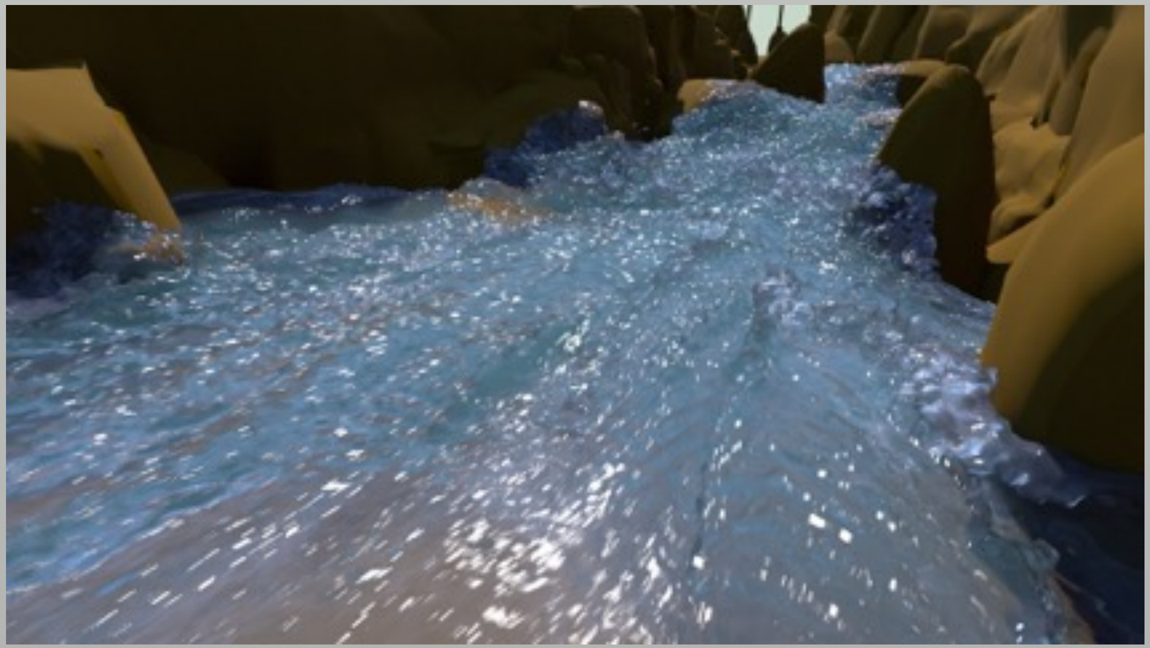
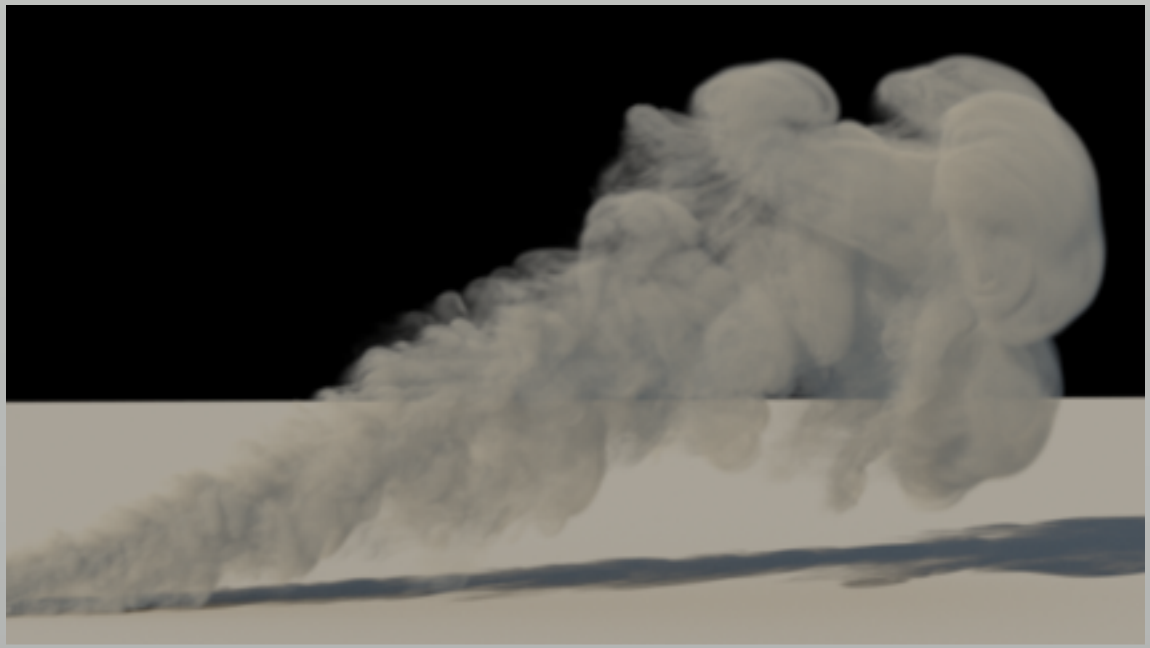
A Schur Complement Preconditioner for Scalable Parallel Fluid Simulation





Jieyu Chu
Shanghai Jiao Tong University
Oriental DreamWorks

Nafees Bin Zafar
Oriental DreamWorks

Xubo Yang
Shanghai Jiao Tong University

Contributions

- A novel Schur Complement preconditioner
- A framework to apply different solvers to inner subdomains
- High performance method to solve the pressure projection problem for incompressible flows

Overview

- Previous efforts in production
- Schur complement method and existing preconditioners
- Our new preconditioner
- Choosing subdomain solvers
- Poisson solver tests and fluid simulation tests
- Limitations and future work

Production Inspiration

- ILM
- Scanline
- Weta
- Simplifies
  sims

- But we usually use ad-hoc solutions
  - Requires clever artists
  - Tweaked for each shot

Incompressible Euler Equations

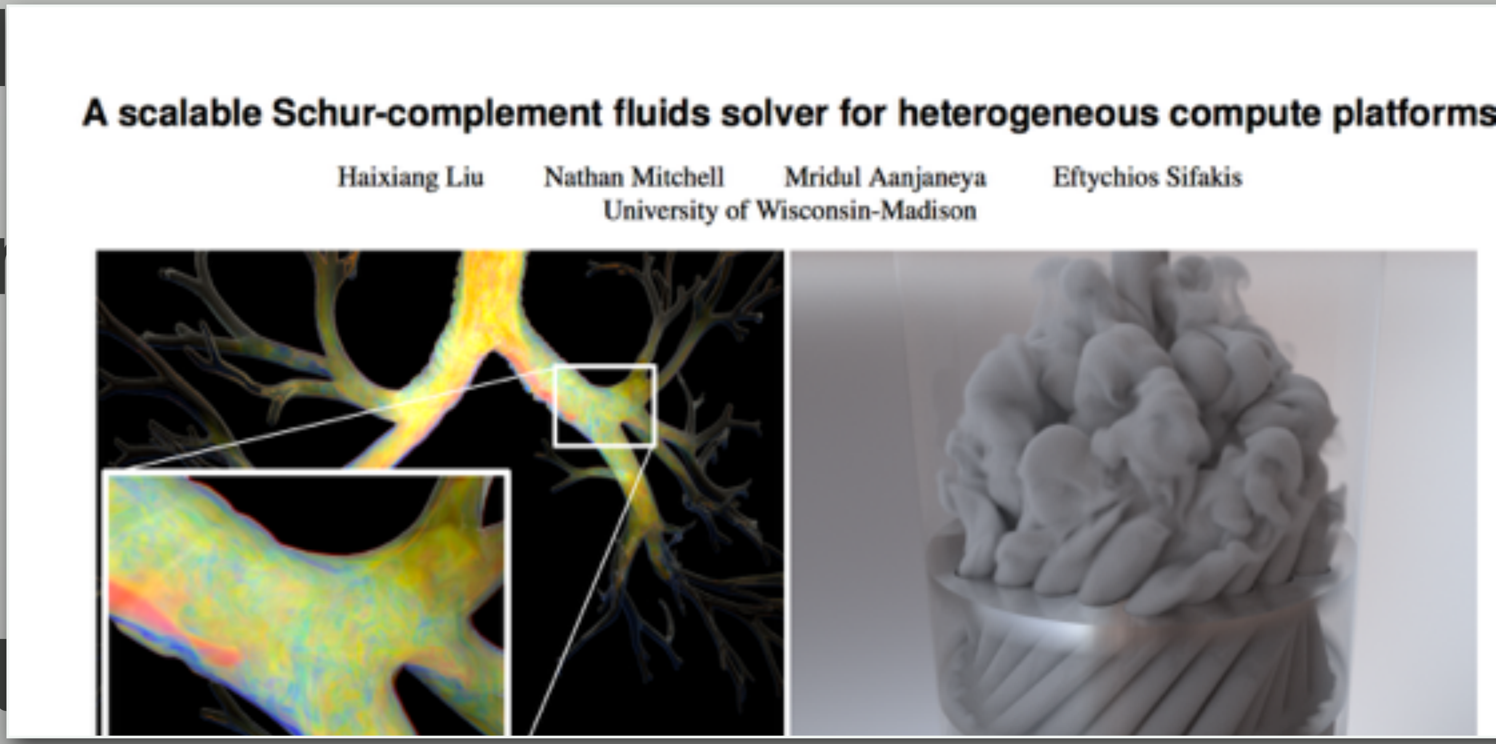
$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla \mathbf{p} + \mathbf{f}$$

$$\nabla \cdot \mathbf{u} = \mathbf{0}$$

Slow!

