Università della Svizzera italiana

Institute of Computational Science ICS

Università della Svizzera italiana

CARDIOCENTROTICINO

Center for Computational Medicine in Cardiology CCMC
Institute of Computational Science ICS
Faculty of Informatics

SCCER SoE
SWISS COMPETENCE CENTER for ENERGY RESEARCH
SUPPLY of ELECTRICITY

FNSNF
FONDS NATIONAL SUISSE
SCHWEIZERISCHER NATIONALFONDS
FONDO NAZIONALE SVIZZERO
SWISS NATIONAL SCIENCE FOUNDATION

Eurohack 2018
Utopia an EDSL for scientific computing

Team: Patrick Zulian, Alena Kopanicakova, Maria Nestola,

Mentors: Nur Fadel, Andreas Fink, Carter Edwards (NVIDIA), Christian Trott (SNL)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

NVIDIA

Sandia National Laboratories

October 10th, 2018, Lugano

Introduction

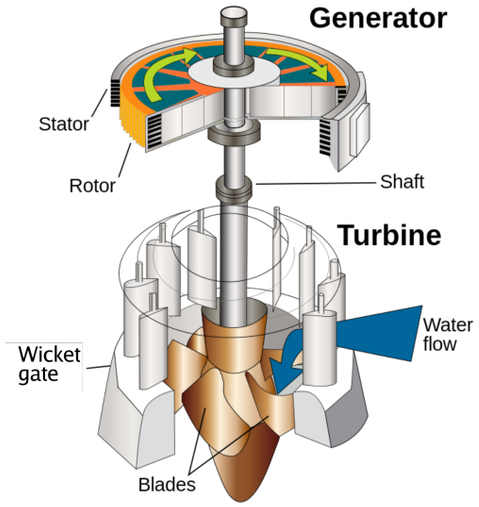
High-performance computing

Usability

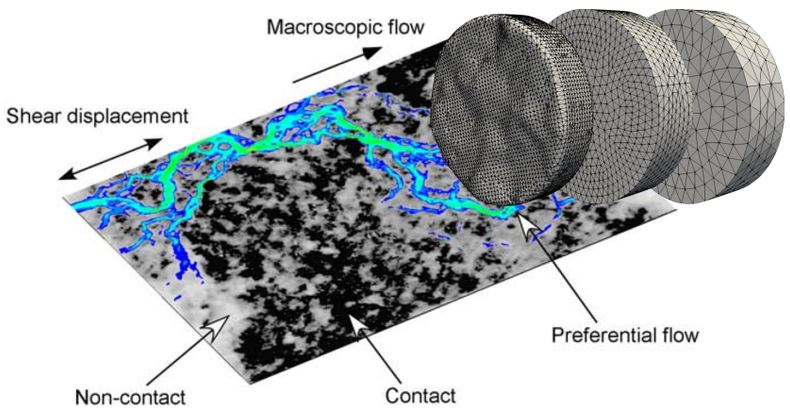
+

Flexibility!

