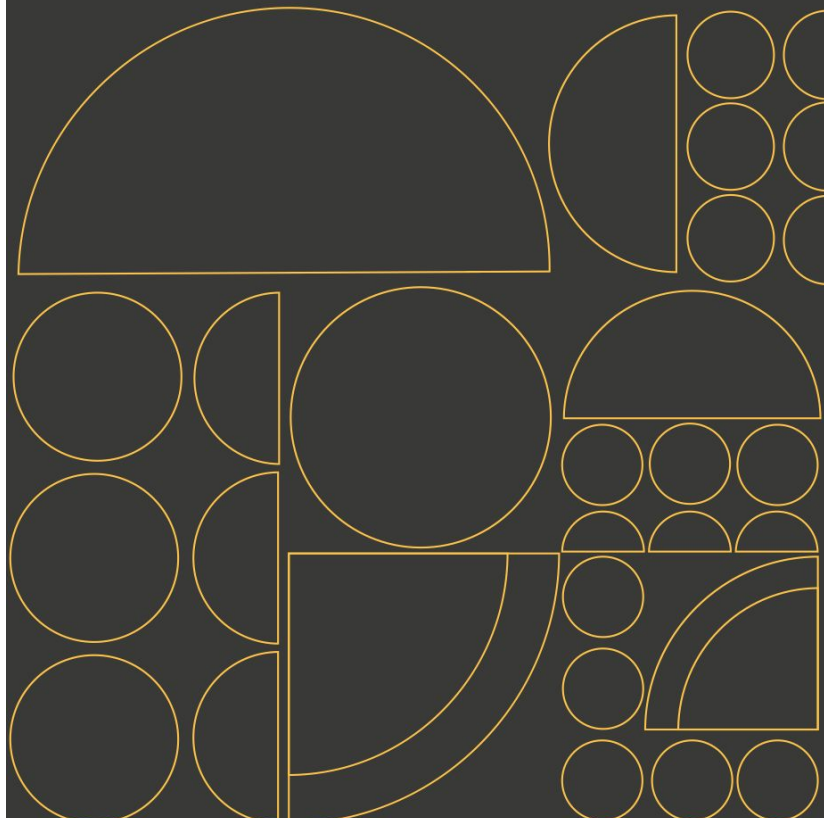
ChEESE

Tandem

Thomas Ulrich (LMU), Alice Gabriel (LMU, UCSD),
Brian Wylie (Jülich Supercomputing), Dave
May (UCSD), Bar Oryan (UCSD), Jeena Yun (UCSD),
James Biemiller (UCSD, USGS)

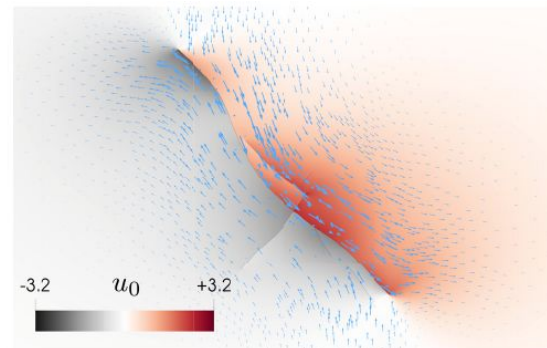
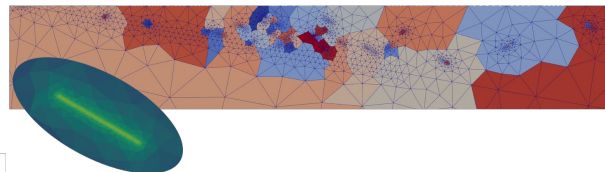
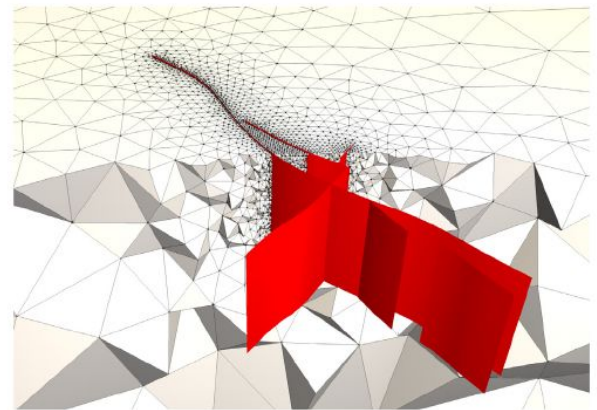
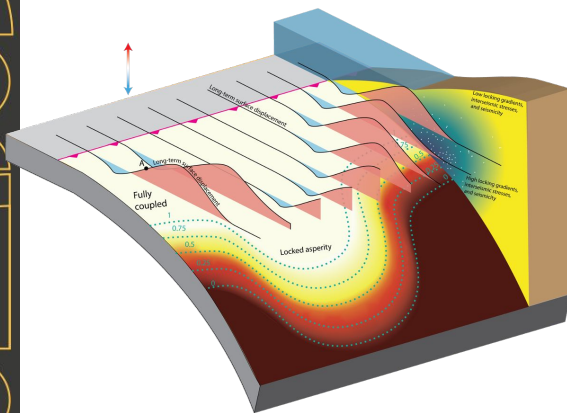