Inspiration

- Lot of prior work in CFD and graphics
- Adaptive fluid simul
- Fast Poisson solver
- Multi-grid
- domain decomposit



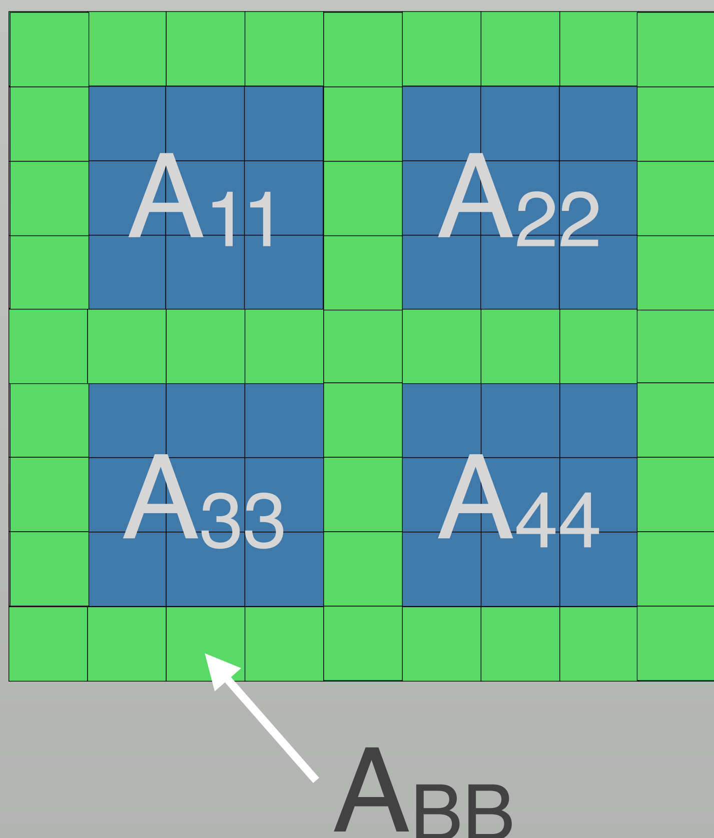
Fast Poisson Solver

- Regular domain: n log n
- Boundaries using IOP [Molemaker et al. 2008]
- Large scale gas sims [Henderson 2012]
- Not usable for liquid simulations
- But PCG has poor parallel scalability
  - Cores are idling

Design Requirements

- 1 billion voxels
- Parallelized for multi-core systems
- Use the Fast Poisson solver algorithm
- PIC/FLIP is popular
- Large sims are slow, and need too much memory
  - Our target resolutions needed NB-FLIP [Ferstl et al. 2016]

Schur Complement Decomposition



Subdomain

Boundary set

$$
\begin{pmatrix}
A_{11} & & & & A_{1B} \\
& A_{22} & & & A_{2B} \\
& & \ddots & & \vdots \\
& & & A_{nn} & A_{nB} \\
A_{1B}^T & A_{2B}^T & \cdots & A_{nB}^T & A_{BB}
\end{pmatrix}
\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_B \end{pmatrix}
=
\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ b_B \end{pmatrix}
$$

Schur Complement Matrix

$$\begin{pmatrix} A_{11} & & & & A_{1B} \\ & A_{22} & & & A_{2B} \\ & & \ddots & & \vdots \\ & & & A_{nn} & A_{nB} \\ A_{1B}^T & A_{2B}^T & \cdots & A_{nB}^T & A_{BB} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_B \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ b_B \end{pmatrix}$$

$$\begin{cases} A_{11}x_1 + A_{1B}x_B &= b_1, \\ A_{22}x_2 + A_{2B}x_B &= b_2, \\ \qquad \cdots\cdots, \\ A_{nn}x_n + A_{nB}x_B &= b_n\,. \end{cases}$$

$$\boxed{x_i = A_{ii}^{-1}(b_i - A_{iB}x_B)}$$

Schur Complement Matrix

$$\begin{pmatrix} A_{11} & & & & A_{1B} \\ & A_{22} & & & A_{2B} \\ & & \ddots & & \vdots \\ & & & A_{nn} & A_{nB} \\ A_{1B}^T & A_{2B}^T & \cdots & A_{nB}^T & A_{BB} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ x_B \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \\ b_B \end{pmatrix}$$

$$A_{1B}^T x_1 + A_{2B}^T x_2 + \ldots + A_{nB}^T x_n + A_{BB} x_B = b_B$$

$$S x_B = b$$

$$S = A_{BB} - \sum_{i=1}^{n} A_{iB}^T A_{ii}^{-1} A_{iB}$$

$$b = b_B - \sum_{i=1}^{n} A_{iB}^T A_{ii}^{-1} b_i$$

Schur Complement Solver

$$Sx_B = b$$

$$S = A_{BB} - \sum_{i=1}^{n} A_{iB}^T A_{ii}^{-1} A_{iB}$$

- S is also symmetric positive definite
- We don't need to form S explicitly
- We use PCG to solve this
- We only need to multiply a vector with S

Schur Complement Solver

$$Sx_B = b$$

$$S = A_{BB} - \sum_{i=1}^{n} A_{iB}^T \boxed{A_{ii}^{-1} A_{iB}}$$

- Get the multiply results of this part and one vector

$$A_{ii} w_i = A_{iB} w_B$$

$$S w_B = A_{BB} w_B - \sum_{i=1}^{n} A_{iB}^T w_i$$

- Actually solve the subdomain equation

Subdomain Independence

$$Sx_B = b$$

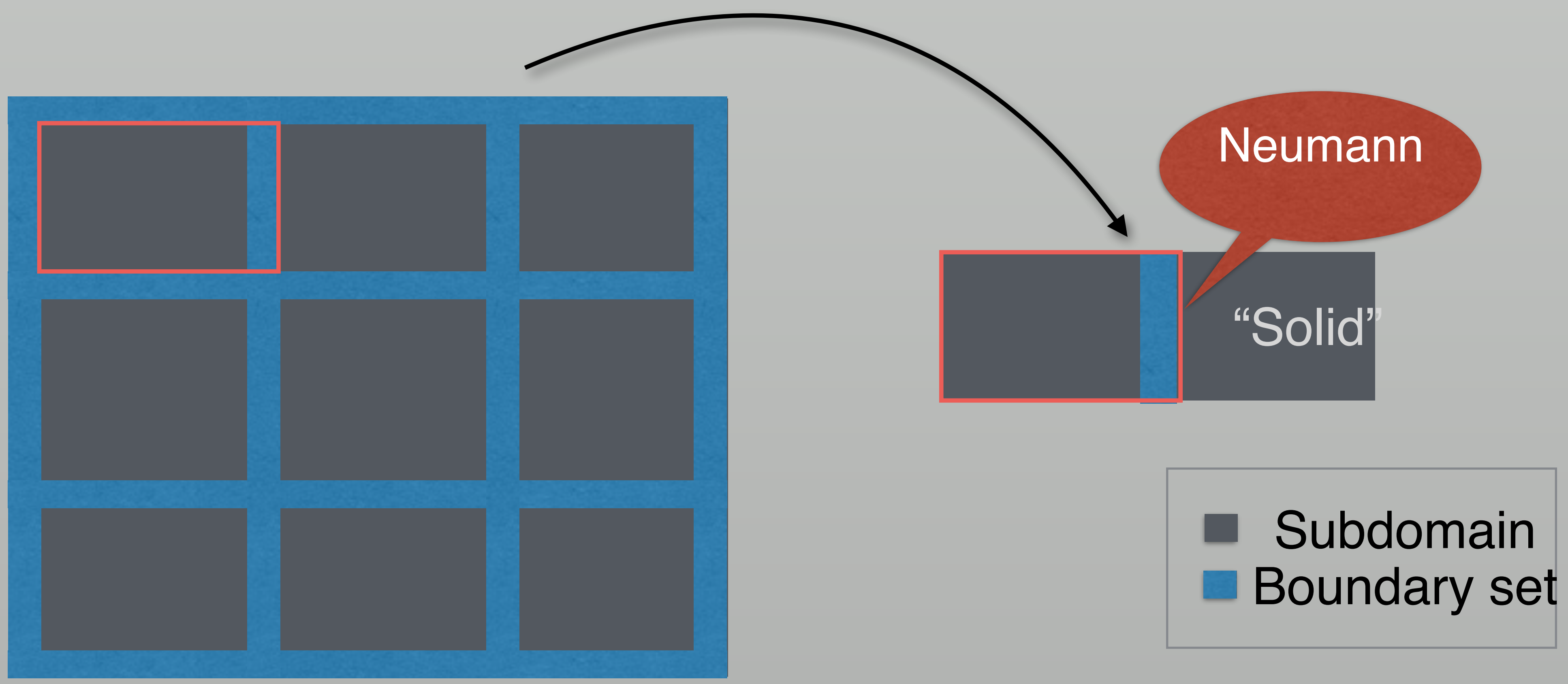
$$S = A_{BB} - \sum_{i=1}^{n} A_{iB}^T A_{ii}^{-1} A_{iB}$$

