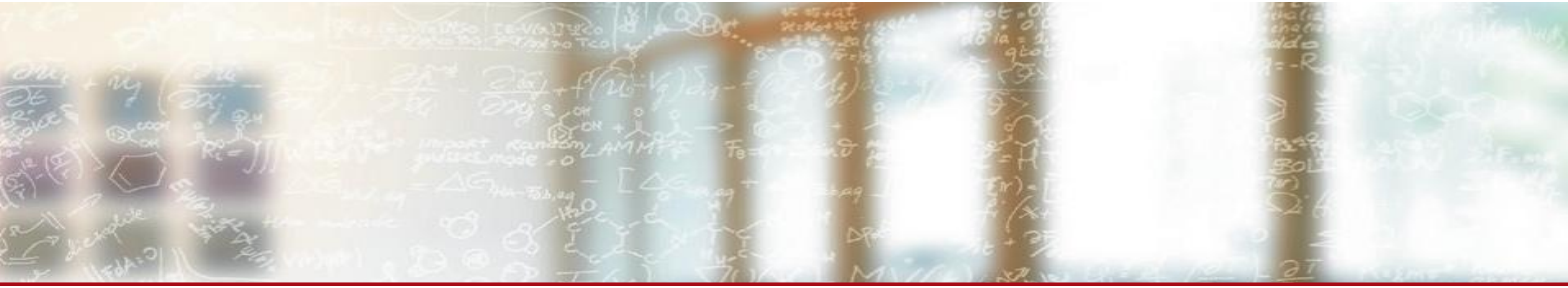
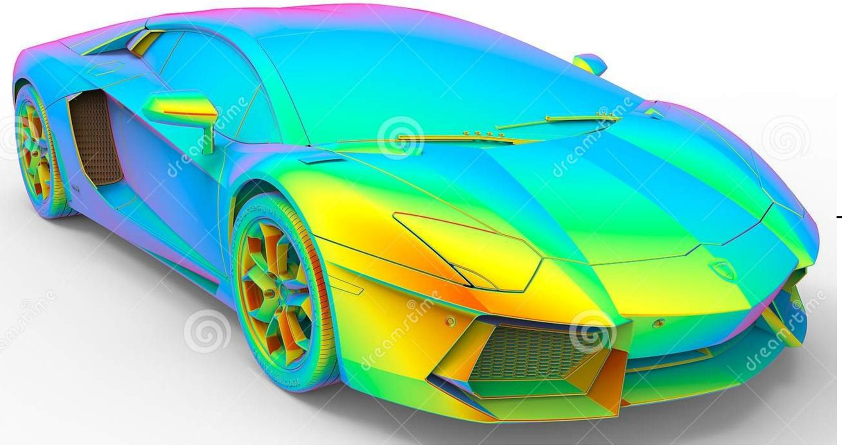


Utopia: a Performance-Portable C++ Library for Non-Linear Algebra

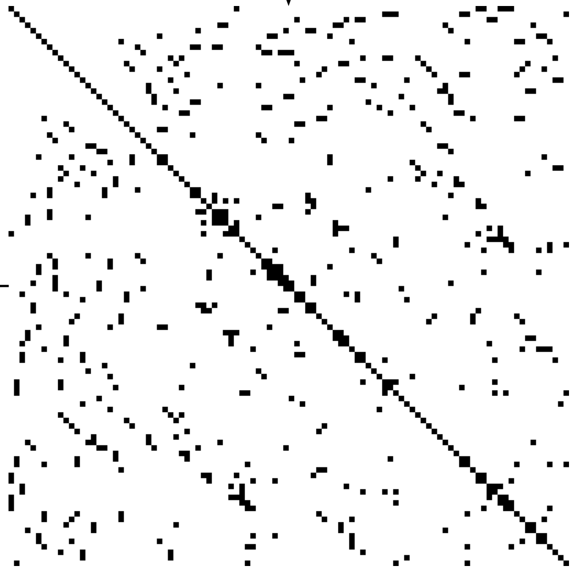
Nur A. Fadel, Patrick Zulian, Alena Kopaničáková, Andreas Fink, Daniel Ganellari, Rolf Krause

