High-Performance Computing for Flows in Porous Media

Peter Bastian[1], Eike Müller[2], Steffen Müthing[1]

[1]Interdisciplinary Center for Scientific Computing, U Heidelberg, Germany
[2]University of Bath, UK

Funded by DFG within SPP 1648 (SPPEXA)

KAUST Workshop on Scalable Hierarchical Algorithms
9–11 May 2016

Challenges in Modern Computer Architectures

- Power wall
  - Power consumption is **the** limiting factor for exascale!
  - Tianhe 2: 33,8 PF (Linpack), 17.8 MW $\Rightarrow$ 500 MW
  - Clock rate stagnates but Moore's law is still valid
- Memory wall
  - Bandwidth insufficient to sustain peak performance
  - (Still) significant latency for main memory access
  - Memory access is energetically costly
  - Memory is hierarchical: register, cache, local, remote, ...
- ILP wall
  - Automatic extraction of instruction level parallelism stagnates
  - Revival of vectorization in form of SIMD instructions
- Other
  - Heterogeneous node architectures: CPU + coprocessors
  - Reduced reliability due to immense number of components?

EXA-SCALE $\Rightarrow 10^6$ **processors** $\times\ 10^3$ **threads** $\times\ 10^9$ **FLOPS**

What Does This Mean for PDE Numerics?

- Exploit three levels of parallelism:
  MPI + shared memory multithreading + SIMD instructions
- FLOPS are "for free", memory access is expensive
- Matrix-based methods
  - **Robust preconditioners** available, e.g. **AMG**
  - Poor floating point performance, bound by memory bandwidth
- Matrix-free methods
  - Explicit time-stepping or certain solvers, e.g. CG, GMG
  - Low-order FE schemes:
    - Exploit problem structure: linear transformation, regular grids, constant coefficients, "stencil codes"
    - Vectorization over several elements
  - **High-order (DG) FE schemes**: naive complexity $O(p^{\alpha d} h^{-d})$
    - Complexity reduction through **tensor-product approach**
    - Medium high order: Vectorization within one element
    - Very high order: Several threads to one element
- Other: mixed precission, increased asynchronicity

Contents

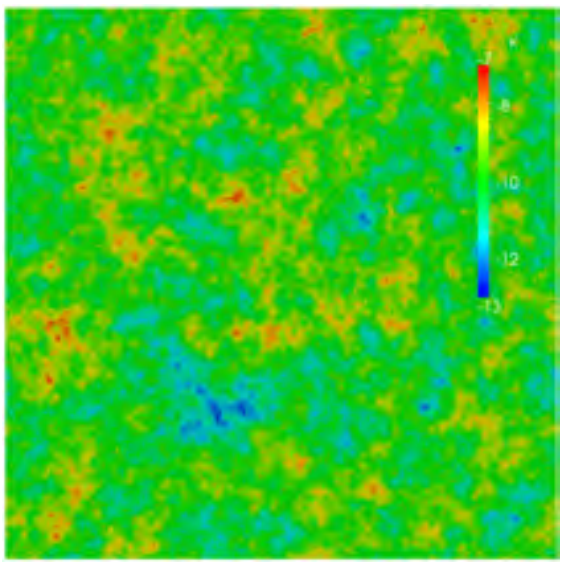
Contents

Single Phase Flow and Transport

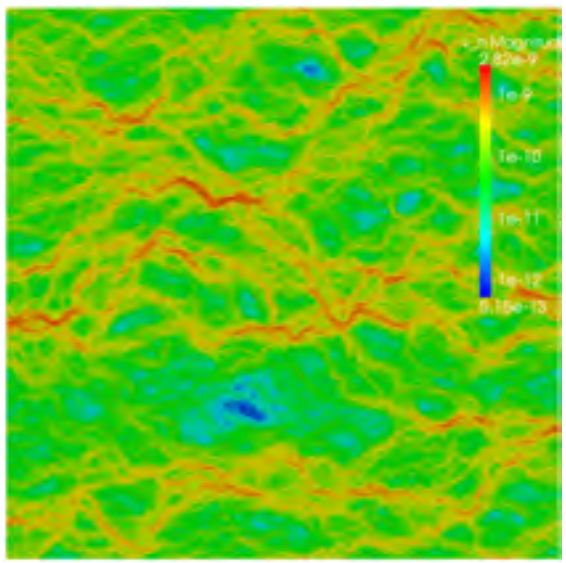
Parabolic/hyperbolic problem coupled to elliptic problem:

$$\nabla \cdot v = f, \qquad v = -\frac{K}{\mu}(\nabla p - \rho g),$$
$$\partial_t(\Phi c) + \nabla \cdot (cv - D(v)\nabla c) = q.$$

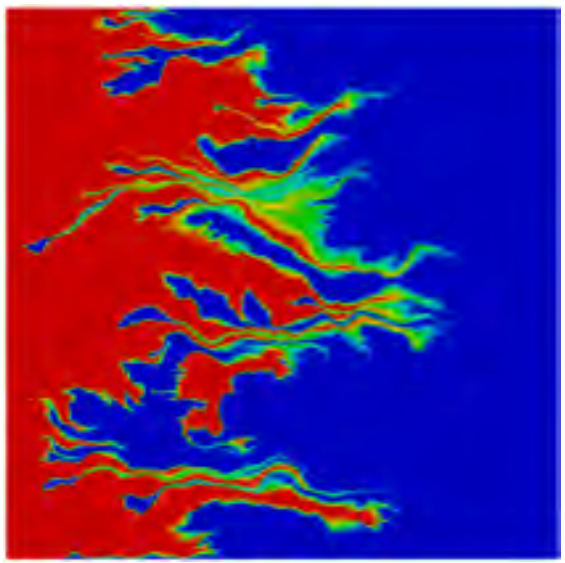
Tensor K, D, heterogeneity, convection-dominated
effective macro-scale descriptions, uncertainty of parameters



Permeability K



Velocity magnitude $\|u\|$



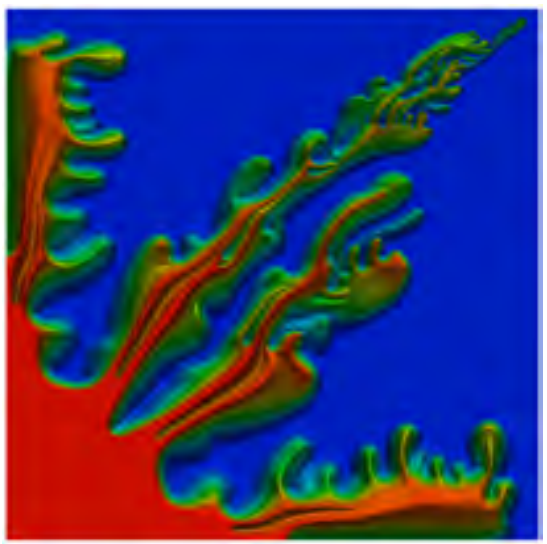
Concentration c

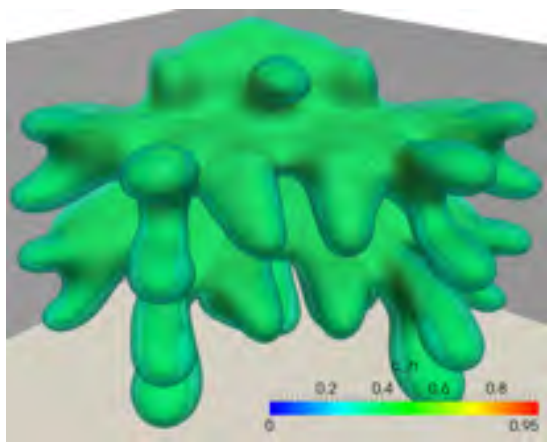
Application: Unstable Flows in Porous Media

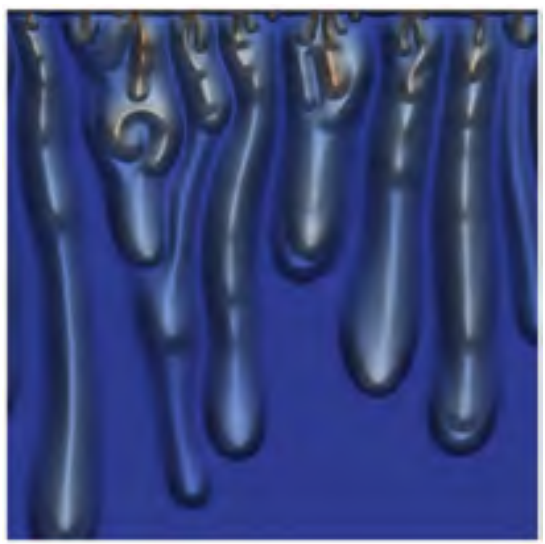
$$\nabla \cdot v = 0, \qquad v = -\frac{K}{\mu(c)}(\nabla p - \rho(c)g)$$

$$\partial_t(\Phi c) + \nabla \cdot (cv - D(v)\nabla c) = 0$$

Density driven flow, miscible discplacement
Can exploit high resolution schemes







Incompressible Immiscible Two-phase Flow

Total fluid conservation:

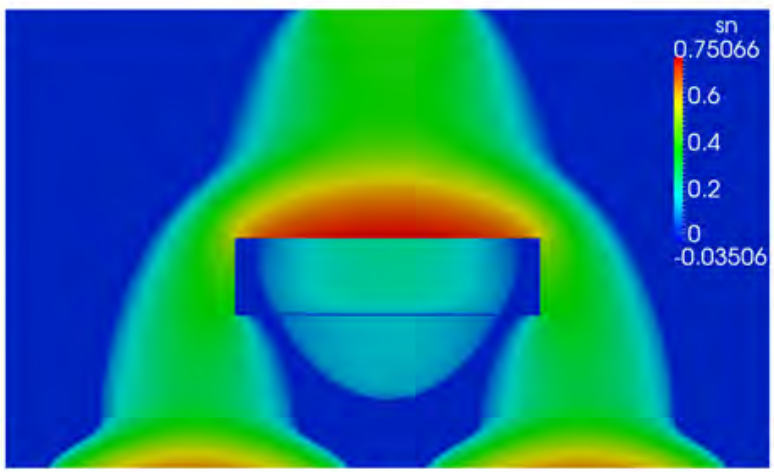
$$\nabla \cdot (v_a - \lambda_n K \nabla \phi_c) = q, \qquad v_a = -\lambda_t K \nabla \phi_w.$$

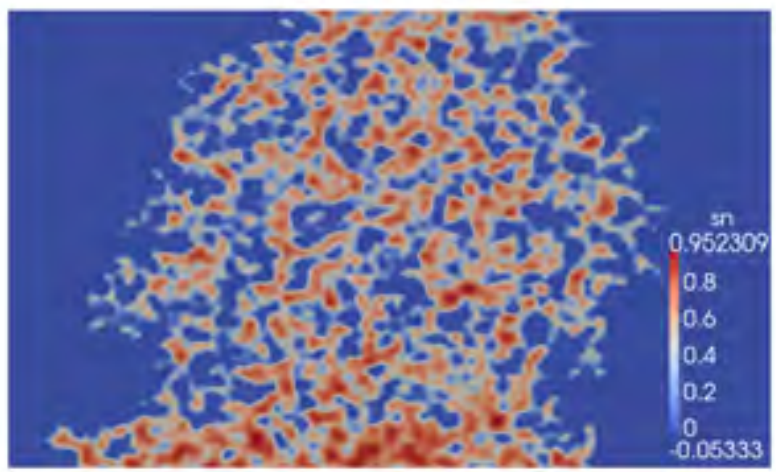
Conservation of non-wetting phase:

$$\Phi \partial_t (1 - \psi(\phi_c)) + \nabla \cdot (f_n(1 - \psi(\phi_c)) v_a - \lambda_n (1 - \psi(\phi_c)) K \nabla \phi_c) = q_n.$$

Primary variables: $\phi_w = p_w - \rho_w g d$, $\phi_c = p_c - (\rho_n - \rho_w) g d$

$\psi(\phi_c)$, $\lambda_n(\phi_c)$, $\lambda_t(\phi_c)$, $f_n(\phi_c)$ are nonlinear functions





Discontinuous Galerkin (DG) Finite Element Methods

- Advantages
  - Potentially high order convergence
  - High-order methods beneficial even with low regularity
  - Element-wise conservative
  - Full permeability tensors, discontinuous coefficients
  - Handles elliptic, parabolic and hyperbolic equations equally well
  - Unstructured grids, nonconforming refinement
  - Contiguous memory access pattern
  - Block matrix structure: less indirect access, BLAS level 3

- Disadvantages
  - No monotonicity (for 2nd order problems)
  - Free parameter to tune
  - Higher number of DOFs compared to standard FE or FV
  - Linear systems considered to be more difficult to solve

DG for Groundwater Flow Equation

SIPG + weighted averages (*Di Pietro, Ern, Guermond '2008*):

$$a_h(p,w;c) = \sum_{T\in\mathcal{T}_h}\int_T [\frac{K}{\mu(c)}(\nabla p - \rho(c)g)]\cdot\nabla w - \sum_{F\in\mathcal{F}_h^{iD}}\int_F \nu_F\cdot\{\frac{K}{\mu(c)}(\nabla p-\rho(c)g)\}_\omega [\![w]\!]$$

$$- \sum_{F\in\mathcal{F}_h^{iD}}\int_F \theta\nu_F\cdot\{\frac{K}{\mu(c)}\nabla w\}_\omega [\![p]\!] + \sum_{F\in\mathcal{F}_h^{iD}}\int_F \gamma_{F,f}[\![w]\!][\![p]\!]$$

$$l(w) = \sum_{T\in\mathcal{T}_h}\int_T q_f w - \sum_{F\in\mathcal{F}_h^{N}}\int_F j_f w \qquad \theta = 1 \text{ (SIPG)},\ \theta = 0 \text{ (OBB)},\ \theta = -1 \text{ (NIPG)}$$

$$\delta_{Kn}^{\pm} = \nu_F^t K^{\pm}\nu_F/\mu(c)^{\pm},\quad \omega^- = \frac{\delta_{Kn}^+}{\delta_{Kn}^- + \delta_{Kn}^+},\quad \omega^+ = \frac{\delta_{Kn}^-}{\delta_{Kn}^- + \delta_{Kn}^+}.$$

Penalty term ($\langle\cdot,\cdot\rangle$ harmonic average, p polynomial degree, d space dimension):

$$\gamma_{F,f} = \alpha\langle\delta_{Kn}^-,\delta_{Kn}^+\rangle M,\quad M = p(p+d-1)\frac{|F|}{\min(|T^-(F)|,|T^+(F)|)}$$

Contents

Multigrid Method

- Wish to solve large $Ax = b$
- Assume hierarchy of matrices

$$A_l \in \mathbb{R}^{I_l \times I_l} \qquad l = 0, \ldots, L, \; A_L = A$$

  $I_l = \{1, \ldots, n_l\}$ denotes the index set on level l
- Transfer matrices

$$R_l \in \mathbb{R}^{I_{l-1} \times I_l}, \; P_l \in \mathbb{R}^{I_l \times I_{l-1}} \qquad l = 1, \ldots, L$$

- Geometric multigrid: A_l, R_l, P_l related to hierarchy of finite element spaces

$$V_l = V(\mathcal{T}_l) = \text{span}\{\varphi_{i,l} : i \in I_l\} \qquad l = 0, \ldots, L$$

  constructed from a mesh hierarchy

$$\mathcal{T}_l \qquad l = 0, \ldots, L$$

- Algebraic multigrid: Only A and b are given

Multiplicative Multigrid V-Cycle

Require: x approximation of solution of linear system $Ax = b$
Ensure: x improved approximation

$d_L \leftarrow b - Ax$; $v_L \leftarrow 0$; ▷ Compute defect
for $l \in \{L, \ldots, 1\}$ **do**
  Smooth(v_l, d_l, ν_1); ▷ Pre-smoothing (updates v_l)
  $d \leftarrow d_l - A_l v_l$. ▷ Compute defect
  $d_{l-1} \leftarrow R_l d$; ▷ Restrict defect
  $v_{l-1} \leftarrow 0$; ▷ Prepare coarser level
end for
Solve $A_0 v_0 = d_0$; ▷ Exact coarsest level solve
for $l \in \{1, \ldots, L\}$ **do**
  $v_l \leftarrow v_l + \xi P_l v_{l-1}$; ▷ Prolongate coarse grid correction
  Smooth(v_l, d_l, ν_2); ▷ Post-smoothing (updates v_l)
end for
$x \leftarrow x + v_L$;