$$x_i = A_{ii}^{-1}(b_i - A_{iB}x_B)$$

$$b = b_B - \sum_{i=1}^{n} A_{iB}^T A_{ii}^{-1} b_i$$

- Aii subdomains are all independent
- Subdomains can be solved in parallel

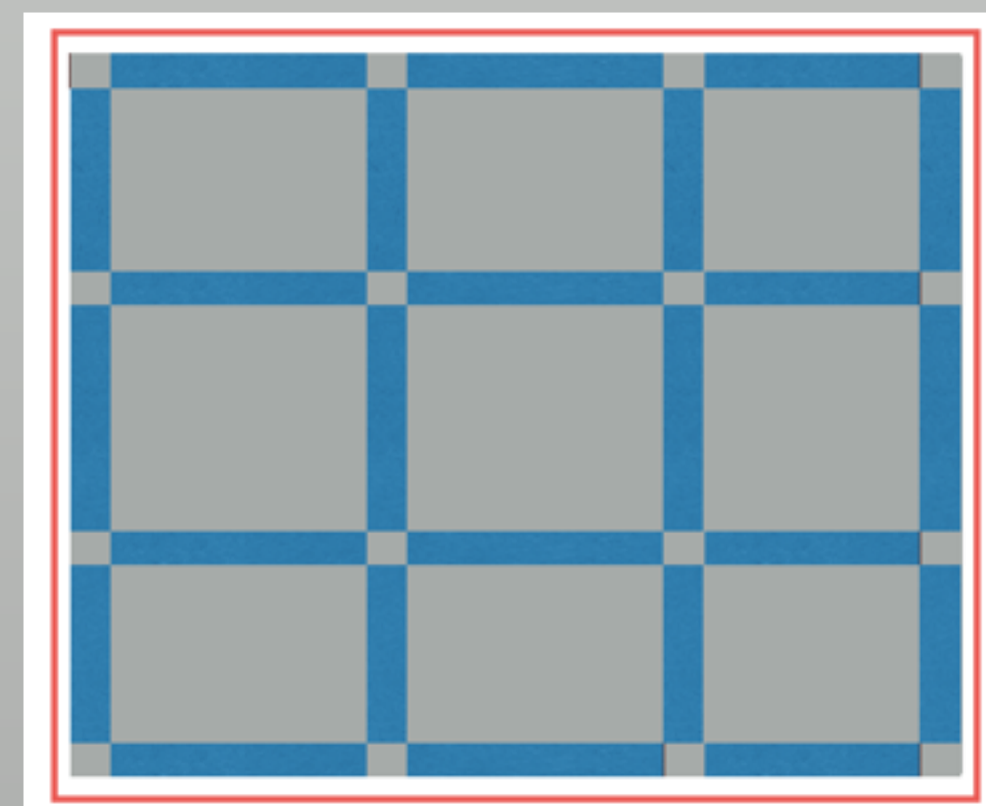
Block Jacobi Preconditioner



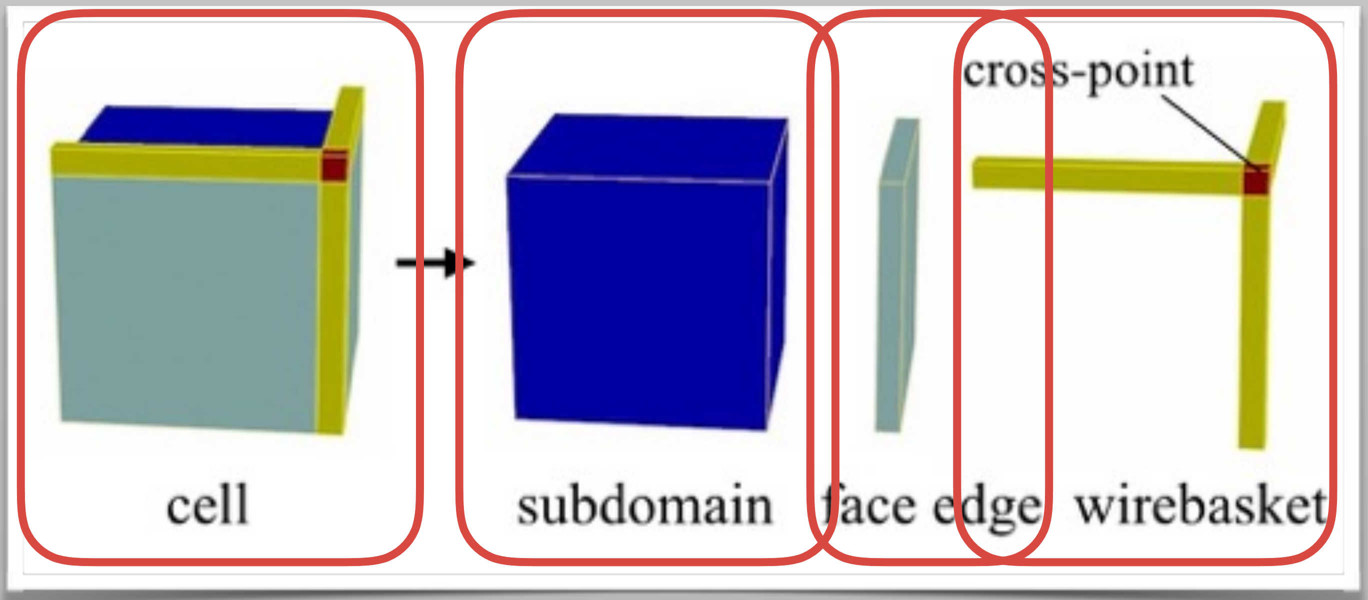
Cross Points Subdomain

- The cross points are disconnected with subdomains
- Treat them as another subdomain
- Modified system

$$\begin{pmatrix} A_{11} & & & & A_{1B'} \\ & \ddots & & & \vdots \\ & & A_{nn} & & A_{nB'} \\ & & & A_{WW} & A_{WB'} \\ A_{1B'}^T & \cdots & A_{nB'}^T & A_{WB'}^T & A_{B'B'} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ x_W \\ x_{B'} \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \\ b_W \\ b_{B'} \end{pmatrix}$$