12th June, 2020

Motivation



$$Ax = b$$



Parallel linear algebra library

PETSc

TRILINOS

Monolithic code

Current situation
PETSc and Trilinos

- Easy to use
- Highly Scalable
- Rich set of parallel nonlinear solvers
- Mostly MPI based
- GPU version is far from being complete and mature
- C based code

- Similar goals as petsc
- Based on several packages
- Major rewrite ongoing
- Now based on Kokkos
- Focused on performance portability (extensive GPU support)
- C++11 based code

Kokkos Core Abstractions

Kokkos

- **Data Structures**
  - Memory Spaces (“Where”)
    - HBM, DDR, Non-Volatile, Scratch
  - Memory Layouts
    - Row/Column-Major, Tiled, Strided
  - Memory Traits (“How”)
    - Streaming, Atomic, Restrict
- **Parallel Execution**
  - Execution Spaces (“Where”)
    - CPU, GPU, Executor Mechanism
  - Execution Patterns
    - parallel_for/reduce/scan, task-spawn
  - Execution Policies (“How”)
    - Range, Team, Task-Graph

CSCS : Swiss National Supercomputer Centre

Early adopter of GPU technology for HPC

Flagship : Piz Daint : one of the most powerful supercomputers in the world
5700 GPU accelerated nodes, Aries network, Cray XC50
1800 Dual socket intel nodes, XC40

CSCS is accessible for scientists worldwide via open peer reviewed calls:
Excellent science and efficient use of the GPU accelerated infrastructure
- ‘small’ proposals (<1M node hours) → via national call
- ‘large’ proposals (>=1M node hours) → via PRACE

UTOPIA

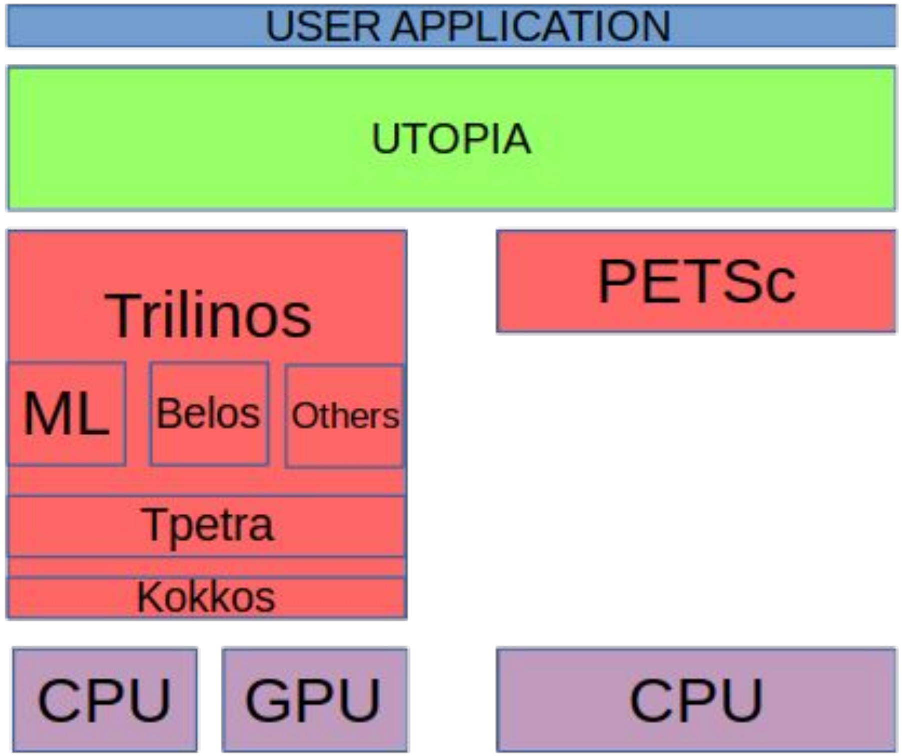
Università della Svizzera italiana
Institute of Computational Science ICS

