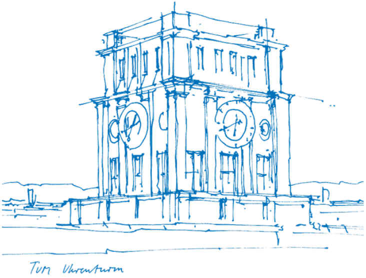
A variable-precision implementation of the ADER-DG algorithm

IEEE High Performance Extreme Computing Virtual Conference

Marc Marot-Lassauzaie & Michael Bader
Technical University of Munich

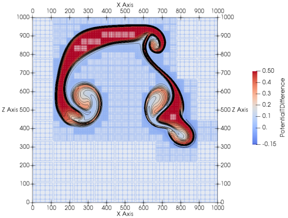
19th September 2025



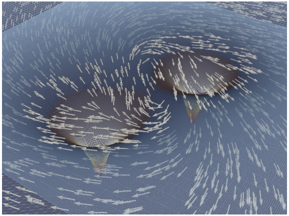
ExaHyPE – an “Exascale PDE Engine”

Goal: a PDE “engine” (as in “game engine”) ⇝ Reinarz et al.[1]

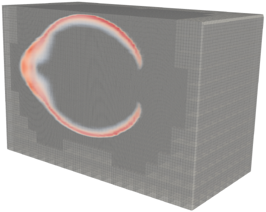
- Fixed numerics and mesh infrastructure, but *stay flexible w.r.t. PDE* (focus on hyperbolic conservation laws and high-order DG)
- Load balancing and adaptive mesh refinement via **Peano4**



atmospheric flows



relativistic astrophysics (Durham University)

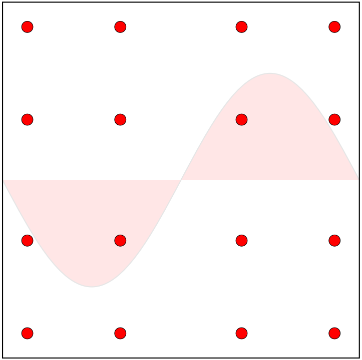


earthquake dynamic rupture

[1] Reinarz et. al., ExaHyPE: An engine for parallel dynamically adaptive simulations of wave problems. Comp. Phys. Comm. 254, 2020. `https://doi.org/10.1016/j.cpc.2020.107251`

ADER-DG

- High-order hyperbolic PDE solver
- Discontinuous Galerkin with ADER time stepping
- Piecewise polynomials within cells
- One data exchange per timestep
- Predictor-Corrector scheme



$$\int \partial_t U * \phi \, dx = \int F * \nabla \cdot \phi \, dx - \oint (F * \phi) \cdot \vec{n} \, ds \quad (1)$$

Why Reduced Precision?

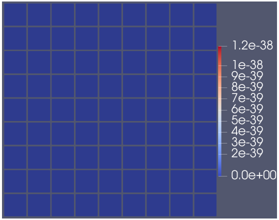
Speedup may result from:

- higher effective vectorization
- reduced bandwidth
- holding data in lower caches

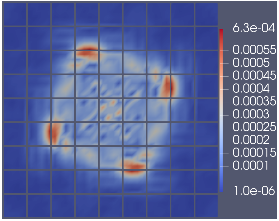
Precision	Significand bits	Exponents bits	Max. exponent	Decimal digits
bf16	7	8	127	≈ 2.5
fp16	10	5	15	≈ 3.3
fp32	23	8	127	≈ 7.2
fp64	52	11	1023	≈ 16
fp128	112	15	16383	≈ 34

