Exploring unstructured Poisson solvers for FDS

Dr. Susanne Kilian

hhpberlin - Ingenieure für Brandschutz
10245 Berlin - Germany

Agenda

1 Discretization of Poisson equation

2 Solvers for Poisson equation

3 Numerical Tests

4 Conclusions

Discretization of the Poisson equation

Structured versus unstructured Cartesian grids

Pressure equation in FDS

1

Discretization of Poisson equation

Elliptic partial differential equation of type „Poisson“

$$\nabla^2 \mathcal{H} = -\frac{\partial(\nabla \cdot \mathbf{u})}{\partial t} - \nabla \cdot \mathbf{F} \qquad + \qquad \text{Boundary conditions}$$

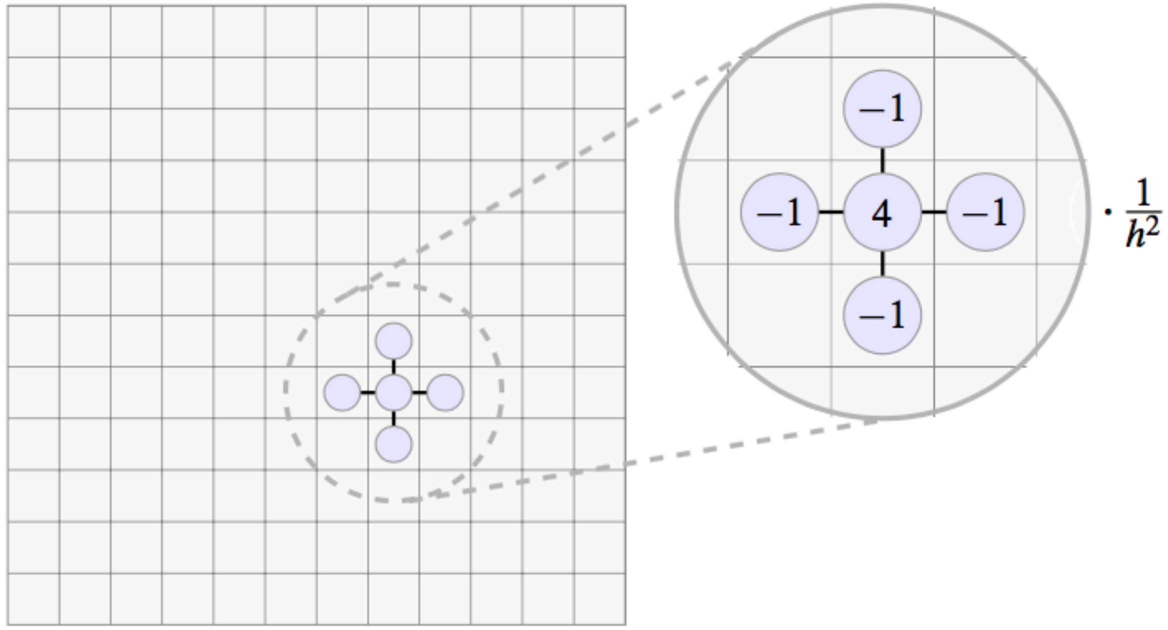
Source terms of previous time step
(radiation, combustion, etc.)

- must be solved at least twice per time step
- strongly coupled with velocity field

Finite difference discretization

Discretization stencil in 2D:

$$\frac{1}{h^2}\left(-\mathcal{H}_{i,k-1} - \mathcal{H}_{i-1,k} + 4\,\mathcal{H}_{i,k} - \mathcal{H}_{i,k+1} - \mathcal{H}_{i+1,k}\right) = R_{i,k}$$



- cell-centered
- specifies physical relations between single cells

Subdivision into meshes

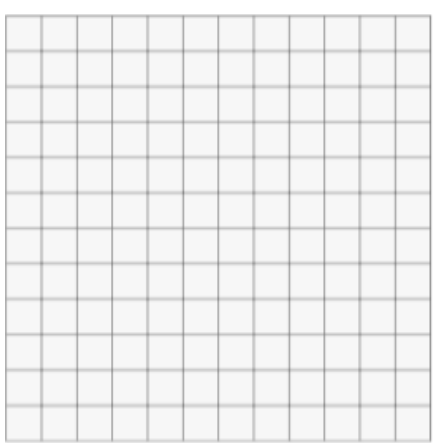
1

Discretization of Poisson equation

Single-Mesh:

1 global system of equations

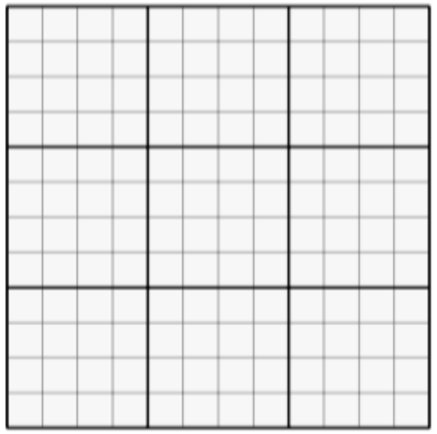
$$A x = b$$



Multi-Mesh:

