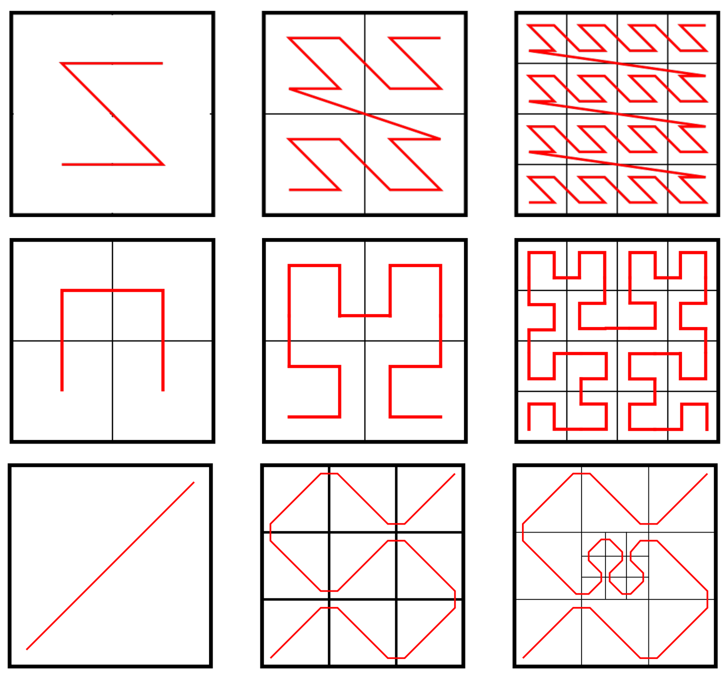
UNIVERSITY OF EXETER www.exeter.ac.uk

Towards a Cluster-based Parallelization with High-dimensional Adaptive Cartesian Grids

Martin Schreiber
M.Schreiber@exeter.ac.uk

Grid generation and data access

Candidates for D-dimensional space-filling curves for dynamically adaptive Cartesian grids:



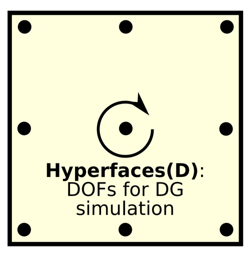
Avoid recursive grid traversal
Only traverse grid cells (see e.g. [5])

Exposing interfaces for high-dimensional dynamically adaptive grids to scientists:

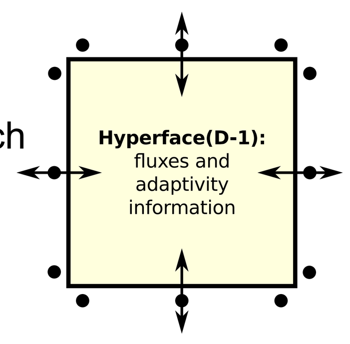
For a D-dimensional unit hypercube,
let $H_i(d_a)$ be the i-th hyperface of dimension d_a.
Then, access patterns can be described
by a communication matrix ($H_i(d_a)$ x $H_j(d_b)$)

Finite Volume/Discontinuous Galerkin DoF update, ...
$H_0(D)$->$H_0(D)$ timestep update

Flux computation, adaptivitivity markers, ...
$H_0(D)$->$H_i(D-1)$ send DoFs
$H_i(D-1)$->$H_i(D-1)$ compute flux
$H_i(D-1)$->$H_0(D)$ recv DoFs

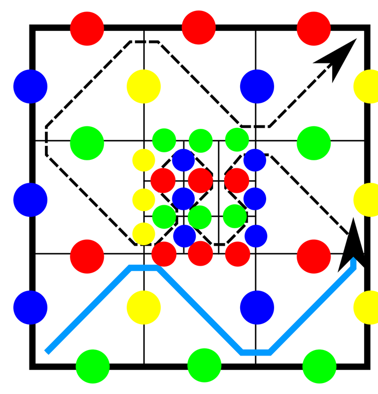


Visualization, limiters, ...
$H_0(D)$->$H_i(0)$ first/middle/last touch for DoFs [1,2,3]
$H_0(0)$->$H_0(D)$ recv DoFs