The respective distribution of bits in different signed floating-point formats

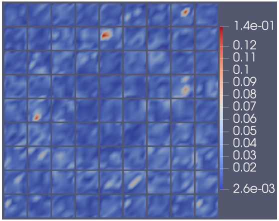
Cost of Reduced Precision



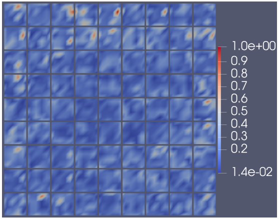
fp64



fp32



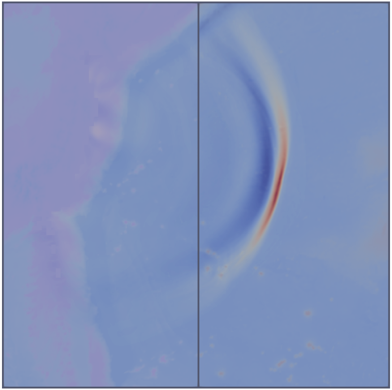
fp16



bf16

Variable precision

- Different precisions in different parts of domain
- Analogous to adaptive refinement
- Capture complicated but local phenomena with higher accuracy, and use lower precision for the rest of the domain



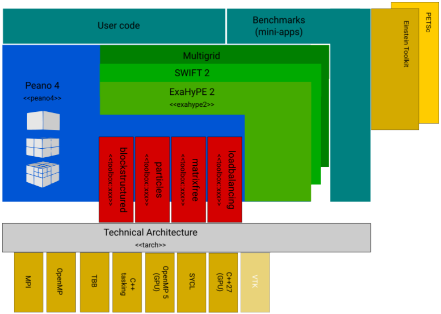
Variable precision simulation of the Tohoku, model see e.g. [3]

[3] Rannabauer et. al., ADER-DG with a-posteriori finite-volume limiting to simulate tsunamis in a parallel adaptive mesh refinement framework. Computers & Fluids 173, DOI: 10.1016/j.compfluid.2018.01.031

How to implement variable precision

- ExaHyPE 2 *generates* `C++` code using `Python`
- Code generation relies on Jinja2 templates

1. Conditionally deactivate each solver on parts of the domain
2. At the interface send projections to other solver.
3. Synchronize timesteps

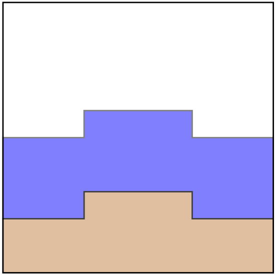


Architecture of the Peano 4 project [4]

[4] https://hpcsoftware.pages.gitlab.lrz.de/Peano/db/d3f/page_architecture_home.html

SWE

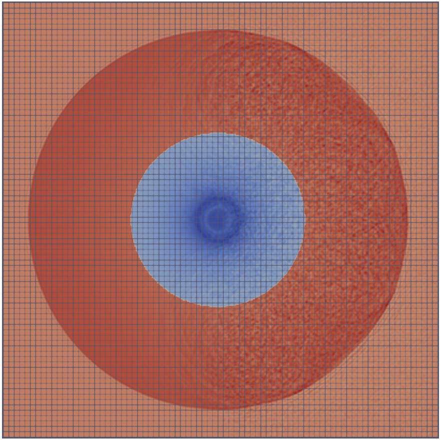
- Movement of a shallow fluid
- Constant water height over circular bathymetry
- Should lead to symmetric waves moving outward



The circular dam break initial condition used for verification of the shallow water equations.

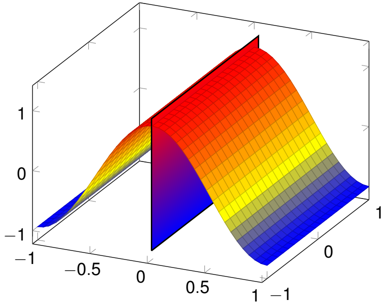
SWE

- Clear difference in quality between both halves of the domain
- `fp16` produces spurious oscillations, particularly along wave fronts
- Oscillations propagate back into the left half
- about half of the domain each in `fp16` and `fp64`