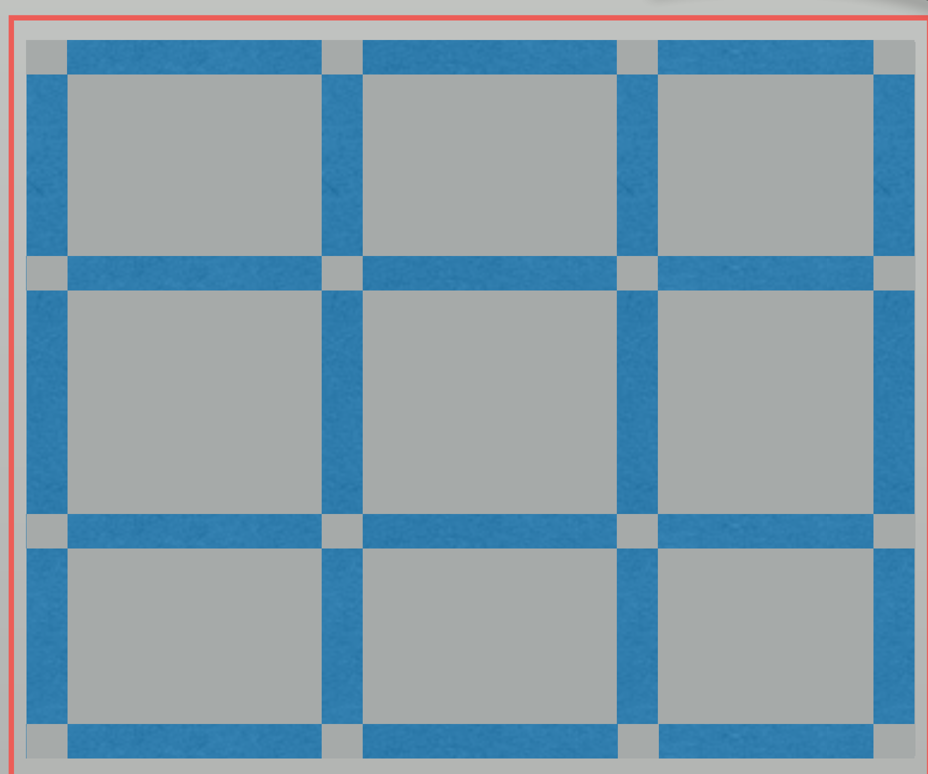


The 3D View

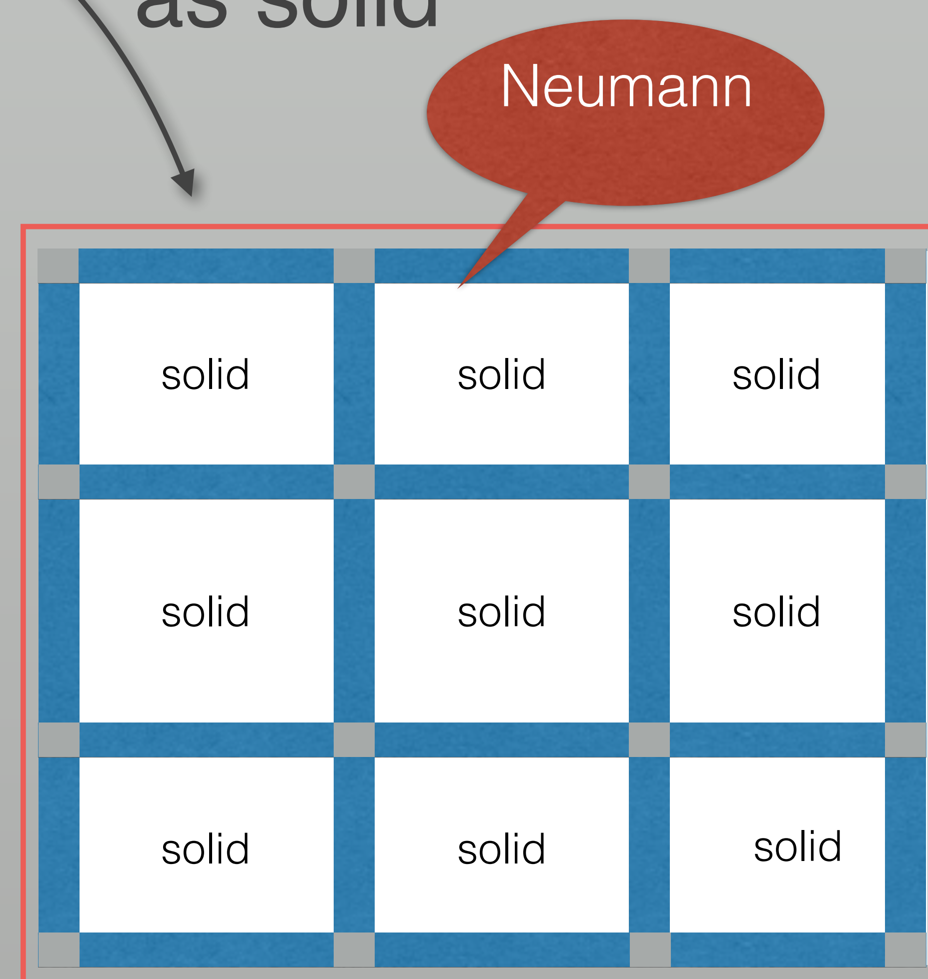


- Wirebasket and cross points are disconnected from the subdomains
- Cross points are also disconnected from face sets

Our New Preconditioner



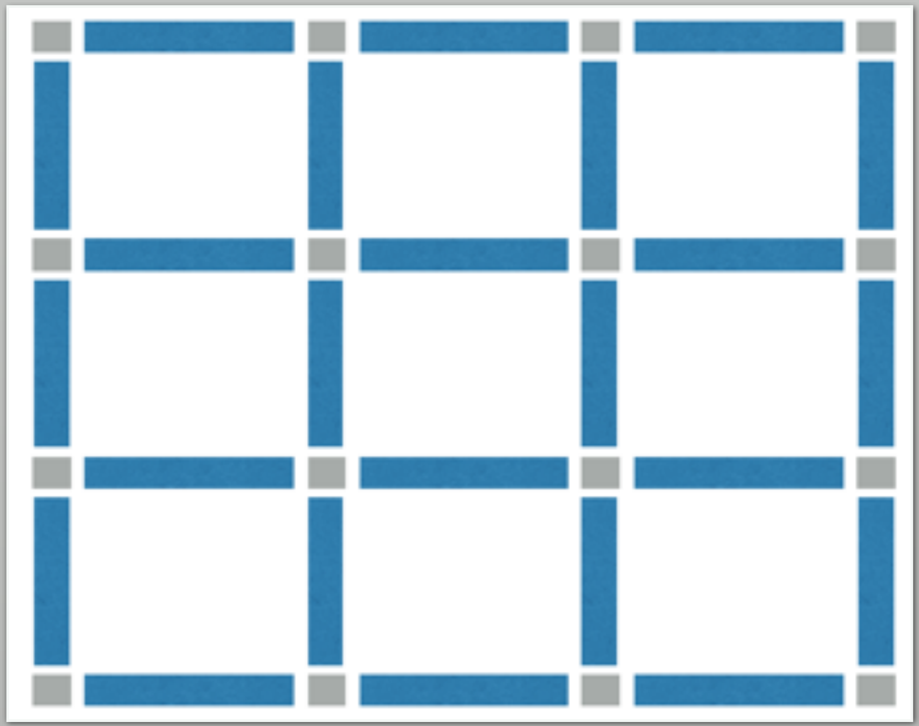
View the subdomain as solid



$$\begin{pmatrix} A_{WW} & A_{WB'} \\ A_{WB'}^{T} & N_{B'B'} \end{pmatrix} \begin{pmatrix} v_W \\ v_{B'} \end{pmatrix} = \begin{pmatrix} 0 \\ r_{B'} \end{pmatrix}$$

The “Aha” Moment

$$\begin{pmatrix} A_{WW} & A_{WB'} \\ A_{WB'}^T & N_{B'B'} \end{pmatrix} \begin{pmatrix} v_W \\ v_{B'} \end{pmatrix} = \begin{pmatrix} 0 \\ r_{B'} \end{pmatrix}$$



- This is still too slow to solve
- Edge sets are disconnected
- Formulate the preconditioner as a Schur complement problem
- And we can parallelize the preconditioner too!

Our New Preconditioner

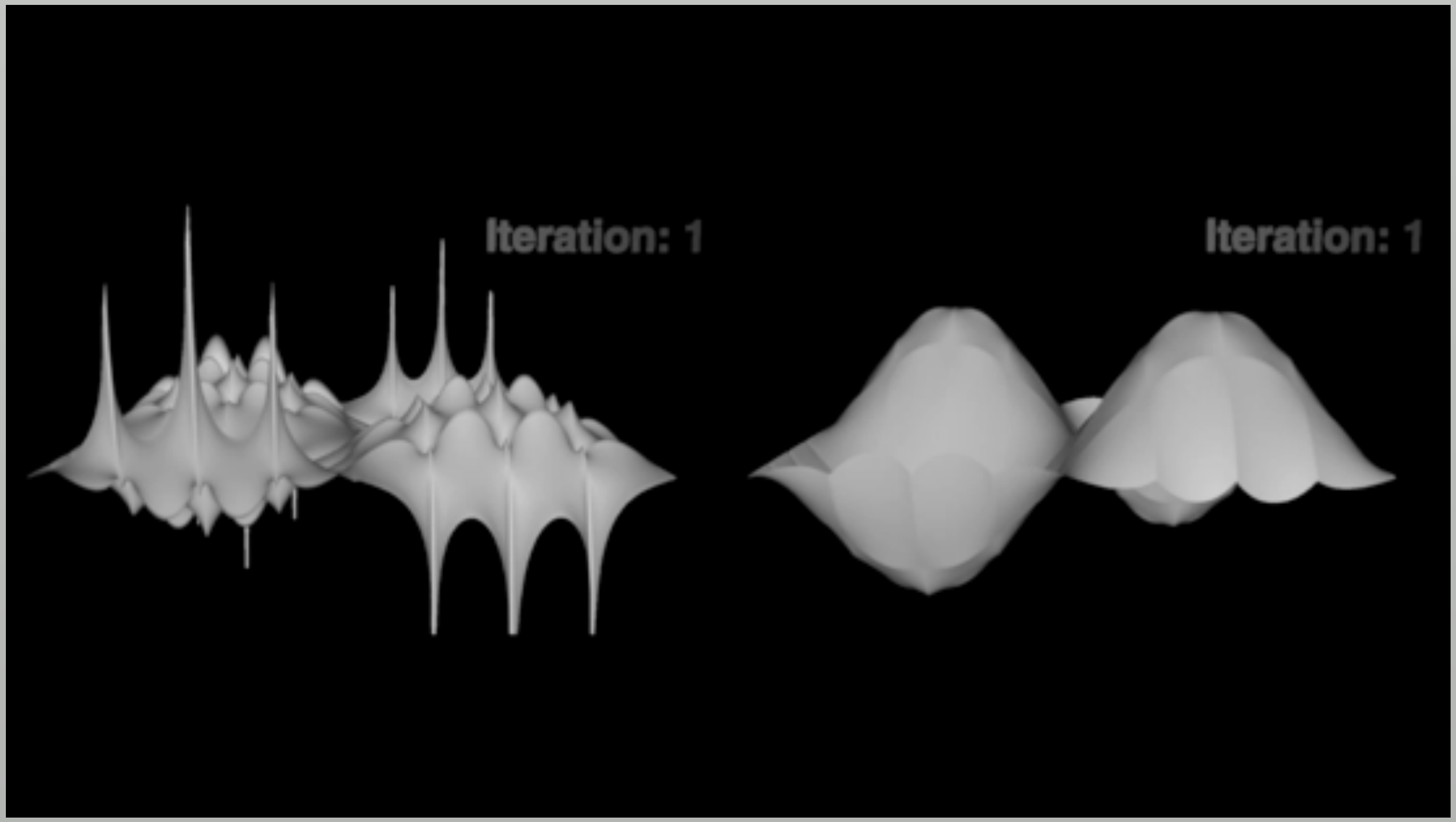
$$\begin{pmatrix} A_{WW} & A_{WB'} \\ A_{WB'}^T & N_{B'B'} \end{pmatrix} \begin{pmatrix} v_W \\ v_{B'} \end{pmatrix} = \begin{pmatrix} 0 \\ r_{B'} \end{pmatrix}$$

$\Big\downarrow$ Make another Schur system

$$\begin{pmatrix} N_{F_1F_1} & & & & A_{F_1W'} \\ & \ddots & & & \vdots \\ & & N_{F_qF_q} & & A_{F_qW'} \\ & & & A_{\nu\nu} & A_{\nu W'} \\ A_{F_1W'}^T & \cdots & A_{F_qW'}^T & A_{\nu W'}^T & A_{W'W'} \end{pmatrix} \begin{pmatrix} x_{F_1} \\ \vdots \\ x_{F_q} \\ x_\nu \\ x_{W'} \end{pmatrix} = \begin{pmatrix} b_{F_1} \\ \vdots \\ b_{F_q} \\ b_\nu \\ b_{W'} \end{pmatrix}$$

Face sets are 2D problems.
Only a 5-point stencil needed.

