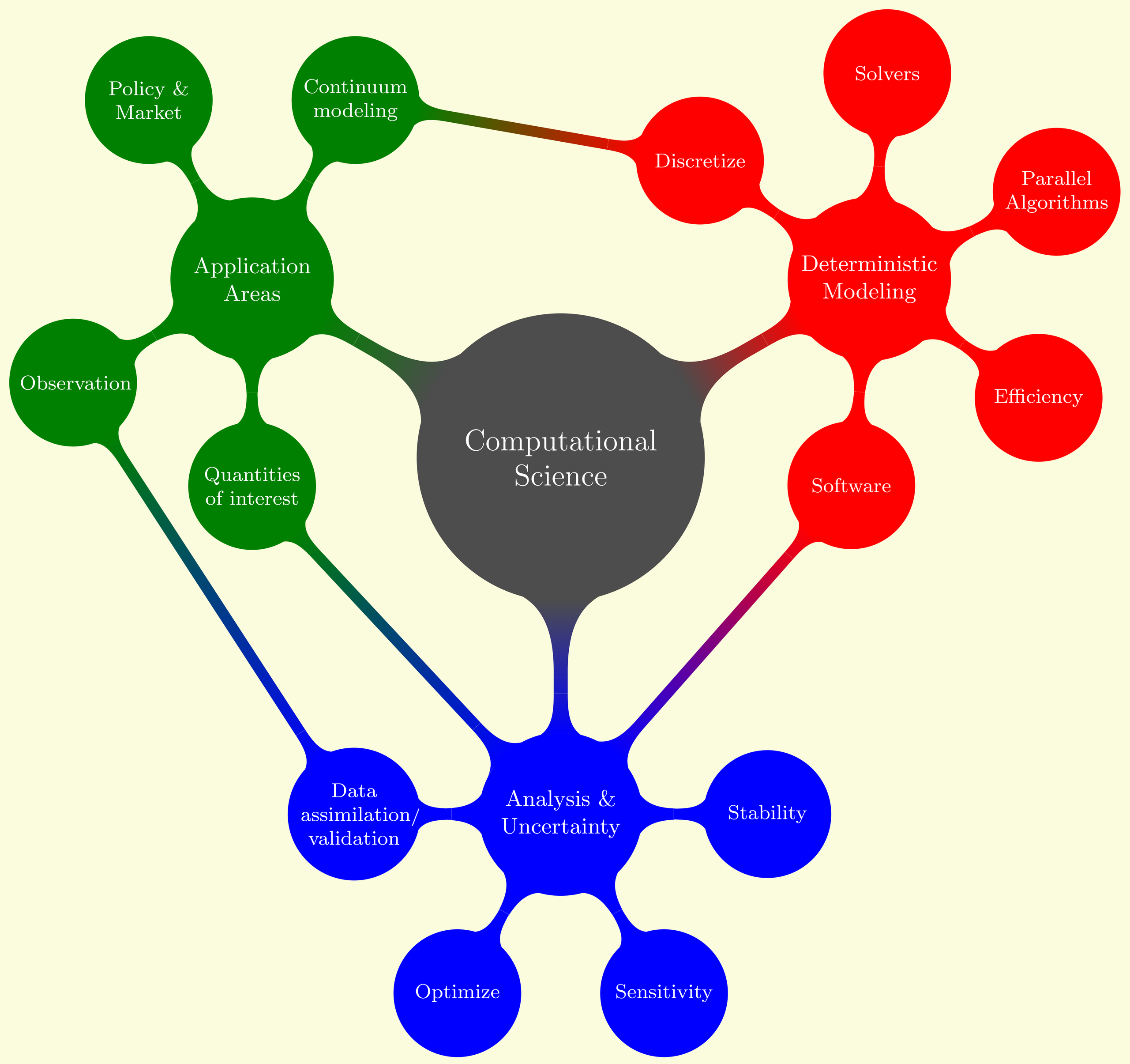
Modern Algorithms Through Libraries

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

Download this poster from http://59A2.org/files/201210-EarthCube.pdf

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”.



The Quest for Performance Portability

- Longer vector registers versus SIMT?
- Prefetched cache-based or GPU-style concurrent threads?
- How non-uniform will memory access be?
- Which operations make sense to run on an accelerator?
- Which coarse-grid distribution is network-efficient on my architecture?

Why Libraries?