Geometric and Algebraic Multigrid

- **Geometric multigrid**
  - $\mathcal{T}_l, 0 \leq l \leq L$: nested sequence of meshes
  - $V_l = V(\mathcal{T}_l)$ nested sequence of finite element spaces
  - Basis representation: $\phi_{i,l-1} = \sum_{j \in I_l} (R_l)_{i,j} \phi_{j,l}$ gives R_l
  - Then $A_{l-1} = R_l A_l P_l$ with $P_l = R_l^T$ (Galerkin product)
- **Ruge-Stüben algebraic multigrid**
  - Partition index set $I_l = F_l \cup C_l$ and set $I_{l-1} = C_l$
  - Determine entries of prolongation matrix $P_l \in \mathbb{R}^{I_l \times C_l}$
  - Set $A_{l-1} = R_l A_l P_l$ with $R_l = P_l^T$
  - Selection of C_l and interpolation factors depend on A_l
- **Agglomeration algebraic multigrid**
  - Partition index set $I_l = \bigcup_{i=1}^{n_{l-1}} C_{i,l}$ (clustering)
  - Partitioning depends on A_l
  - Set $(R_l)_{i,j} = 1$ if $j \in C_{i,l}$ and zero else ("piecewise constants")
  - Set $A_{l-1} = R_l A_l P_l$ with $P_l = R_l^T$
  - Important: scale coarse grid correction by factor ξ ($\xi \approx 2$ for regular 2^d coarsening of diffusion problem)

Agglomeration AMG: Details I

- References
  - Developed independently in the mid 90s: *Vanek et al (1994)*, *Braess (1995)*, *Raw et al (1996)*
  - Theory and practice by Notay, e.g. *Notay, Napov, JCP, 2015*
  - Combinatorial algebraic multigrid: I. Koutis, G.L. Miller, and R. Peng. arXiv:1102.4842v4
- Our implementation follows *Raw et al (1996)*
  - Symmetric strength of connection criterion (for M-matrices): Edge $(i,j) \in I_l \times I_l$ is called strong iff

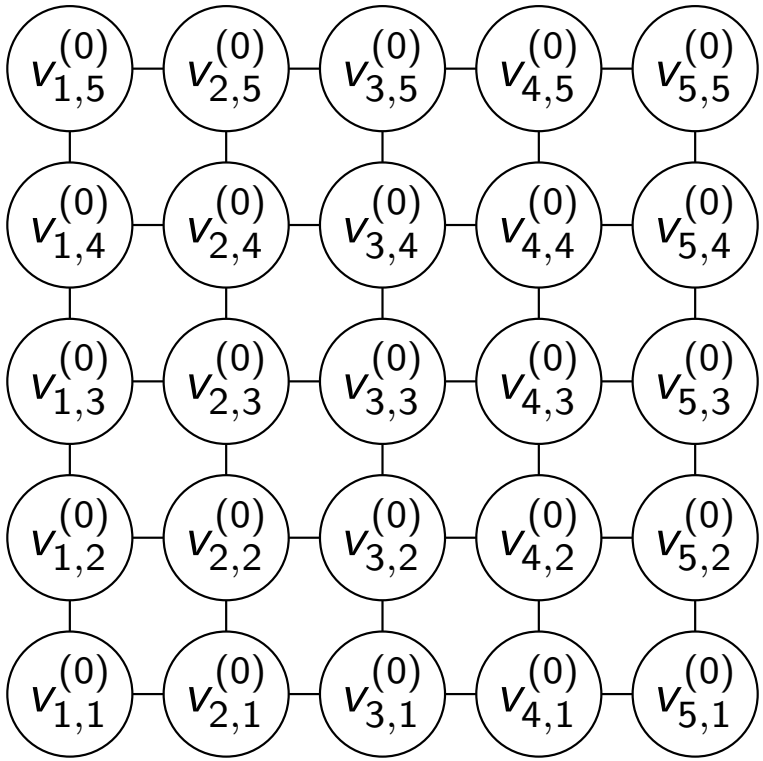
$$\frac{a_{ij}\,a_{ji}}{a_{ii}\,a_{jj}} \geq \delta \min(\eta_i, \eta_j)$$

    with $\eta_i = \max_j \frac{a_{ij}\,a_{ji}}{a_{ii}\,a_{jj}}$ and threshold $0 < \delta < 1$.
  - Greedy region growing algorithm builds clusters of arbitrary size (aggressive coarsening)
  - Fill-in minimization: roundness, merge small clusters
  - Non-M-matrices: ignore positive off-diagonals
  - Systems of PDEs: norm of blocks, guiding component

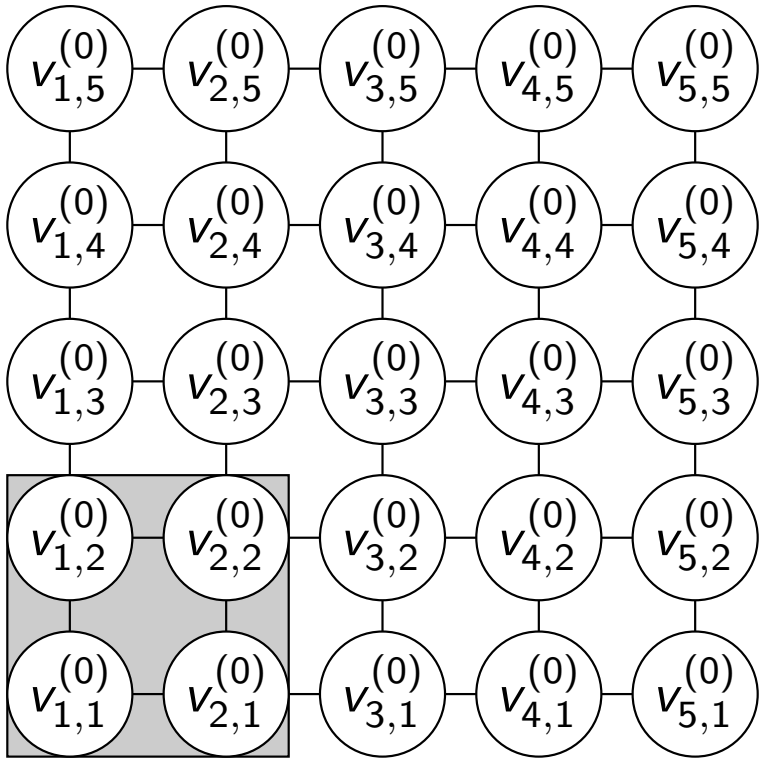
Agglomeration AMG: Details II

- Parallelization
  - Row-wise partitioning of the matrix with 1-overlap
  - Build clusters in parallel and exchange overlap
  - Build coarse system in parallel w/o communication
  - $P \gtrapprox 5000$ requires successive coarsening
- Advantages
  - Quite robust for elliptic problems with heterogeneous coefficients
  - Sparsity pattern is very well preserved on coarser levels
  - Conjecture: AAMG V-Cycle preconditioner yields $O(L)$ condition number
- Disadvantages
  - Needs to be used as preconditioner
  - Convergence for Poisson equation is generally not optimal (is cured by smoothed aggregation)

