Faculty of Mathematics
and Physics
Charles University

Scalable Two-Level Schwarz Preconditioners for Discontinuous Galerkin method

---

Tomáš Hammerbauer

Department of Numerical Mathematics, Charles University

May 19, 2026

Outline

1. Motivation & Background
2. Scalability of Preconditioners
3. Hybrid Two-Level Schwarz
4. Conclusions

- *Motivation & Background*
- *Scalability of Preconditioners*
- *Hybrid Two-Level Schwarz*
- *Conclusions*

Motivation & Background

Motivation: Solving PDEs

PDEs model a broad range of physical phenomena — fluid flow, heat transfer, elasticity, and electromagnetic fields. Accurate and efficient solvers for **large-scale problems** are therefore essential.

Model problem — elliptic diffusion equation:

$$-\nabla \cdot (a(x)\nabla u(x)) = f(x) \quad \text{in} \quad \Omega, \quad u(x) = u_D(x) \quad \text{on} \quad \partial\Omega$$

where there $\exists a_0, a_1 > 0$, such that

$$a_0|\xi| \leq |a(x)\xi| \leq a_1|\xi|, \quad \forall x \in \Omega, \forall \xi \in \mathbb{R}^d$$

Note: Fine or adaptively refined meshes lead to **very large linear systems** — efficient preconditioning is essential.

Goal: Develop robust DD preconditioners for general nonlinear PDE problems (e.g., Navier–Stokes, Richards equation).

From FEM to Discontinuous Galerkin

We discretise Ω into a mesh $\mathscr{T}_h$ of triangles (simplexes).

Finite Element Method (FEM)

- Continuous piecewise polynomials:

$$V_h = \{v \in C(\Omega); v|_K \in P(K), \forall K \in \mathscr{T}_h\}$$

- DOFs are **globally coupled** — difficult to parallelise

Discontinuous Galerkin (DG)

- Fully **discontinuous** piecewise polynomials:

$$S_{hp} = \left\{v \in L^2(\Omega); v|_K \in P_{p_K}(K), \forall K \in \mathscr{T}_h\right\}$$

- Enables **local operations** and p-adaptivity
- Discontinuity balanced by a **penalty term**

Discrete problem: Find $u_h \in S_{hp}$ such that

$$\mathcal{A}_h(u_h, v) = (f, v) \qquad \forall v \in S_{hp}$$

Using the basis $\{\varphi_i\}_{i=1}^n$ of S_{hp}, this is equivalent to the linear system $\mathbf{A}\boldsymbol{u} = \boldsymbol{g}$, $\quad \mathbf{A} \in \mathbb{R}^{n\times n}$ **SPD.**

Definition 1 (SIPG bilinear form).

$$\mathcal{A}_h(u,v) = \sum_{K\in\mathscr{T}_h}\int_K \nabla u\cdot\nabla v\mathrm{d}x - \sum_{\gamma\in\mathscr{F}_h^I}\int_\gamma \{\nabla u\}\cdot[v]\cdot\vec{\mathbf{n}}\,\mathrm{d}S - \sum_{\gamma\in\mathscr{F}_h^I}\int_\gamma \{\nabla v\}\cdot[u]\cdot\vec{\mathbf{n}}\,\mathrm{d}S + \sum_{\gamma\in\mathscr{F}_h^I}\frac{\sigma}{h_\gamma}\int_\gamma [u][v]\,\mathrm{d}S$$

Preconditioning the Linear System

Solving $\mathbf{A}\boldsymbol{u} = \boldsymbol{g}$ is typically done by **Krylov iterative methods** (CG, GMRES, ...).

To accelerate convergence we apply **preconditioning**:

$$\mathrm{M}^{-1}\mathbf{A}\boldsymbol{u} = \mathrm{M}^{-1}\boldsymbol{g}$$

We want $\mathbf{M}^{-1} \approx \mathbf{A}^{-1}$ while the application of $\mathbf{M}^{-1}$ is **cheap**. In practice, we solve $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{x}$ at each iteration.

Theorem 1 (CG convergence). The error of the preconditioned CG method satisfies:

$$\|\boldsymbol{u}_* - \boldsymbol{u}_k\| \leq 2 \left[\frac{\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{M}^{-1}\mathbf{A})} + 1} \right]^k \|\boldsymbol{u}_* - \boldsymbol{u}_0\|$$

where $\kappa(\mathbf{M}^{-1}\mathbf{A}) = \|\mathbf{M}^{-1}\mathbf{A}\| \left\| (\mathbf{M}^{-1}\mathbf{A})^{-1} \right\|$.

Note: Minimising $\kappa(\mathbf{M}^{-1}\mathbf{A})$ directly accelerates convergence — **this is the goal of our preconditioners**.