Project funded by EuroHPC under the grant agreement No
101093038.



Tandem: Overview

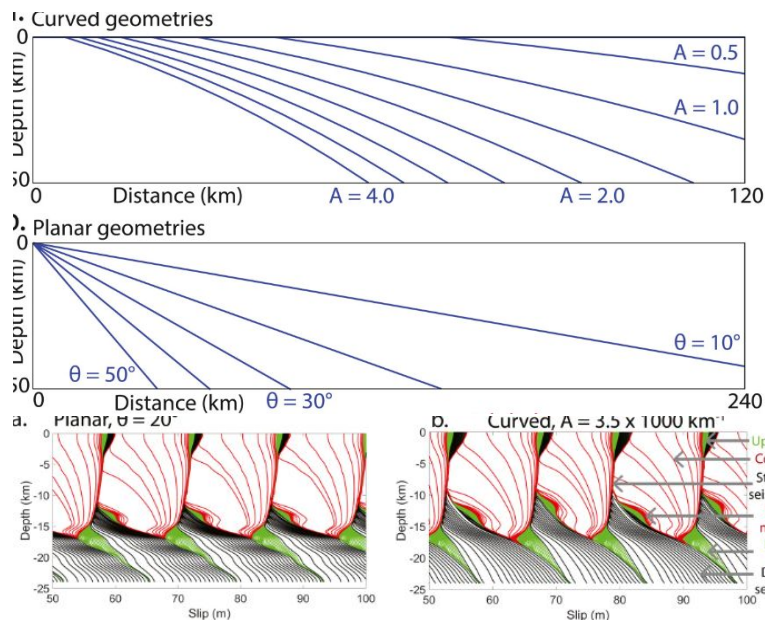
- **First HPC seismic cycle simulation framework**
- Symmetric interior penalty discontinuous Galerkin (SIPG) method, Uphoff et al. (2023)
- Simulates Sequences of Earthquakes and Aseismic Slip (SEAS) in 2D or 3D, including the complex geometries and (sub-elemental) heterogeneities of the subsurface
- Open source, <https://github.com/TEAR-ERC/tandem/>
- Unstructured curvilinear grids
- Arbitrary polynomial degree
- C++17 parallelized with MPI (& CUDA), built upon **PETSc**
- Extensive suite of verification problems



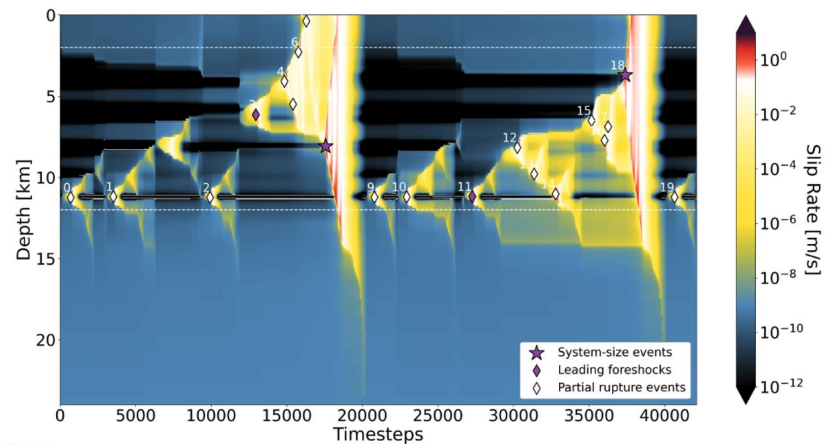
(a) x displacement (u_0)

Tandem: Cutting-edge applications

Biemiller, Gabriel, May, Staisch, *JGR - Solid Earth* (2024): **Subduction Zone Geometry Modulates the Megathrust Earthquake Cycle: Magnitude, Recurrence, and Variability**



Yun, Gabriel, May & Fialko (in prep.): **Controls of Dynamic and Static Stress Changes and Aseismic Slip on Delayed Earthquake Triggering in Rate-and-State Simulations of the 2019 Ridgecrest Earthquake Sequence**