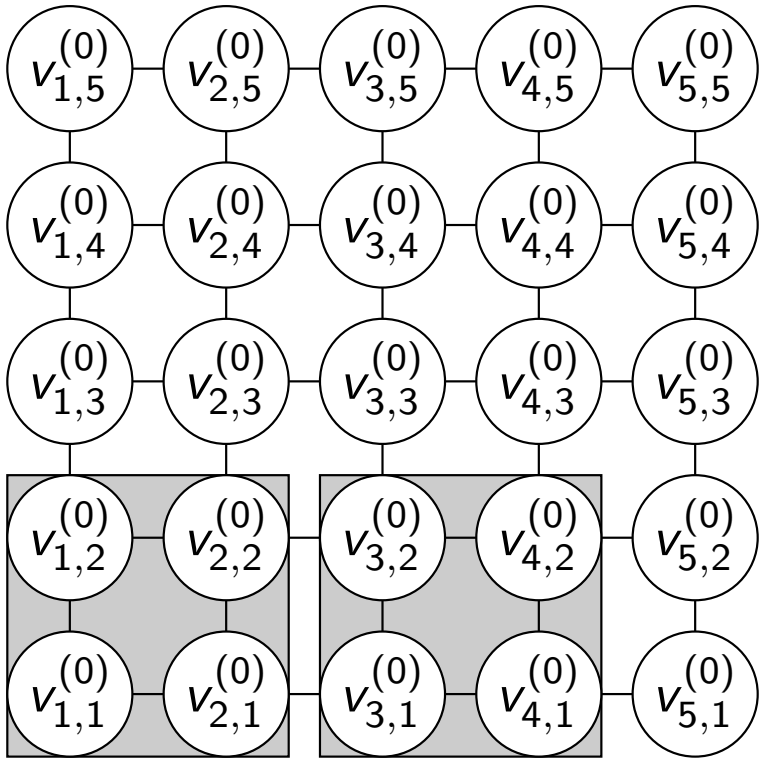
Sequential Agglomeration



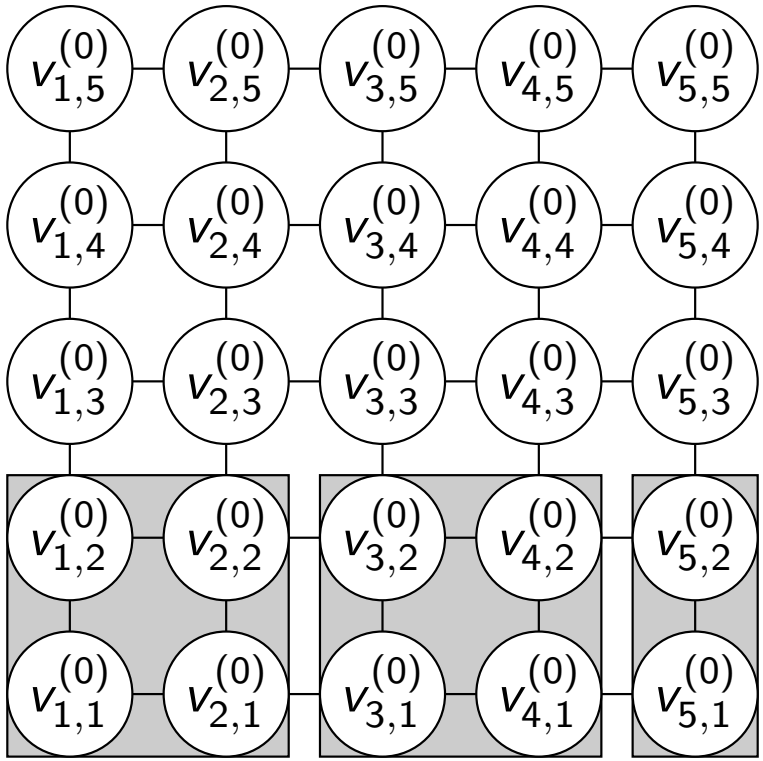
Sequential Agglomeration



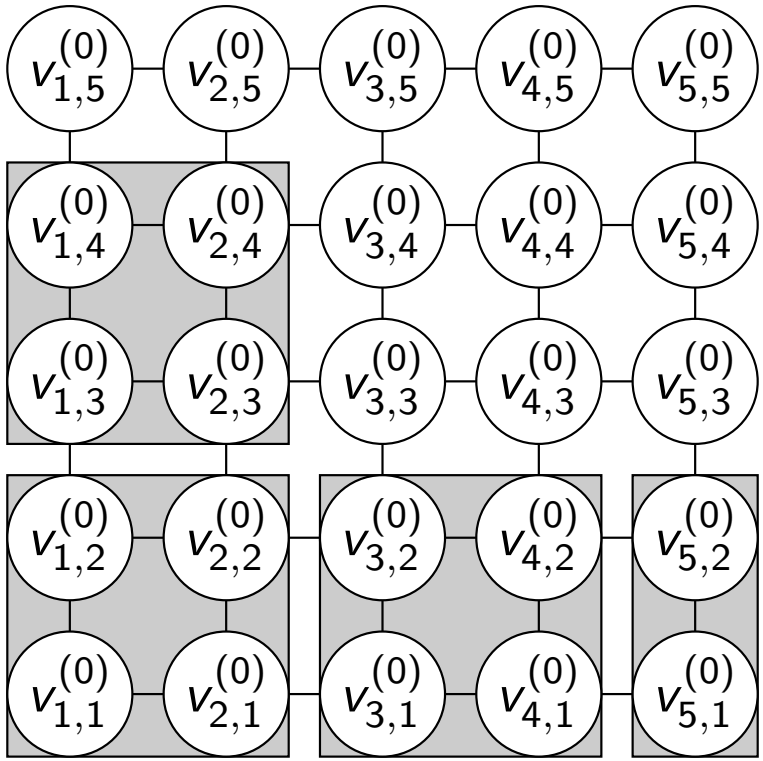
Sequential Agglomeration



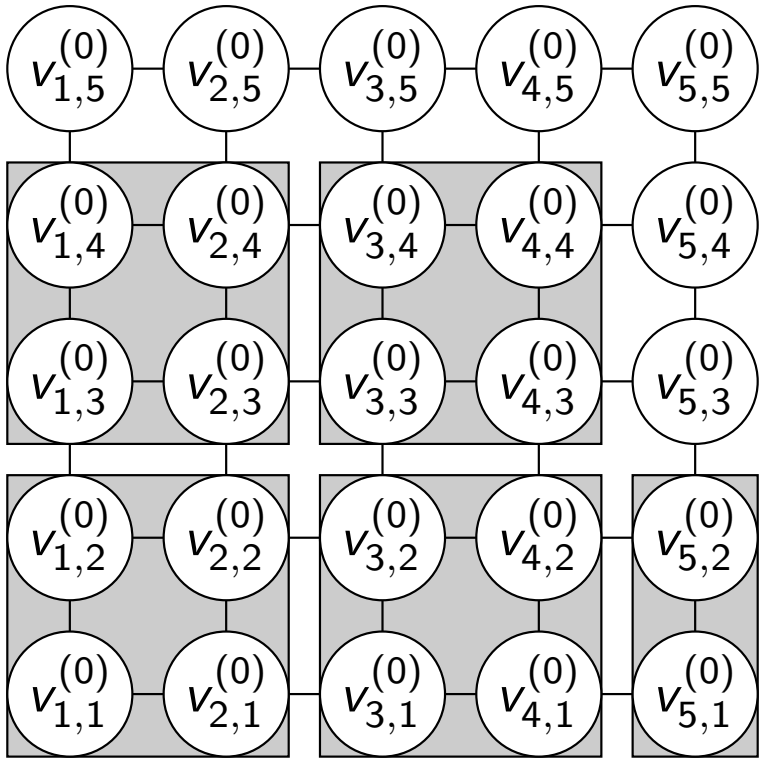
Sequential Agglomeration



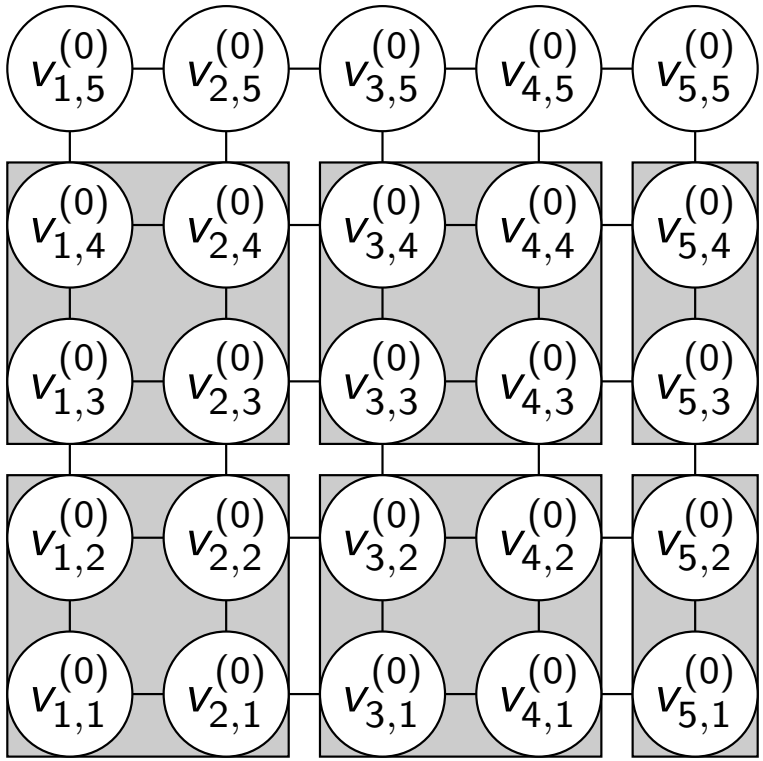
Sequential Agglomeration



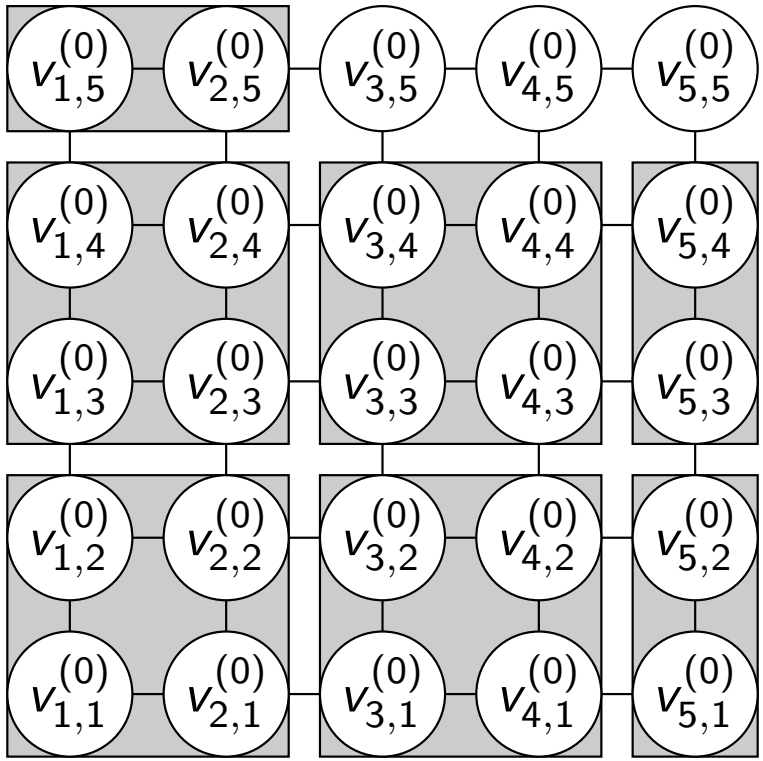
Sequential Agglomeration



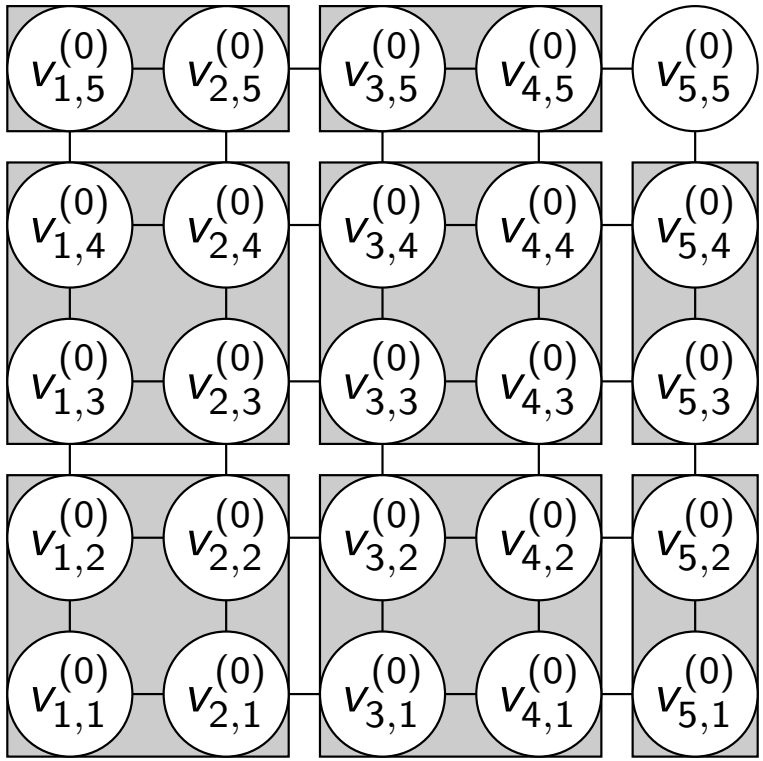
Sequential Agglomeration



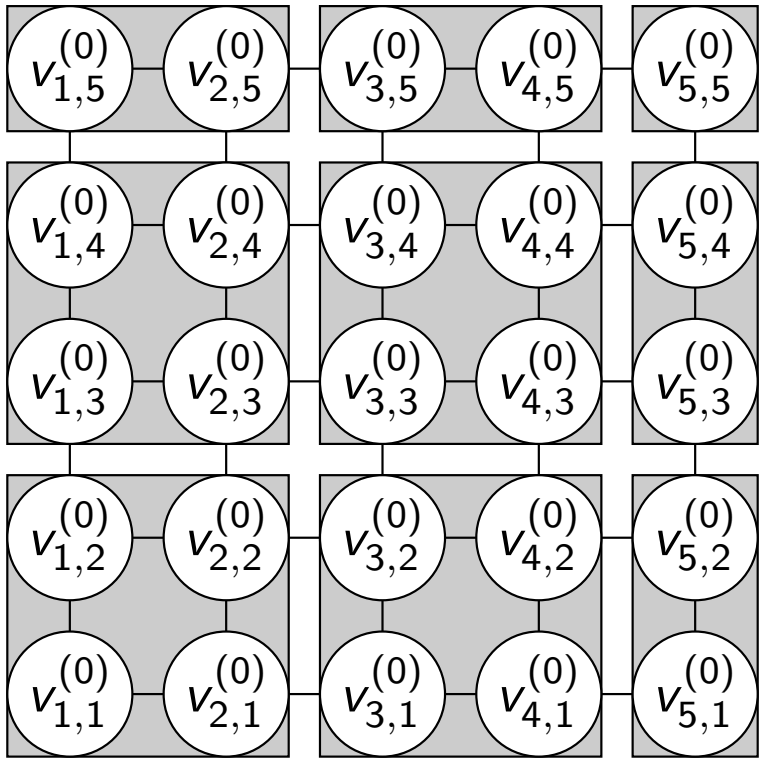
Sequential Agglomeration



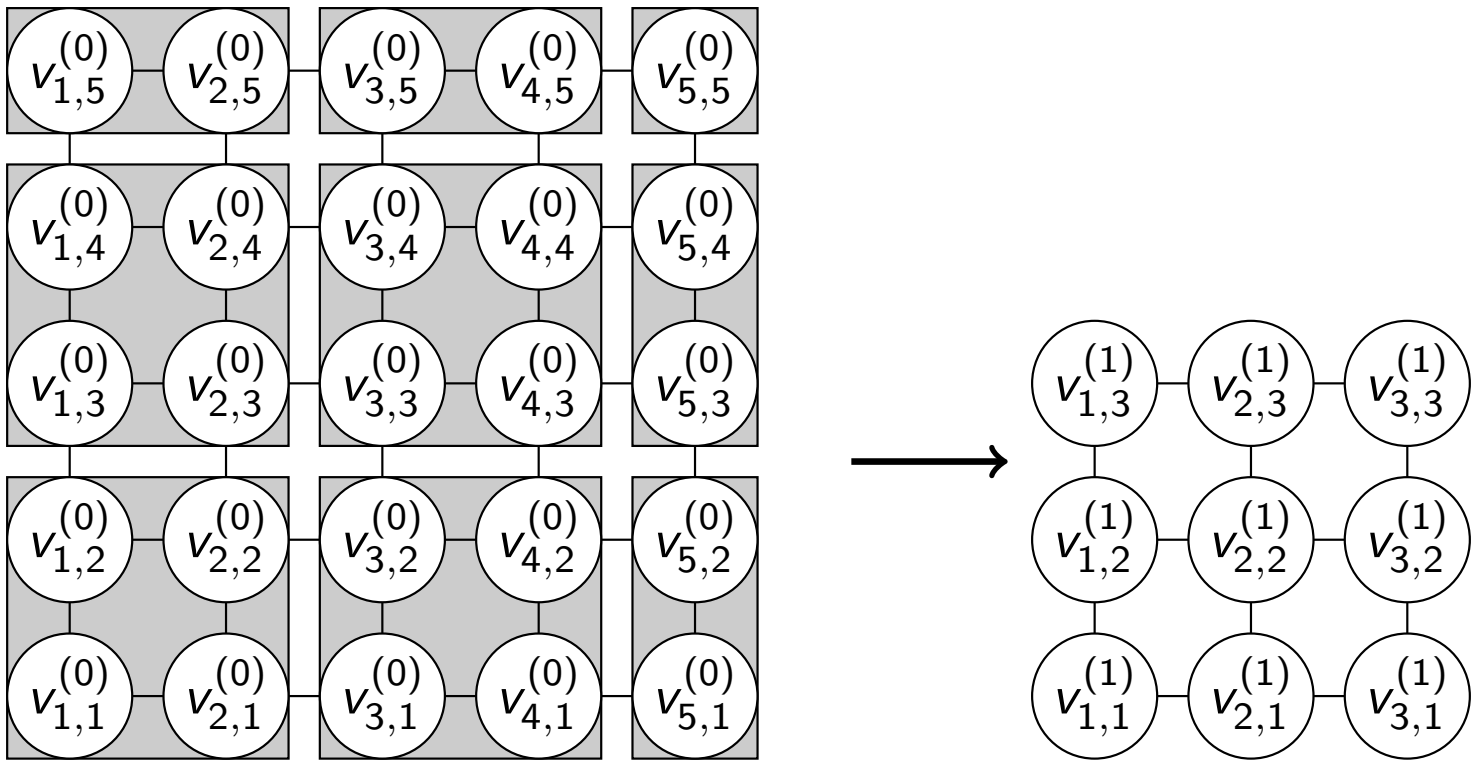
Sequential Agglomeration



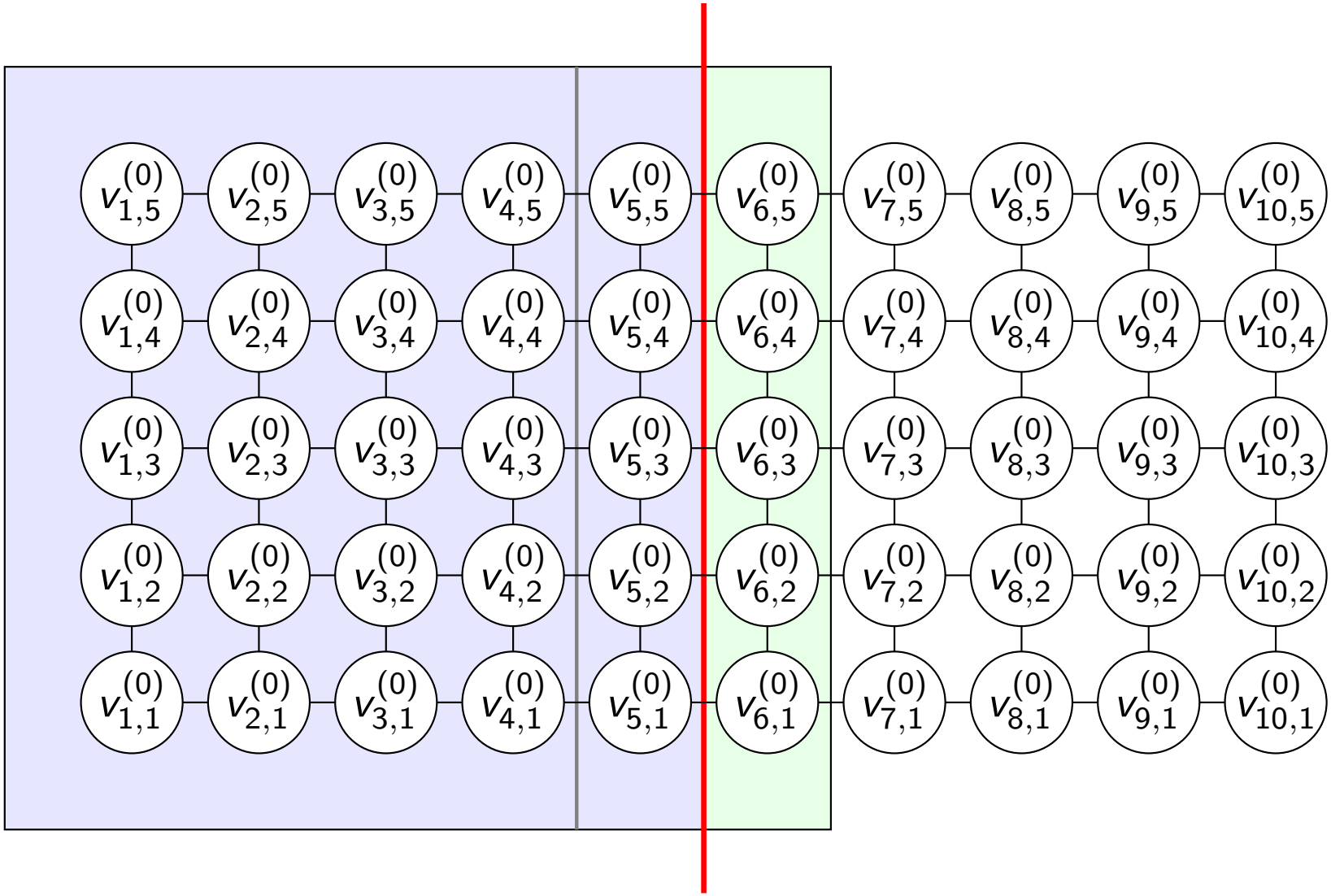
Sequential Agglomeration



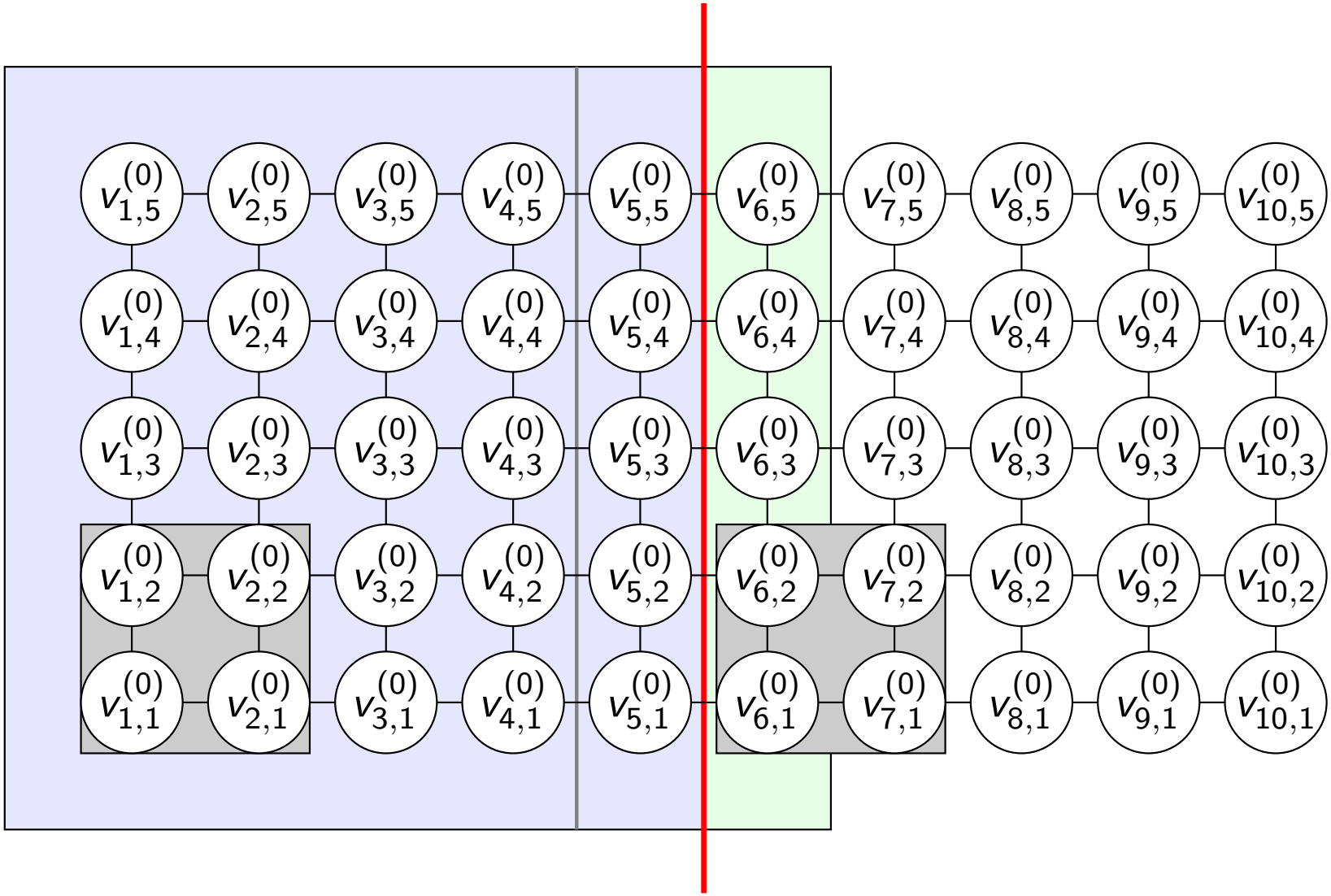
Sequential Agglomeration



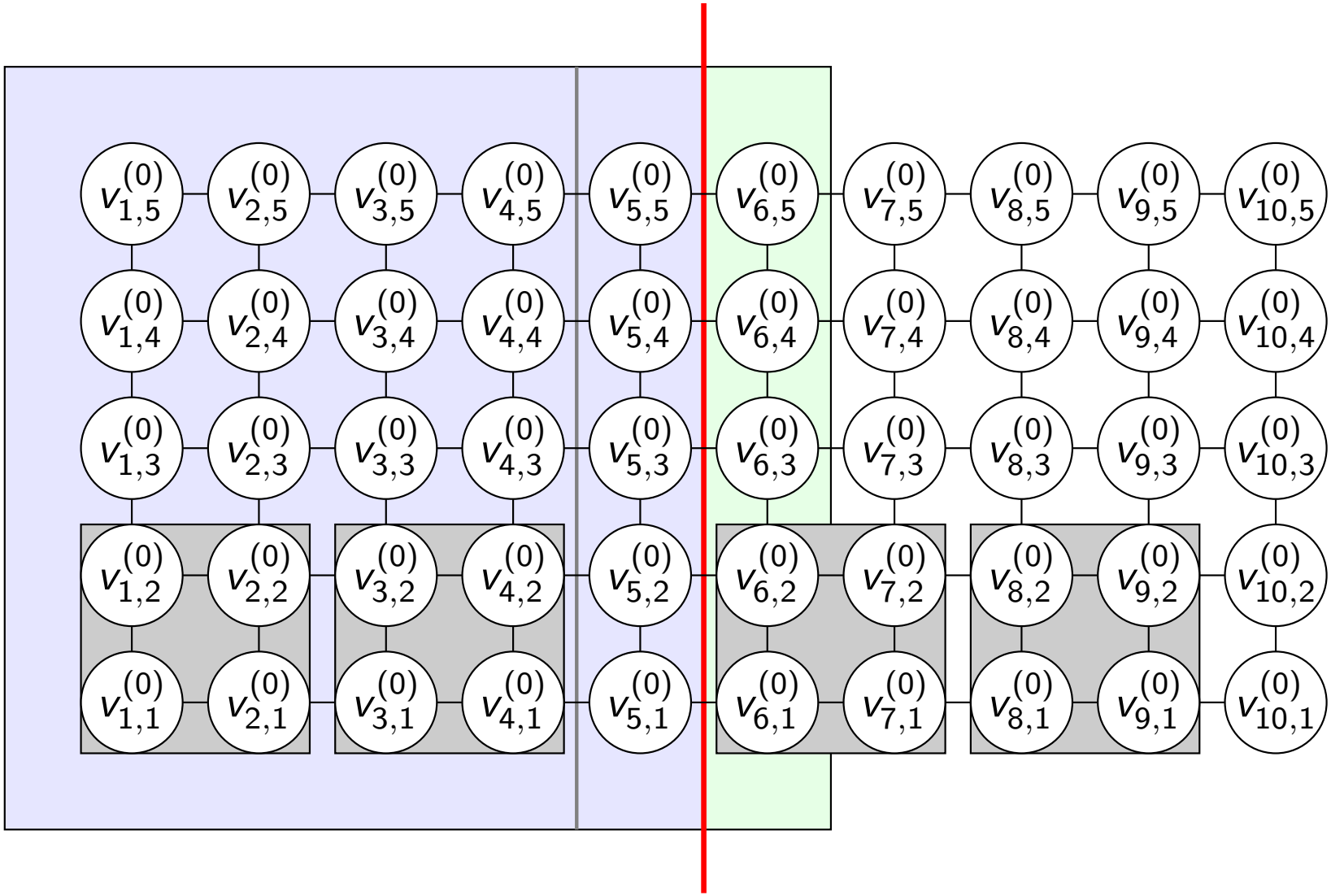
Parallel Agglomeration



