



March 18, 2026 | HPSF Conference 2026

Trilinos in deal.II

PRESENTED BY

Daniel Arndt

Oak Ridge National Laboratory



U.S. DEPARTMENT
of ENERGY

ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

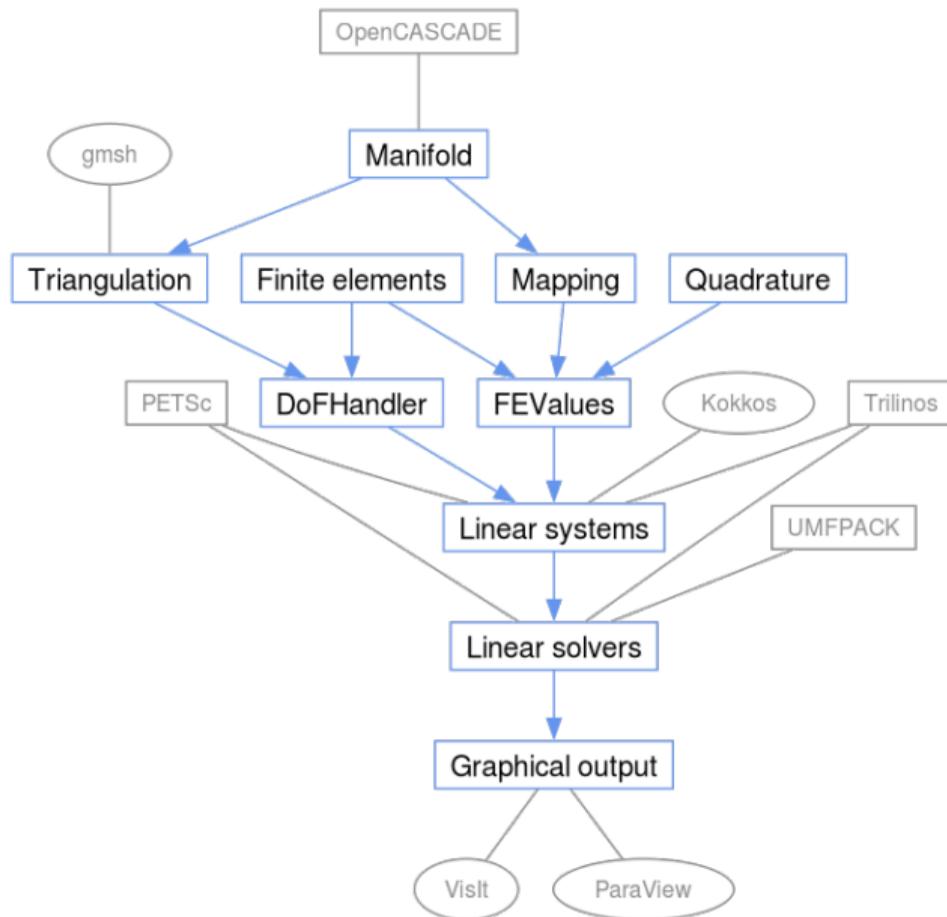


What is deal.II?



- One of the largest open-source C++ finite element libraries
- Focus on:
 - Large-scale PDE solvers
 - Adaptive mesh refinement
 - Modern C++ abstractions for FEM
- SIAM/ACM Prize in 2025
- Broad scope:
 - Commonly used for teaching FEM (100 tutorials)
 - Used in very large runs on supercomputers (10^{10} DoFs)
- Mostly used on CPU-only systems
- Community project, no direct funding for development

Parts of deal.II



deal.II's dependencies

- ADOL-C
- ArborX
- ARPACK
- assimp
- Boost
- CGAL
- Ginkgo
- Gmsh
- GSL
- HDF5
- Kokkos
- LAPACK
- Magic Enum
- METIS
- MPI
- MUMPS
- MUPARSER
- OpenASCADE
- p4est
- PETSc
- PSBLAS
- SCALAPACK
- SLEPC
- SUNDIALS
- SYMENGINE
- TASKFLOW
- TBB
- TRILINOS
- UMFPACK
- VTK
- ZLIB