- Outsource risk investing in algorithms and architectures
- Benefit from algorithmic innovation
- Defer data format and accelerator choices, often to run-time
- Specialized optimizations and vendor tuning

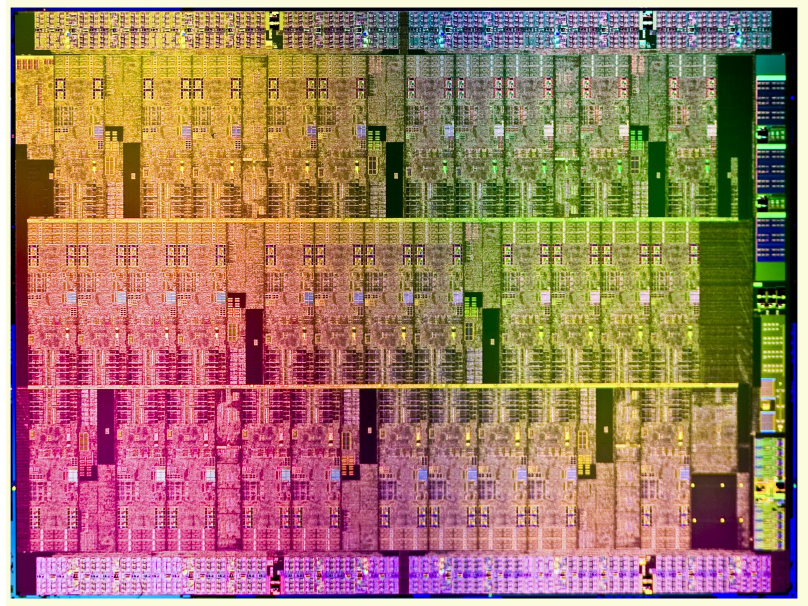


Figure: Intel MIC die

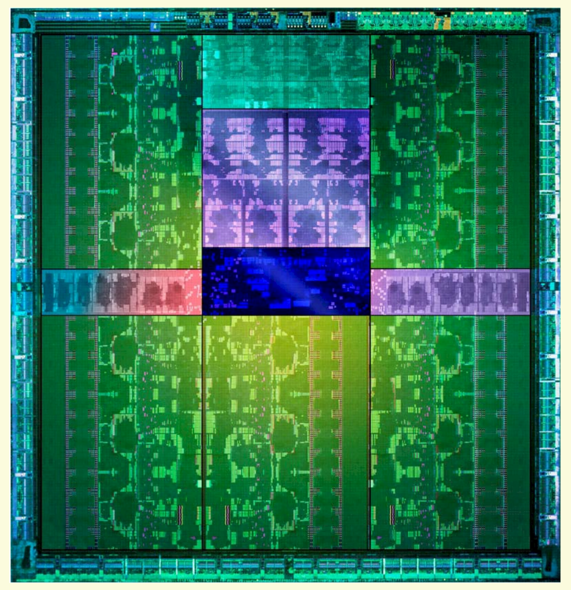


Figure: NVidia Kepler GK110 die

Rapid Algorithmic Experimentation

- matrix formats: `-mat_type [bstrm,cusp,opencl]`
- preconditioners: `-pc_type [gamg,fieldsplit,asm]`
- linear solvers: `-ksp_type [pgmres,ifbicgs]`
- nonlinear solvers: `-snes_type [ngmres,qn,ncg,fas]`
- time integration methods: `-ts_type [arkimex,rosw]`
- order and compatibility of finite element spaces:

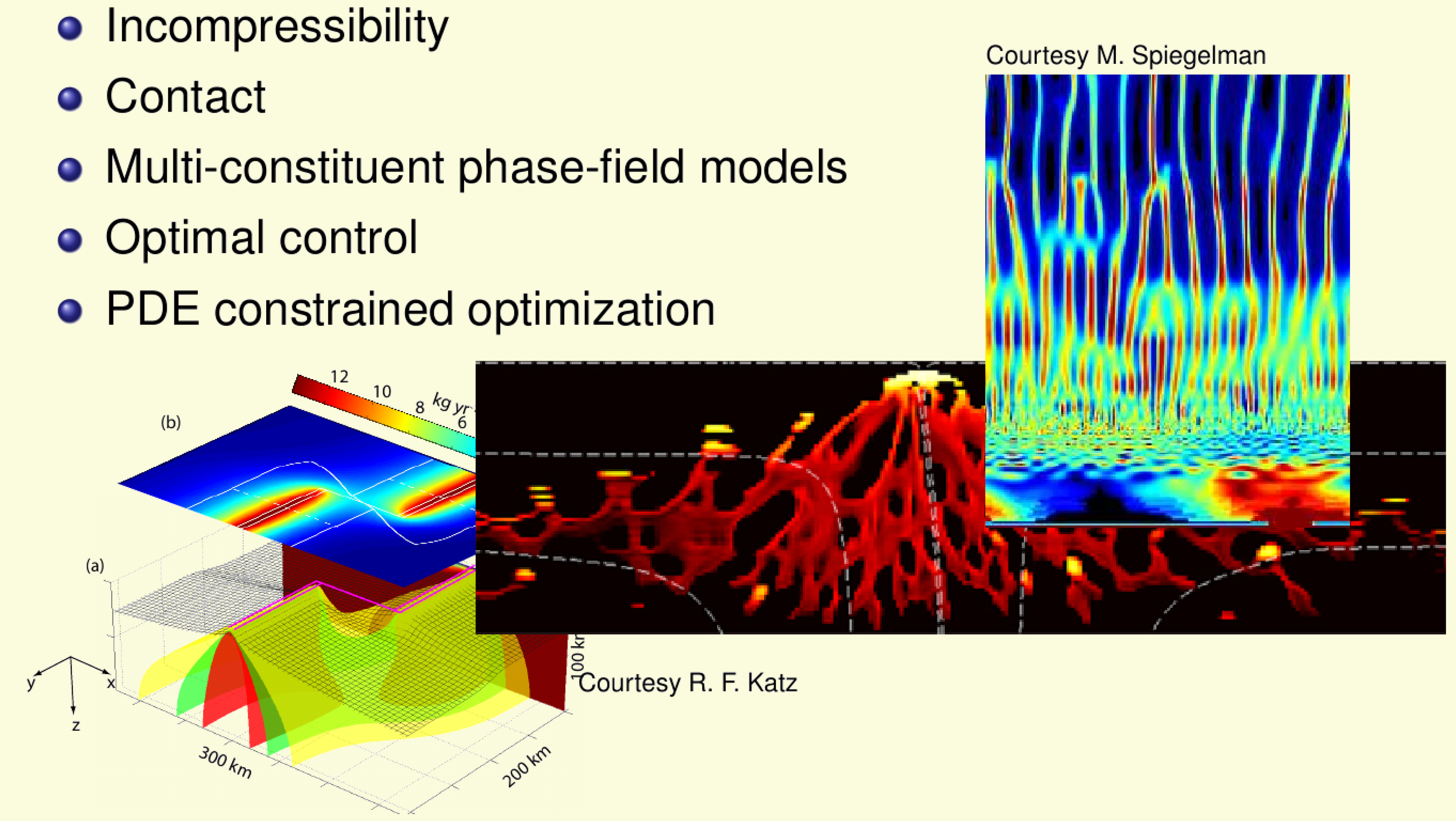
```
W = FiniteElement("BDM", triangle, 1)
    * FiniteElement("DG", triangle, 0)
```

- 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.

Reproducibility and Community

- Libraries provide a common language for precisely discussing methods.
- Reliably reproduce your algorithm in their application.
- Suite of diagnostics for understanding performance.

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 unassembled 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 plugin loads 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

- J. Brown, M. G. Knepley, D. A. May, L. C. McInnes, and B. F. Smith. Composable linear solvers for multiphysics. In *Proceeedings of the 11th International Symposium on Parallel and Distributed Computing (ISPDC 2012)*, 2012.
- Peter Brune, Mathew Knepley, Barry Smith, and Xuemin Tu. Composing scalable nonlinear solvers. Preprint ANL/MCS-P2010-0112, Argonne National Laboratory, 2012.
- William D. Gropp. Exploiting existing software in libraries: Successes, failures, and reasons why. In *Proceedings of the SIAM Workshop on Object Oriented Methods for Inter-operable Scientific and Engineering Computing*, pages 21–29. SIAM, 1999.
- David E. Keyes, Lois Curfman McInnes, Carol Woodward, et al. Multiphysics Simulations: Challenges and Opportunities. Technical Report ANL/MCS-TM-321, Revision 1.1, Argonne National Laboratory, Oct 2012. To appear as a special issue of the *International Journal of High Performance Computing Applications*.
- Matthew G. Knepley and Andy R. Terrel. Finite element integration on GPUs. *ACM Transactions on Mathematical Software*, 39(2).
- Victor Minden, Barry F. Smith, and Matthew G. Knepley. Preliminary implementation of PETSc using GPUs. In *Proceedings of the 2010 Workshop of GPU Solutions to Multiscale Problems in Science and Engineering*, 2012.

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