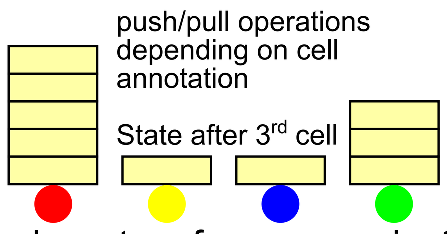


Realization

Communication: Non-recursive and stack-based for D-1 hyperfaces



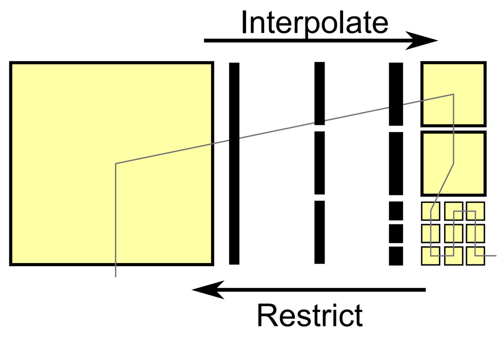
Stack-based communication based on coloring of grid primitives (see [1,2,3, 4] with recursive implementation & node embedding)

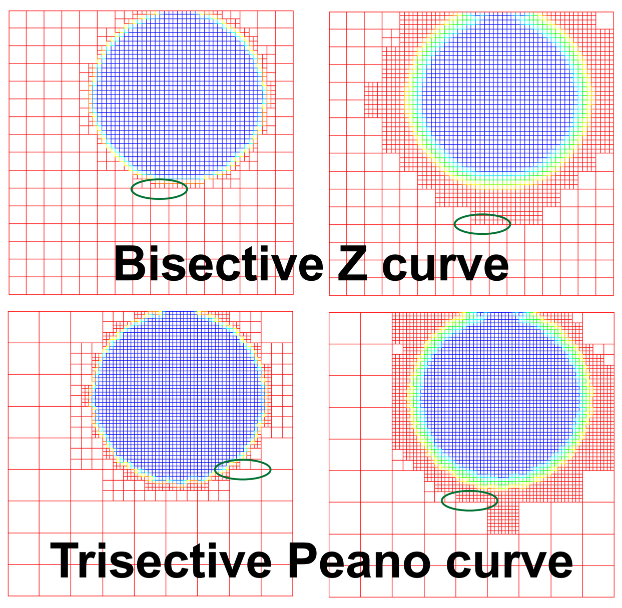


Stack system for communication

Hanging hyperfaces:

- Support **non 1:S balanced** hyperfaces with S > 3
- **Reduce/interpolate operations** on data communication stacks

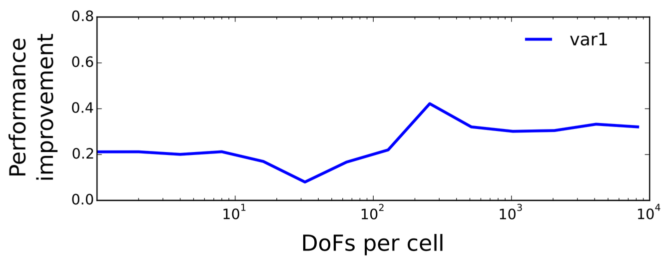




Example: Solving the **shallow-water equation** with finite volume on **dynamically adaptive grid**

Microbenchmark: Data exchange

Different stack-based communication scheme
a) forwarding array of DoFs (traditional one)
b) forward only pointers to DoFs