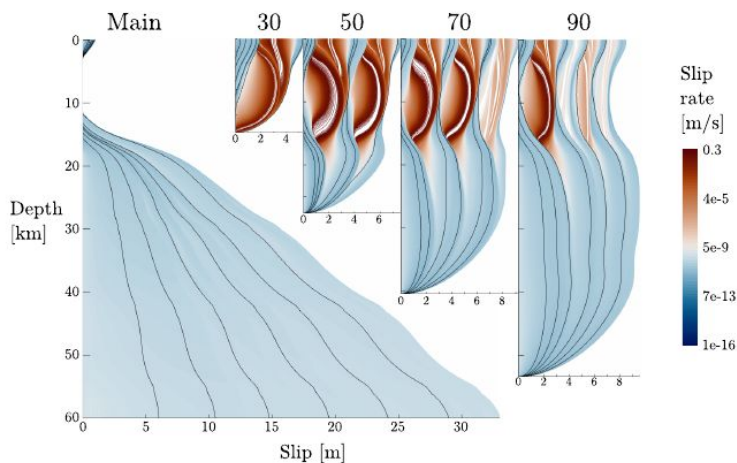
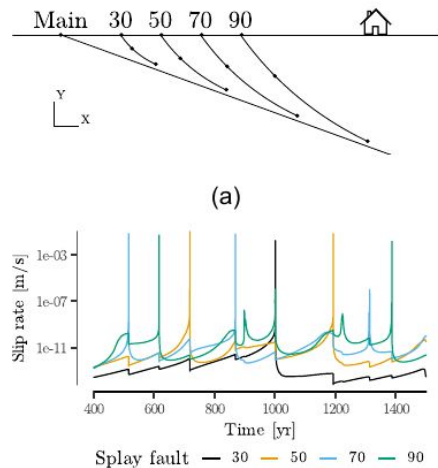


Oryan, Gabriel, May, *in prep*: **Breaking the elastic mold: The effect of upper plate permanent deformation on earthquake cycles**

WP2: Optimisation & new functionalities

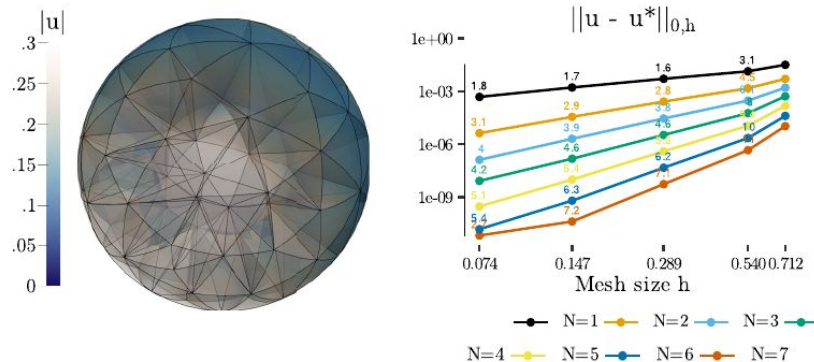
- **Checkpointing** of Green's functions (GFs): store/load GFs to and from disk (done)
- **Checkpointing** of PETSc TS object (time stepping) (done)
- **Weighted mesh partitioning**, penalizing fault elements in development (in dev.)
- **Fault-tagging** functionality for segment-wise initial and boundary condition (in dev.)
- Improved **multi-grid preconditioner** (in dev.)

Audit-driven developments towards optimizations: manual code annotation to better distinguished PETSc-related calls, and filter out non-completed communication



WP2: Portability campaign

- Portability to **NVIDIA GPU effective**: tandem runs on our local cluster with 2 GPU (NVIDIA GeForce RTX 3090)
- Will be tested on higher number of nVidia GPUs on **Leonardo**
- Problems installing the **software stack** with spack on **LUMI-G** (now fixed)
- Multigrid on GPU yields **different convergence history** to CPU, same PETSc options
 - float vs double precision, different linear solvers or preconditioners?
 - Convergence to be investigated with **verification problems**