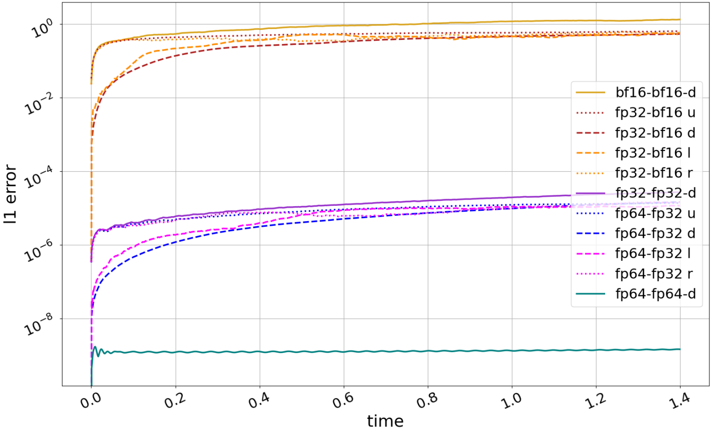
Elastic Planar Waves

- Propagation of waves through heterogeneous media
- Sinusoidal starting conditions propagate through the domain without deformation
- We simulate two full grid traversals
- about half of the domain each in `bf16` and `fp32`

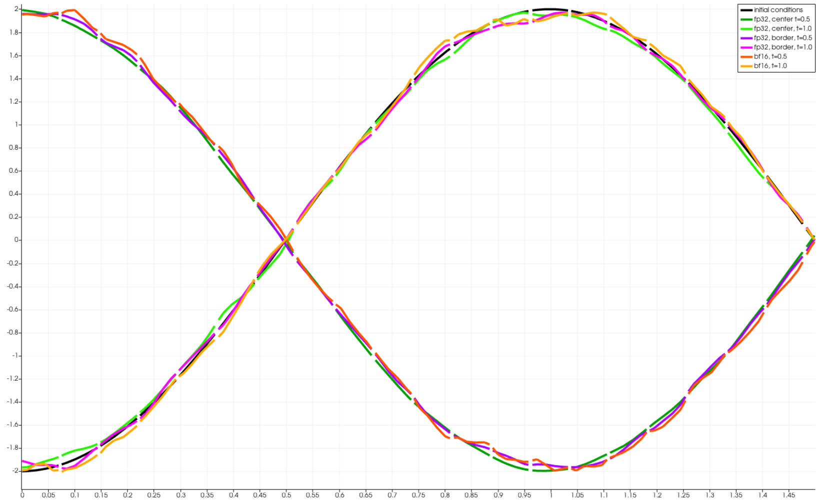


The planar-wave initial condition used for verification of the elastic equations.
$\cos(-\pi * x)$

Elastic Planar Waves

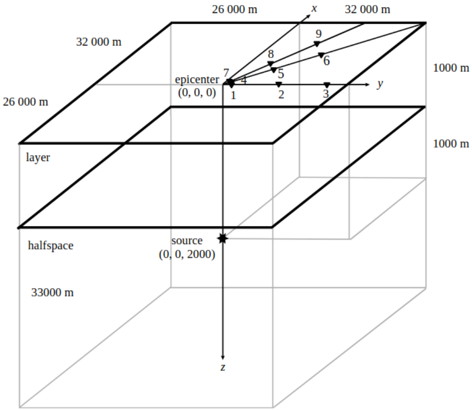


Elastic Planar Waves perpendicular to the split



The LOH1 Benchmark

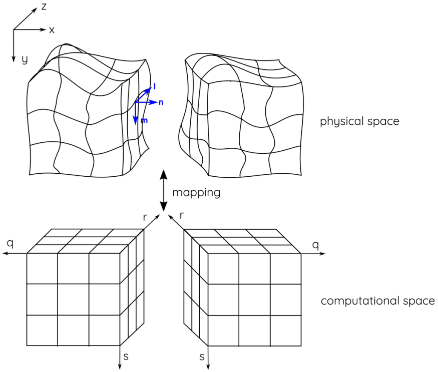
- Elastic wave propagation
- Single point source in an infinite domain
- Free surface at the top
- One layer with different material properties



Geometry of the LOH1 problem as defined in the SISMOWINE collection (http://www.sismowine.org/)

