Society for Industrial and Applied Mathematics

Optimizing Stencil-based Algorithms

–a double minisymposium–

Organizers:

Jan Treibig

Georg Hager

Gerhard Wellein

Erlangen Regional Computing Center, Germany

David Keyes

King Abdullah Univ. of Sci. & Tech. (KAUST), Saudi Arabia

Issues addressed over the 8 talks

- The variety of stencil-based computations
- Performance prediction
- Performance engineering
- Communication reduction
- Compiler-based automation
- Domain-specific languages
- Architecture obliviousness

… many of the themes of Thursday's "Forward Looking" panel, specialized to some of Berkeley "Seven Dwarfs" (http://view.eecs.berkeley.edu/wiki/Dwarfs)

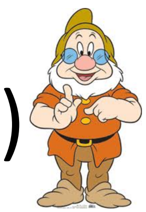
This opening talk: 2 parts

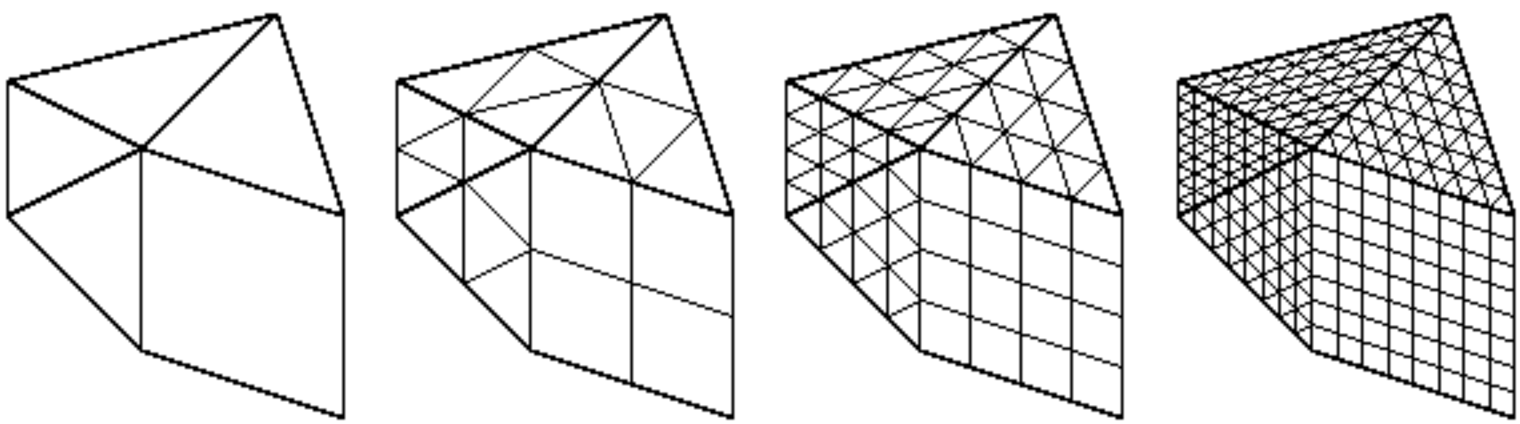
- Overview of the minisymposium
  - Theme 1: exploiting kernel and architecture structure to get high performance
  - Theme 2: adaptivity to strong scaling within a node and weak scaling across nodes
- New results on a merger of two popular techniques
  - increasing on-node temporal locality
  - decreasing off-node communication frequency
  - overlapping communication with computation

Models addressed

- Updates for explicit methods
  - typically, wave equations in 2nd-order form
- SpMVs for implicit methods
  - typically, diffusion-convection-reaction systems
    - sparse matrix-vector products of Krylov methods
    - smoothing recurrences of multigrid