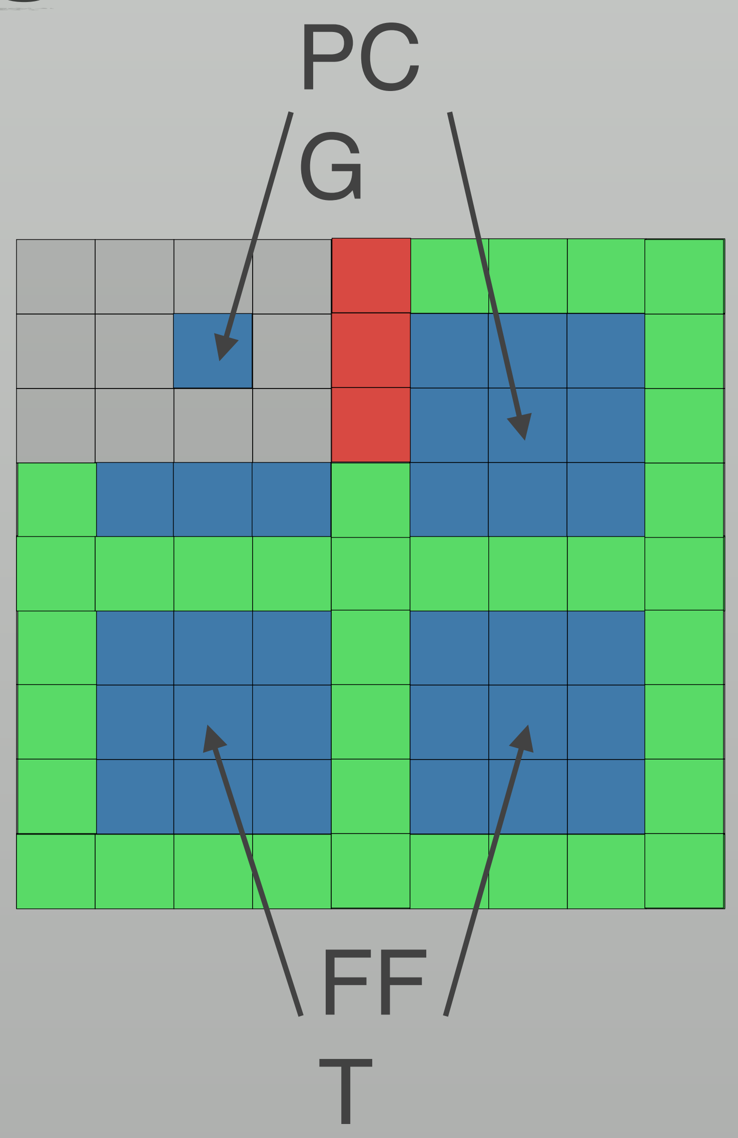
Comparison of Preconditioners



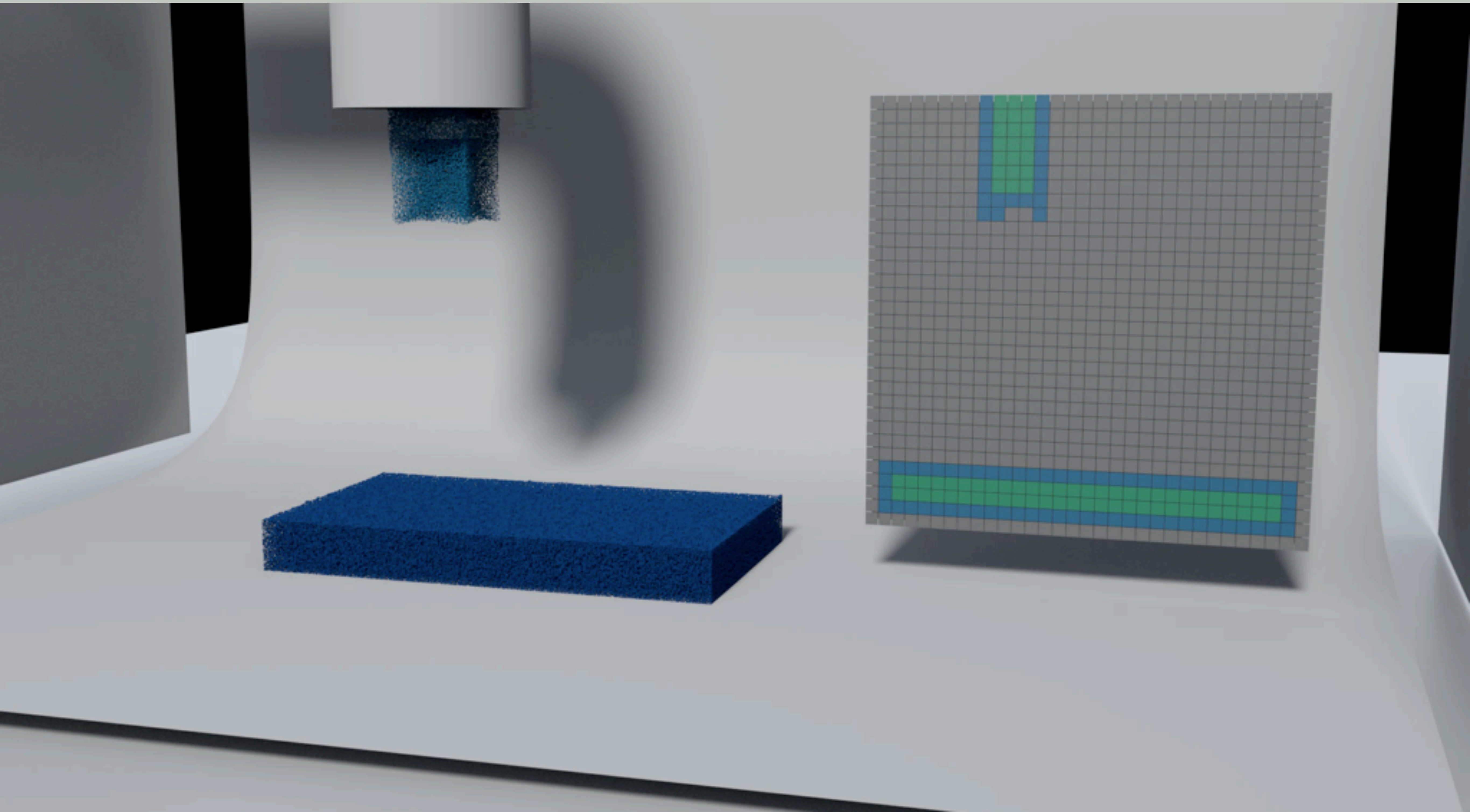
512^3 System	Iterations	Time
Block Jacobi with Dirichlet-Neumann	78	4977s
Block Jacobi with Neumann-Neumann	78	7882s
Our preconditioner	10	243s

Subdomain Solvers

- A more realistic situation
- FFT in the interior
- PCG for weird boundaries



Subdomain Solvers



Memory Issue

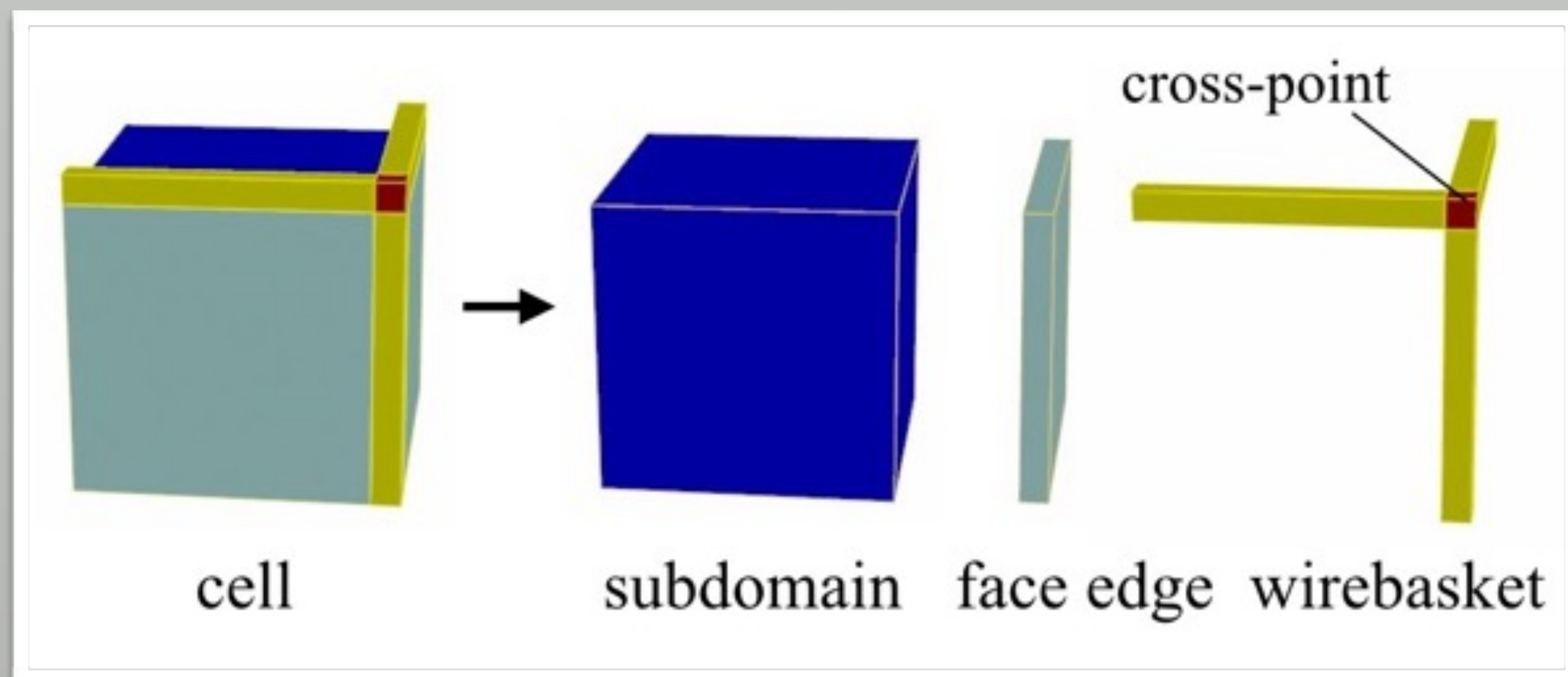
- Save the memory as well
    - FFT inner solver