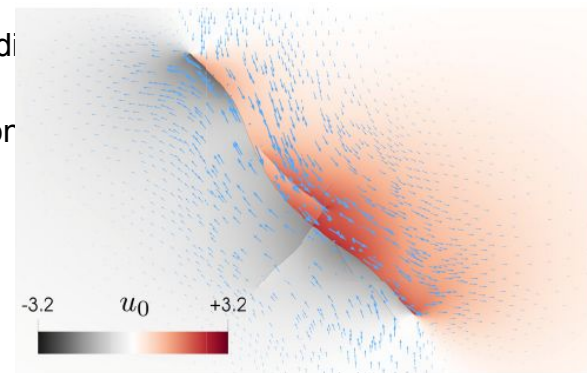


WP3: mini-app

Legend
Not tested
Fetch the code
Compile code
Run test cases
Validate results

			x86 profiling status	HBM study	ARM porting	ARM profiling	SVE study	Optimisation status + comments
Tandem	Mini-Tandem CPU testing feedback	Mini-Tandem GPU testing feedback	Not started	Not started	Not started	Not started	Not started	Not started

- Workflow for verifying the planned optimizations in developpement/di
 - No analytical solution for the given use case
 - We will validate with simpler use cases with analytical solution optimized vs non-optimized results.
 - Implementing automatized (github) workflow?
 - GPU verification pending



(a) x displacement (u_0)

Many technical and SC advancements!

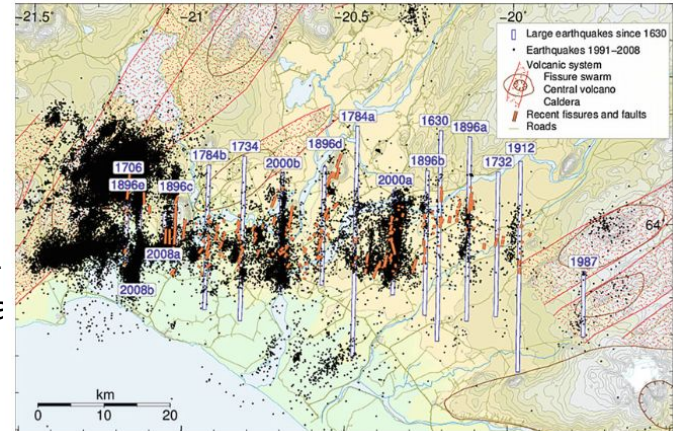
Workflow and software development

- github workflow for testing the code compiles on PR
- tandem spack package created
- clang-format/ clang-tidy in preparation
- Improved teaching material and introductory material, fully documented examples in development
- tandem app in the Quakeworx science gateway (SCEC)
- [qaaas.eosc silver badge](#)

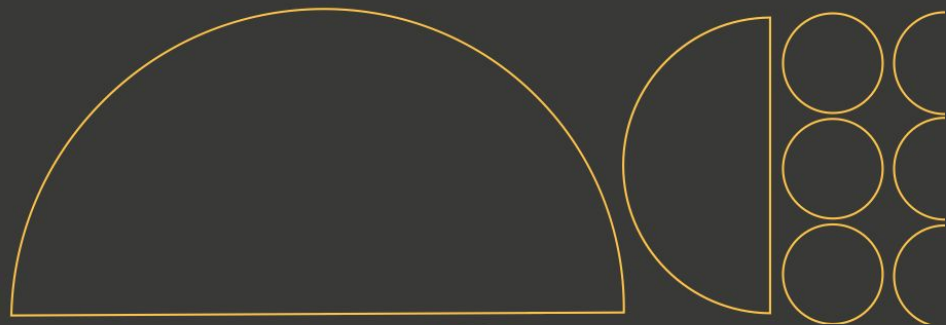


Involvement in SCs

- SC1.2, part 3 : SW Iceland seismic cycle simulation
- planned application on CPU-based machine, 65k nodes.hours
- constrained by the historical earthquake catalog (~100-deformation measurements and earthquake hypocentre
- modeling directly feed into SC1.3 (shaking scenarios)



Thank you!



Tandem

Thomas Ulrich (LMU), Alice Gabriel (LMU, UCSD),
Brian Wylie (Jülich Supercomputing), Dave
May (UCSD), Bar Oryan (UCSD), Jeena Yun (UCSD),
James Biemiller (UCSD, USGS)