Discretizations

- Structured grids (dwarf #5) 
- Unstructured grids (dwarf #6) 
- Hierarchies of grids (i.e., multigrid, dwarf #2) 
- Semi-structured grids (e.g., Hierarchical Hybrid Grids)



http://www10.informatik.uni-erlangen.de/en/Research/Projects/HHG/

Structured grids 

- Points updated using values from a local neighborhood
- Updates are logically concurrent
- Updates in place *or* using two copies to avoid read-write conflicts, giving vectorizability/SIMT
- Data access patterns regular and statically determinable
- Prefetching effective
- High spatial locality for long cache lines
- Some temporal locality within a full-grid update, with additional temporal locality across full-grid updates
- Data decomposed by domain, with ghost cell exchanges to complete neighborhoods
- Deep ghost-cell regions trade bandwidth to hide additional latency
- Parallel efficiency determined by the surface to volume ratio

Unstructured grids 

- Considerations of update ordering with multicoloring and partitioning minimizing surface-to-volume ratios mimic those of structured grid, but require graph algorithms
- Updates logically concurrent, but accesses require indirection, which adds memory for the indexing information and results in poorer spatial locality
- Vectorization/SIMT possible, but requires scatter-gather memory accesses
- Memory hierarchies less effective: spatial locality usually limited to one of the entity at a time, as related entities are accessed indirectly
- Potential for temporal locality, as a given entity is used in several different neighborhoods
- Mesh divided into "chunky" contiguous subdomains
- Mesh partitioning performed as a preprocessing step
- Communication and synchronization mostly between neighbors

Variety of stencil-based computations

- Issues arising from physics
  - Constant vs. variable coefficients (in space and time)
  - Number of spatial dimensions
  - Number of components
  - Scalar vs. tensor coefficients, isotropic vs. anisotropic
- Issues arising from numerical modeling
  - Discretization order
  - Structured vs. unstructured
  - Star vs. box
  - Centered vs. staggered
  - Formulaic or tabular coefficient
- Issues arising from hardware/software architecture
  - Precision
  - Vector/cache/GPU
  - Shared/distributed

Issues arising from physics (1/4)

- Constant vs. variable coefficient
  - Is the largest working set the field *vector* being operated on, or the *operator*, itself?
  - Spatially constant coefficient case is common in industrial seismic codes, where

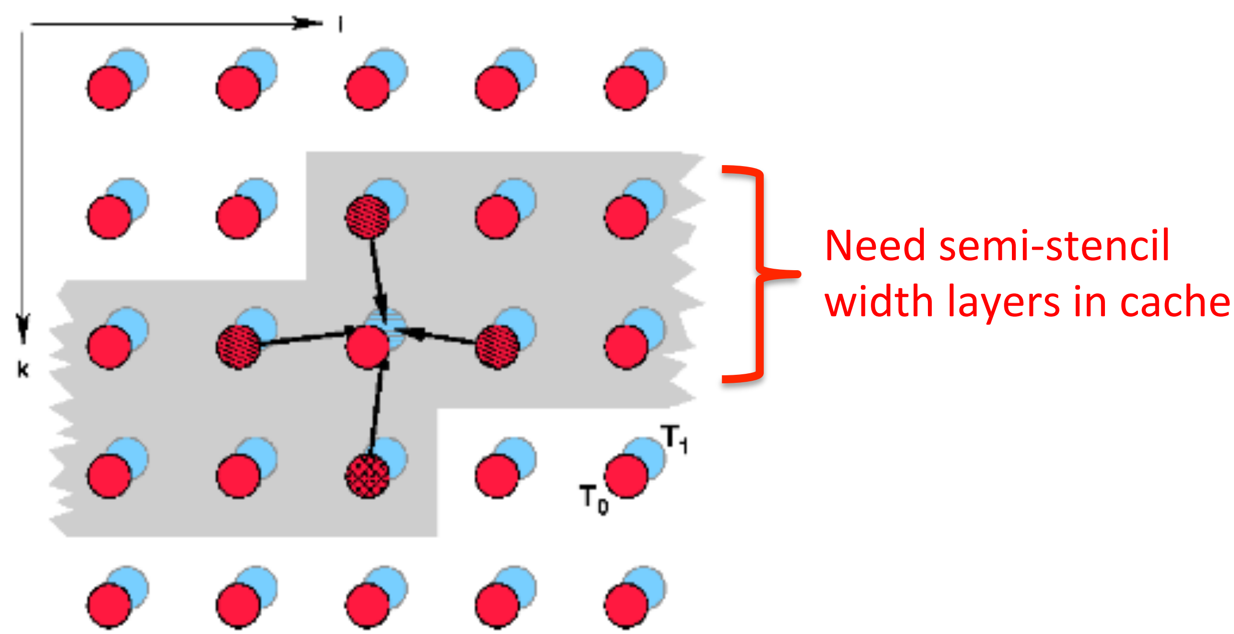
$$\rho(x)\frac{\partial^2 u}{\partial t^2} - \nabla \cdot K(x)\nabla u = f(x,t)$$