M local systems of equations

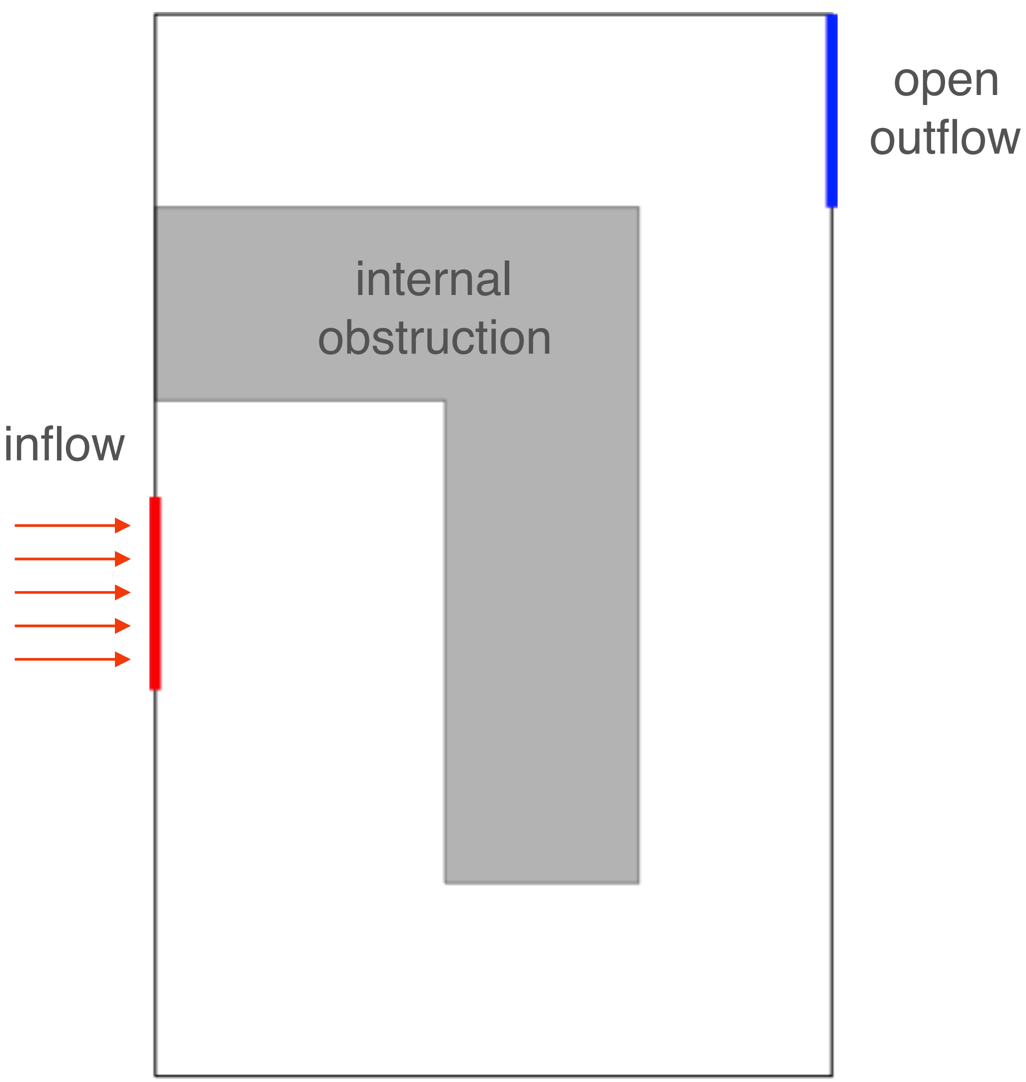
$$A_m x_m = b_m$$



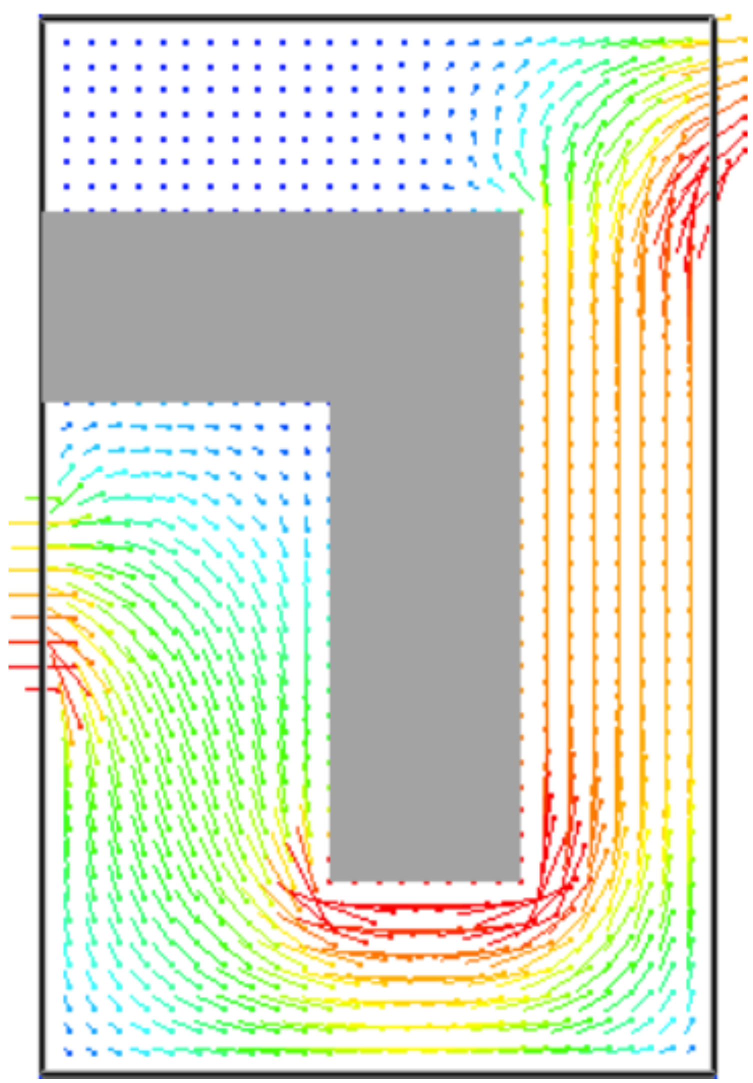
A and A_m are sparse matrices (only very few non-zeros entries)

Treatment of internal obstructions

Simple 2D-domain



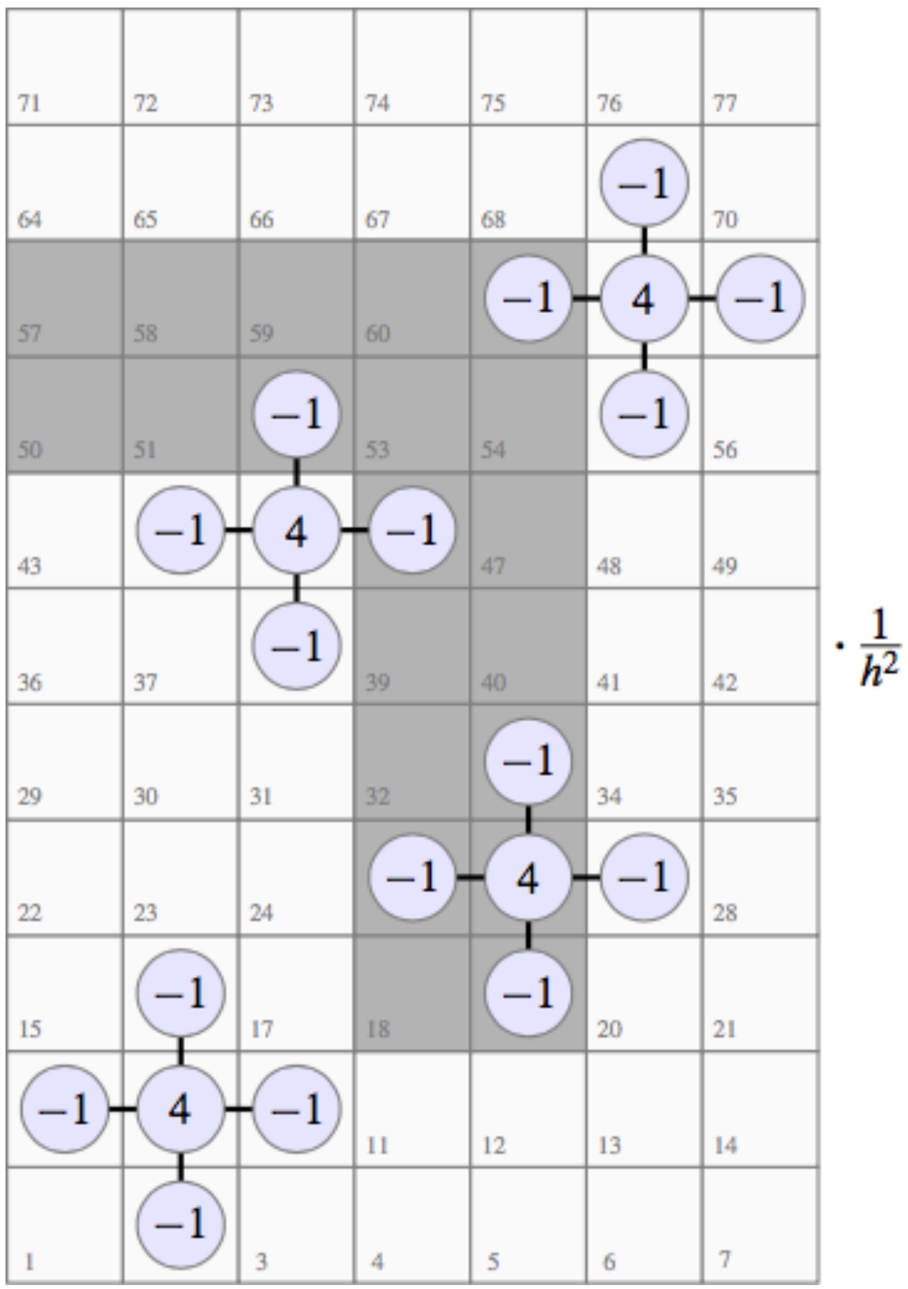
FDS velocity field



Structured Cartesian grids

1

Discretization of Poisson equation

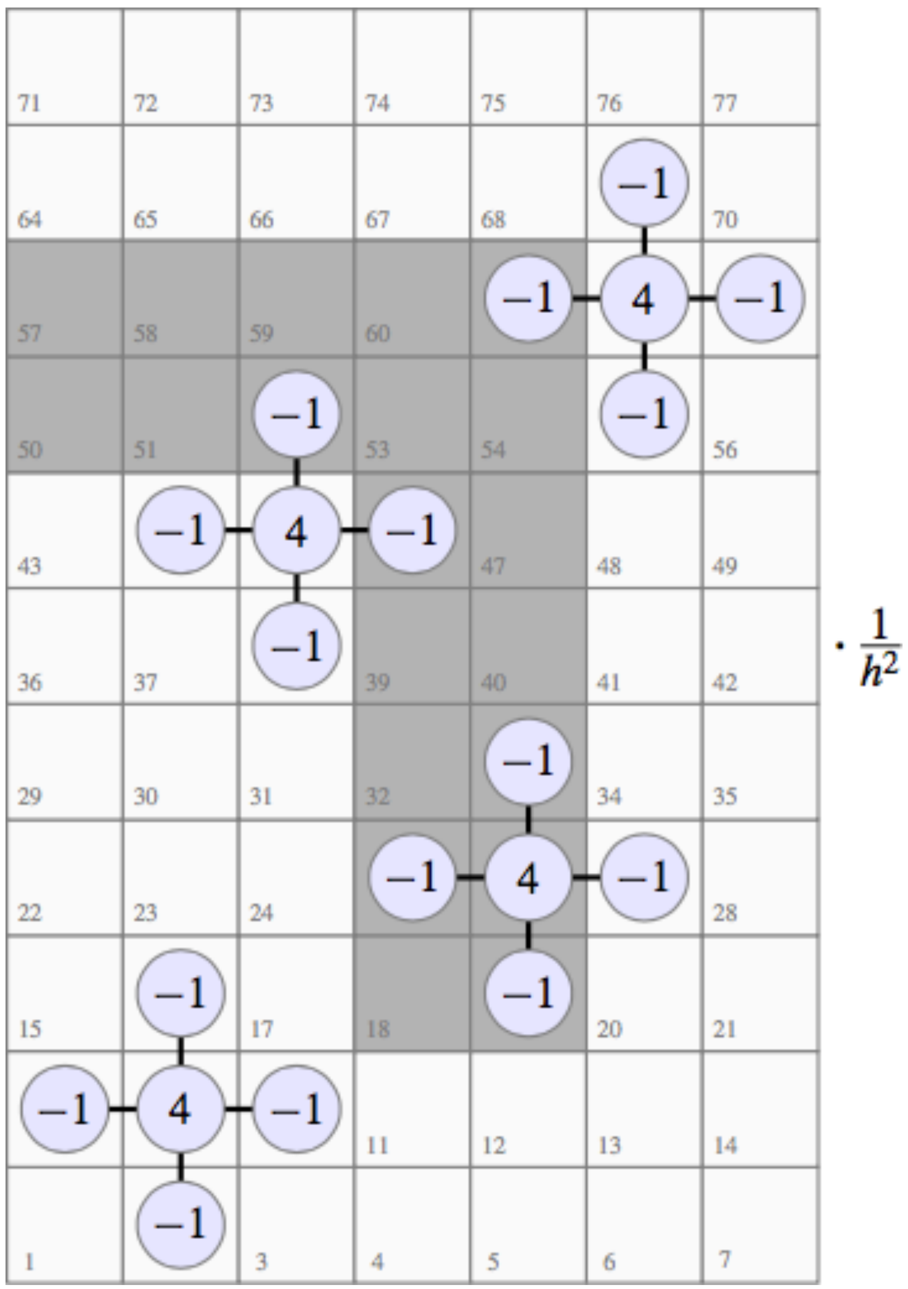


<u>„Gasphase“ **and** „Solid“-cells:</u>

- uniform matrix stencils regardless of inner obstructions
- cells interior to obstructions are part of system of equations