Trilinos in deal.II (since 2008)

<https://dealii.org/current/external-libraries/trilinos.html>

- Amesos
- AztecOO
- Epetra
- EpetraExt (optional)
- Ifpack
- ML
- MueLu (optional)
- NOX (optional)
- ROL (optional)
- Sacado (optional)
- SEACAS (optional)
- Teuchos
- Tpetra (optional)
- Zoltan (optional)

Supported Trilinos versions: 13.2 - 16.2

Trilinos 17 doesn't support Epetra-based stack anymore

→ Need to rethink Trilinos interface

Relevance of supporting Tpetra

- Both Trilinos and PETSc provide MPI-enabled sparse matrix classes.
- Coarsest level in a geometric multigrid solver still needs algebraic preconditioner (which needs a matrix) even if the rest of the setup is matrix-free.
- Trilinos 17 was just released, not many users have demand for it
- Tpetra supports GPUs.
- Tpetra is still developed and maintained.
- xSDK/S4PST want to build software stacks with latest versions.

Issues with enabling Tpetra in deal.II

- GlobalOrdinal and dealii::types::signed_global_dof_index need to match (<https://github.com/dealii/dealii/issues/16528>). Trilinos 13.4 changed the default type. Only one global ordinal type can be instantiated in Tpetra.
- Issues enabling Epetra and Tpetra at the same time or generally using Trilinos with some compilers
<https://github.com/trilinos/Trilinos/pull/12757>.
- Issues with Tpetra_INST_INT_INT=ON:
<https://github.com/dealii/dealii/issues/18461>:
`/tmp/trilinos-release-16-2-0/packages/xpetra/src/Utils/Xpetra_ThyraUtils_decl.hpp:65
fatal error: Thyra_EpetraLinearOp.hpp: No such file or directory`

Transition from Epetra to Tpetra

Can we change from Epetra to Tpetra without changes in user codes?

Approach:

- Copy TrilinosWrappers classes' implementation to TpetraWrappers classes (done for the most part with differences in interface; minor annoyance: Why do we need RCP objects everywhere?)
- Make TrilinosWrappers classes aliases for TpetraWrapper classes with Number=double and MemorySpace=Host (<https://github.com/dealii/dealii/pull/18791>).
- Deprecate TpetraWrappers at some point and give TrilinosWrappers template arguments at some point?

Relevant Linear Algebra Types in deal.II ¹

| Epetra | Tpetra |
|--|--|
| <code>LA::EpetraWrappers::Vector</code> | (none) |
| <code>TrilinosWrappers::MPI::Vector</code> | <code>LA::TpetraWrappers::Vector</code> |
| <code>TrilinosWrappers::MPI::BlockVector</code> | <code>LA::TpetraWrappers::BlockVector</code> |
| <code>TrilinosWrappers::BlockSparseMatrix</code> | <code>LA::TpetraWrappers::BlockSparseMatrix</code> |
| <code>TrilinosWrappers::SparseMatrix</code> | <code>LA::TpetraWrappers::SparseMatrix</code> |
| <code>TrilinosWrappers::SparsityPattern</code> | <code>LA::TpetraWrappers::SparsityPattern</code> |

and various solver and preconditioner classes.

¹LinearAlgebra abbreviated as LA

Timeline

- 2019: Started with `TpetraWrappers::Vector` in 2019 slow progress since.
- 2023: Started working on `TpetraWrappers::SparsityPattern`
- 2024: Jan-Philip Thiele and Sebastian Kinnewig completed a first version of `TpetraWrappers::SparsityPattern` together with introducing `TpetraWrappers::SparseMatrix`
- 2025: Completed `TpetraWrappers::PreconditionAMGMueLu`

Assembly in deal.II

Assembly approaches in deal.II

- Manually write the assembly of the matrix
- `MatrixCreator`
- `MeshWorker/MeshLoop`
- `MatrixFree`
- `Portable::MatrixFree`