Irregular Subdomain Solvers

- Sparse Cholesky factorization
- PCG

Degrees of freedom	2D		3D	
	N	Condition Number	N	Condition Number
64	8	196.829	4	94.9691
729	27	62741.6	9	1399.99
4096	64	79567.4	16	9696.85
15625	125	565988	25	43741.8

Irregular Subdomain Solvers



subdomain matrix in Schur system A_{ii}^{-1}

$N_{F_l F_l}^{-1}$ subdomain matrix in our preconditioner's Schur system

7-point stencil matrix
Modified Incomplete Choleskey PCG

5-point stencial matrix
sparse Choleskey factorization

Implementation

- Multi-threading
- Fluid solver
- Domain partitioning
- Solver parameters

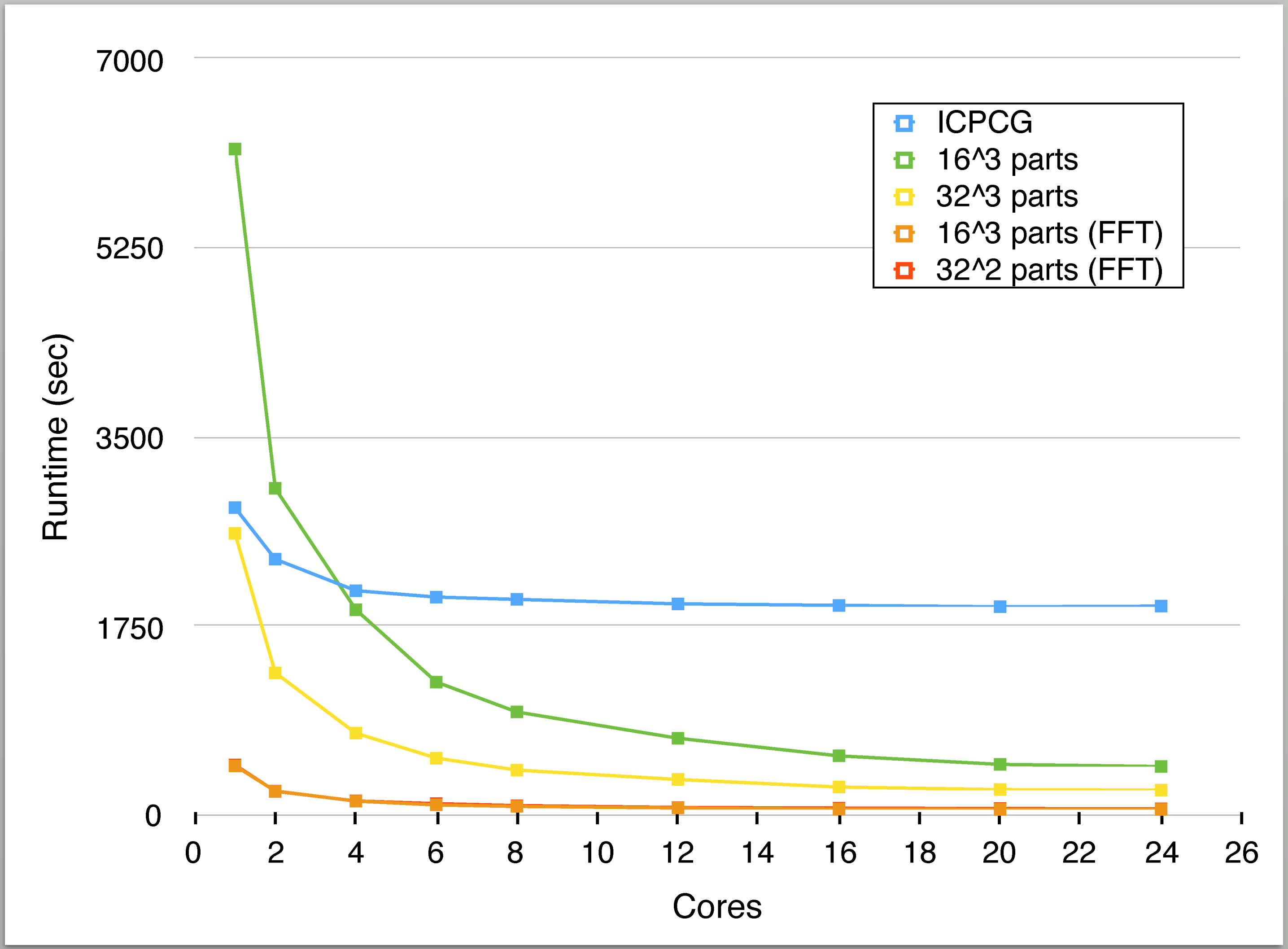
Poisson Solver Test

Resolution: 512^3	CPU			
Inner solver: ICPCG	1	8	16	24
ICPCG	2846.18	1997.04	1942.07	1934.64
8*8*8 subdomains	6166.51	954.67	548.42	449.71
16*16*16 subdomains	2607.32	417.15	259.83	230.91
8*8*8 subdomains(FFT)	456.75	80.40	55.82	52.43
16*16*16 subdomains(FFT)	463.29	87.92	65.15	58.35

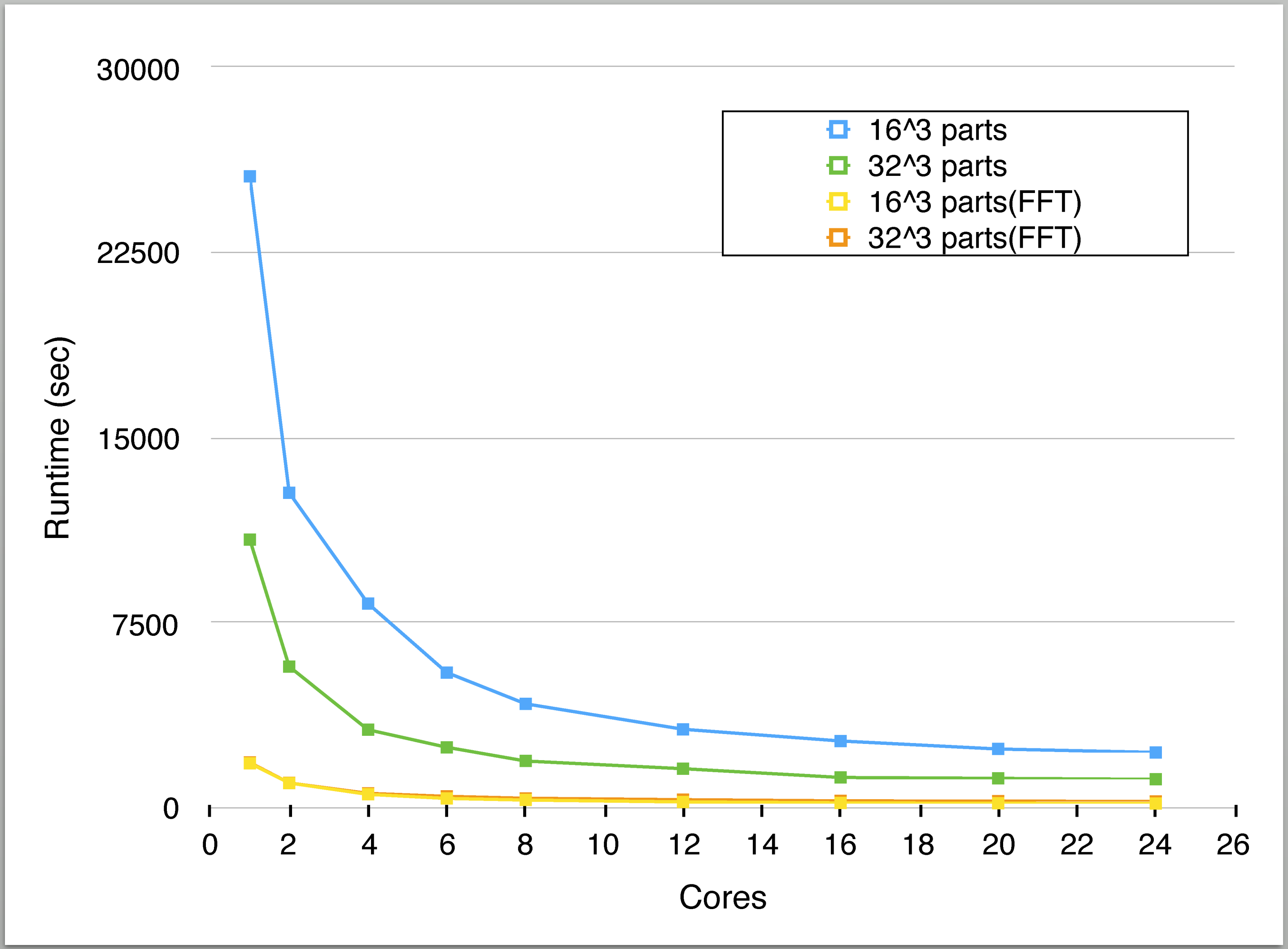
Machine: 2.5GHZ, 24 core,

2 processor system, 128GB memory.