Matrix stencils don't care about obstructions

Structured Cartesian grids



<u>Advantages:</u>

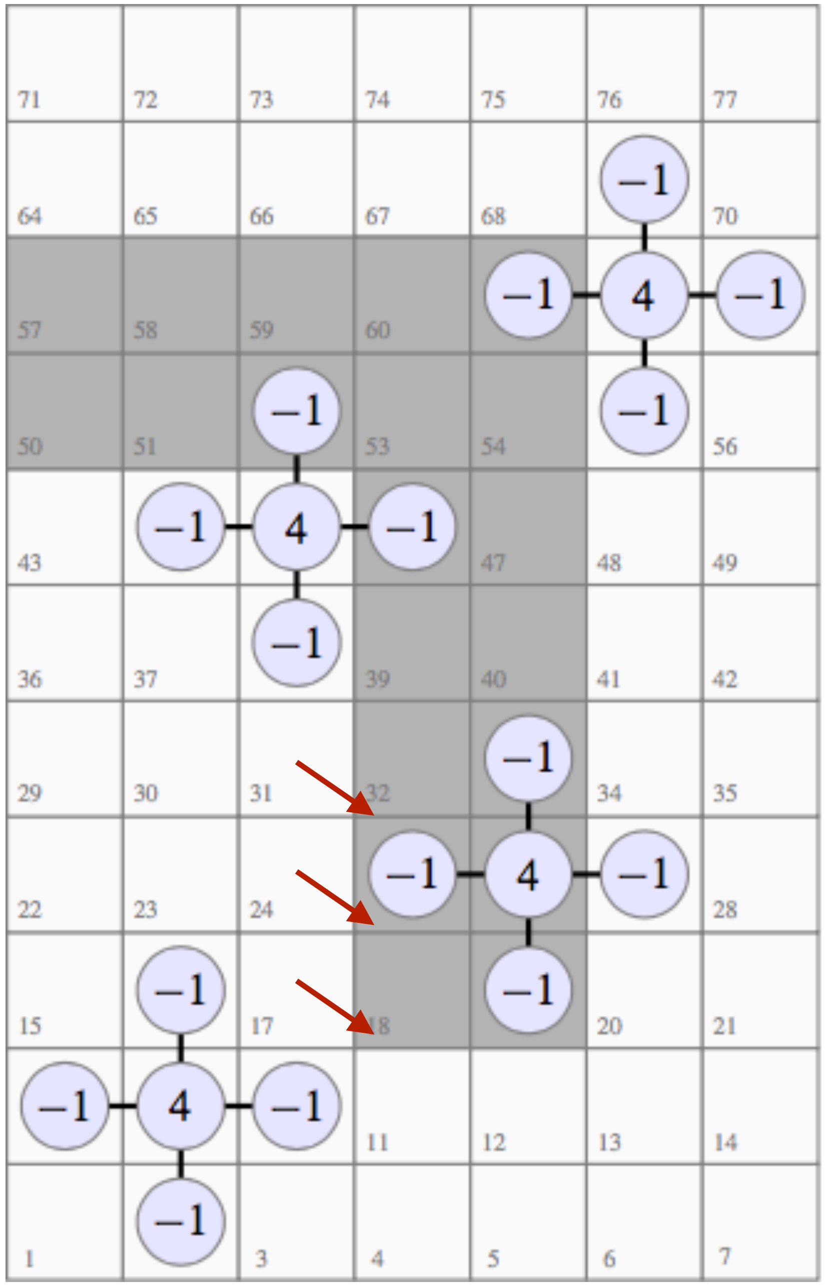
- very regular matrix structure (uniform numbering between neighboring cells)
- can be exploited efficiently in solution process (Example: FFT)

Use of highly optimized solvers possible

Structured Cartesian grids

1

Discretization of Poisson equation

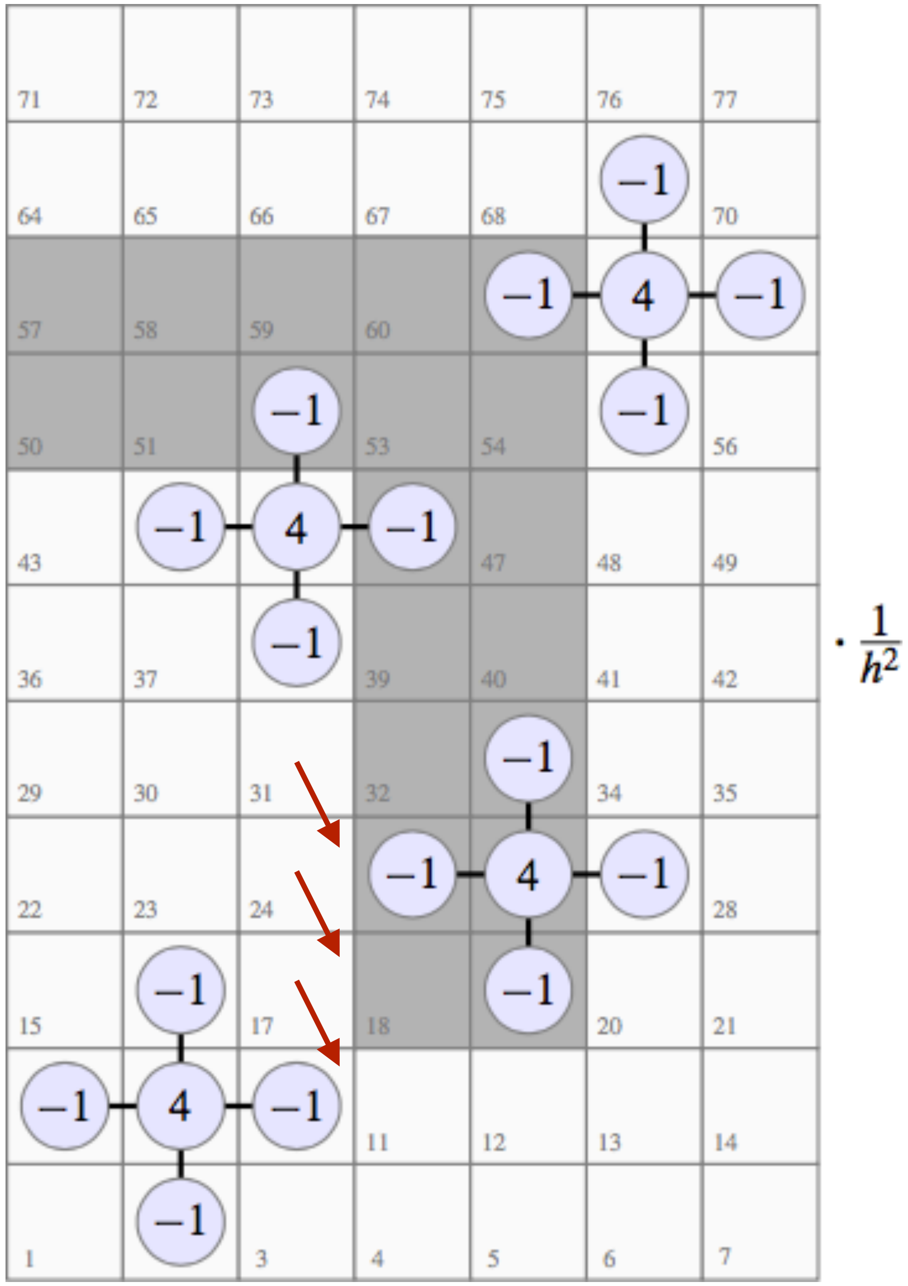


$\cdot \frac{1}{h^2}$

<u>Disadvantages:</u>

- incorrect treatment of interior boundaries
- possible penetration of velocity field into internal solids

