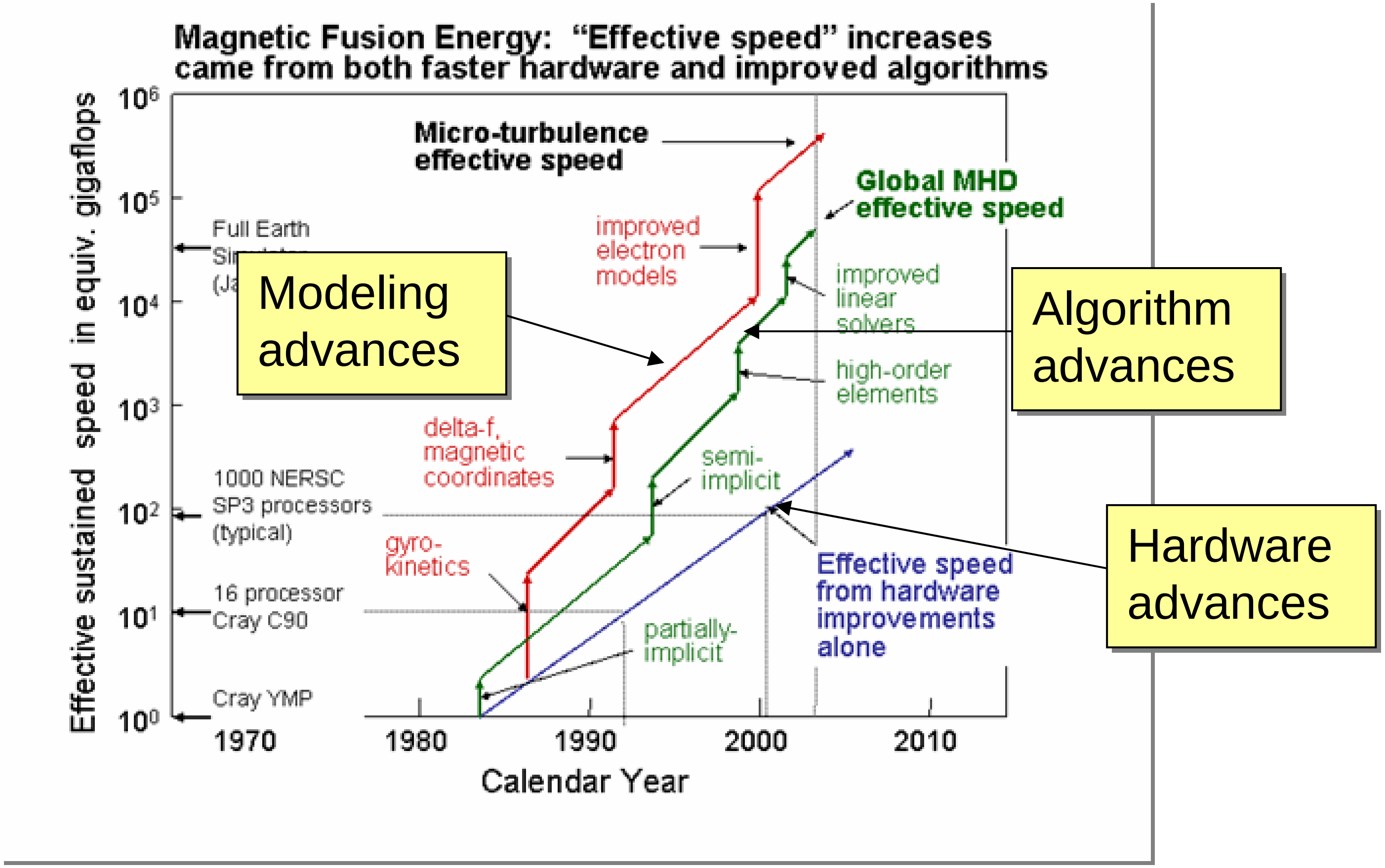
Solvers Discussion
FDM3D Workshop

Moderated by
David Keyes* and Mark Adams
TOPS Project & Columbia University

*in over my head and under my rest

Consider the full ecosystem

"You can never do just one thing" – Barry Commoner, late of UCSB



Metaphor for this discussion

- Imagine a conductor who is put over an orchestra of marvelous individual players
- He doesn’t quite know what to ask them to play together
- … but he knows that they are good!
- So, he decides to show them off with individual solos to generate ideas and to allow them to hear each other

Ten-minute “seeds”

- Alan Glasser, LANL
  – FETI-DP (Finite element tearing and interconnection, dual-primal formulation)
- Ravi Samtaney, PPPL
  – Preconditioned Jacobian-free Newton-Krylov
- Mark Adams, Columbia
  – Multigrid for hyperbolic problems
- Steve Jardin, PPPL
  – ADI-based on direct inversion
- Guoyong Fu
  – Implicit treatment of fast and Alfven waves