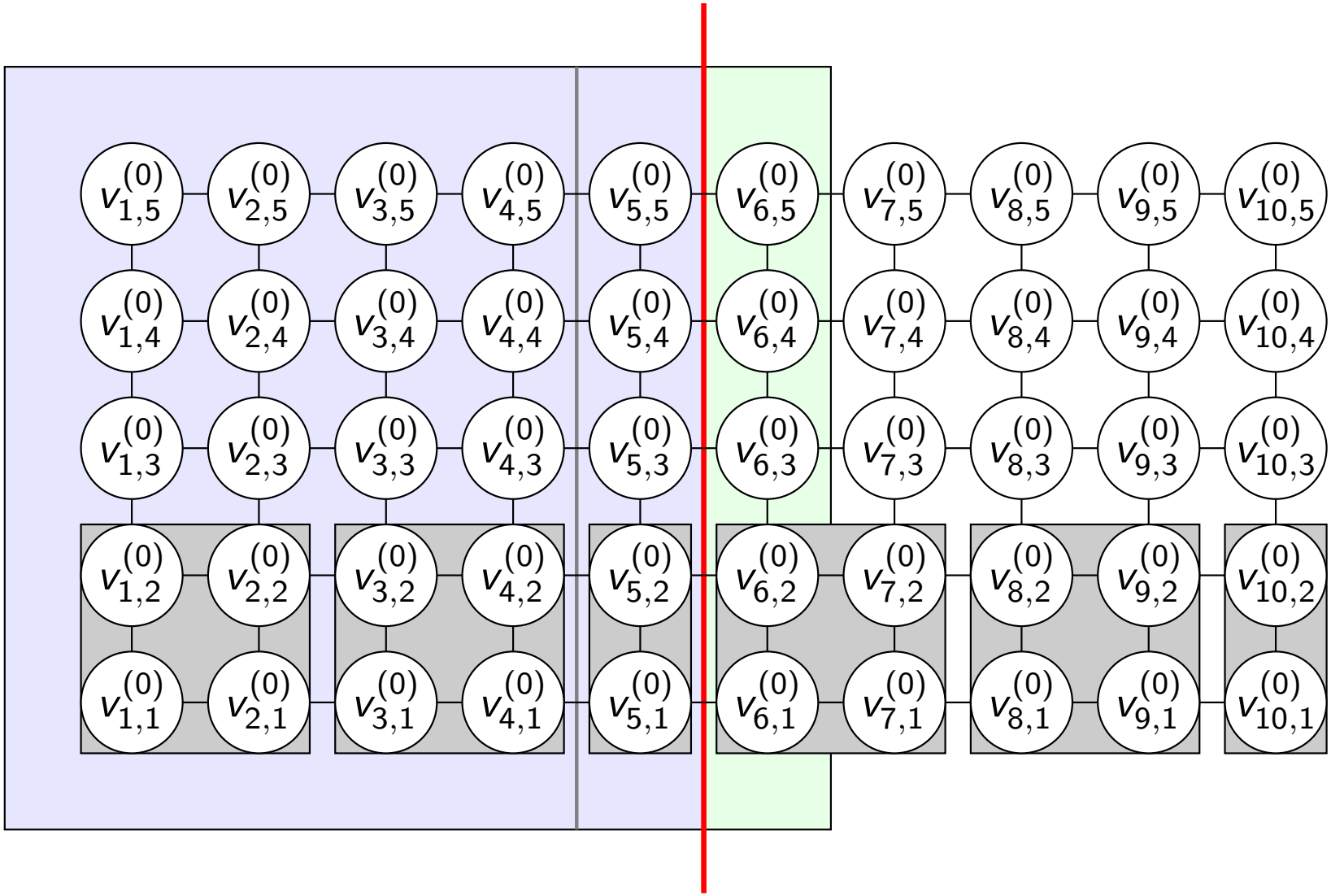
Parallel Agglomeration



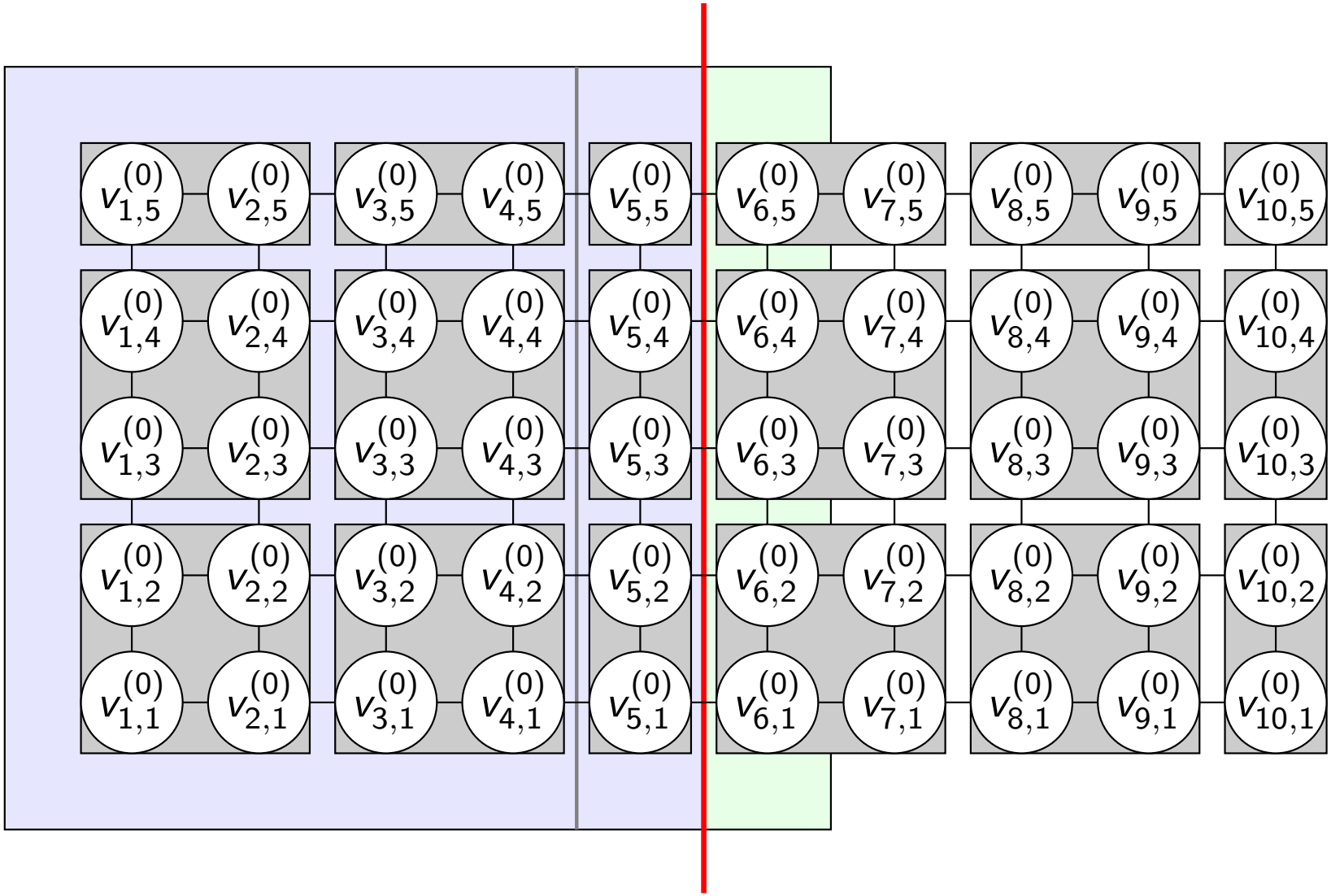
Parallel Agglomeration



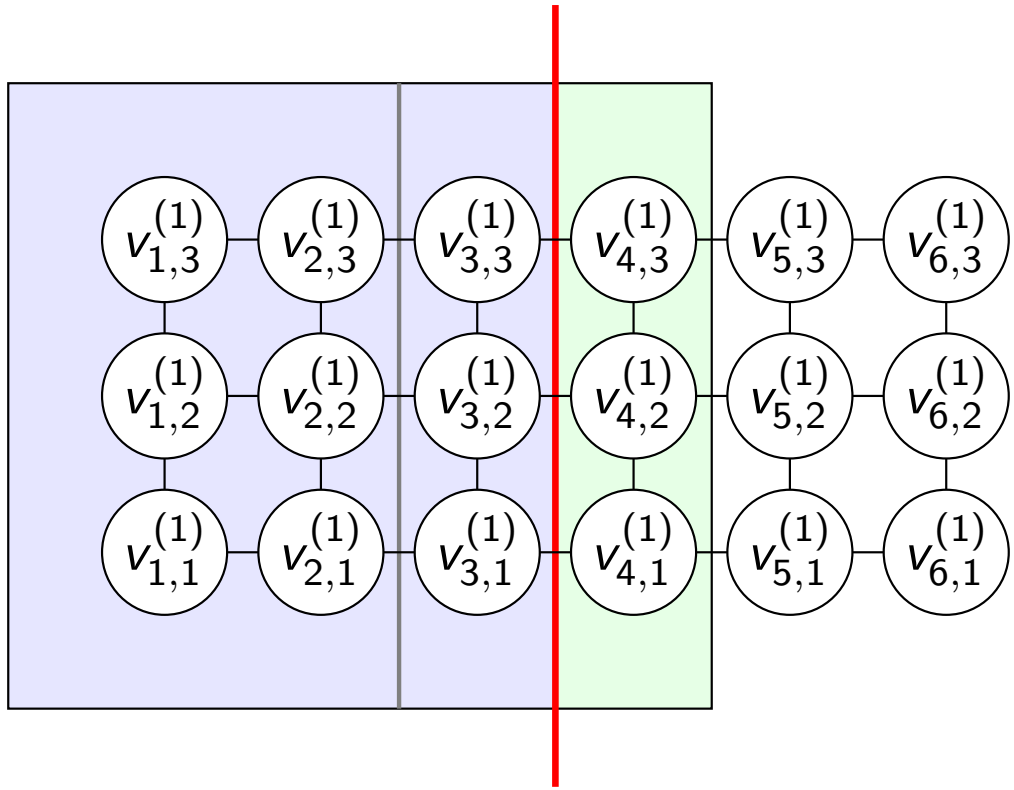
Parallel Agglomeration



Parallel Agglomeration



Parallel Agglomeration



AMG Weak Scaling Results

- AAMG and results presented is work of **Markus Blatt**
- BlueGene/P at Jülich Supercomputing Center
- $P \cdot 80^3$ degrees of freedom (5120^3 finest mesh), CCFV
- Poisson problem, 10^{-8} reduction
- AMG used as preconditioner in BiCGStab (2 V-Cycles!)
- **Scaling starts at 1**

procs	1/h	lev.	TB	TS	NIT	TIT	TT
1	80	5	19.86	31.91	8	3.989	51.77
8	160	6	27.7	46.4	10	4.64	74.2
64	320	7	74.1	49.3	10	4.93	123
512	640	8	76.91	60.2	12	5.017	137.1
4096	1280	10	81.31	64.45	13	4.958	145.8
32768	2560	11	92.75	65.55	13	5.042	158.3
262144	5120	12	188.5	67.66	13	5.205	256.2