All apart from the last one only use CPUs

step-40 setup systems

```
void LaplaceProblem<dim>::setup_system() {
    [...]
    locally_relevant_solution.reinit(locally_owned_dofs,
                                     locally_relevant_dofs,
                                     mpi_communicator);

    system_rhs.reinit(locally_owned_dofs, mpi_communicator);
    DynamicSparsityPattern dsp(locally_relevant_dofs);
    DoFTools::make_sparsity_pattern(dof_handler, dsp, constraints, false);
    SparsityTools::distribute_sparsity_pattern(
        dsp, dof_handler.locally_owned_dofs(), mpi_communicator,
        locally_relevant_dofs);
    system_matrix.reinit(
        locally_owned_dofs, locally_owned_dofs, dsp, mpi_communicator);
}
```

step-40 setup system alternative reusing sparsity pattern

```
TrilinosWrappers::SparsityPattern dsp(  
    locally_owned_dofs, locally_owned_dofs, locally_relevant_dofs,  
    mpi_communicator);  
DoFTools::make_sparsity_pattern(dof_handler, dsp, constraints);  
dsp.compress();  
system_matrix.reinit(dsp);
```

- Epetra doesn't require specifying entries per row, Tpetra does
- Should the whole class be deprecated?
- Should the constructor without guess for number of entries be deprecated?
(<https://github.com/dealii/dealii/pull/19106>)
- Should we switch to using DynamicSparsityPattern under the hood?
(<https://github.com/dealii/dealii/pull/19146>)

step-40 assembly, ignore rhs

```
template <int dim>
void LaplaceProblem<dim>::assemble_system() {
    [...]
    for (const auto &cell : dof_handler.active_cell_iterators())
        if (cell->is_locally_owned()) {
            fe_values.reinit(cell);
            cell_matrix = 0.;
            for (unsigned int q_p = 0; q_p < n_q_ps; ++q_p) {
                for (unsigned int i = 0; i < dofs_per_cell; ++i) {
                    for (unsigned int j = 0; j < dofs_per_cell; ++j)
                        cell_matrix(i, j) += fe_v.shape_grad(i, q_p) *
                                                fe_v.shape_grad(j, q_p) * fe_v.JxW(q_p);
                }
            }
            constraints.distribute_local_to_global(cell_matrix, local_dof_indices, system_matrix)
        }
    system_matrix.compress(VectorOperation::add);
}
```

step-40 assembly, ignore rhs

Approach:

- Start assembly on any processor without coordination
- Call `compress` to communicate entries

Issues:

- `resumeFill` is documented to be a collective but isn't really.
- `fillComplete` is collective and can't be called multiple times in a row (while that works for `Epetra`).
- Recording the local status for now sufficient.

Question:

- All assembly is on the CPU. Is there a better assembly approach that is portable across deal.II matrix and vector types?

Future work, preconditioners and solvers

Preconditioners:

- MueLu doesn't have the same options as ML.
<https://github.com/dealii/dealii/pull/17331>
- Can't create preconditioners from const matrix objects
<https://github.com/dealii/dealii/pull/19317>.

Future work:

- FROSch Interface for deal.II. Why does that need to use the Xpetra interface?
- Replace Tpetra::CrsGraph with Tpetra::FECrsGraph

Performance - additive manufacturing example

- 10^4 DoFs
- 100 time steps
- CG solve
- Identity or SSOR preconditioner

| | Tpetra(HIP) | Tpetra(Serial) | Epetra |
|----------|-------------|----------------|--------|
| Identity | 94s | 136s | 103s |
| SSOR | 94s | 102s | 72s |

Summary

- The biggest push for the move from Epetra to Tpetra is Trilinos 17 at this point
- Allowing deal.II to use Trilinos 17, i.e., configuring Trilinos without Epetra but only Tpetra, seems feasible
- Most problems just appear in unit tests and not in user code and by wanting to provide a smooth transition for users.
- Some incompatibilities seem unavoidable.
- Future for TrilinosWrappers namespace still open.
- `ctest -R "trilinos"`
[...]
76% tests passed, 127 tests failed out of 531

Questions?

Acknowledgments

- This manuscript has been authored by UT-Battelle, LLC, under Contract No. DE-AC0500OR22725 with the U.S. Department of Energy.