Structured Cartesian grids

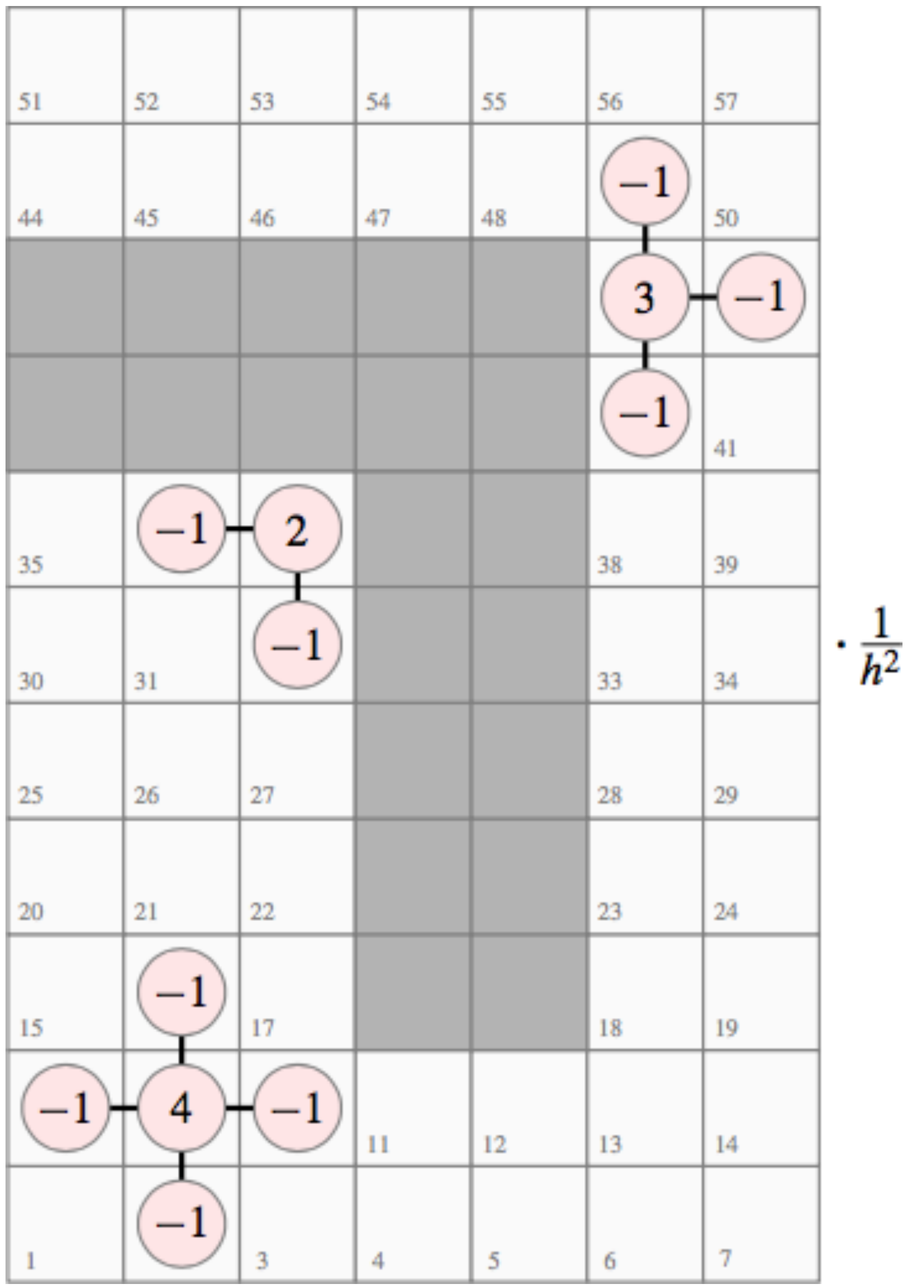


<u>Disadvantages:</u>

- incorrect treatment of interior boundaries
- possible penetration of velocity field into internal solids
- need of additional correction

Losses of efficiency and accuracy

Unstructured Cartesian grids

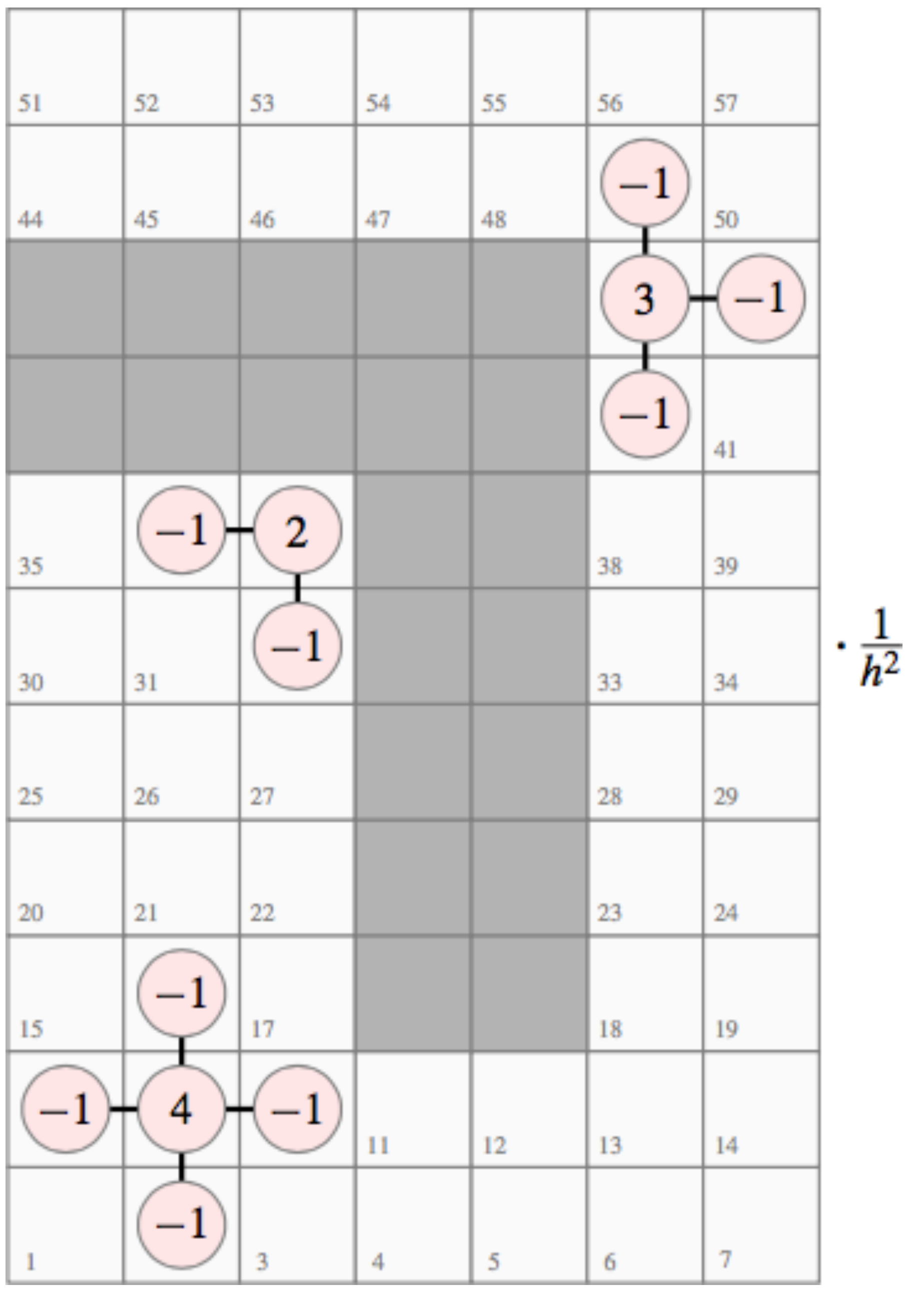


Only „Gasphase"-cells:

- individual matrix stencils by omitting internal obstructions
- cells interior to obstructions are not part of system of equations