Computational energy



Water turbines

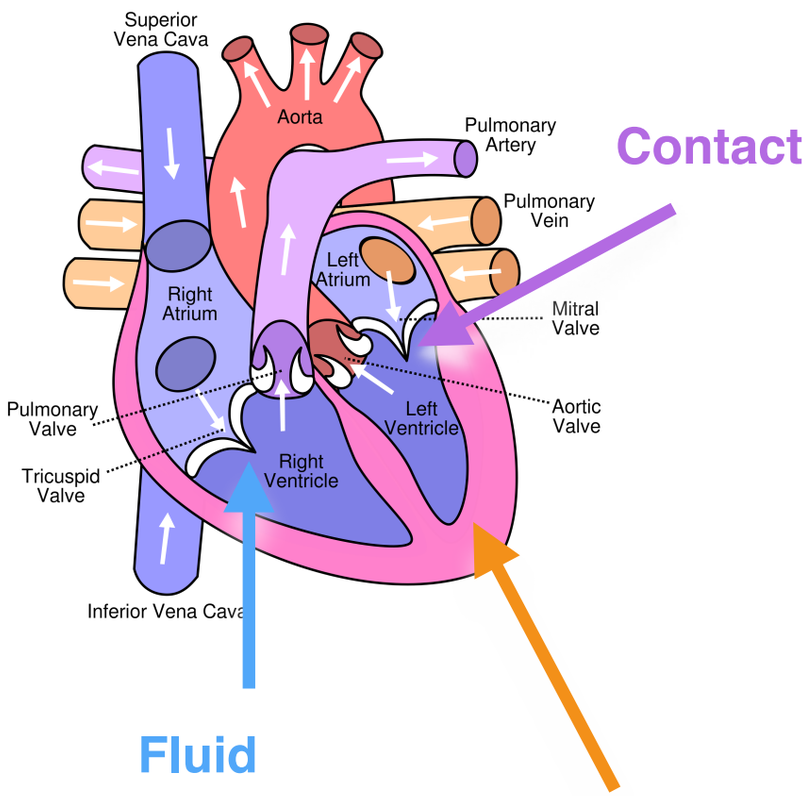


Computational Geophysics

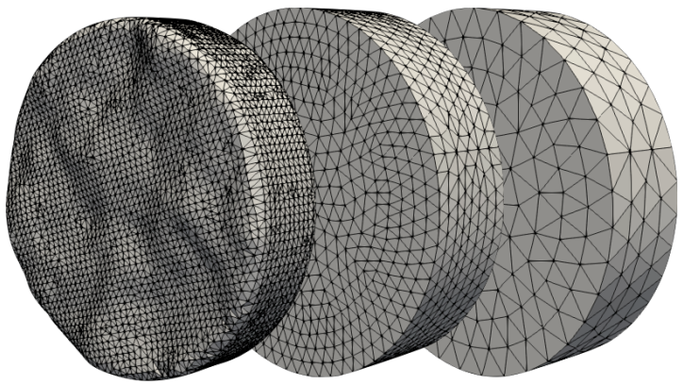
SCCER SoE
SWISS COMPETENCE CENTER for ENERGY RESEARCH
SUPPLY of ELECTRICITY