is often approximated to:

$$\frac{1}{c^2(x)}\frac{\partial^2 u}{\partial t^2} - \nabla^2 u = \bar{f}(x,t)$$

Issues arising from physics (2/4)

- Number of spatial dimensions
  - Affects the amount of cache required to satisfy the "layer condition" of maximal temporal reuse of field value at each stencil location



Wellein, *et al. Efficient Temporal Blocking for Stencil Computations by Multicore-Aware Wavefront Parallelization*, COMSAC 2009.

Issues arising from physics (3/4)

- Number of components
  - DOFs at a stencil point tend to interact with each other in a local dense block
  - Reasonable values for DOFs range from *scalar* (e.g., acoustics), to *several* (e.g., mechanics), to *hundreds* (e.g., detailed kinetics combustion)
  - Cross-component coupling at central or neighboring points can contribute to temporal reuse as much as spatial coupling to components of same type

Issues arising from physics (4/4)

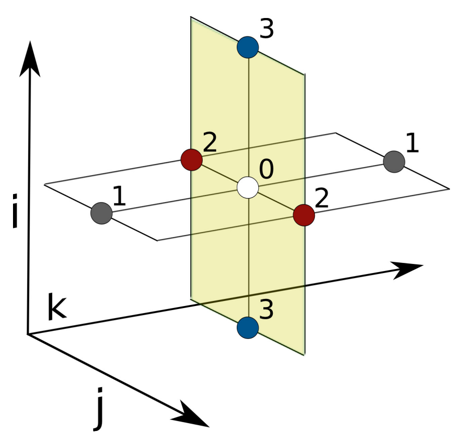
- Scalar vs. tensor coefficient, isotropic vs. anisotropic
  - Affects intensity of cross-coupling and number of distinct nonzero coefficients to be stored per point

$$\nabla \cdot \overleftrightarrow{D} \nabla u$$

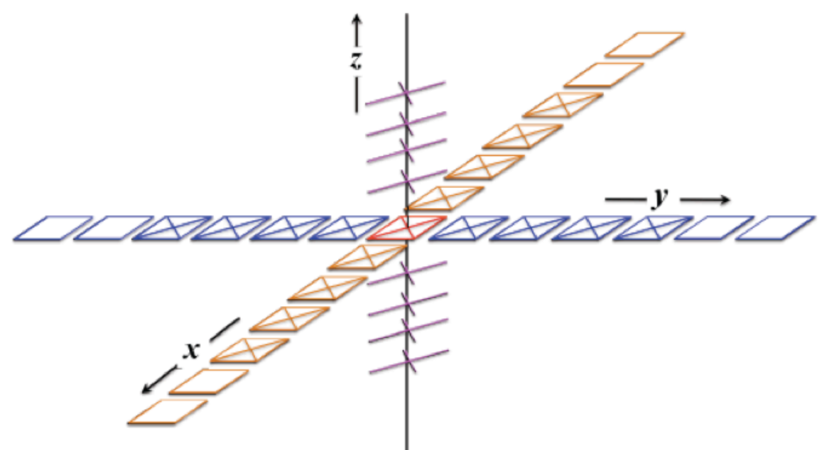
$$\overleftrightarrow{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} \quad \overleftrightarrow{D} = \begin{bmatrix} D_{xx} & 0 & 0 \\ 0 & D_{yy} & 0 \\ 0 & 0 & D_{zz} \end{bmatrix} \quad \overleftrightarrow{D} = \begin{bmatrix} D & 0 & 0 \\ 0 & D & 0 \\ 0 & 0 & D \end{bmatrix}$$

Issues arising from modeling (1/5)