Matrix stencils care about obstructions

Unstructured Cartesian grids

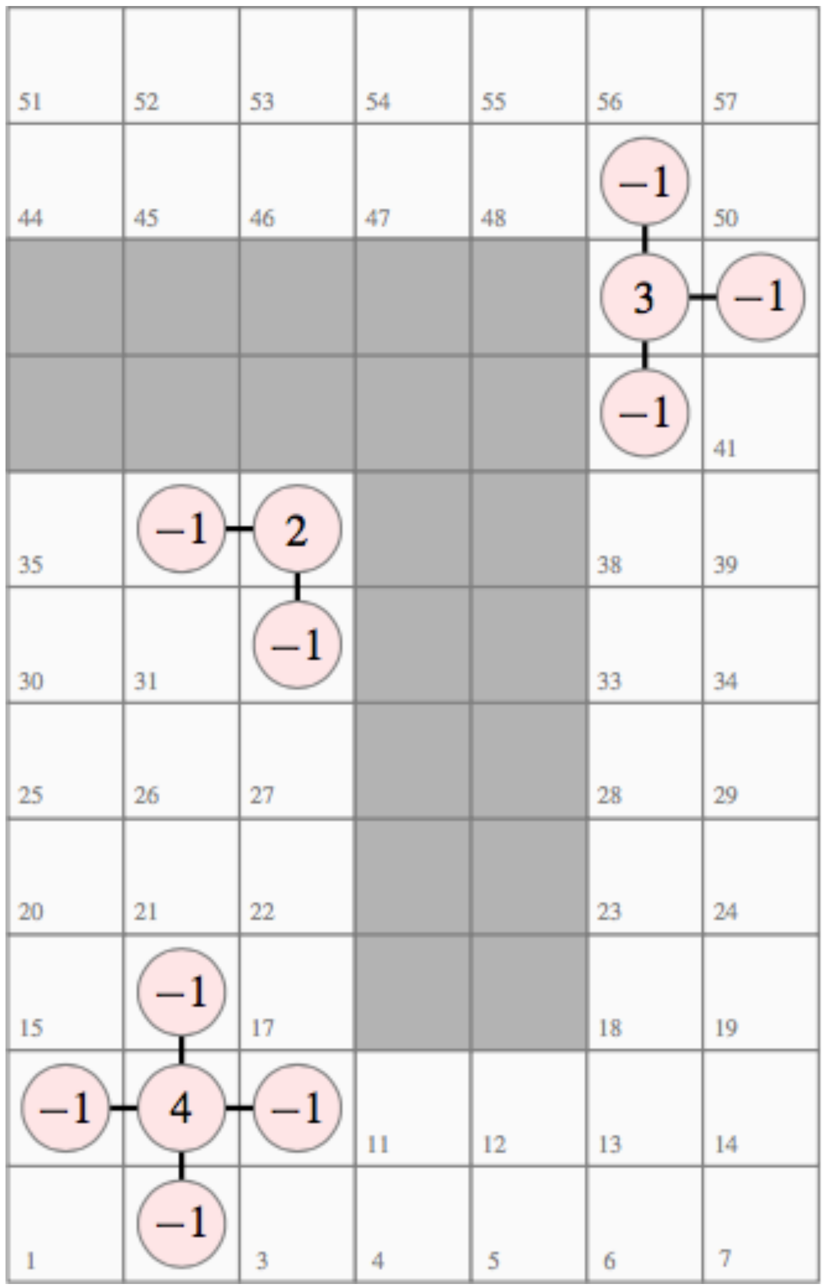


Advantages:

- correct setting of interior boundary conditions possible (**homogeneous Neumann)**
- less grid cells

Higher accuracy,
no additional correction

Unstructured Cartesian grids



$\cdot \frac{1}{h^2}$

<u>Disadvantages:</u>

- loss of regular matrix structure (cells must store its neighbors)
- more general solvers needed (FFT doesn't work anymore)

Application of optimized solvers difficult

hhpberlin

Solvers for the Poisson equation

Presentation of different strategies

Fast Fourier Transformation:
FFT(tol) with velocity correction

FFT(tol)

$$x_m \overset{\text{FFT}}{=} A_m^{-1}\, b_m \,, \qquad x = \tilde{\sum}_{m=1}^{M} x_m$$

Condition 1 ?

Condition 2 ?

Condition 1:
„Internal obstructions“

normal velocity components < **tol**

Condition 2:
„Mesh interfaces“

difference of neighboring
normal velocity components < **tol**

- FFT-solutions on single meshes are highly efficient and fast
- usable for structured grids only

Parallel LU-Decomposition:
Cluster interface of Intel MKL Pardiso

MKL - Init
$LU = \sum_{m=1}^{M} A_m$

MKL - Solve
$Ly = b\,, \qquad Ux = y$

Initialization:

- first „reordering“ of matrix structure
- then distributed LU-factorization

Pressure solution per time step:

- simple forward/backward substitution

- also praised to be very efficient
- usable for structured **and** unstructured grids

Scalable Recursive Clustering (ScaRC): Block-CG and -GMG Methods

ScaRC-CG / ScaRC-GMG
Preconditioning/Smoothing:
Block-SSOR, Block-MKL
Solution of coarse grid problem:
Global CG, MKL

Meshwise strategies with 1 cell overlap

Conjugate Gradient Methods (CG):

- solve equivalent minimization problem

Geometric Multigrid Methods (GMG):

- use complete grid hierarchy with exact solution on coarsest grid level

- reasonable convergence rates and scalability properties
- usable for structured **and** unstructured grids

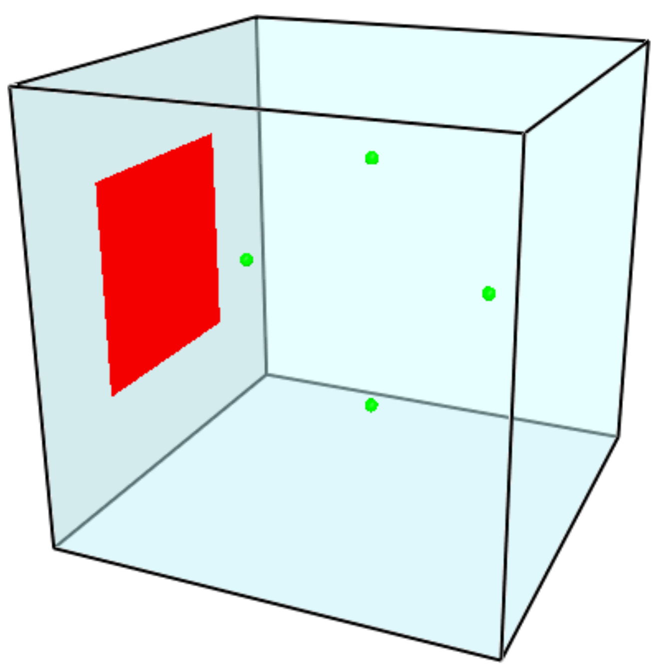
Numerical tests

Comparison of solvers on different geometries

Basic test geometries

Cube⁻

Cube without obstruction



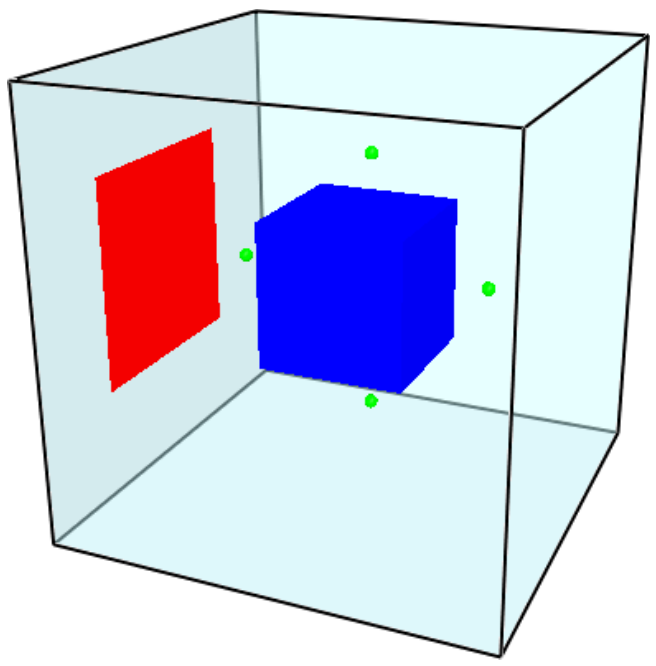
Cells per cube:

24^3, 48^3, 96^3,

192^3, 240^3, 288^3

Cube⁺

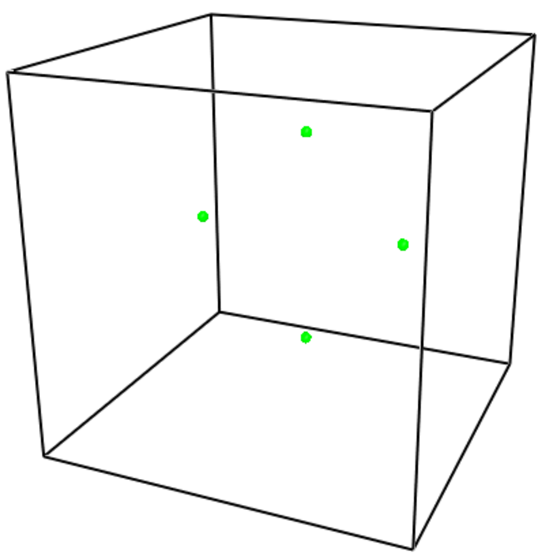
Cube with obstruction

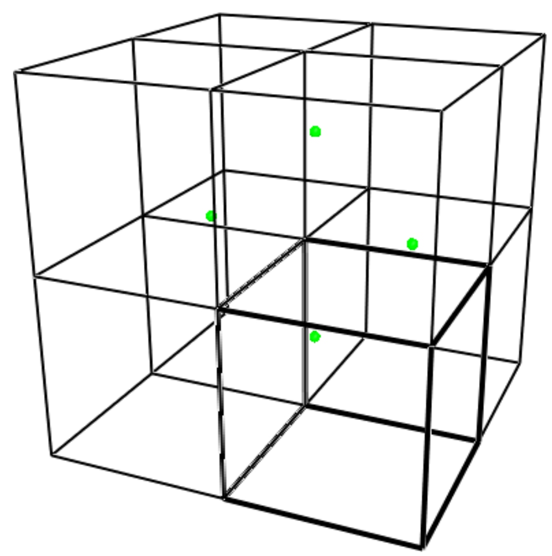


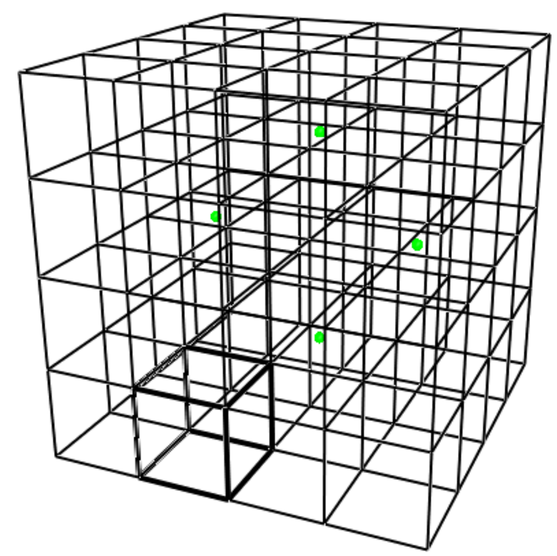
- constant inflow of 1 m/s from the left, open outflow on the right
- comparison of structured FFT(tol) versus unstructured MKL und ScaRC

Different mesh decompositions

1-Mesh	8-Mesh	64-Mesh
1x1x1	2x2x2	4x4x4



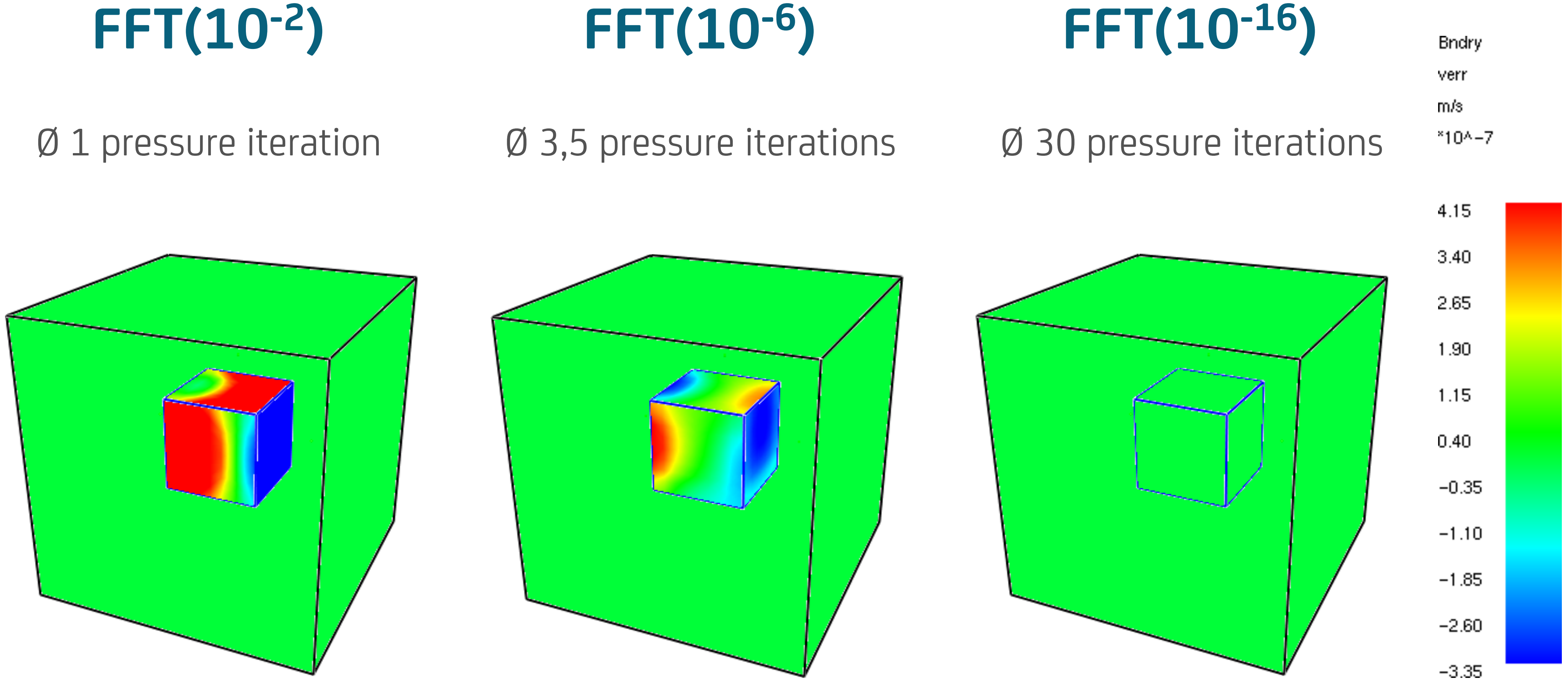




- notations: **Cube**$^-$**(M)** and **Cube**$^+$**(M)** for corresponding M-mesh geometry
- comparison of all solvers on both geometries for M=1, 8, 64

Cube$^+$(1): Velocity error

FFT(10^{-2})	**FFT(10^{-6})**	**FFT(10^{-16})**
Ø 1 pressure iteration	Ø 3,5 pressure iterations	Ø 30 pressure iterations



24^3 Cells, same simulation time and display range for all cases

- velocity correction successfully reduces error along internal obstructions
- number of pressure iterations increases if tolerance is driven to zero

$Cube^-(M)$ vs. $Cube^+(M)$: FFT(10^{-6})

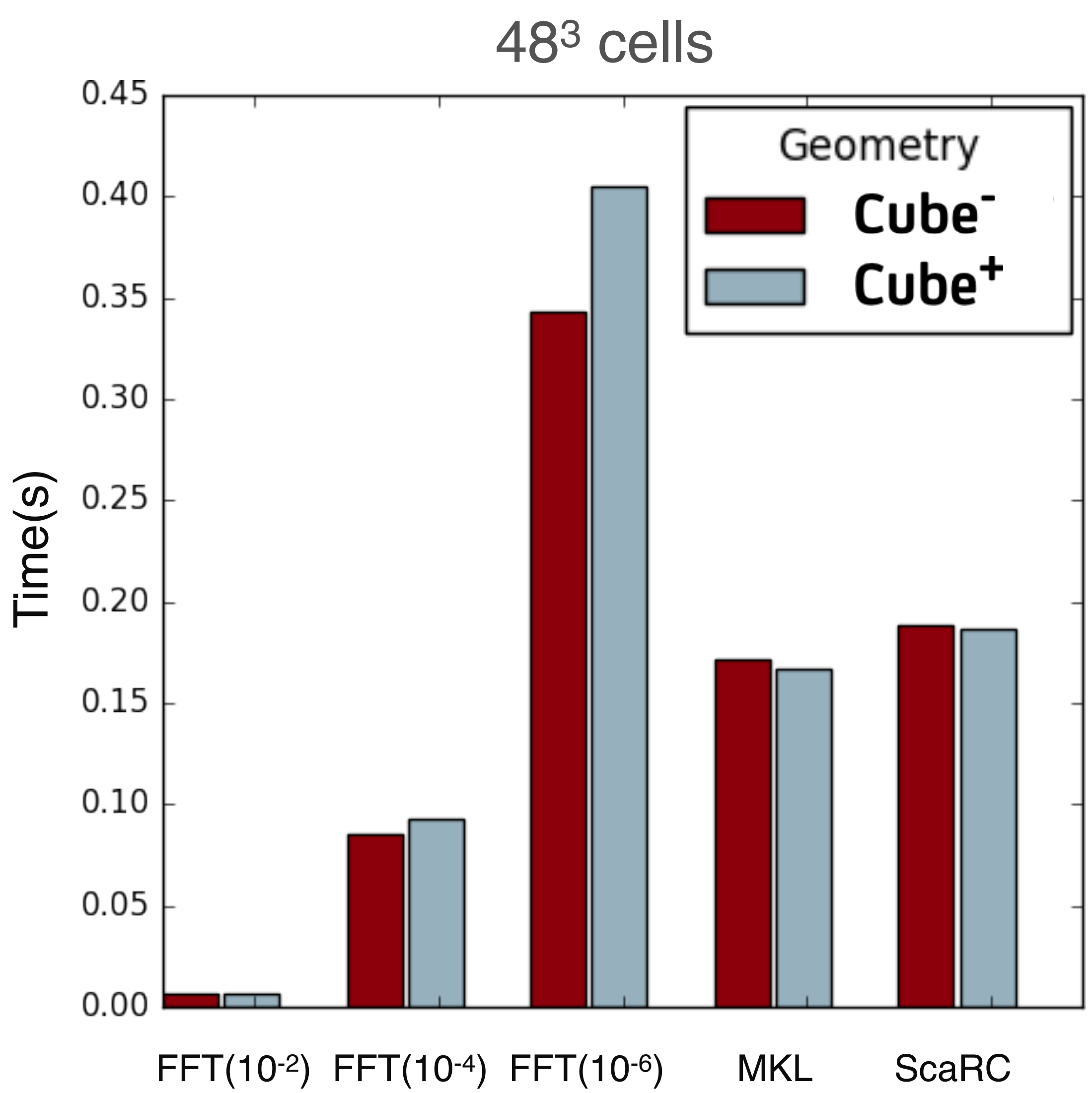
Average of pressure iterations per time step for increasing M:

Geometry 48^3 cells	Number of meshes M		
	1	8	64
$Cube^-(M)$	1	106	222
$Cube^+(M)$	8	123	254

- increasing number of pressure iterations if number of meshes is increased
- mesh decomposition causes higher rise than internal obstruction

Cube⁻(8) vs. Cube⁺(8): All solvers

Average time for 1 pressure solution:



FFT(tol):

- extremely fast for coarse tol
- increasing costs for finer tol

MKL:

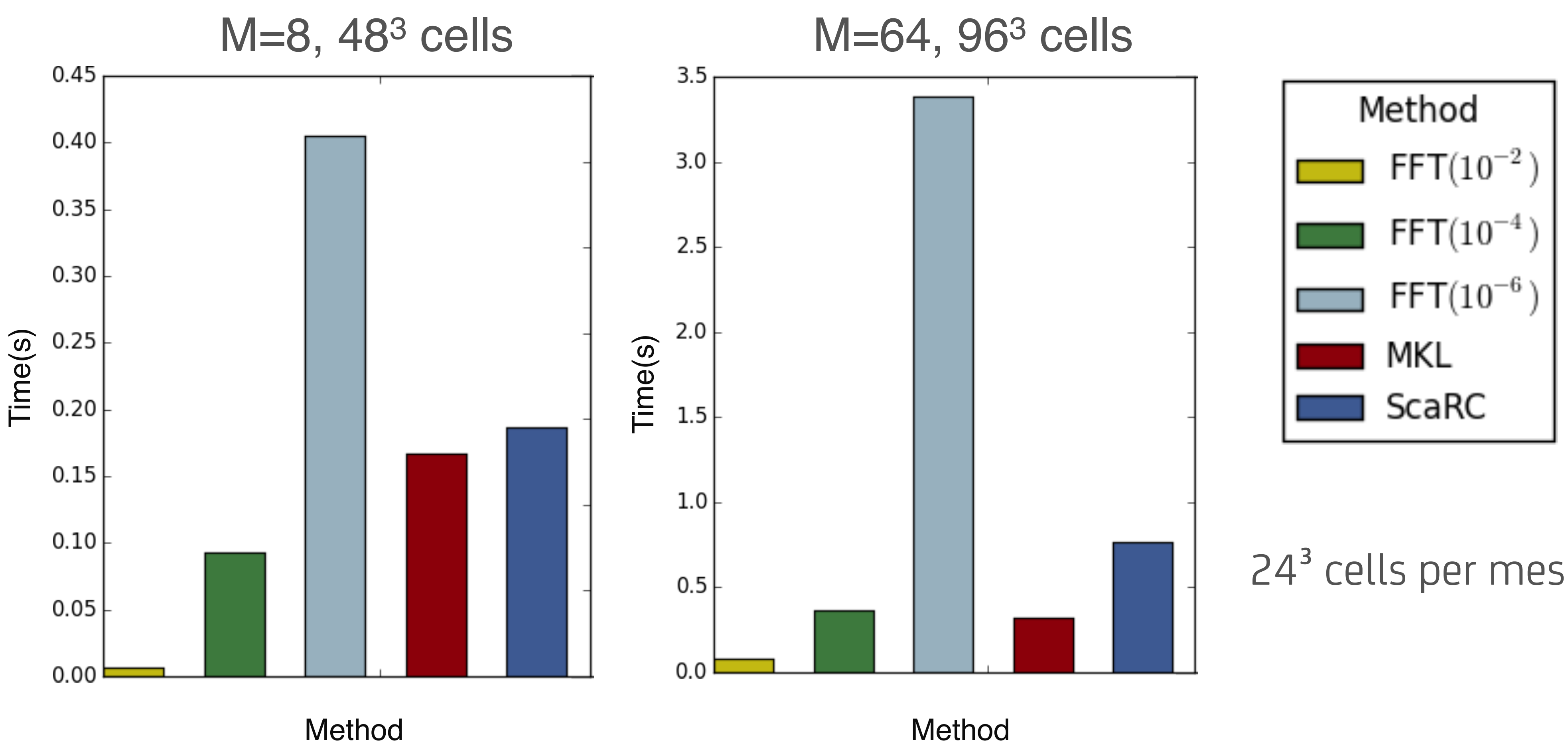
- best computing times (~ zero tol)

ScaRC:

- good computing times (~ zero tol)

Cube^{+}(8) vs. Cube^{+}(64): All solvers

Average time for 1 pressure solution, growing problem size:

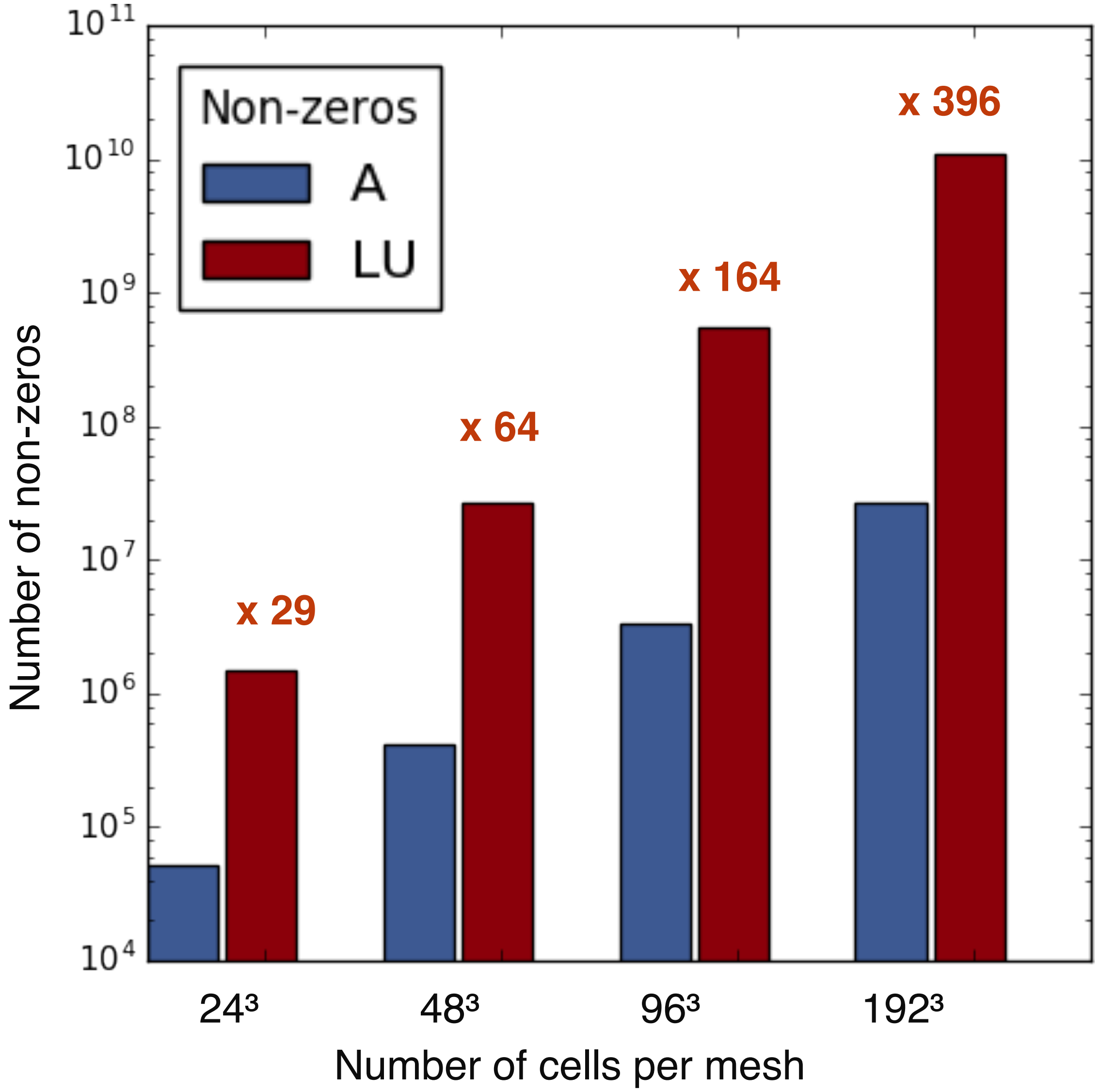


24^3 cells per mesh

scalability gets worse if number of meshes is increased at constant load

Cube⁺(8): Costs MKL-method

Logarithmic scale !!