Multiplicative AMG-DG Preconditioner

Require: x_F approximation of solution of DG system $A_F x_F = b_F$

Ensure: x_F improved approximation

$d_F \leftarrow b_F - A_F x_F$; $v_F \leftarrow 0$; ▷ Compute defect

Smooth(v_F, d_F, ν_1); ▷ Pre-smoothing on DG system

$d_C \leftarrow R_F(d_F - A_F v_F)$; ▷ Restrict defect to subspace

$v_C \leftarrow 0$; ▷ Prepare coarse level

AMG(v_C, d_C); ▷ Apply AMG V-Cycle

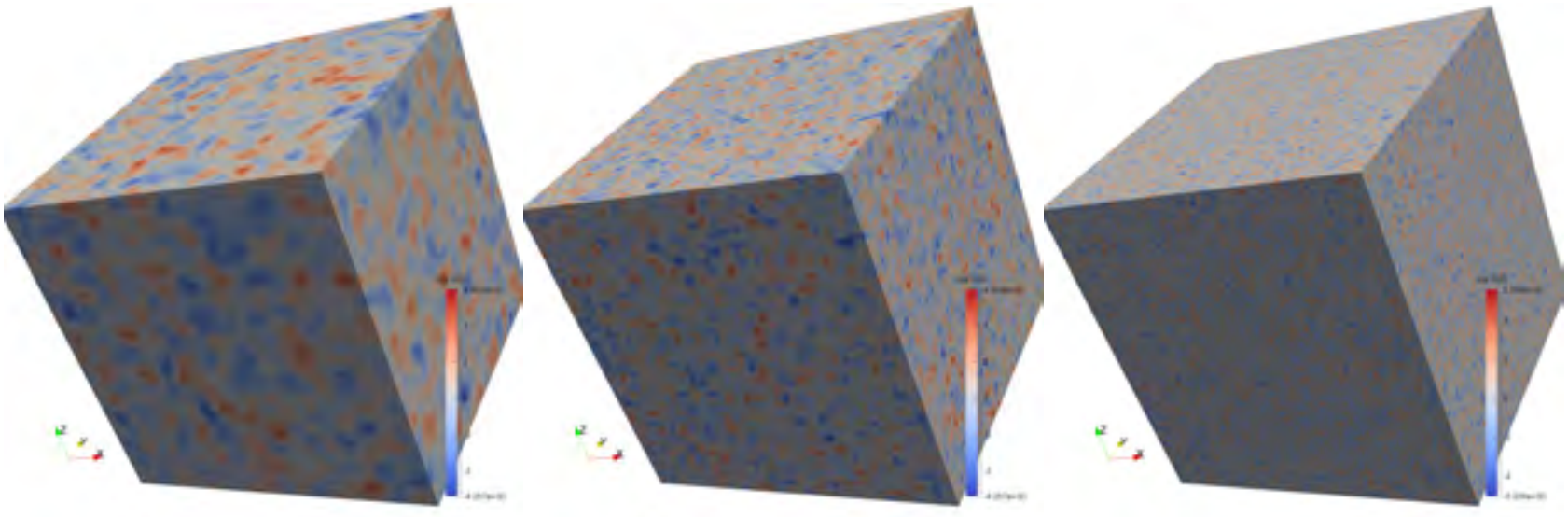
$v_F \leftarrow v_F + R_F^T v_C$ ▷ Add coarse grid correction

Smooth(v_F, d_F, ν_2); ▷ Post-smoothing on DG system

$x_F \leftarrow x_F + v_F$; ▷ Provide result

- R_F obtained from embedding $V_C \subset V_F = V_{DG}$
- V_C may be Q_1 conforming or P_0
- Method is not purely algebraic
- Smoothing step takes majority of computing time
- Smoothing step may be implemented matrix-free
- See *B., Blatt, Scheichl, NLAA, 2012*

AMG Weak Scaling Results: Heterogeneous Problems



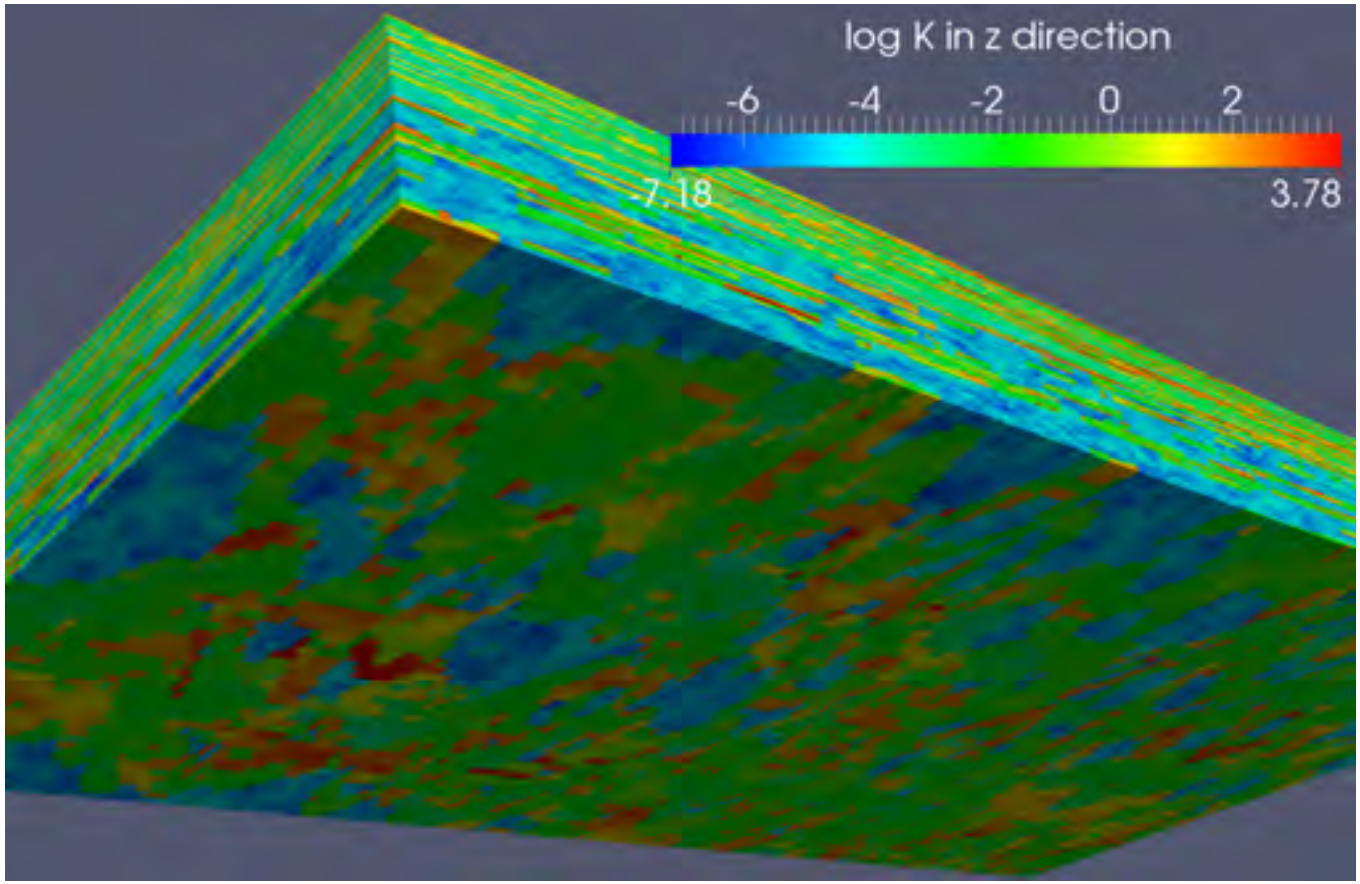
- Gaussian random field with Gaussian covariance function
- Picture: correlation length $2h$, variance 5
- 10 orders of magnitude variation
- Problem gets more difficult with refinement

AMG Weak Scaling Results: Hybrid DG-AMG

Problem	P	TIT	NIT	TIT	NIT
		Q1		DG1/Q1	
CL=$20h$	1	0.24	10	3.90	9
VAR=1	8	0.32	14	4.95	12
	64	0.35	16	5.19	15
	512	0.37	17	5.24	18
	4096	0.61	19	5.88	18
		CCFV		DG1/P0	
CL=$2h$	1	0.08	18	2.41	32
VAR=5	8	0.12	25	3.11	49
	64	0.14	25	3.30	55
	512	0.15	40	3.36	74
	4096	0.75	50	3.92	94

- $60^3 = 216000$ cells per core, 10^{-8} reduction
- Xeon CPU E5-2630 v3 @ 2.40GHz "Haswell", 8 cores
- 416 nodes, 2 CPUs per node

SPE10 Test Problem



- Domain $1200 \times 2200 \times 170$ ft^3
- $60 \times 220 \times 85$ mesh, 1.1 Mio cells
- Anisotropic mesh, anisotropic diagonal permeability tensor
- 11 orders of magnitude variation

SPE10 Results

- $-\nabla \cdot (K \nabla u) = 0$, Dirichlet on sides, Neumann top/bottom
- 10^{-6} reduction from zero initial value

Strong Scaling

		# cores			
Scheme	What	1	4	16	80
CCFV	NIT	23	24	27	27
	TIT	0.27891	0.08955	0.02714	0.00907
	Overall Speedup		3.0	8.8	26.1
DG1/P0	NIT	35	38	37	42
	TIT	10.058	3.026	1.001	0.289
	Overall Speedup		3.1	9.5	29.0

Iteration numbers for h refinement, $P = 80$

# cells	$1.1 \cdot 10^6$	$8.8 \cdot 10^6$	$70.4 \cdot 10^6$
CCFV	27	33	39
DG1/P0	42	54	66

Contents

Exploiting Tensor Product Structure

Tensor product basis: $\phi_i(x) = \theta_{i_d}(x_d) \dots \theta_{i_1}(x_1)$

Tensor product quadrature: $\Xi_j = (\xi_{j_1}, \dots, \xi_{j_d})^T$

where $i = (i_1, \dots, i_d) \in I^d$ and $j = (j_1, \dots, j_d) \in J^d$

Evaluate finite element function on ref elem at quadrature points:

$$
\begin{aligned}
u_h(\Xi_i) &= \sum_{j \in J^d} c_j \phi_j(\Xi_i) && (i \in I^d) \\
&= \sum_{j_d \in J} \dots \sum_{j_1 \in J} \theta_{j_d}(\xi_{i_d}) \dots \theta_{j_1}(\xi_{i_1}) \; c_{j_1,\dots,j_d} && \text{(tensor product struct.)} \\
&= \sum_{j_d \in J} \dots \sum_{j_1 \in J} A^{(d)}_{i_d,j_d} \dots A^{(1)}_{i_1,j_1} \; c_{j_1,\dots,j_d} && (A^{(k)} \in \mathbb{R}^{m \times n})
\end{aligned}
$$

Evaluation at all quadrature points is matrix-vector product:

$$y = Ax = (A^{(d)} \otimes \dots \otimes A^{(1)})\, x, \qquad A_{(i_1,\dots,i_d),(j_1,\dots,j_d)} = \Pi_{k=1}^{d} A^{(k)}_{i_k,j_k}$$

Same structure obtained for BLF evaluation: $a(u_h, \phi_i)$, $i \in I^d$

Sum Factorization

Extract common factors:

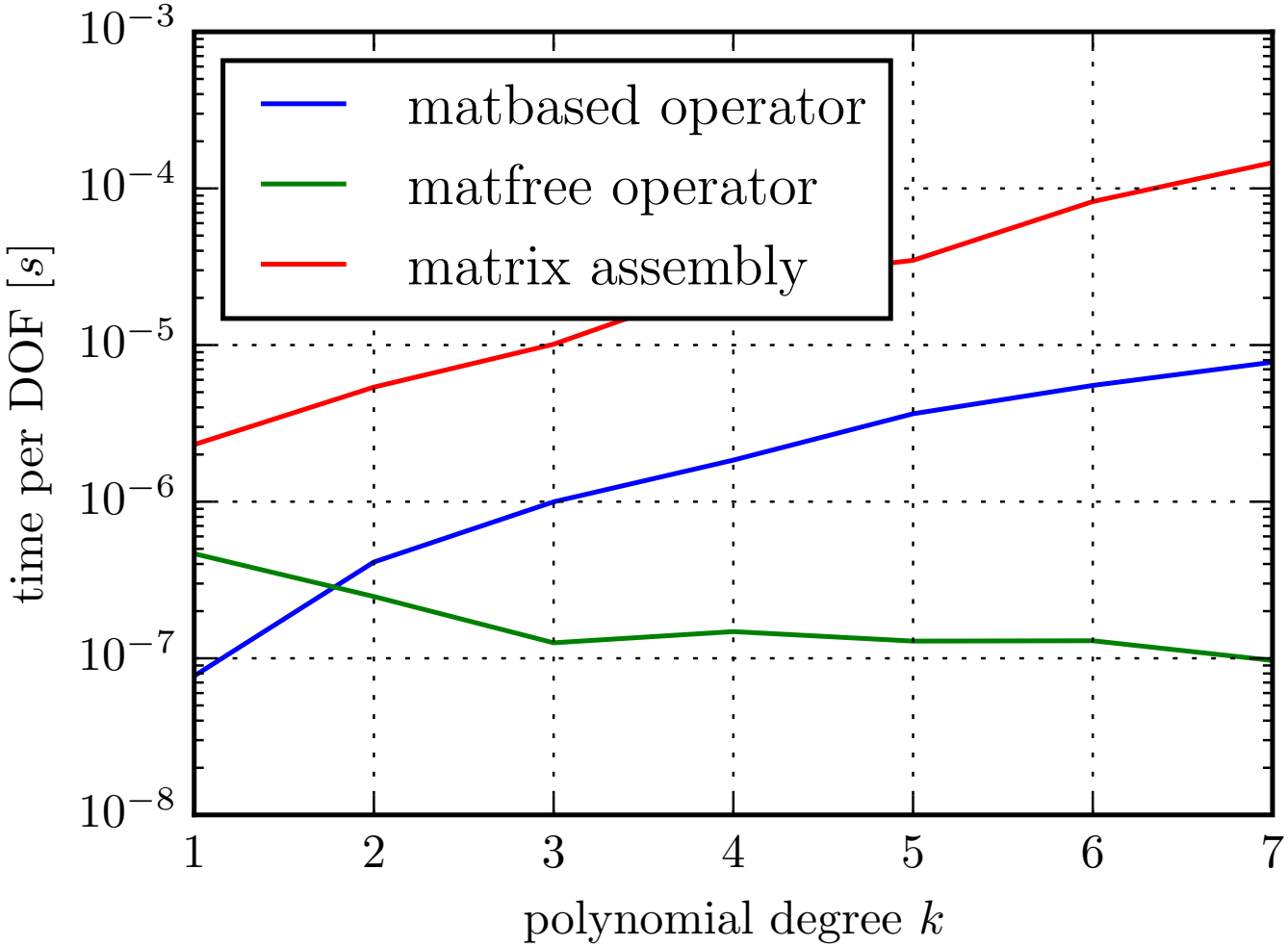
$$
\begin{aligned}
y_{i_1,\ldots,i_d} &= \sum_{j_d \in J} \cdots \sum_{j_1 \in J} A^{(d)}_{i_d,j_d} \cdots A^{(1)}_{i_1,j_1} x^{(0)}_{j_1,\ldots,j_d} \\
&= \sum_{j_d \in J} \cdots \sum_{j_2 \in J} A^{(d)}_{i_d,j_d} \cdots A^{(2)}_{i_2,j_2} \underbrace{\left(\sum_{j_1 \in J} A^{(1)}_{i_1,j_1} x^{(0)}_{j_1,\ldots,j_d} \right)}_{x^{(1)}_{i_1,j_2,\ldots,j_d}} \\
&= \sum_{j_d \in J} \cdots \sum_{j_3 \in J} A^{(d)}_{i_d,j_d} \cdots A^{(3)}_{i_3,j_3} \underbrace{\left(\sum_{j_2 \in J} A^{(2)}_{i_2,j_2} x^{(1)}_{i_1,j_2,\ldots,j_d} \right)}_{x^{(2)}_{i_2,i_1,j_3,\ldots,j_d}} \\
&= \ldots = \sum_{j_d \in J} A^{(d)}_{i_d,j_d} x^{(d-1)}_{i_{d-1},\ldots,i_1,j_d}
\end{aligned}
$$