- Discretization order
  - Temporal order affects number of copies of spatial array
  - Spatial order affects number of layers needed for reuse, size of tiles in wavefront ordering, and width of halo regions in distributed memory



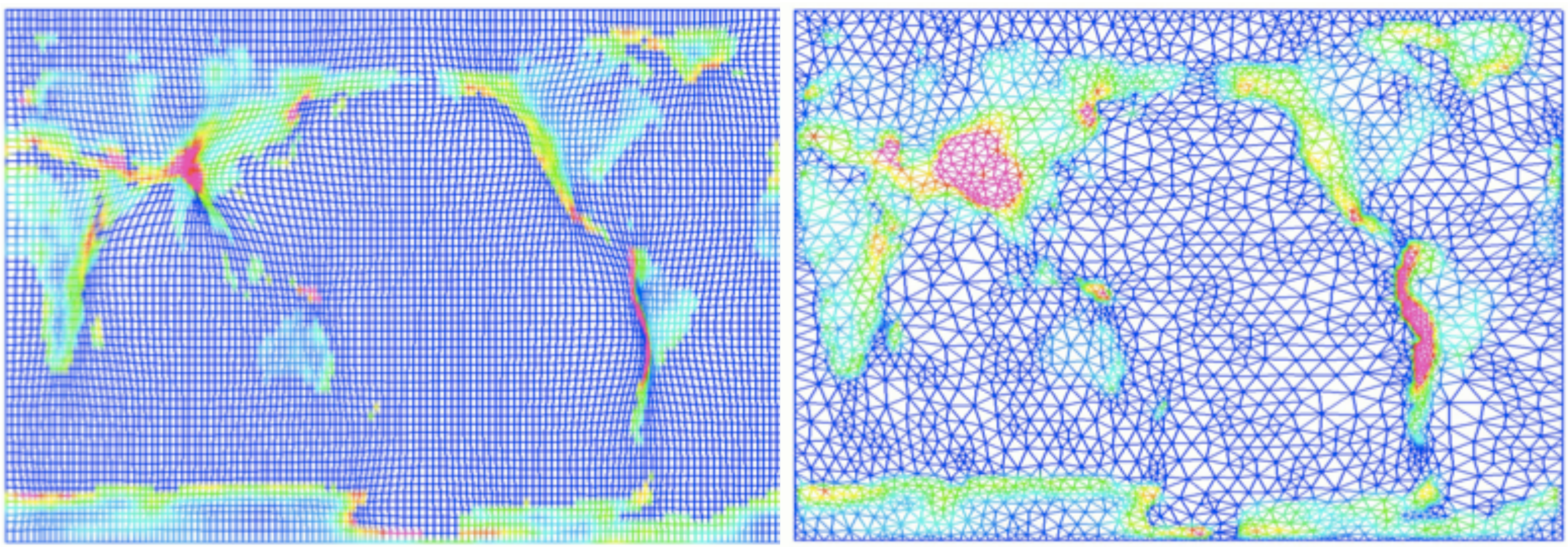
second order in space



eighth order in space

Issues arising from modeling (2/5)

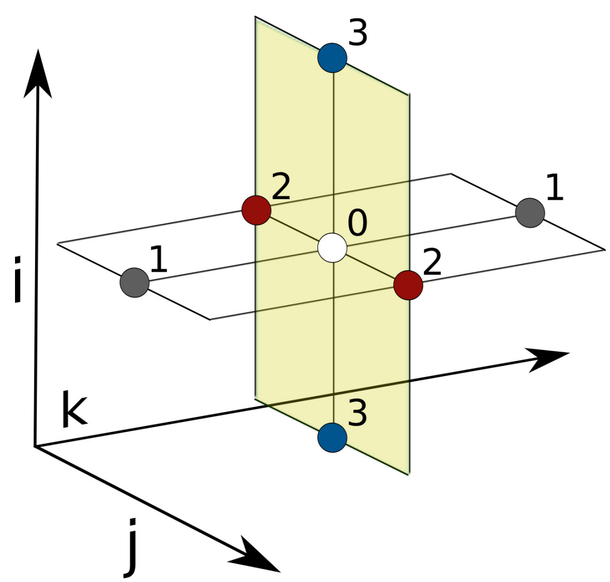
- Structured vs. unstructured
  - Affects regularity of addressing and number of integers required for connectivity (flop intensity)
  - Affects ability to do static unrolling