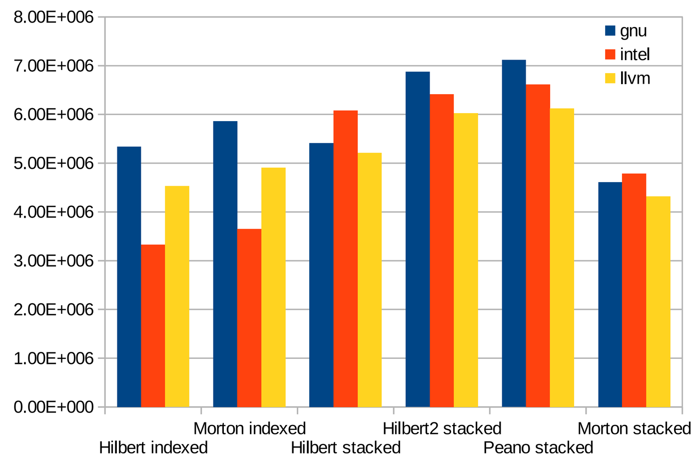


Avoid obsolete cache read/writes by forwarding pointers results in **performance improvements for most cases!**

Optimizations applicable to discontinuous Galerkin, extruded/multi-layer 2D meshes, etc.

Benchmarks & Outlook

Micro benchmark: Communication with different SFCs and regular grid:



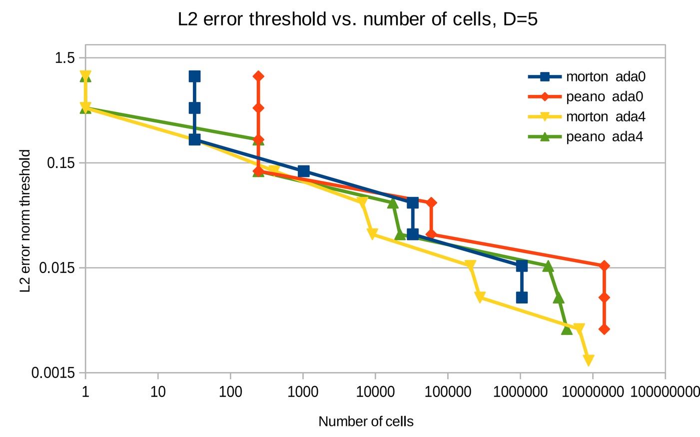
- **Stack-based communication faster** for Hilbert & Peano SFC compared to indexed grids
- **Morton-order stack-based communication about 10% slower** compared to Hilbert curve, however applicable in high dimensions

Number of cells for high-dimensional problems:

Let the following solution to a 5D Poisson problem be given:

$$\nabla \cdot (2\sin(\pi x_0)\sin(\pi x_1)\cos(2\pi x_2/3)\sinh(\sqrt{2}\pi x_4)\cosh(2\pi x_3/3))/32 = 0$$

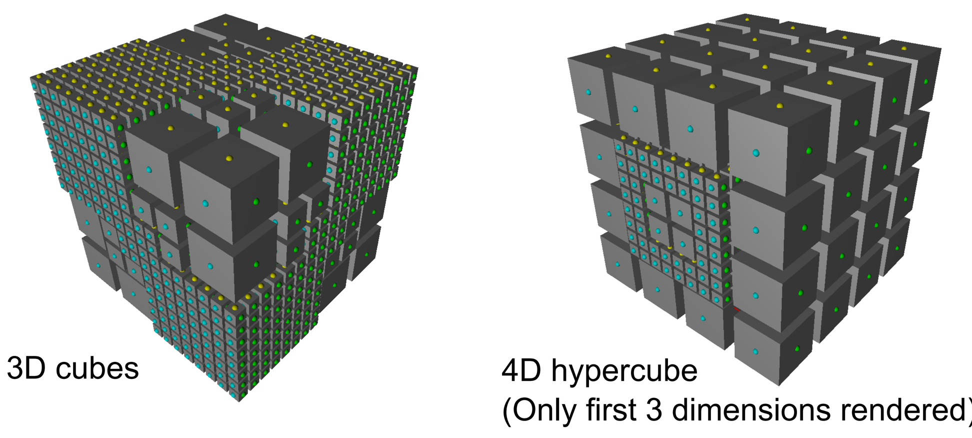
How many cells are required to approximate the 5D solution?
Local refinement criteria only