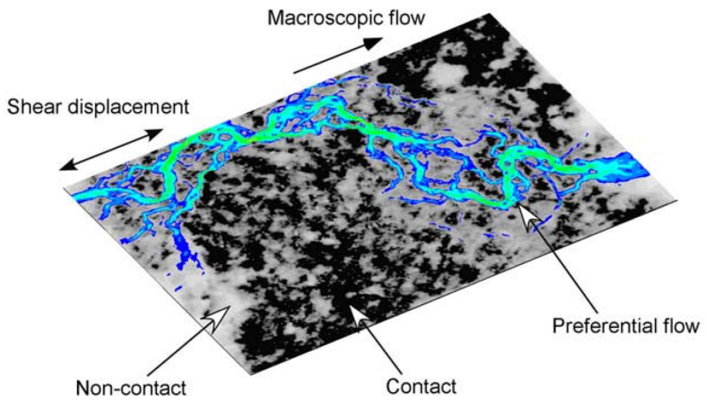
Other applications

Simulation of the heart

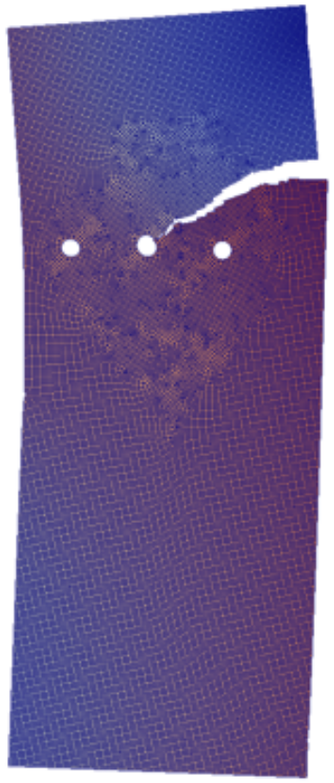


Multilevel solution methods

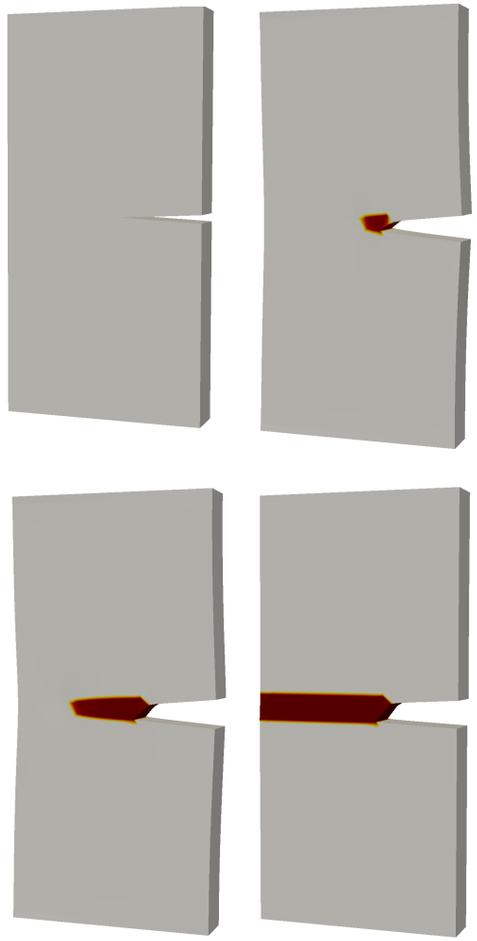




Multigrid solver for **computational geology** (C. Von Planta)



Multigrid solver for **crack simulations** with XFEM[1] (H. Kothari)