Curvilinear models

- Perfectly matched layers (PML) as boundary conditions
- Curvilinear meshes approximated by coordinate transformation
- Requires e.g. fluxes to take into account the projection
- In particular the Riemann solver is more numerically complicated

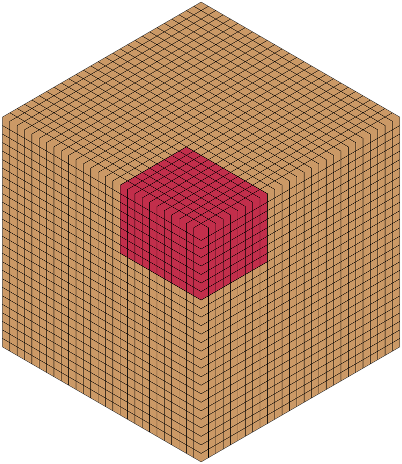


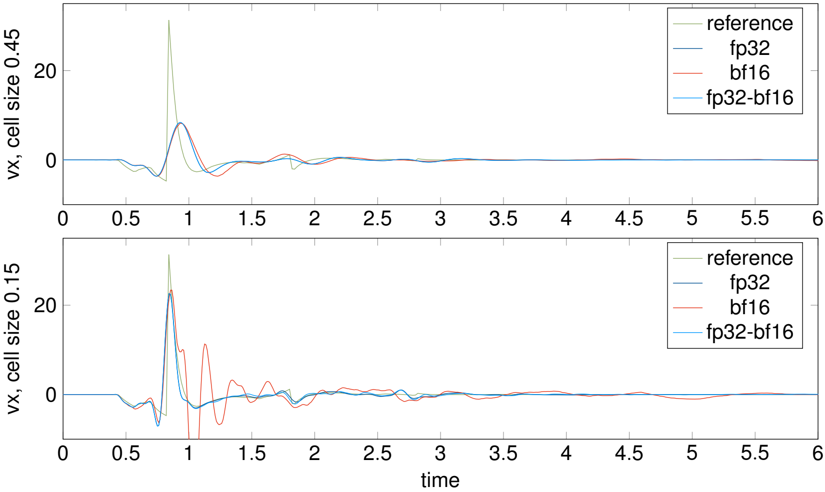
Curvilinear meshes and coordinate transformation [6]

[6] Duru et al. A stable discontinuous Galerkin method for the perfectly matched layer for elastodynamics in first order form. Numer. Math. 146, 2020. DOI: 10.1007/s00211-020-01160-w

LOH1

- $\frac{9\times8\times11}{27^3}$ or $\frac{14\times20\times19}{81^3}$ fp32 cells around point source and receiver
- about respectively 4% or 1% of domain in fp32
- All else computed in bf16

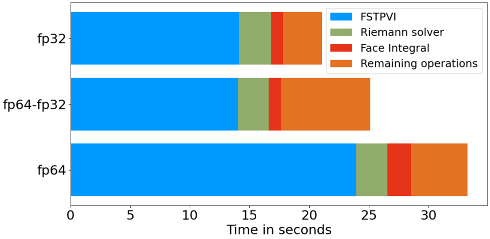




vx, cell size 0.45
20
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
reference
fp32
bf16
fp32-bf16
vx, cell size 0.15
20
0
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
time
reference
fp32
bf16
fp32-bf16

Impact on Performance

- CPU time of the three most time-intensive ADER-DG kernels, measured using Intel VTune
- 27^3 cells, with about 4% of domain in fp32, rest in fp64
- Variable-precision about 25% faster than fp64, as opposed to about 37% reduction for pure fp32



Conclusion

- By coupling two separate solvers in ExaHyPE 2, we can create a *variable precision* implementation of ADER-DG.
- This method works, and has no negative impacts on the solution
- However, artificial waves caused by the low-precision propagate into the high-precision region, affecting the solution there
- *Variable precision* is therefore most beneficial when a specific *area of interest* can be defined, and local features of the solution are the main sources of error.

Thank you for your attention