Modern Algorithms Through Libraries

Jed Brown jedbrown@mcs.anl.gov and Matt Knepley


SCALING PEOPLE

It is unsustainable for every project to find individuals that understand the best available methods throughout the computational science cycle. Libraries reuse expertise, freeing users to understand only high-level properties and focus on "their business".

Rapid Algorithmic Experimentation

- matrix formats: -mat_type [bstrm, cusp, opencl]
- preconditioners: -pc_type [gmres, ifbicgs]
- linear solvers: -ksp_type [pagem, ifbicgs]
- nonlinear solvers: -nss_type [ngmres, qn, ncg, fas]
- time integration methods: -ts_type [arkimex, rosw]
- order and compatibility of finite element spaces:
  \[ W = \text{FiniteElement}("BDN", \text{triangle}, 1) \]  
  \* FiniteElement("DG", \text{triangle}, 0)
- adaptive refinement criteria

CASE STUDY: Stokes/KKT preconditioners for Geodynamics

The saddle-point matrix is a canonical form for handling constraints:
- Incompressibility
- Contact
- Multi-constituent phase-field models
- Optimal control
- PDE constrained optimization

Many algorithms to solve Stokes problems can be classified as monolithic or split, but they are rarely compared. The best choice depends on the underlying physics, other coupled processes, and the machine architecture. We use run-time options to compare
- Block triangular preconditioners with different Schur complement approximations (Elman, Silvester, Wathen, ...)
- Schur complement reduction (preconditioned accelerated Uzawa-type)
- Algebraic or geometric multigrid in above
- Monolithic multigrid with split smoothers, block smoothers (Vanka-type), or distributive smoothers

Development is incremental. Try the method out today, then write specialized low-memory/matrix-free version if the algorithm performs well for your problem.

Model coupling

- Most simulation codes start as models of a single physical process.
- Coupling requires either (a) different control flow or (b) great pain.
- Libraries encourage more modular control flow and provide common abstractions for coupling.
- (Good) libraries encourage better software design.

Evaluating Analysis-Friendly Applications and Libraries

State isolation. A model component represents a physical process, not a snapshot of state that the process operates on. Isolating state improves composability and eases coupling.

Humility. Side-effects and implicit dependencies significantly increase coupling effort and maintenance costs.

Extensibility. "Library functionality" can be implemented outside of library. Communicators. Provide explicit scope for distributed objects. Error handling. Complete clean-up may be impossible, but a consistent and debuggable error handling mechanism is invaluable.

Extensibility and Plugins: Not Your Grandpa’s Libraries

- Modern libraries are toolboxes more than black boxes.
- The interface is more important than a specific implementation.
- An application with unique structure can extend library features by providing their "special sauce".

Examples

- High-order finite element computation defines new matrix format that applies operator in un assembled form, reduces memory bandwidth requirement by 30x compared to assembled operator and works with all Krylov methods and nonlinear solvers.
- Vendor provides matrix format and preconditioner plugins as compiled shared object. An application that was compiled with no knowledge of the plug load it as a first-class implementation, used in a sub-system of a multi-physics solve.
- Application with critical behavior provides adaptive controller for time integration, available to all families of ODE/DAE integrators.
- Application uses analytically reduced nonlinear 2D coarse level in multigrid algorithm applied to 3D problem.

REFERENCES

> Composable linear solvers for multiphysics.
> Matthew G. Knepley and Andy R. Terrel.
> Preliminary implementation of PETSc using GPUs.

This work was supported by the Department of Energy Office of Advanced Scientific Computing Research.