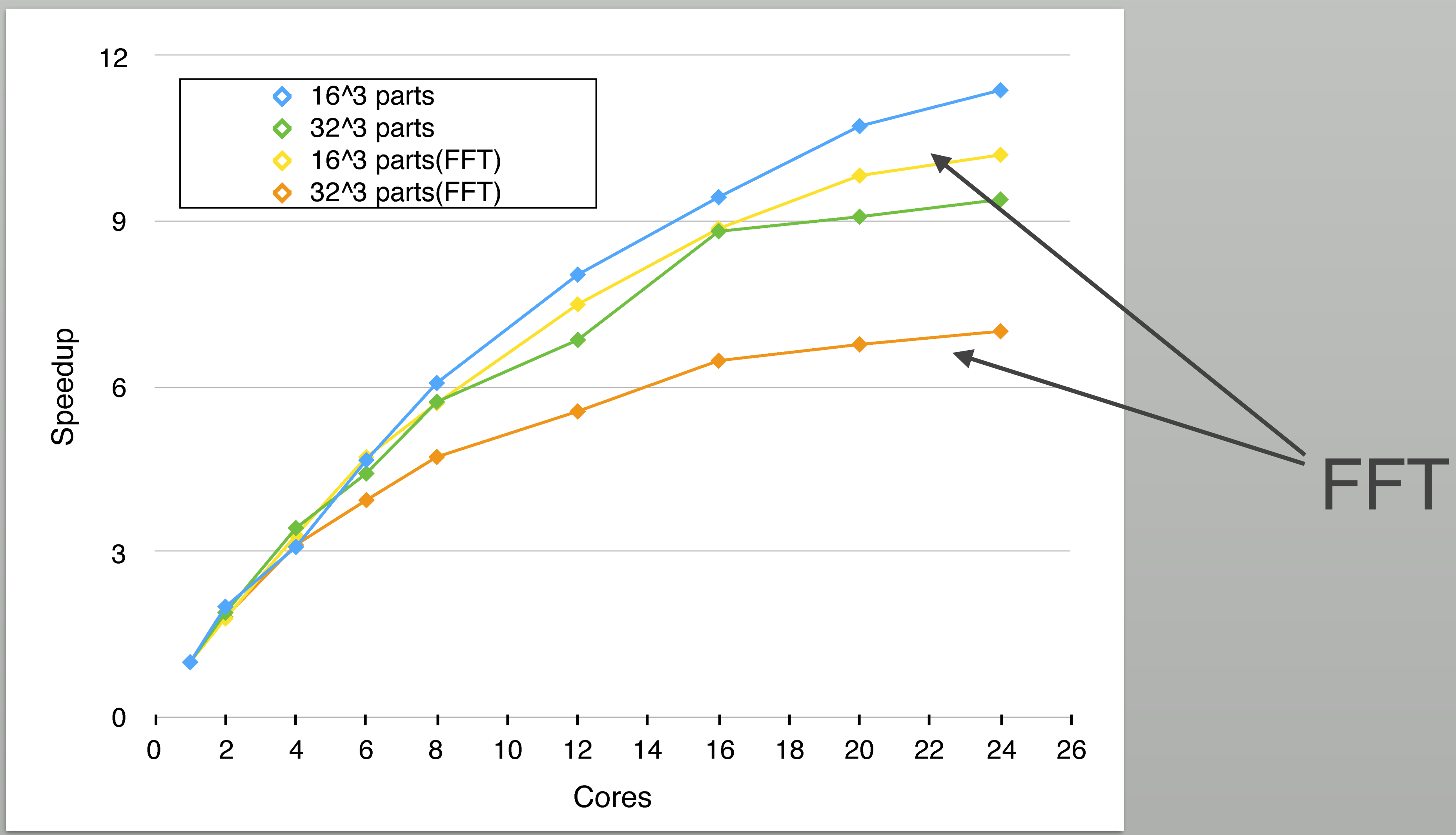
Runtime for 512^3



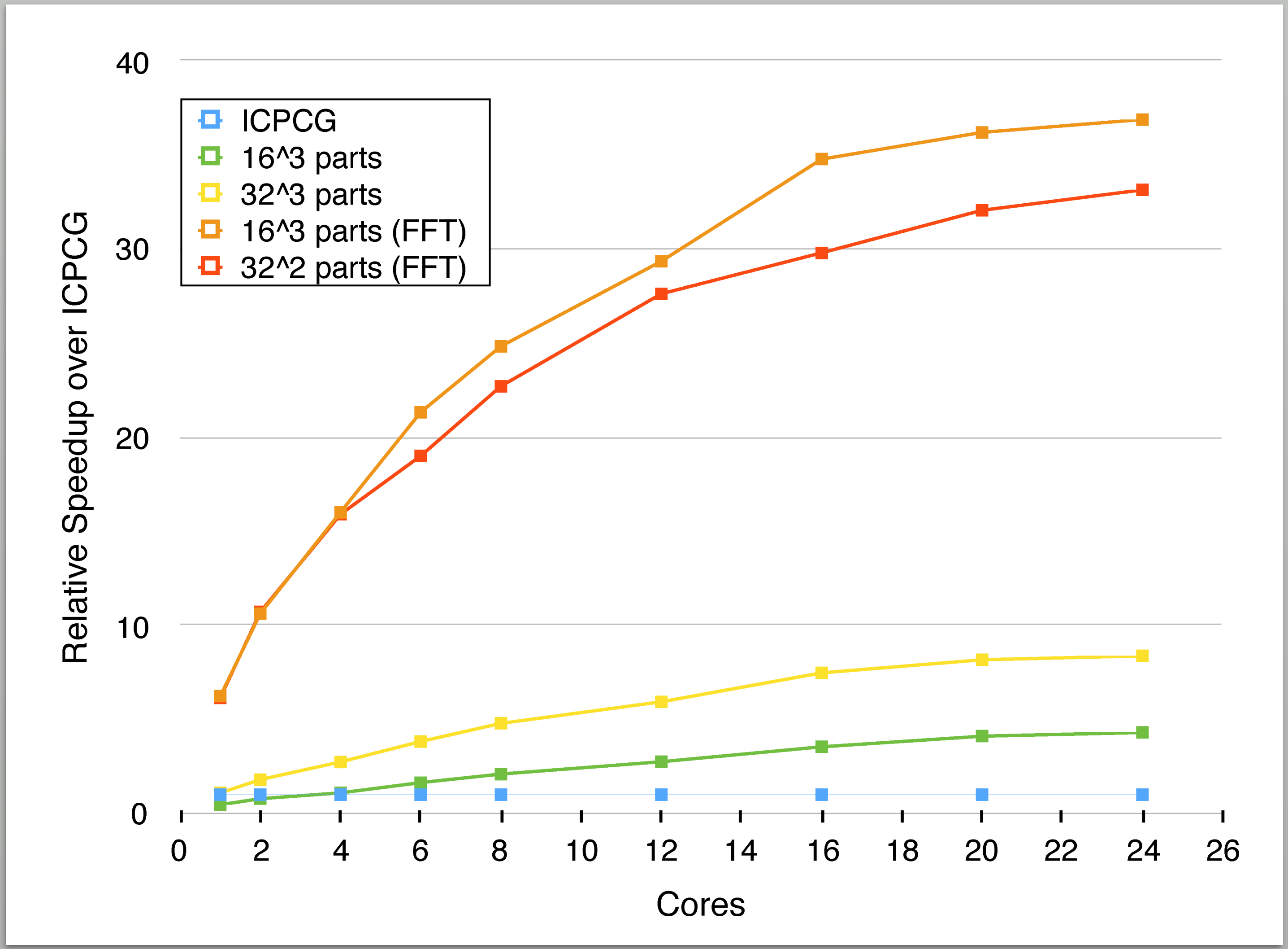
Runtime for 1024^3



Parallel Speedup for 1024^3



Speedup over PCG for 512^3



Comparison with MGPCG

Table VIII. Comparison of iterations and runtimes between MGPCG and Schur complement solver (16^3 subdomains).

	MGPCG iter	MGPCG time	Schur iter	Schur time
Scene1	23	43.35s	15	24.56s
Scene2	24	44.03s	14	22.99s

The results show moderate performance gains for our method.