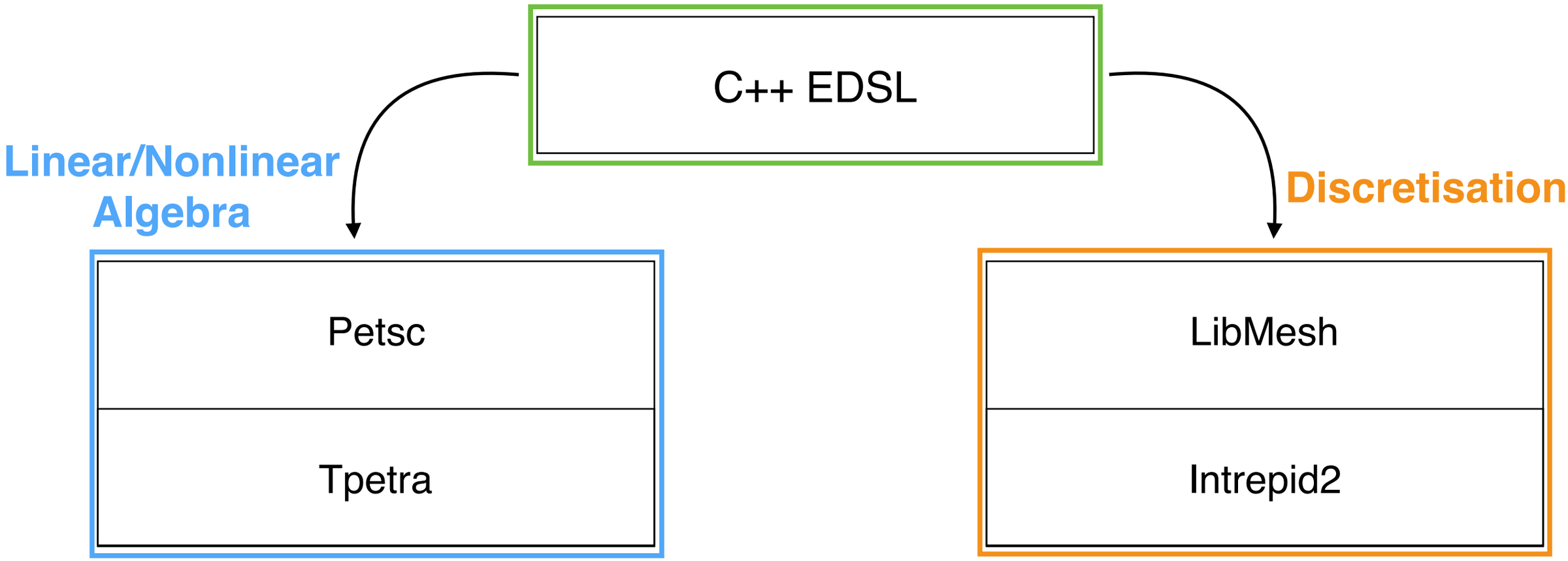


Multigrid solver for **Phase-field method**[2] (A. Kopaničáková)

Literature: **1** Z. Goangseup, and T. Belytschko, 2003.
2 C. Miehe, F. Welschinger, and M. Hofacker, 2010.