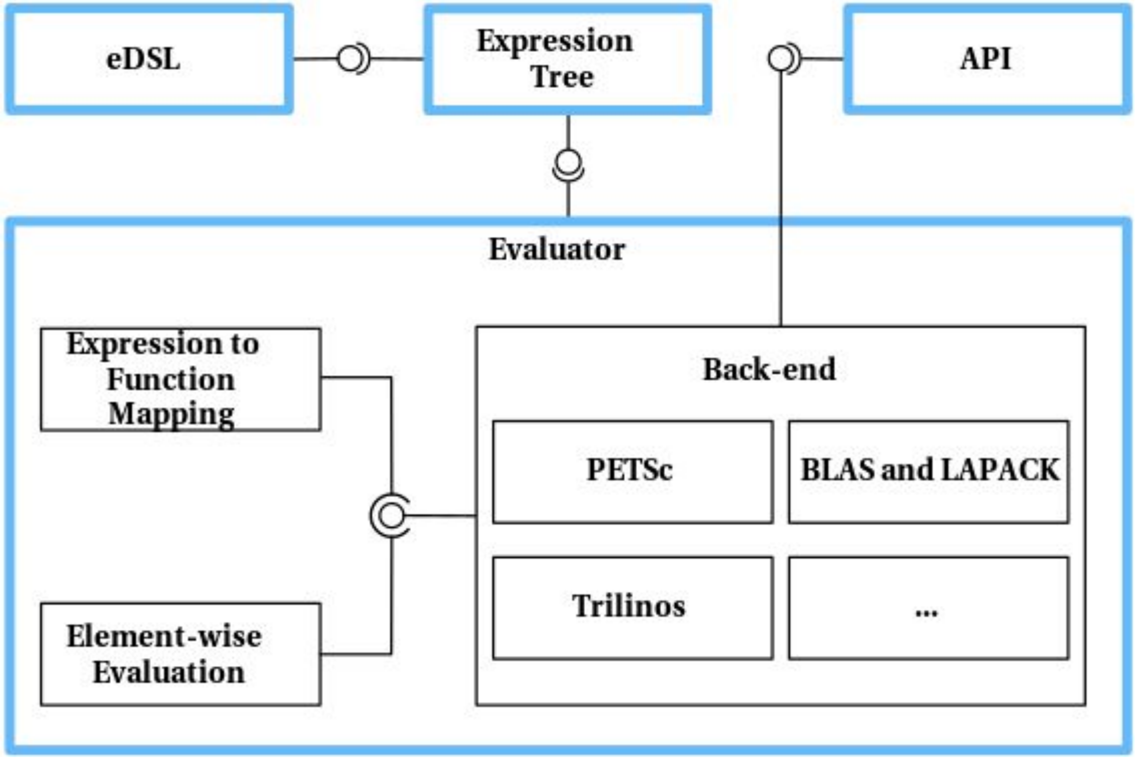
- C++11 code
- Open source – license BSD 3 Clause
  https://bitbucket.org/zulianp/utopia/src/master/
- Has multiple backends
  - PETSc
  - Trilinos (Kokkos - www.kokkos.org)
- Design based on self contained backends

- Defines a eDSL (embedded domain specific language):
- Freedom to build on top of the eDSL
- Freedom to add new backends
- Currently the DSL covers linear and non-linear algebra

UTOPIA

Library for parallel sparse linear and non linear algebra

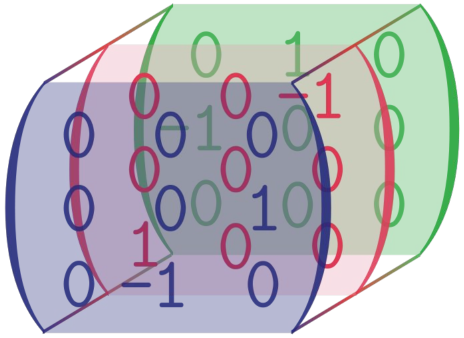




eDSL - Language for expressing mathematics

AIM: separate idea from actual implementation

$\epsilon_{ijk} =$




```
template<class Derived, int Order>
class Tensor
```

Interaction between Tensors
is defined through
the Utopia primitives

generates an expression
tree

Lazy evaluation
evaluated when assigned to
another tensor obj

Example of code

```
// types
Matrix A, B, C;
// initialize matrices here …

Scalar alpha, beta;
auto expr = alpha * A * B + beta + C;
// the expression is evaluated here
Matrix value = expr;
```

