Integration with External PDE Solvers

René Fritze

October 6, 2021

Overview

Available Bindings

- FEniCS
- NGSolve
- deal.II
- Dune XT

Thermalblock Demo

Easily switch the same reduction code from one discretization to another.
Here: FEniCS+ NGSolve+ pyMOR

Generic Algorithms via Abstraction

- VectorSpaceInterface
- VectorArrayInterface
- OperatorInterface

Bindings Setup

- Vector Implementation derived from
  pymor.vectorarrays.list.CopyOnWriteVector
- VectorSpace Implementation derived from
  pymor.vectorarrays.list.ListVectorSpace
- Operator Implementation derived from
  pymor.operators.basic.OperatorBase

Requirements: VectorSpace

- zero_vector()
- make_vector(obj)

Requirements: Vector

- copy()
- _scal(alpha), _axpy(alpha, x), dot(other)
- l1_norm(), l2_norm(), l2_norm2()
- dofs(dof_indices)
- amax()
- to_numpy(ensure_copy=False)

Requirements: Operator

- ___init___(self, op)
- apply(self, U, mu=None)

Bindings Types

- Pure Python NGSolve, FEniCS, pyMOR
- Python bindings for native (C/C ++) data + pyMOR Wrappers deal.II, minimal C ++ Demo
- Python wrapper for on-disk serialized data
- Python wrapper for remote discretization server

Tutorial