Do this for all $(i_1, \ldots, i_d)$ simultaneously

Sum Factorization for Finite Elements

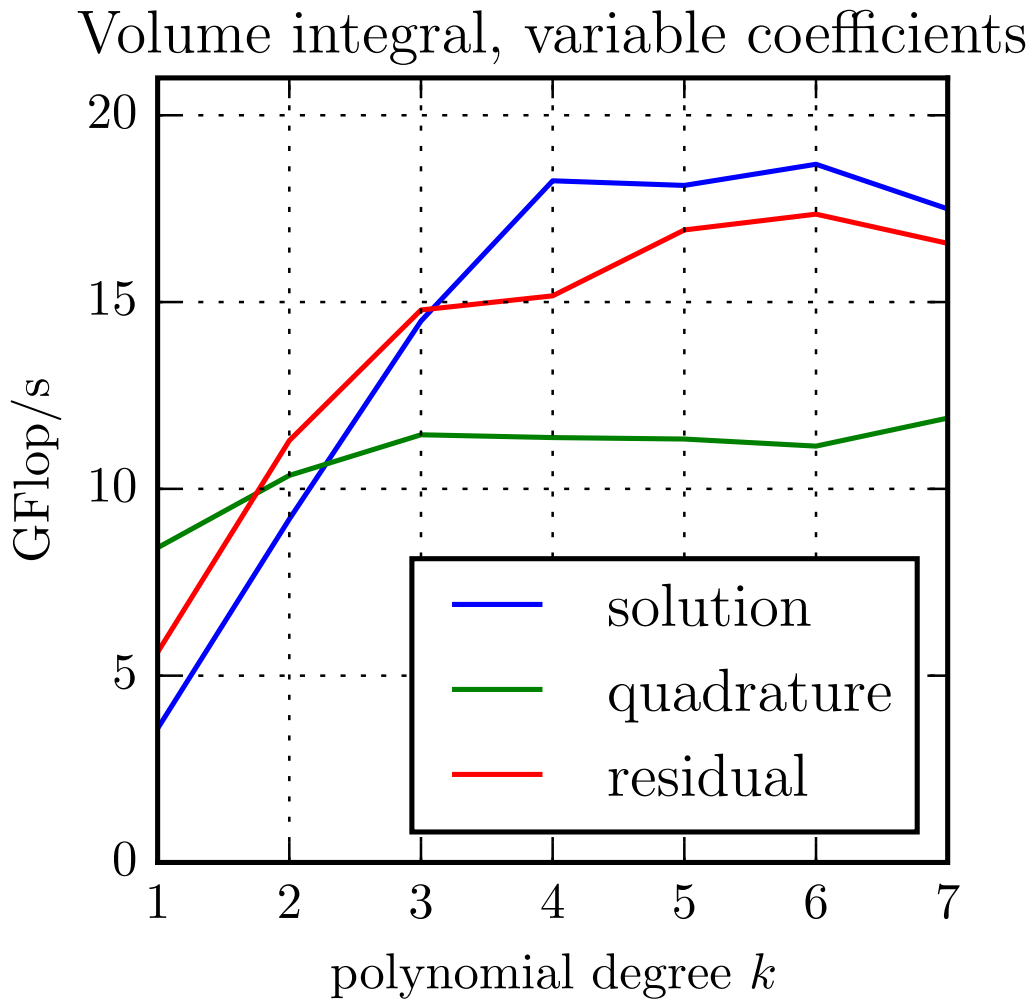
- Sum factorization algorithm by *Buis, Dyksen (1996)*:
- For $k = 1, \dots d$ do („*dim by dim approach*")
  - compute **matrix-matrix product** $X^{(k)} = A^{(k)} X^{(k-1)}$
  - "rotate" $X^{(k)}$ to make j_{k+1} the row index (transposition in 2d)
- Complexity reduction ($n = m = p + 1$):
  - $O(p^{2d}) \rightarrow O(p^{d+1})$ for operator application (mat-vec)
  - $O(p^{3d}) \rightarrow \ O(p^{2d+1})$ for Jacobian assembly
- Small matrices, e.g. $(4 \times 4) \cdot (4 \times 16)$ for $\mathbb{Q}_3$ in 3d
- FE operator evaluation, by e.g. *Melenk, Gerdes, Schwab (2001)*, *Kronbichler, Korman (2012)*:
  1. compute $\hat{u}_h$, $\nabla \hat{u}_h$ at quadrature points, $O(p^{d+1})$
  2. compute coefficients, geometry factors at quad. points, $O(p^d)$
  3. compute residuals $O(p^{d+1})$
- Applicable to nonlinear problems
- Applicable to non-affine transformations
- In DG face terms are dominating: $O(p^d)$ complexity observed

Time per DOF Single Core



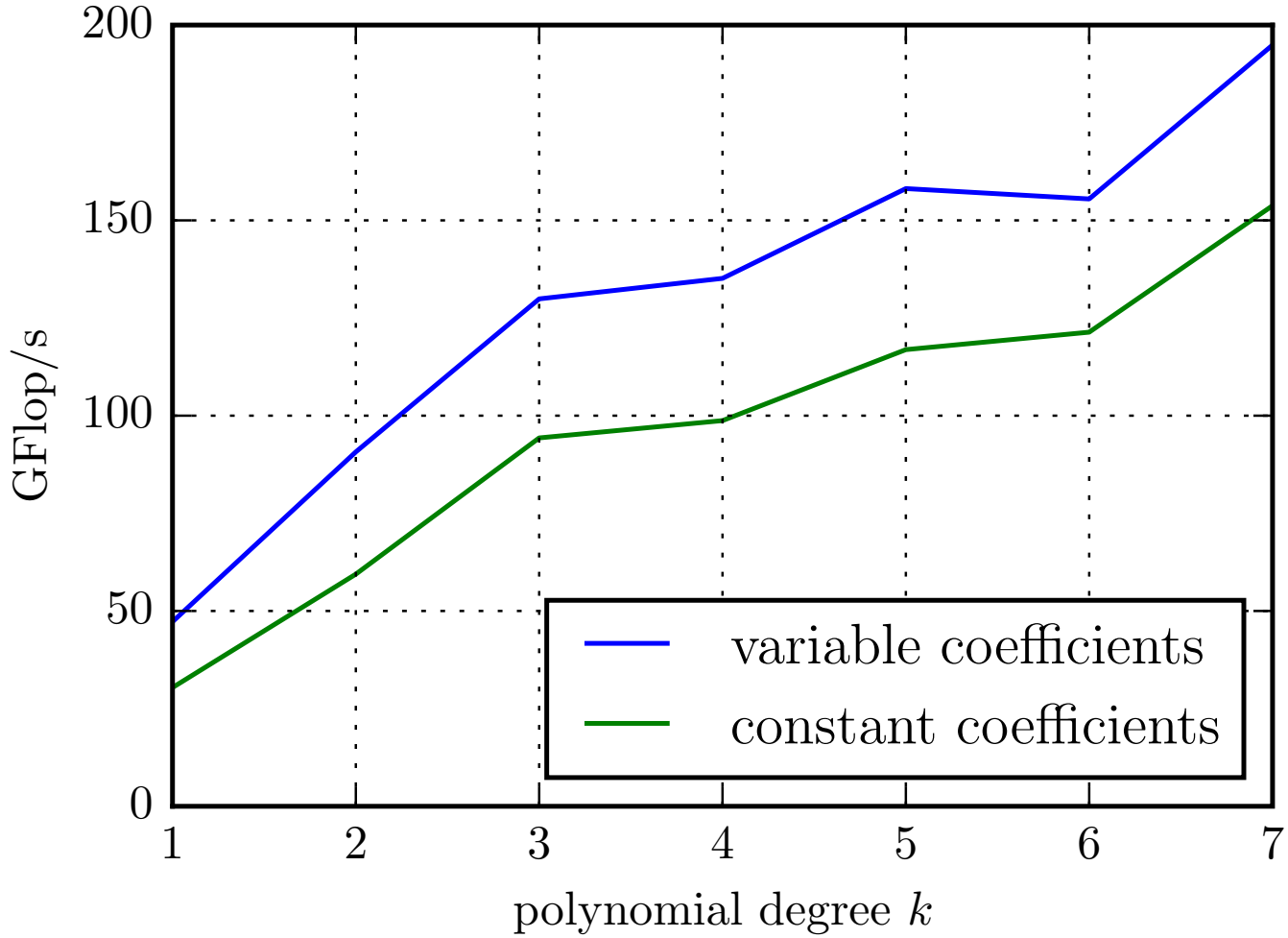
- **All results by Steffen Müthing**
- SIPG for 3d stationary convection-diffusion operator
- Matrix-based uses blocked SELL-C-σ matrix format
- Matrix-assembly is sum-factorized!

GFlop Rates for Volume Term



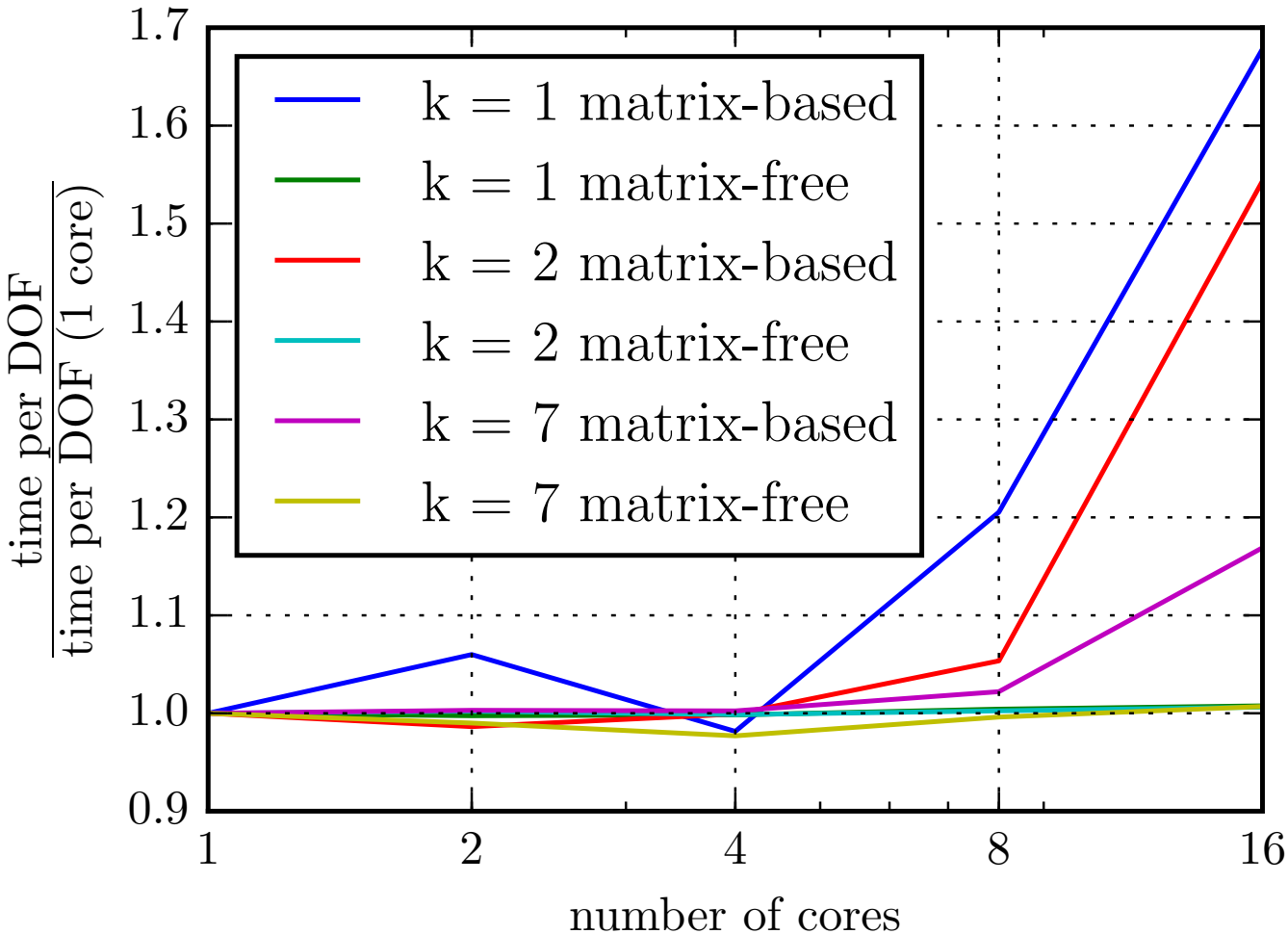
- SIPG for 3d stationary convection-diffusion operator
- Xeon E5-2698v3 (Haswell), Peak $\approx$ 40 GFlop/s per core
- Variable diffusion tensor and velocity field

GFlop Rates on One Socket



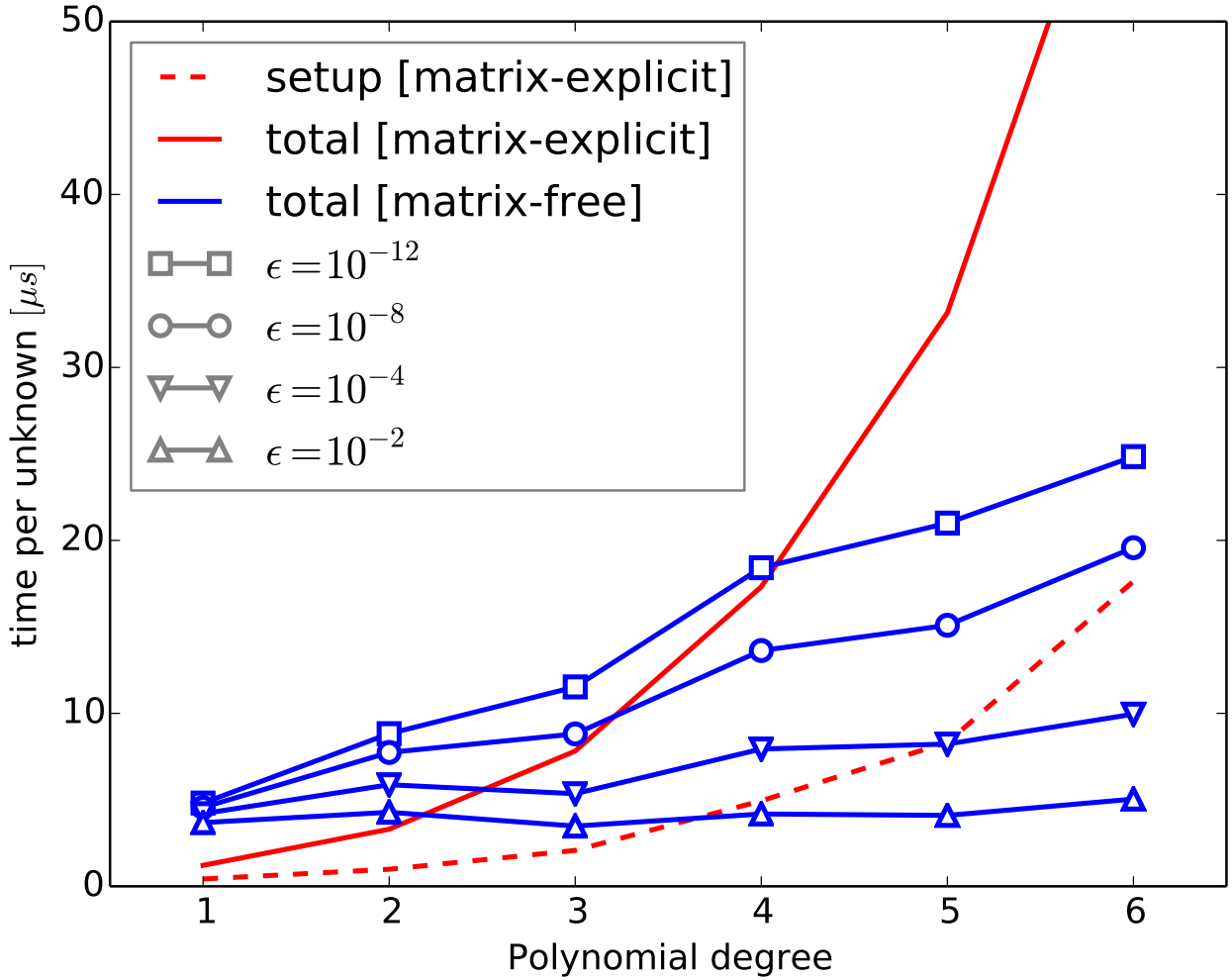
- SIPG for 3d stationary convection-diffusion operator
- Xeon E5-2698v3 16 cores (Haswell), Peak $\approx$ 650 GFlop/s
- Variable diffusion tensor and velocity field