Example of Power method iteration

```
Matrix A;
Vector b_k, b_k1;
// initialize A and b_k here …
for (int i=0, i< num_iterations, i++){
    // calculate the matrix-vector product Ab
    b_k1 = A * b_k;
    //calculate the norm
    Scalar b_k1_norm = norm2(b_k1);
    //re normalize the vector
    b_k = b_k1 / b_k1_norm;
    if (b_k1_norm <= tol) break;
}
```

UTOPIA - benefits

- "Write once, use everywhere" model portability
  - run on CPU and GPU
  - run with either PETSc or Trilinos backend

HW independent

SW independent

- Performance portability:
  - near native performance version thanks to kokkos[1]

- Extensibility (advanced usage):
  - possible to create new solvers based on utopia interface

[1] Deakin and oth., GPU-STREAM v2.0 Benchmarking the achievable memory bandwidth of many-core processors across diverse parallel programming models, P3MA, ISC’16

UTOPIA - Features

Provided by back-ends:

- Direct solvers
- Iterative solvers
- Direct preconditioners
- Multigrid preconditioners
- Nonlinear solvers

Built on top of back ends:

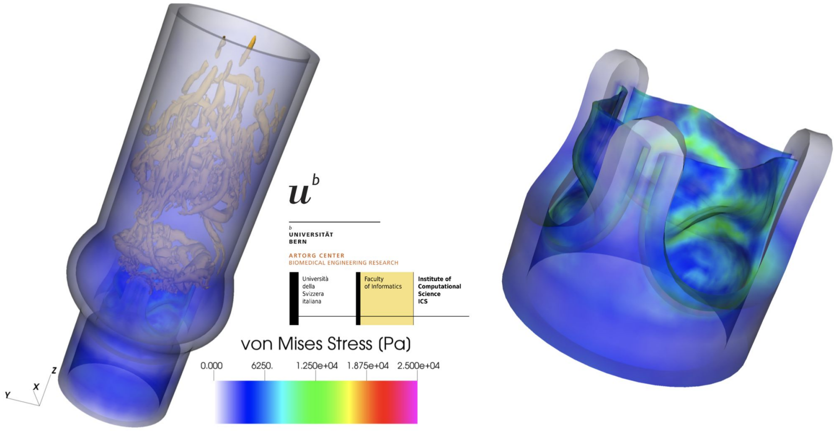
- Possibility to mix solver of different back-ends (*)
- Matrix free methods
- Multilevel nonlinear solvers
- Hessian free solvers