Other material to run through

- Xiao-Chuan Cai, UColorado-Boulder & K.
  - Nonlinear Schwarz
- Luis Chacon, LANL
  - Physics-based preconditioning
- Paul Fischer, ANL
  - Scalability estimates for global spectral and domain decomposed multilevel methods for elliptic kernels
- Sherry Li, LBNL
  - Parallel direct methods
- Tom Manteuffel, UColorado-Boulder
  - Preconditioning high-order methods, FOSLS
- Chi-Wang Shu, Brown University
  - Discontinuous Galerkin discretizations

Metrics for algorithmic comparison

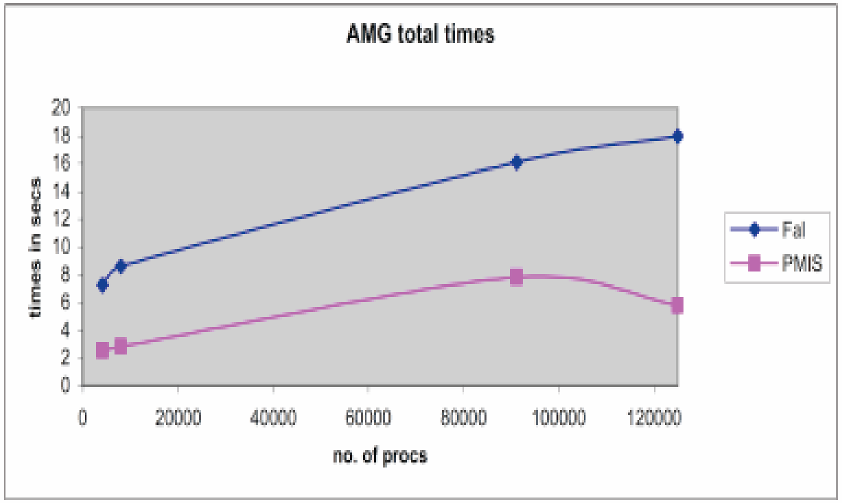
- **Convergence rate for the algebraic problem**
  - “Convergence factor” per iteration, related to “condition number,” any “preconditioning” and any “acceleration” scheme, based on some norm
- **Cost per iteration for the algebraic problem**
  - “Operator Complexity” relative to fine-grid “work unit”
- **Approximation effectiveness per degree of freedom of the algebraic problem**
  - “Order of convergence” of the discretization (relevant for smooth problems)
- **Implementation efficiency of the algorithm on a distributed, hierarchical memory computer**
  - Communication-to-computation “volume”
  - Number of communication startups and synchronizations
  - Spatial and temporal cache locality
- **Set-up versus reuse complexities and implementation efficiencies**
  - Matrix assembly and storage (or cost of function eval. if matrix-free)
  - Number of times set-up for a given system is reused
- **Extensibility to complex geometry, multiple components, problems with “bad” features (indefiniteness, nonsymmetry, inhomogeneity, anisotropy)**
- **Fragility with respect to local lack of smoothness**

Algebraic multigrid on BG/L

- Algebraic multigrid a key algorithmic technology
  – Discrete operator defined for finest grid by the application, itself, *and* for many recursively derived levels with successively fewer degrees of freedom, for solver purposes
  – Unlike geometric multigrid, AMG not restricted to problems with "natural" coarsenings derived from grid alone
- Optimality (cost per cycle) intimately tied to the ability to coarsen aggressively
- Convergence scalability (numbe
- While much research and development remains, multigrid will clearly be practical at BG/L-scale concurrency

Figure shows weak scaling result for AMG out to 120K processors, with one 25×25×25block per processor (up to 1.875B dofs)

c/o U. M. Yang, LLNL



7-pt Laplacian, total execution time, AMG-CG, total problem size ~2 billion

Major decision drivers for solvers

- Separation of scales and opportunity for implicitness to suppress physically uninteresting but explicit stability-limiting modes
- Exploitability of high-order discretizations
- Cost of partitioning and load balancing gridfunction and matrix operator objects following adaptation steps

Types of performance improvements

- More flops per second
  - Better per-processor performance
  - Better scalability
- More “science” per flop
  - Better formulations
  - Better discretizations
  - Better solution algorithms
- These potential improvements are usually not “orthogonal”
  - Some synergistic, some mutually interfering

Other objectives for a new code

- New capabilities
  - Sensitivities built in from the beginning will make the code useful for V&V, UQ, optimization, control, and inversion
  - Well defined interfaces will prepare the code for multiphysics uses
- New platforms
  - Any code written today will have to run on multicore processors (homogeneous and heterogeneous)

Our environment

- Adaptive or fully unstructured grids with local discretizations
  - Sparse, irregular data structures
- Distributed, hierarchical memory computers
  - MPI programming model, cache-awareness
- Premium on weak scaling, due to multiscale, multirate physics and to funding politics
  - Optimal, implicit methods based on domain decomposition
  - High-order spatial discretization and time integration schemes

Other directions for discussion

- Review of the other three math centers under SciDAC-2
  - APDEC
  - CSCAPES
  - ITAPS
- Review of the petascale hardware roadmaps
  - BlueGene: ANL, LLNL
  - Cell: LANL
  - XT: LBNL, ORNL, SNL