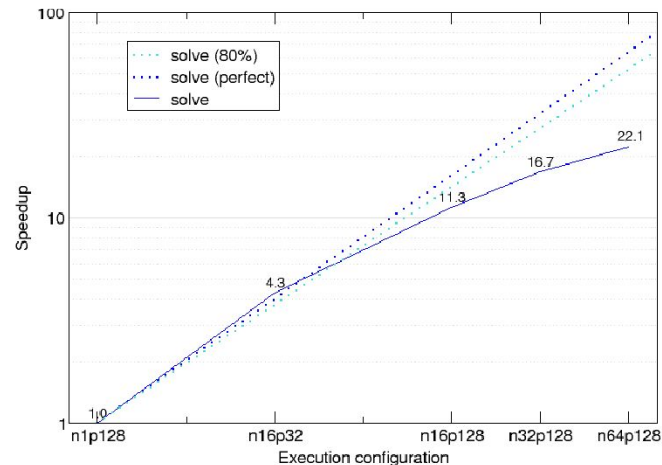
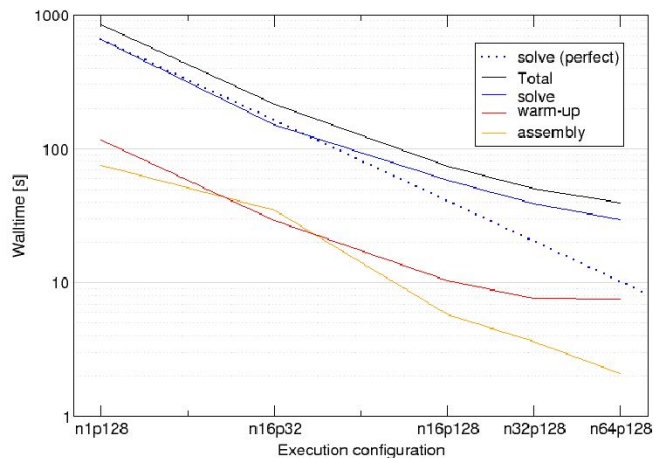
Upcoming Hack-a-thon Nov 5-7!



D2.1: Audit

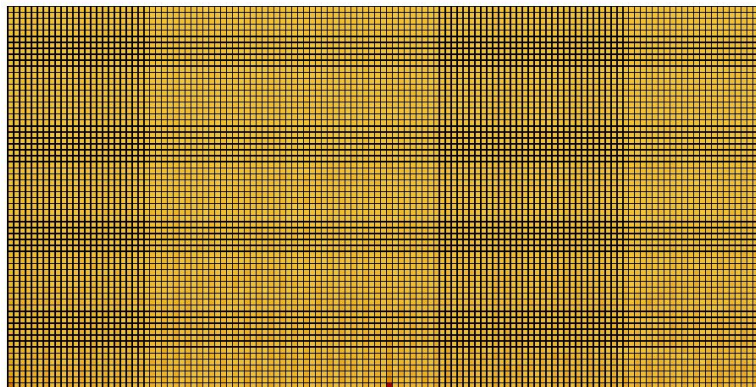
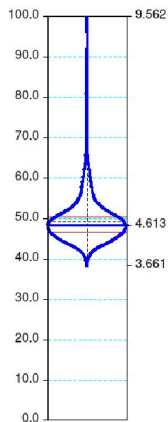
- Test case: Mini-app (static), 3-D intersecting multi fault scenario of **elastostatic** instantaneous displacement of the 2019 **Ridgecrest** earthquake
- Measurements with Scalasca/2.6.1 using **Score-P/8.3**
- **LUMI-C** (development access to LUMI-C, LUMI-G and Leonardo)
- 2800 GB of memory required (largemem (1 node) and standard partition used)
- Solve phase dominates execution time, efficiency drops **below 80% after 512 MPI processes**

MPI ranks	128	512	2048	4096	8192
MPI configuration	n1p128	n16p32	n16p128	n32p128	n64p128
Solve iterations	18	18	19	20	20
Wall-clock time [s]	850.78	216.77	74.76	50.73	39.42
Computation time [s]	95671.33	74046.30	76451.98	78419.05	78618.84
Global scaling efficiency	0.879	0.862	0.624	0.461	0.296
- Computation time scaling	1.000	1.292	1.251	1.221	1.217
- Parallel efficiency	0.879	0.667	0.499	0.377	0.243
- Load balance efficiency	0.904	0.882	0.735	0.783	0.638
- Communication efficiency	0.971	0.756	0.679	0.482	0.382



D2.1: Audit

- Strong scaling severely limited by both **growing load imbalance** and rapidly growing **MPI point-to-point communication**. Most data transferred in MPI **collective communication**.
- Solve based on **PETSc** and related call structure unfortunately **not distinguished in measurement**
 - would requires **manual annotation** of code with Score-P marker routines and/or an instrumentation wrapper generated for PETSc API routines
- PETSc uses MPI_Iprobe, MPI_Test & MPI_Testall **prolifically**, makes Score-P tracing impractical
 - would require option to **avoid tracing non-completed communication** operations
- With 8192 MPI ranks, **one specific rank** sends and receives much more data than others and takes more than twice compute time. Similar, but less extreme, for the other smaller configurations



Compute time distribution. Each row correspond to an MPI rank.