We consider that multigrid method has some difficulties in the special treatments required to support thin boundaries.

Fluid Simulation Test

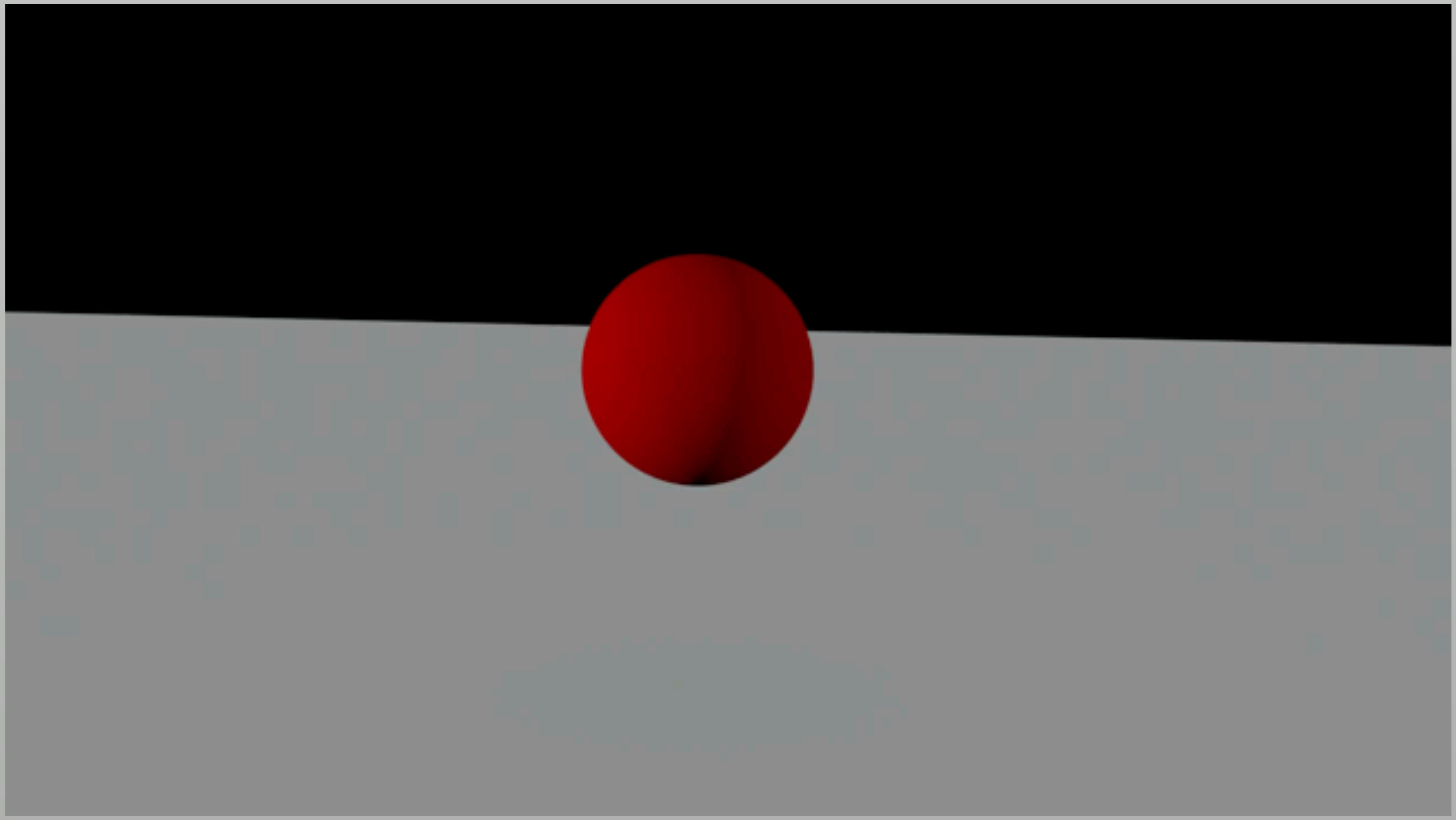
Table IX. Liquid Simulation Time. Resolution 512^3

Fluid Scene	MIC0-PCG ParalletT	Schur SerialT	Schur ParallelT	Speedup over MIC0-PCG
Dam break	261.49s	128.59s	14.96s	17.47
Double dam break	160.44s	126.85S	14.51s	11.05
Drop objects	163.63s	92.83s	11.47s	14.26
Obstacles	371.50s	187.66s	20.08s	18.50

Table X. Smoke Simulation Time. Resolution $512 \times 768 \times 512$

Smoke scene	MIC0-PCG ParallelT	Schur ParallelT	Speedup over MIC0-PCG
Plume	678.99s	47.36s	14.33
Plume with sphere	1108.16s	49.66s	22.31

512 x 768 x 512



49.66 sec / timestep

Conclusion

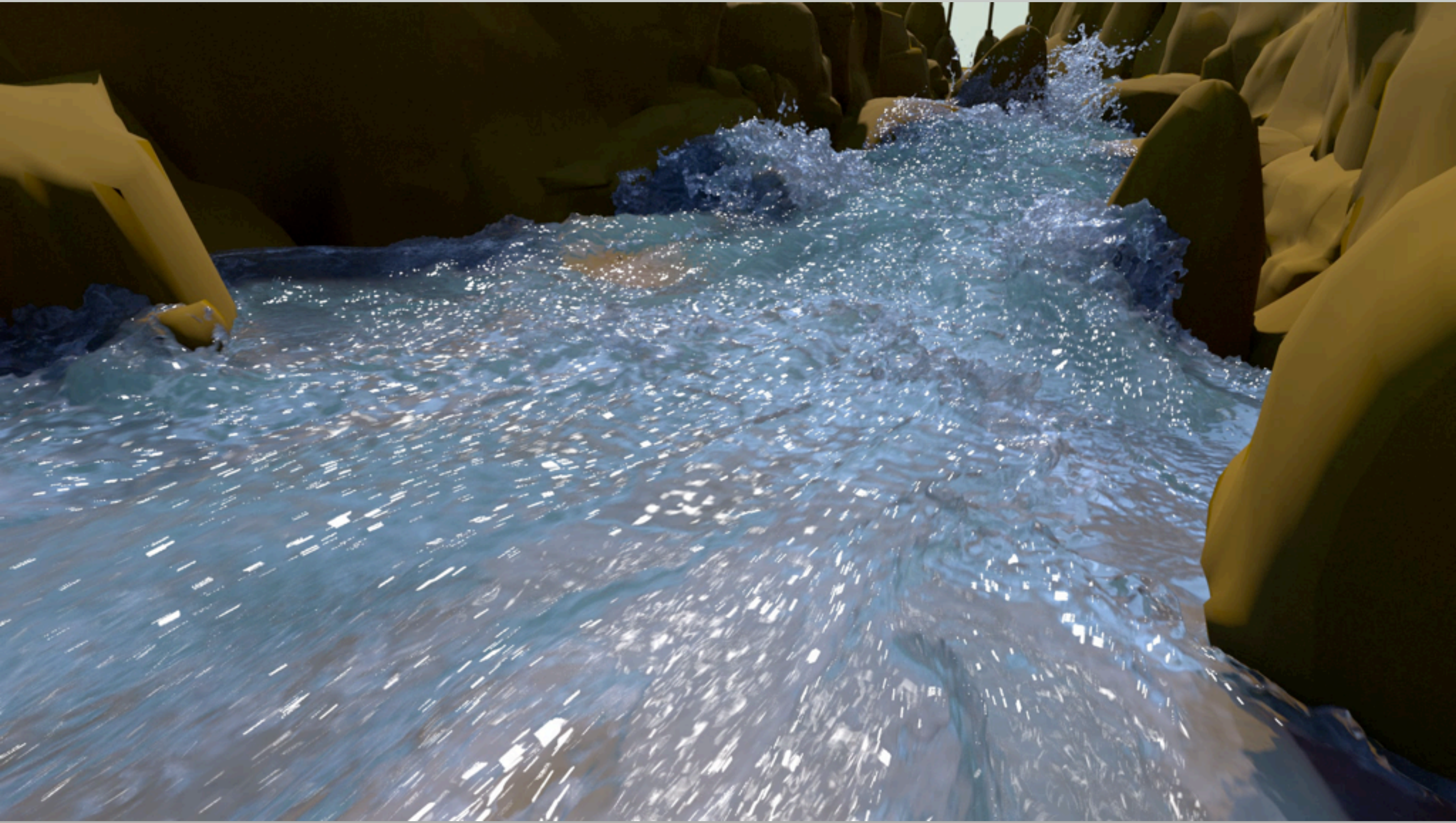
- We use a domain decomposition approach for greater parallelism
- Create a novel Schur complement preconditioner with high convergence rate
- The use of different linear equation solvers in different flow regions
- High parallelism, low computation time and memory cost

Limitations

- Does this actually work?
    - Seems to. But we haven’t proved it.
- We still use PCG, so there’s a serial portion.
- Optimal performance for FFT requires powers of 2 DoF
- High overhead for low to mid-res sims

Still Not Convinced?

- Domain partitioning: efficient implementation
- Domain partitioning: non-uniform subdomains
- Domain partitioning: optimal tile sizes
- Heterogenous computing like Liu et al. 2016
- Distributed computing?
    - Lot of data transfer…
- Adaptive grid methods



谢谢

Thank you!