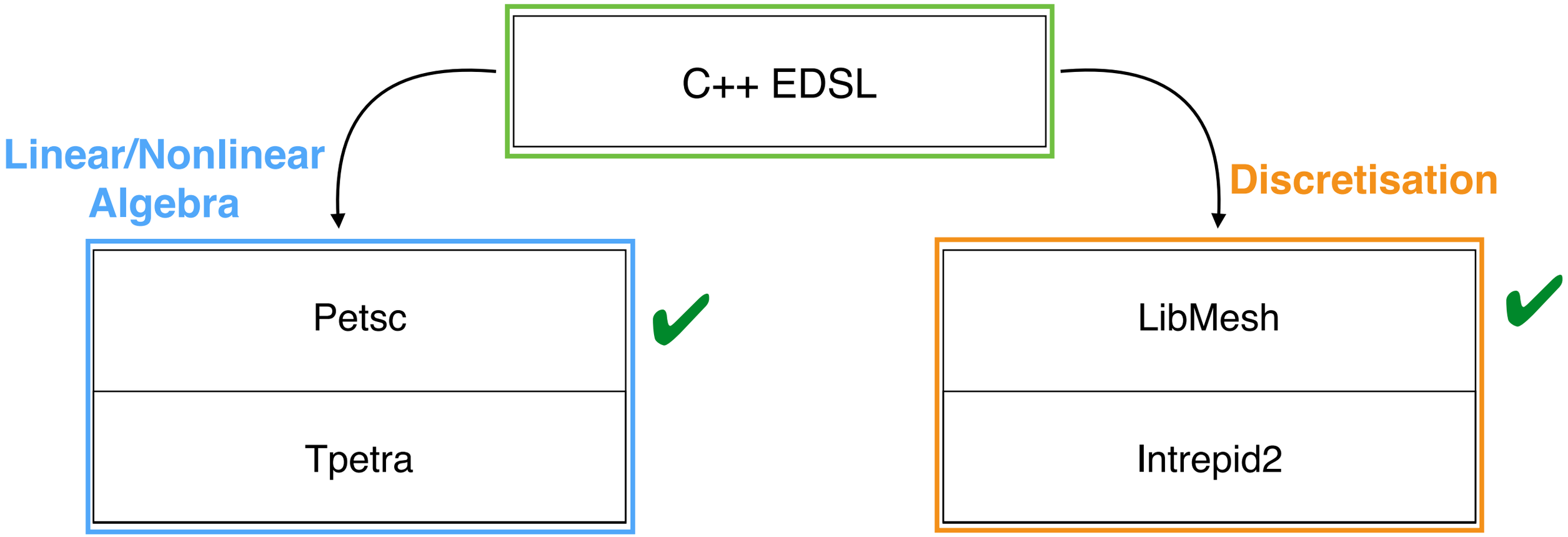
Overview

Architecture



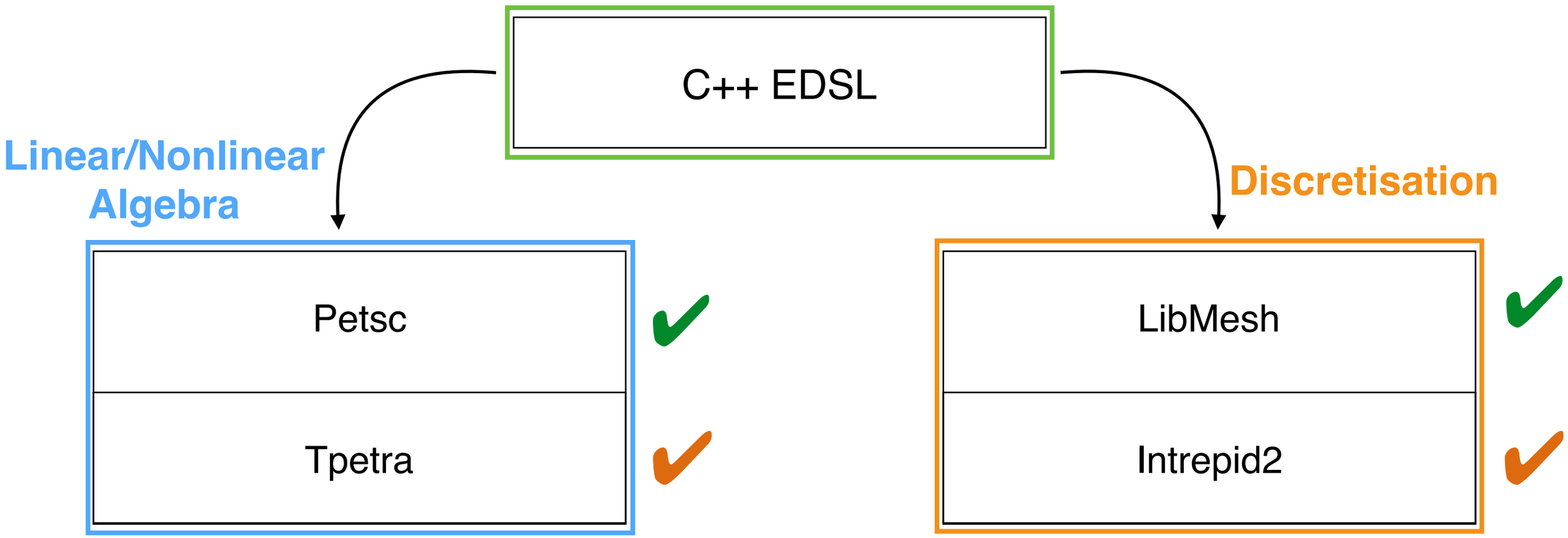
Overview

Architecture



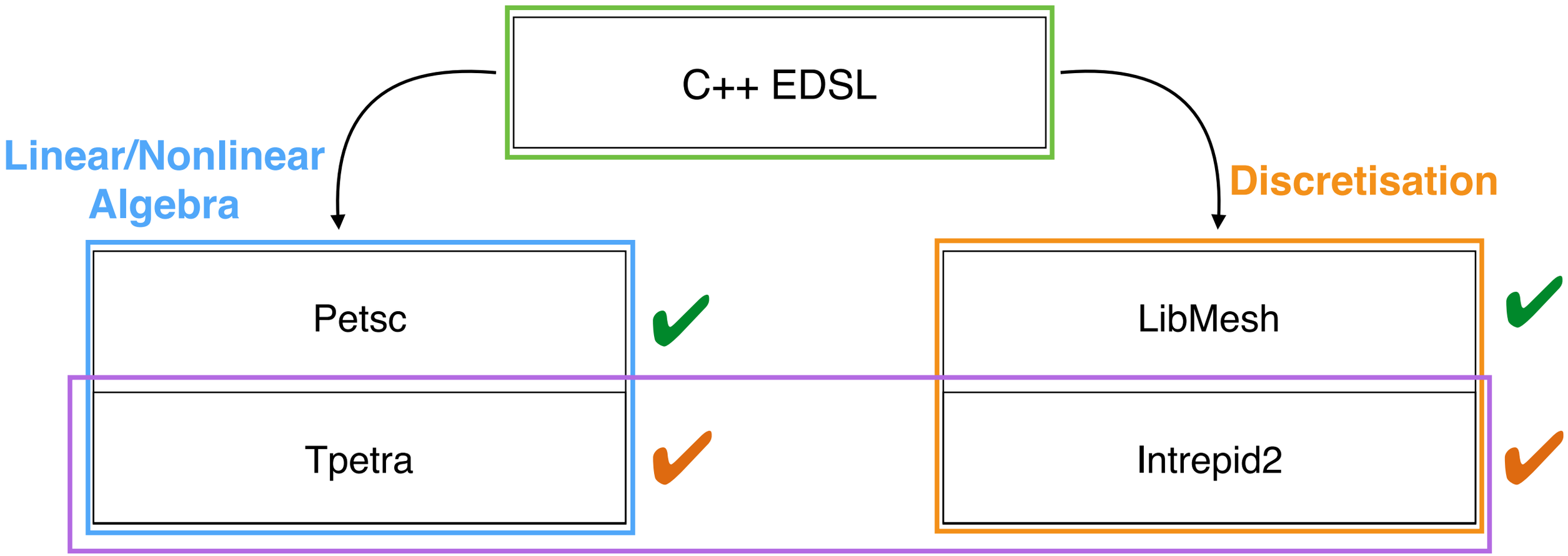
Overview

Architecture



Overview

Architecture



Goals partially reached for the hackathon

Starting point

CPU

- Advanced implementation using **Petsc**
- Initial implementation using **Tpetra** (very buggy and imcomplete)
- Compiling only for CPU on daint
- Finite element assembly using **LibMesh**

GPU

- **PetscCuda** is incomplete with respect to our requirements, (inefficient) workarounds in place just to make the code not crash

Porting to GPU

Porting all the algebra routines to GPU

- Make **Tpetra** based backend work compile and work on GPU
- Add missing functionalities using **Kokkos**