Storage

High memory needs due to „fill-in"

LU has much more non-zeros than A

(FFT/ScaRC: very less memory needs)

Runtime

Expensive initialization

Example: 8 Meshes with 96^3 cells

- MKL-Init: ~ 5000 s
- MKL-Solve: 17 s

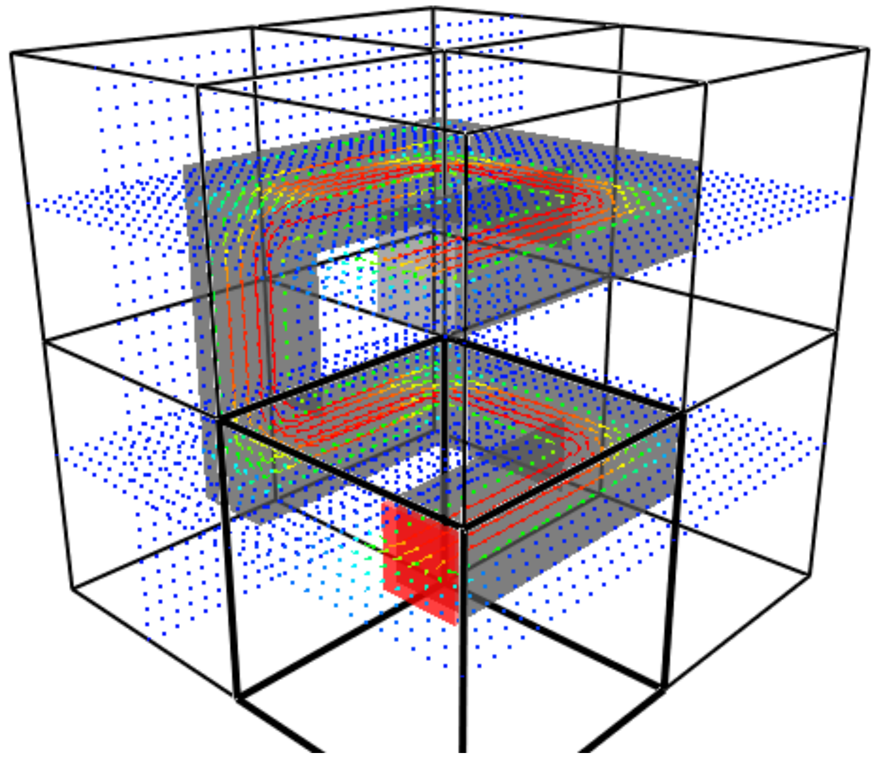
FFT and ScaRC can solve finer problems than MKL on given ressources

(Example: FFT und ScaRC run for 288^3, MKL already fails for 240^3)

hhpberlin

Duct_Flow: Flow through a pipe

Case from FDS Verification Guide:



8 Meshes, 128^3 cells

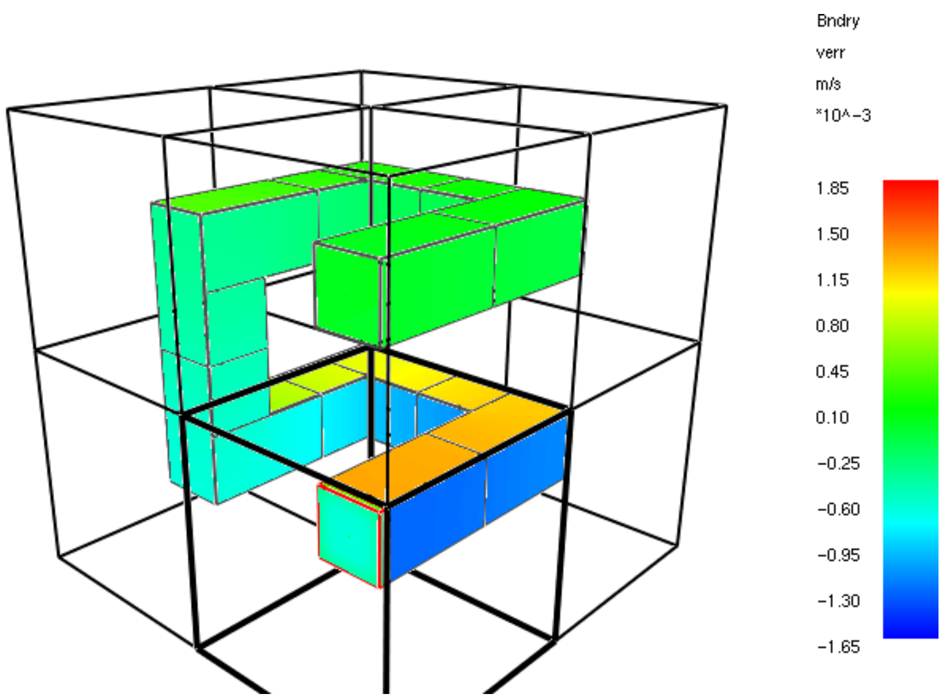
Method	Average time for 1 pressure solution
FFT(10^{-4})	41.3 s
MKL	4.4 s
ScaRC	7.5 s

- comparison of structured FFT(tol) versus unstructured MKL and ScaRC
- best times for MKL, reasonable times for ScaRC

Duct_Flow: Flow through a pipe

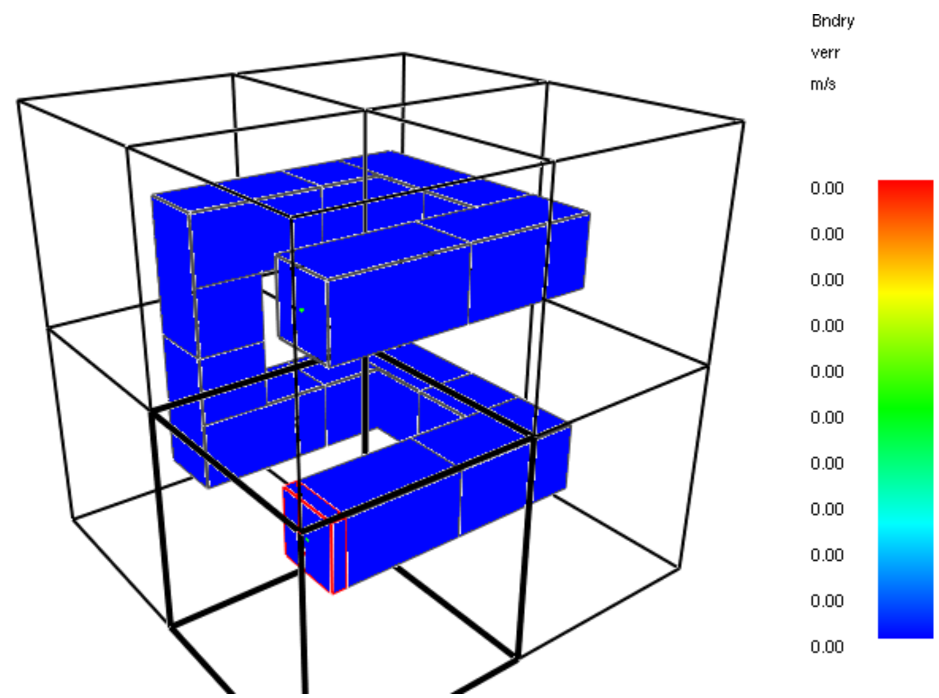
FFT(10-4)

8 Meshes, 64^3 Cells



MKL / ScaRC

8 Meshes, 64^3 Cells



- FFT(tol): velocity correction slow (tol=10^{-4} needs ~1000 iterations)
- MKL / ScaRC: zero velocity error along pipe walls

Conclusions

Summary and outlook

Summary and outlook

Summary

- no consistent overall picture yet, still more tests planned
- need to find a clever balance between:
  - accuracy (velocity error?)
  - performance (computational times for 1 Poisson solve?)
  - additional costs (storage, further libraries?)

Outlook

- test unstructured MKL and ScaRC:
  - to solve the implicit advection diffusion problem for scalars on the cut-cell region (IBM-method)
  - to solve the Laplace problem on the unstructured grid (as velocity correction) in combination with a structured FFT solution of the Poisson problem

Thanks a lot for your attention

Questions?

hhpberlin
Ingenieure für Brandschutz GmbH

Hauptsitz
Rotherstraße 19 · 10245 Berlin

Amtsgericht
Berlin-Charlottenburg
Register-Nr.: HRB 78 927
Ust-ID Nr.: DE217656065

Geschäftsführung:
Dipl.-Ing. Karsten Foth
Dipl.-Inf. BW [VWA] Stefan Truthän

Beirat:
Dipl.-Ing. Margot Ehrlicher
Prof. Dr.-Ing. Dietmar Hosser
Dr.-Ing. Karl-Heinz Schubert