Shared Memory Scalability



- Weak scaling: Constant #DOFs per core
- Xeon E5-2698v3 16 cores (Haswell)
- Matrix-based uses blocked SELL-C-σ matrix format

Matrix-free Block-Jacobi



- **All results by Eike Müller**
- SIPG for 3d stationary convection-diffusion equation

Contents

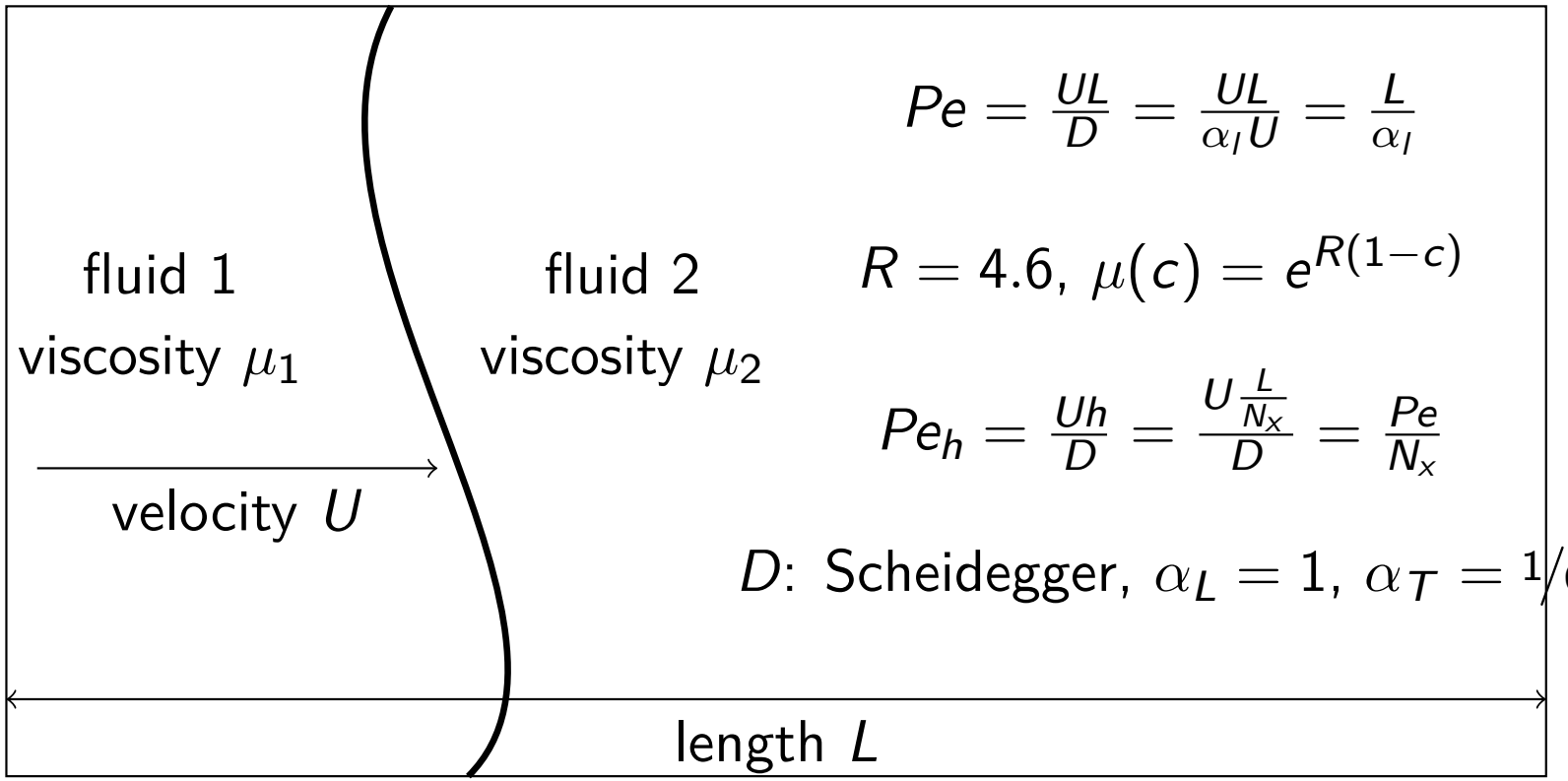
Miscible Displacement: Governing Equations

$$\nabla \cdot v = 0, \qquad v = -\frac{K}{\mu(c)}(\nabla p - \rho g)$$

$$\partial_t(\Phi c) + \nabla \cdot (cv - D(v)\nabla c) = 0$$

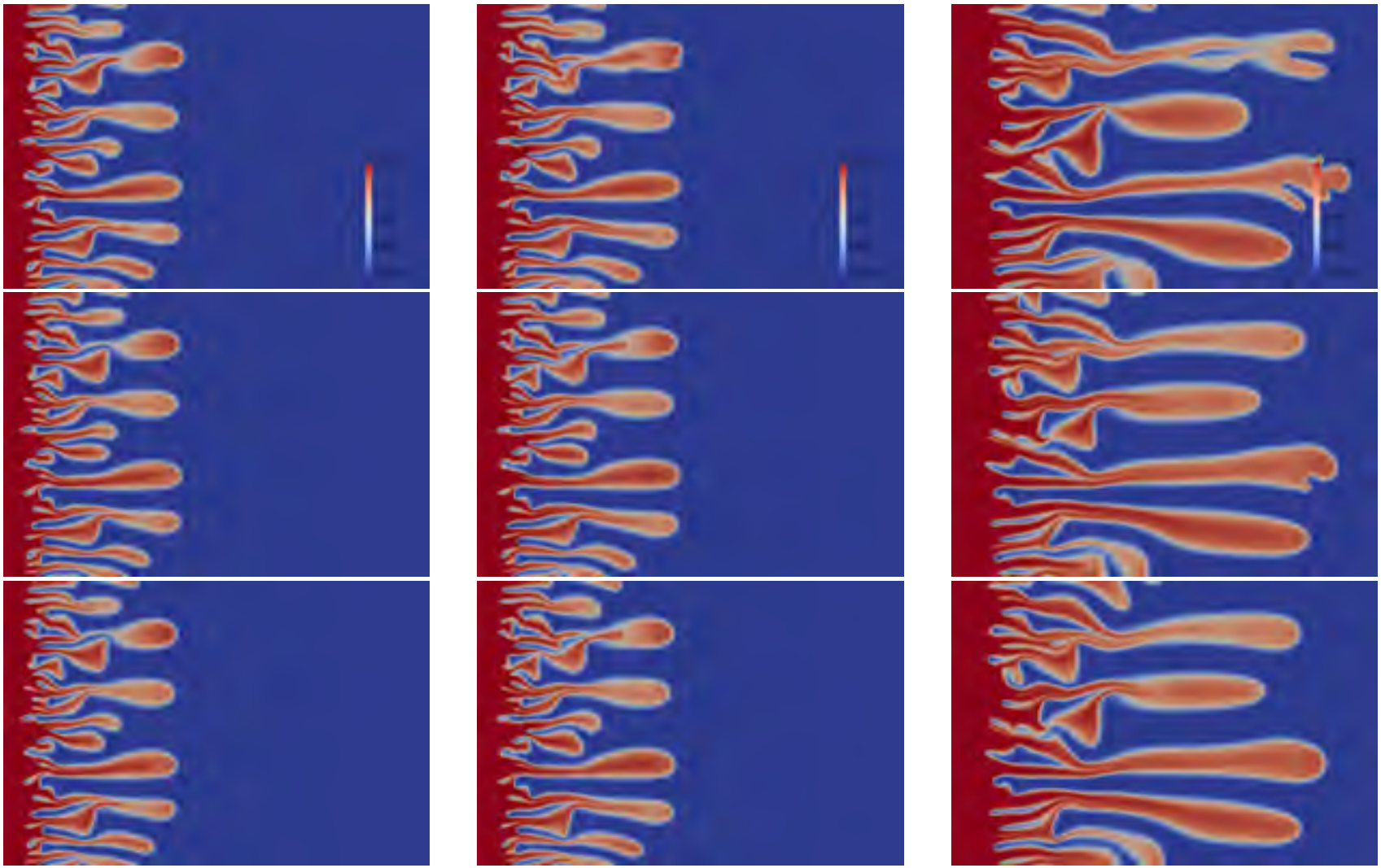
- Application e.g. in polymer flooding
- Unstable for less viscous fluid displacing more viscous fluid
- Decoupled implicit pressure / implicit concentration algorithm

Miscible Displacement: Governing Parameters



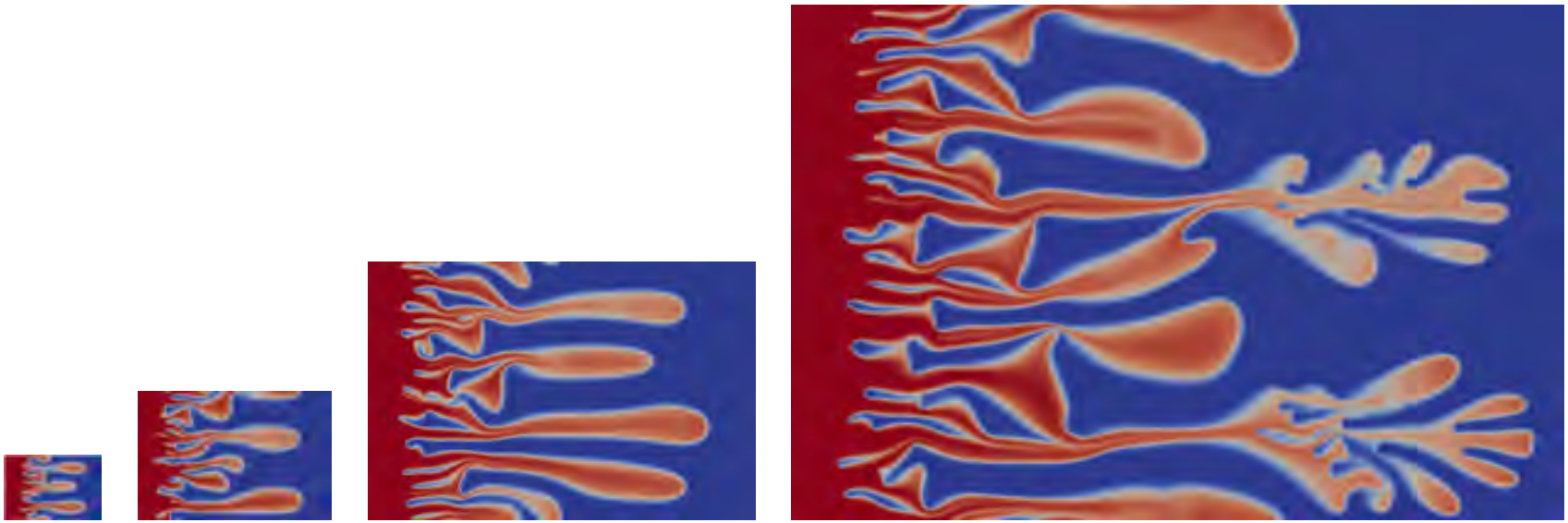
- Unstable flow for $R > 0$
- Need $Pe_h \approx 1$ to avoid unphysical oscillations
- Aspect ratio $A = 3 : 2$, periodic in y

Grid Convergence, $\mathbf{Pe} = \mathbf{1800}, \mathbf{R} = \mathbf{4.6}$



Top to bottom: $Q_1/Q_1, Pe_h = 2,\ Q_1/Q_1, Pe_h = 1,\ Q_2/Q_2, Pe_h = 2$
(All DG methods)