Proof of concept for finite element assembly

- Create prototype using **Intrepid2** (trilinos package)

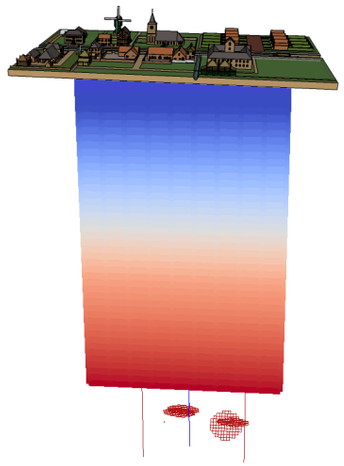
```
auto u  = trial(V);
auto v  = test(V);
auto b_form = inner(grad(u), grad(v)) * dX;
utopia::assemble(b_form, H);
utopia::assemble(inner(coeff(1.), v) * dX, rhs);
```

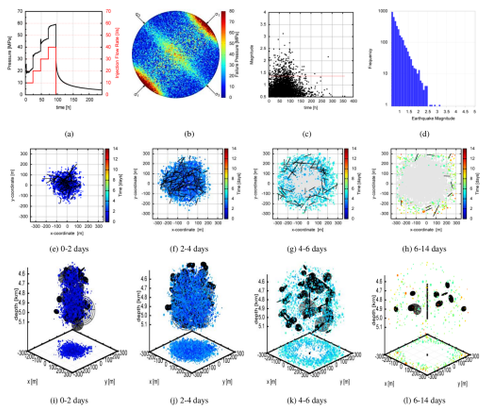
Output (Trilinos backend integration)

- **Goal**: Implement all *TODOs* for the **Tpetra**-backend in utopia
- **Achieved**
  - Most of our code is now running on GPUs (not all of it with hybrid *GPU-MPI*)
  - Large number of tests
  - Fixed most of the critical bugs
  - Added several missing functionalities using **Kokkos**
- **Future steps**: fix remaining bugs and optimise performance

Output (FASTER)