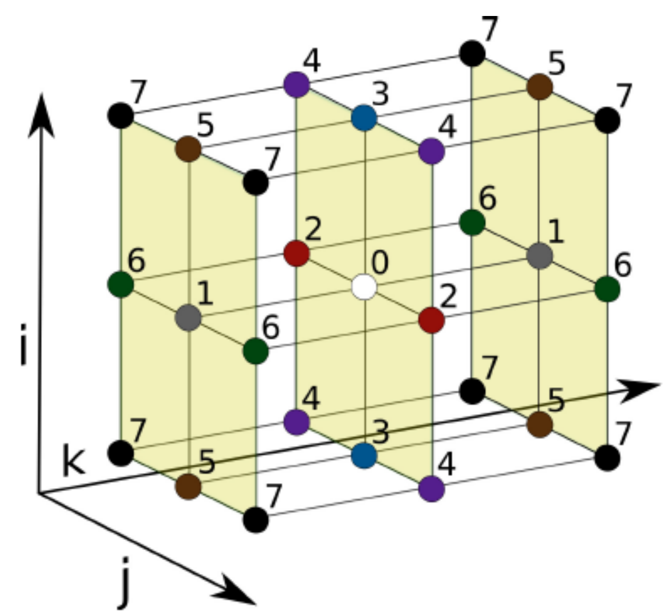
http://www.scidac.gov/ASCR/ASCR_TSTT/reports/TSTT2003Annual2.html

Issues arising from modeling (3/5)

