Composable Streaming Interfaces in PETSc for Extreme-Scale Geometric Multigrid

**Motivation**
- Development of parallel scalable implicit solvers and preconditioners
- Support for various models (PDEs, networks) and parallel architectures
- Careful software design with good maintainability in the future

**Challenges**
- Parallelization of multi-level methods on adaptively refined meshes
- Memory handling for high-order finite elements and matrix-free operators
- Heterogeneous architectures demand complex memory transfers: MPI communication, GPU offloading, etc.

**Hybrid Spectral–Geometric–Algebraic Multigrid**

- High-order finite element discretization
  - High-order finite elements, modal and modal shape functions
  - Adaptive mesh refinement (AMR) to resolve local small features
  - Hexahedral elements allow exploiting the tensor product structure of shape functions to greatly reduce the number of floating point operations
  - Fast, matrix-free application of stiffness and mass matrices reduce memory consumption significantly

- Core-thinning to avoid excessive communication in multigrid cycle
  - Coarse grid solver: AMG invoked on small core counts & small communicator

- High-order projection onto coarser levels: restriction & interpolation are adjoints of each other in $L^2$-sense

**Software design of multigrid**

- Separate hierarchy of meshes and operators
- High-level perspective on design patterns for model–solver interactions

**Motivation**
- Monolithic: model knows and controls solver, e.g. mesh refinement in traditional implementations
- Flexible: model is agnostic of solver, e.g. solver accesses interfaces of model (approach taken by PETSc)

**Design Patterns for DM Objects in PETSc**

- A high-level view on design patterns for model–solver interactions.
- Monolithic: model knows and controls solver
- Flexible: model is agnostic of solver, e.g. solver accesses interfaces of model

**Propose: Model–Controller–Solver Design Patterns**

- The new design is based on the five patterns introduced before and has the goal to realize geometric multigrid implementations (like the MG shown before) and accommodate different parallelism models (distributed & shared memory, MPI+OpenMP or MPI+GPU) via unifying interfaces.

- New “Model” and “Memory” interfaces: Explicitly target separate designs for memory and a structure given to this memory via a model. We introduce the concept of streaming that enables the composition of objects that implement these interfaces. Examples of a stream of models are: (i) export MG levels, (ii) physics decompositions, and (iii) parallel distributions of DOFs.

**Scientific Achievements**
- Implicit solvers for complex PDEs with algorithmic scalability to billions of DOFs and parallel scalability to millions of processor cores
- Generalization of extreme-scale MG to support a wide range of problems
- Software design to efficiently implement multi-level algorithms and their MPI-GPU parallelization

**Significance & Impact**
- Software components that encapsulate mathematical, computational science, and application-specific methods and additionally provide clear interfaces, allow for a more productive collaboration of interdisciplinary teams
- Accessibility of extreme-scale multi-level solvers to a large community of domain specialists, able to leverage state-of-the-art numerical methods

**Argonne Guidelines 022806 ©Argonne National Laboratory. All rights reserved.**

Through consistent use, this logo will accrue the greatest awareness which, in turn, will benefit us most.