> 60% of cells saved compared to adaptive grids

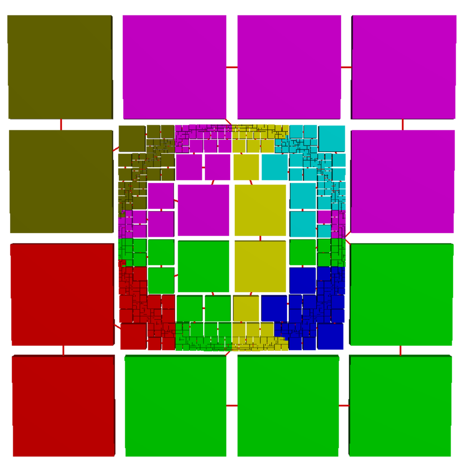
bisection (Z curve) requires less cells than trisection (Peano curve)

Support for high-dimensional communication:



Possible future work and applications

- **Partitioning**: **SFC-cuts**
- **Communication avoiding: No ghost layers**
- **Scalability:** Support of **arbitrary levels** for adjacent cells



Outlook: clustering
(efficient) handling of multiple partitions in a single program context [10]

MPI rank independent cluster generation [6]

Run-length encoded (RLE) connectivity [6]

OpenMP/TBB + MPI parallelization on cluster level [7]

Cluster-based data migration for load balancing [9]

Cluster-based **optimizations** [8]

(Possible) applications:

- Solve nD equations
- Tensor-free tensor comp.
- Simulations in **phase space** (6D and higher)
- **Climate / weather** with discontinuous Galerkin
- **HEVI** for climate/weather
- **Big data** engineering
- **Parallelization in time**
- **Visualization of nD** datasets
- **Exascale: Fault-tolerance** (via clustering **independency** features)

Related work:
[1] F. Günther, M. Mehl, M. Pögl und C. Zenger, A cache-aware algorithm for PDEs on hierarchical data structures based on space-filling curves, SIAM Journal on Scientific Computing, 2006
[2] F. Günther, M. Mehl, M. Pögl und C. Zenger, A Cache-Aware Algorithm for PDEs on Hierarchical Data Structures, PARA 2004
[3] A Parallel Adaptive Cartesian PDE Solver Using Space--Filling Curves, H.-J. Bungartz, M. Mehl und T. Weinzierl, Euro-Par 2006
[4] M. Mehl, T. Weinzierl and C. Zenger: A cache-oblivious self-adaptive full multigrid method, Numerical Linear Algebra with Applications, 2006
[5] C. Burstedde, L. C. Wilcox and O. Ghattas, Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees, SIAM Journal on Scientific Computing, 2011

Related work (two-dimensional *Sierpinski-SFC* generated triangular grids):
[6] M. Schreiber, H.-J. Bungartz, M. Bader: Shared Memory Parallelization of Fully-Adaptive Simulations Using a Dynamic Tree-Split and -Join Approach, HiPC 2012
[7] M. Schreiber, T. Weinzierl and H.-J. Bungartz: SFC-based Communication Metadata Encoding for Adaptive Mesh, ParCo 2013
[8] M. Schreiber, T. Weinzierl and H.-J. Bungartz: Cluster Optimization and Parallelization of Simulations with Dynamically Adaptive Grids, Euro-Par 2013
[9] M. Schreiber, H.-J. Bungartz: Cluster-based communication and load balancing for simulations on dynamically adaptive grids, ICCS 2014
[10] M. Schreiber, Cluster-Based Parallelization of Simulations on Dynamically Adaptive Grids and Dynamic Resource Management, PhD Thesis, 2014