- Star vs. box
  - Affects arithmetic intensity



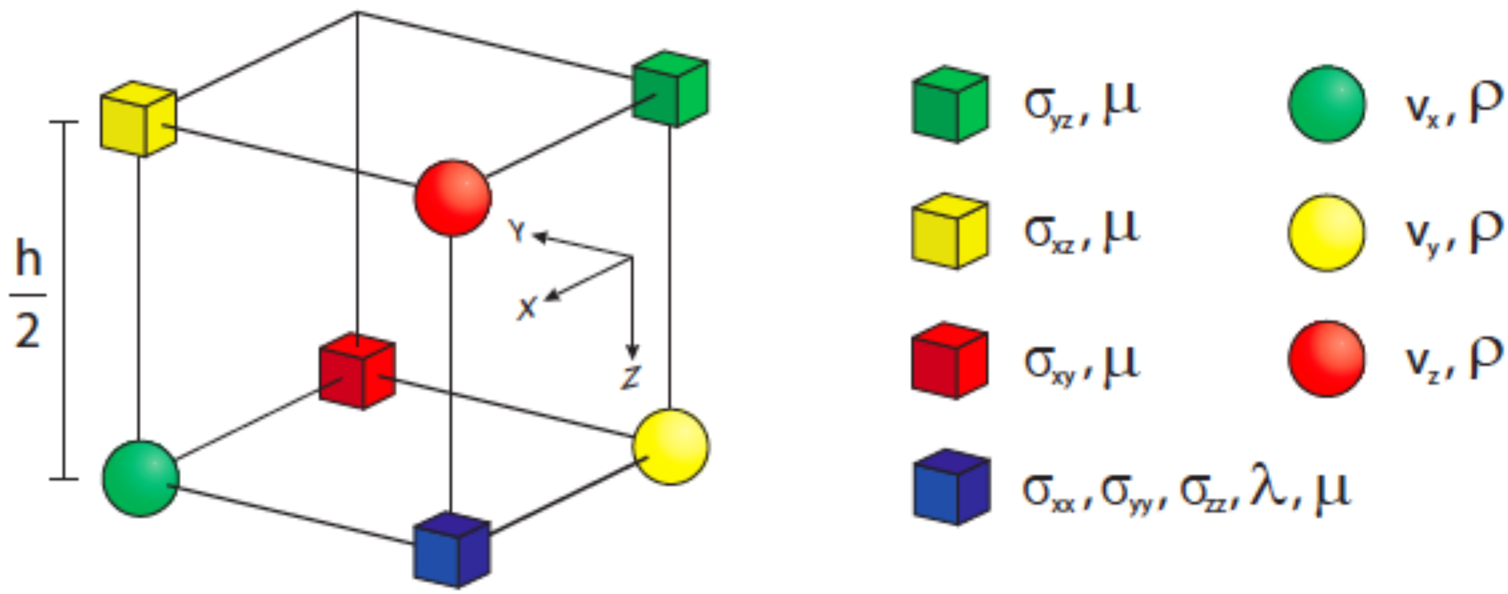
star



box

Issues arising from modeling (4/5)

- Centered vs. staggered
  - Affects code complexity, especially at boundaries of tiles or subdomains

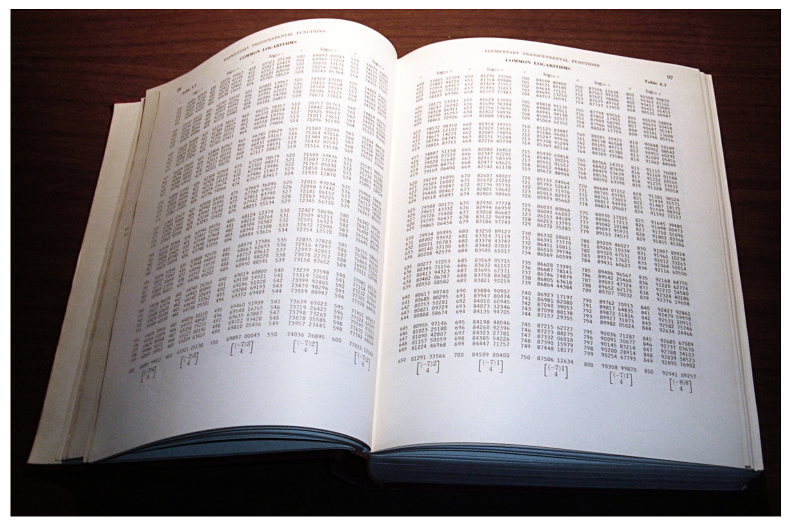


http://www.intechopen.com/download/pdf/26255

Issues arising from modeling (5/5)

- Formulaic or tabular coefficient
  - Affects caching of tabular data into which to interpolate, vs. often computationally intensive empirical or multiphysics first-principles formula to evaluate

$$D = f(x, t, u(x,t)) \quad \text{vs.}$$



Issues arising from architecture (1/3)

- Precision
  - Affects effective memory capacity and memory bandwidth per word
  - Can strongly affect arithmetic performance on today's GPUs

Issues arising from architecture (2/3)

- Vector/cache/GPU
  - Affects code portability
  - Affects performance portability

Issues arising from architecture (3/3)

- Shared/distributed
  - Affects programming model
  - Affects communication overhead
  - Affects synchronization idleness

Techniques used in this mini

- Tiling
- Wavefronts
- Halos
- Redundant computation

Hardware used in this mini

- Intel Sandy Bridge
- Intel Xeon Phi with IMCI
- NVIDIA K20X “Kepler” with OpenACC
- Distributed memory

Schedule – MS 58 (Fri. AM)

- Relevant Stencil Structures for Modern Numerics (this overview)
- Performance Engineering for Stencil Updates on Modern Processors ("ECM" model, a generalization of roofline model)
- Compiler-Automated Communication-Avoiding Optimization of Geometric Multigrid
- Automatic Generation of Algorithms and Data Structures for Geometric Multigrid

Schedule – MS 66 (Fri. PM)

- Stencil Computations: from Academia to Industry
- Evaluating Compiler-driven Parallelization of Stencil Micro-applications on a GPU-enabled Cluster (3 techniques x 10 microkernels on *Piz Daint*, #6 on Nov 2013 Top500)
- Firedrake: a Multilevel Domain Specific Language Approach to Unstructured Mesh Stencil Computations
- Tuning Sparse and Dense Matrix Operators in SeisSol (follow-up to M. Bader's plenary, Intel SNB and Xeon Phi)