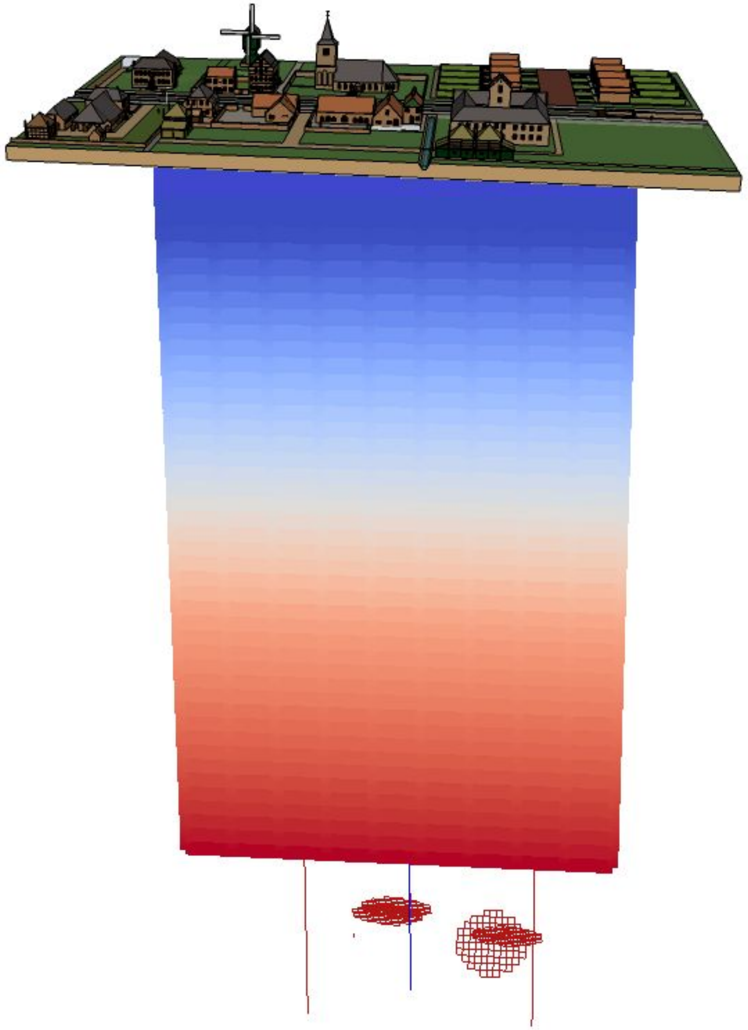
Applications

Applications

Current PASC applications using UTOPIA:

- AV-FLOW[1]
- FASTER
- StagBL





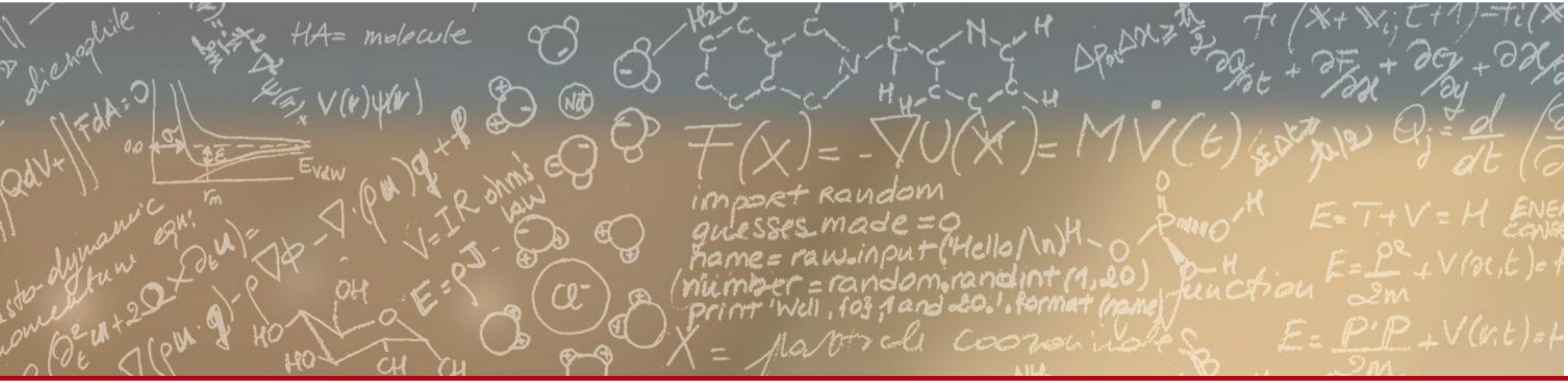
[1]Nestola and oth., 398, Journal of Computational Physics, 2019

Summary

- C++11 Library for sparse (non) linear algebra
- MPI based and Kokkos based (Performant portable)
- Open source
- Support for both PETSc and Trilinos
- Flexible

Future Steps:

- General mesh based discretization interface
- Generic interface to PETSc DM objects - add support of MARS (new mesh handler)
- Add other language bindings
- Investigate multiprecision algorithms
- Parallel in time algorithms



Thank you for your attention.