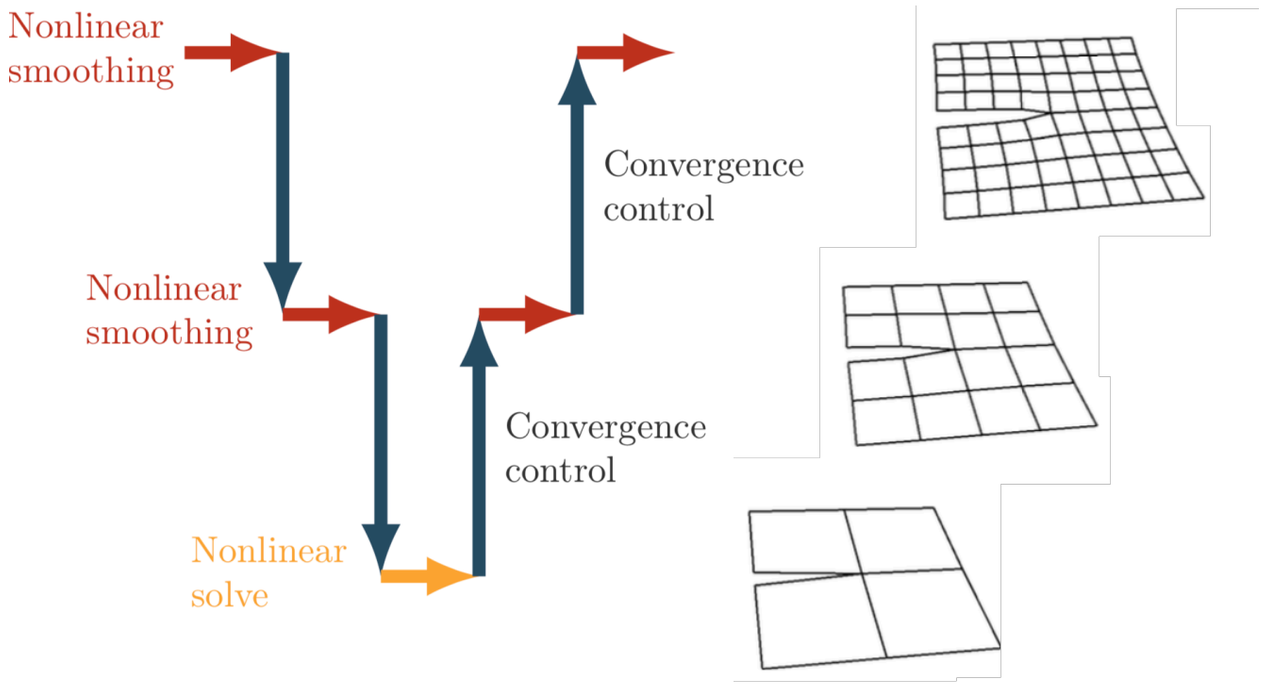
- **Goal**: Use utopia for FASTER instead of Tpetra and optimise it for the least overhead possible
- **Achieved**
  - code simplified
  - **petsc** compatible (different solvers, slower the ones we tested)
  - **Tpetra** *20% slower* in the assembly because we removed backend-specific optimisations
- **Future steps**: add front end abstractions for exploiting backend specific optimisations





Output (RMTR)

- **Goal**: Recursive Multilevel Trust Region aka RMTR on GPU
- **Achieved**
  - Single node GPU port
  - *Serial node* time(**petsc**) = time(**tpetra**)
  - *Serial node* 12 x time(**tpetra-gpu**) = time(**tpetra**)
- **Future steps**: Hybrid GPU-MPI, optimise



Feedback

- **GPU paradigms**: first impact with **Kokkos**
  - Intuitive
  - Easy to integrate

- **Compilers**
  - long compilation times
  - *nvcc* not providing useful error messages

- **Tools**
  - *nvprof* and kernel names from templates hard to read (output simplification/beautification options might help)

- **General**
  - Very well organised and mentors were very helpful

Thank you!!

Università della Svizzera italiana
Institute of Computational Science ICS

Publicly available codes

Utopia	bitbucket.org/zulianp/utopia
ParMOONoLith	bitbucket.org/zulianp/par_moonolith
MFEM-MOONoLith	github.com/mfem/mfem/tree/moonolith-dev

Lawrence Livermore National Laboratory