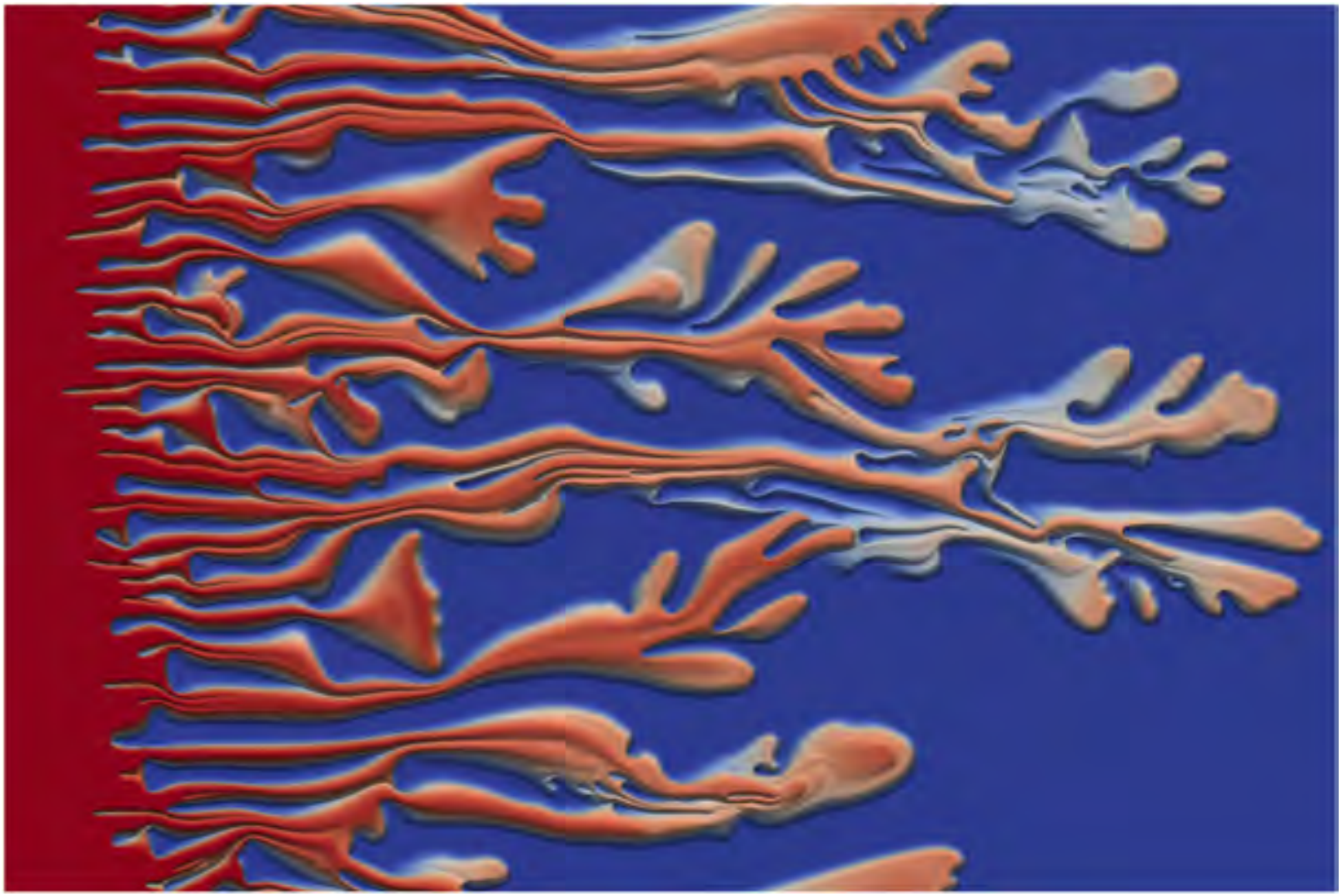
Increasing Peclet Number



$Pe = 450$ $Pe = 900$ $Pe = 1800$ $Pe = 3600$

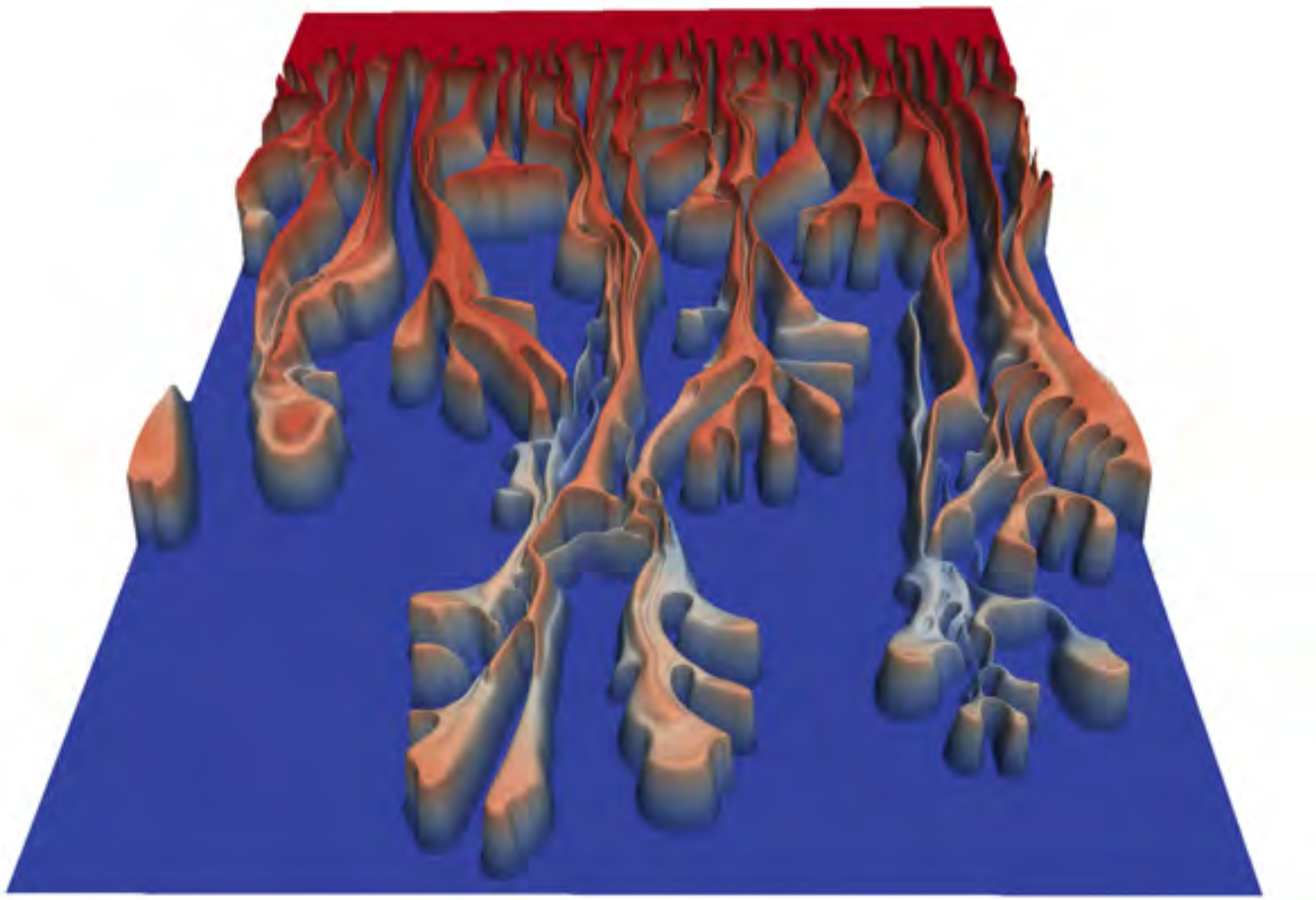
- Increasing Pe corresponds to increasing L and time
- Initially, fingers compete and merge, leading to spikes running on top of existing fingers
- Tip splitting starts at $\approx Pe = 2000$
- Interaction of splitted fingers becomes increasingly complex

$\mathbf{Pe} = \mathbf{7200}$ Top View



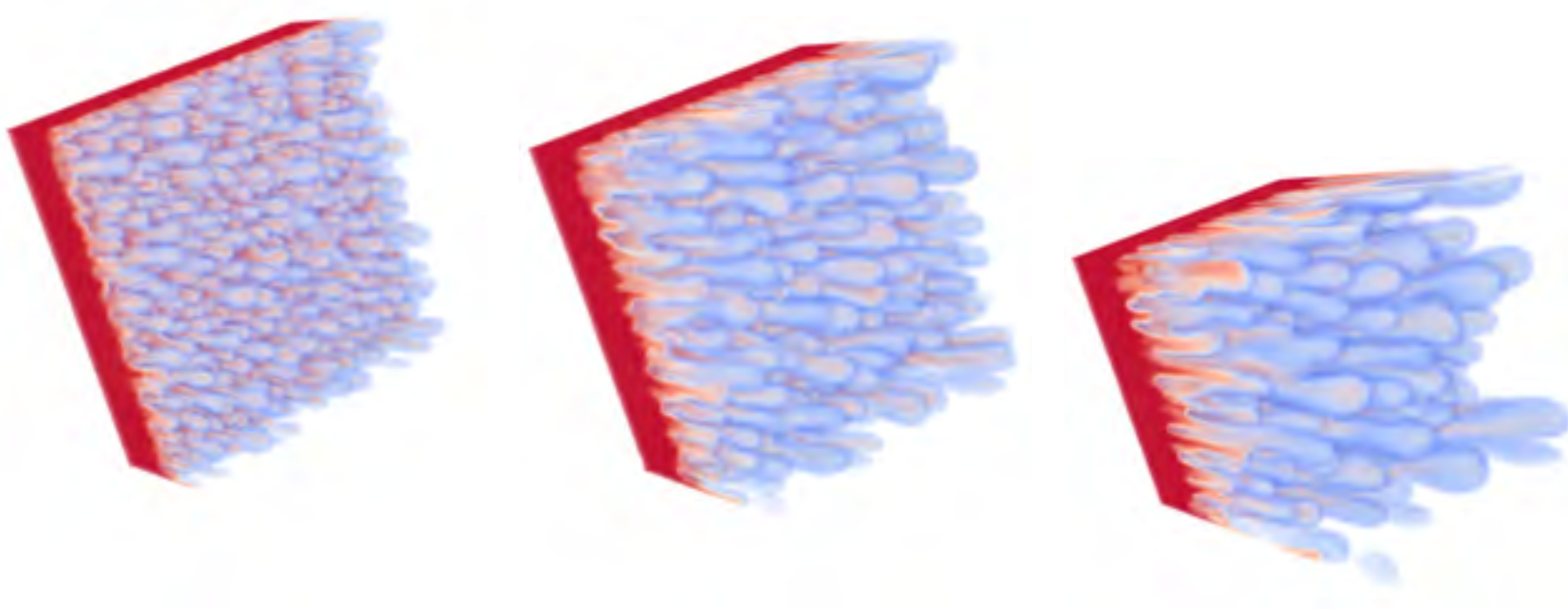
128 $\times$ Xeon E5-2630 v3 8 core, $155 \cdot 10^6$ DOF, Alexander-2, 17000 time steps, 32h

$\mathbf{Pe} = \mathbf{7200}$ Warped View



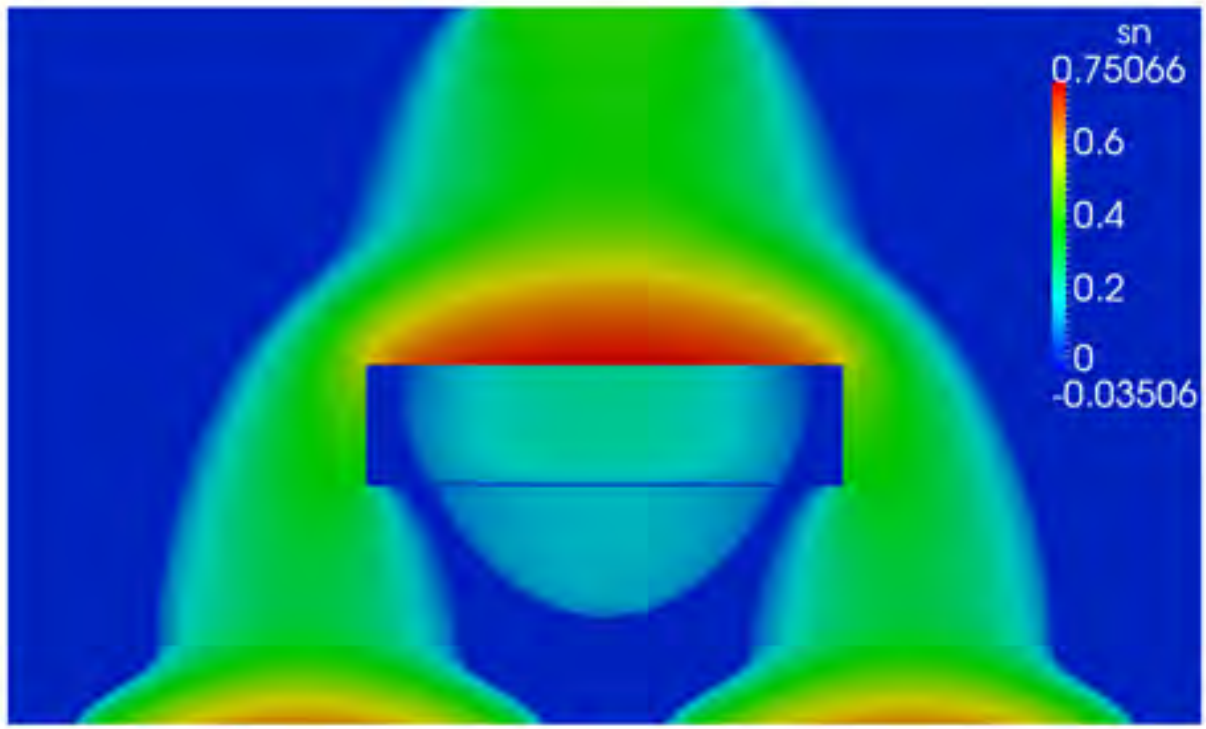
128 $\times$ Xeon E5-2630 v3 8 core, $155 \cdot 10^6$ DOF, Alexander-2, 17000 time steps, 32h

First 3d Results for **Pe = 300**



8 × Xeon E5-2680 v3 10 core, $243 \cdot 10^6$ DOF, Alexander-2

A DNAPL Infiltration Model Problem

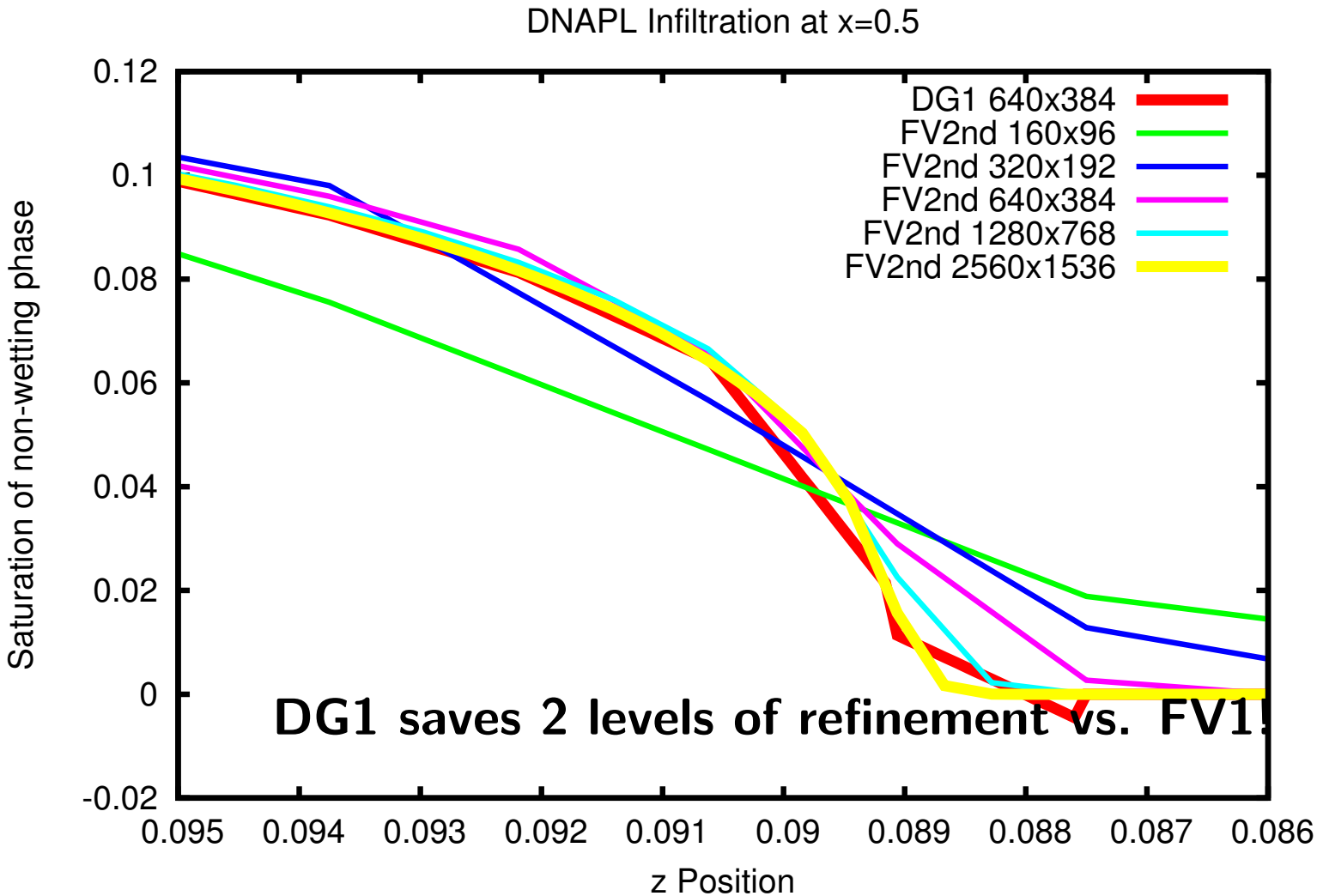


- $k_{r\alpha}$: quadratic
- p_c: Brooks-Corey, $\lambda = 2$
  $p_e^{bulk} = 755$,
  $p_e^{lense} = 1163$

- Fully implicit fully coupled
- DG1 / Alexander(2)
- CCFV0: full upw., impl. Euler
- CCFV1: central, Alex.(2)

At Free Boundary

DG1 640x384 vs. FV1



Weak Scalability for DNAPL Infiltration

$T = [0, 3600s]$, DG1/Alexander(2) vs. FV1/Alexander(2)

	P	h^{-1}	DOF	TS	ANL	ALIN	TT	LTIT
DG1	1	40	$7.7 \cdot 10^3$	30	11.4	1.8	156.8	0.144
	4	80	$3.1 \cdot 10^4$	71	11.0	3.4	466.1	0.052
	16	160	$1.2 \cdot 10^5$	125	12.0	5.7	1187.3	0.062
	64	**320**	$\mathbf{4.9 \cdot 10^5}$	**263**	**11.4**	**9.1**	**3250.5**	**0.067**
	256	640	$2.0 \cdot 10^6$	563	10.7	15.3	11233.0	0.082
	1024	1280	$7.9 \cdot 10^6$	1369	9.2	17.0	41917.4	0.112
FV1	1	160	$3.1 \cdot 10^4$	60	7.7	1.8	191.2	0.043
	4	320	$1.2 \cdot 10^5$	120	8.2	2.2	592.1	0.069
	16	640	$4.9 \cdot 10^5$	246	8.4	2.6	1453.7	0.078
	64	**1280**	$\mathbf{2.0 \cdot 10^6}$	**491**	**8.9**	**3.3**	**6496.1**	**0.151**
	256	2560	$7.9 \cdot 10^6$	1021	9.7	4.5	12774.9	0.156

Accuracy(DG1) = Accuracy(FV1) on 2 times refined mesh

Summary

- Agglomeration Algebraic Multigrid is reasonably robust and scalable
- Sum factorization has reduced complexity at increased performance
- Hybrid AMG-DG preconditioner can overcome memory wall and improve scalability
- All methods presented in this talk have been implemented in the DUNE software framework (`www.dune-project.org`) as part of the **EXA-DUNE** project
- DUNE is also the basis of the *Open Porous Media Initiative* driven by StatOil (`http://opm-project.org`)

Thank you for your attention!