Condition Number for DG

For the DG discretisation, we have the following condition number estimate:

Theorem 2 (DG condition number). For the SIPG method with penalty parameter σ and mesh size h:

$$\kappa(\mathbf{M}^{-1}\mathbf{A}) \leq C\frac{a_1}{a_0}p^4 h^{-2}$$

where p is the maximum polynomial degree and C is a constant independent of h and p.

The condition number grows as $h \to 0$ and $p \to \infty$ – **preconditioning is essential** for large problems.

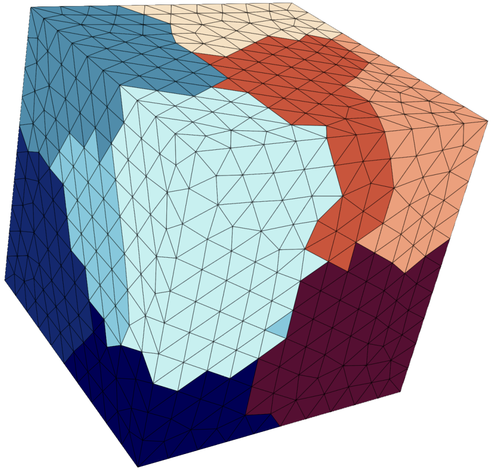
- *Motivation & Background*
- ***Scalability of Preconditioners***
- *Hybrid Two-Level Schwarz*
- *Conclusions*

Scalability of Preconditioners

Domain Decomposition (DD)

Partition the global domain Ω into N **non-overlapping** subdomains:

$$\Omega = \bigcup_{i=1}^{N} \Omega_i$$



Local PDE problems are solved **independently** in each subdomain, yielding local matrices $\mathbf{A}_i$.

Original idea (H.A. Schwarz, 1870): prove existence of solutions on complex domains by alternating between overlapping subdomain problems.

Today: used as efficient **parallel preconditioners** for Krylov methods.

To achieve **scalability** (iteration counts independent of N), we need a **coarse level** to propagate information globally.

For **DG methods**, non-overlapping DD is natural: DOFs are element-local, so restriction to a subdomain is simply selecting the elements $K \in \Omega_i$.

Additive Schwarz Preconditioners

Define restriction matrices $\mathbf{R}_\mathrm{i}$ from Ω to Ω_i and local matrices:

$$\mathbf{A}_i = \mathbf{R}_\mathrm{i} \mathbf{A} \mathbf{R}_\mathrm{i}^T, \quad \forall i = 1, \ldots, N$$

Definition 2 (One-level additive Schwarz (ASM)).

$$\mathbf{M}_{\mathrm{add},1}^{-1} = \sum_{i=1}^{N} \mathbf{R}_\mathrm{i}^T \mathbf{A}_i^{-1} \mathbf{R}_\mathrm{i}$$

The restriction $\mathbf{R}_\mathrm{i}$ selects the rows and columns of $\mathbf{A}$ corresponding to elements in Ω_i.

$$\mathbf{R}_\mathrm{i} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & & \ddots & & \vdots & \vdots & \cdots \\ 0 & \cdots & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n_i \times n}$$

Local problems are solved **independently** in each subdomain.

Two level methods

To propagate information **globally** and achieve scalability, add a **coarse level** ($i = 0$):

Definition 3 (Two-level additive Schwarz).

$$\mathbf{M}_{\mathrm{add},2}^{-1} = \sum_{i=0}^{N} \mathbf{R}_{\mathrm{i}}^{T} \mathbf{A}_{i}^{-1} \mathbf{R}_{\mathrm{i}} = \underbrace{\mathbf{R}_{0} \mathbf{A}_{0}^{-1} \mathbf{R}_{0}^{T}}_{\texttt{coarse correction}} + \sum_{i=1}^{N} \mathbf{R}_{\mathrm{i}}^{T} \mathbf{A}_{i}^{-1} \mathbf{R}_{\mathrm{i}}$$

where $\mathbf{R}_0 : \mathbb{R}^{n_0} \to \mathbb{R}^{n}$ is the **prolongation** operator, $\mathbf{A}_0 = \mathbf{R}_0^T \mathbf{A} \mathbf{R}_0$.

The coarse prolongation $\mathbf{R}_0$ is defined by a **coarse mesh** $\mathscr{T}_H$ with $H \gg h$ and a corresponding coarse space S_{Hq}.

It is constructed as an interpolation operator mapping the coarse space S_{Hq} to the fine DG space S_{hp}.

$$\mathbf{R}_0 \in \mathbb{R}^{n \times n_0}, \quad n_0 \ll n$$

Note: The coarse solve propagates low-frequency information across the entire domain — essential for scalability.

Coarse space construction

Coarse space design is crucial for the performance of the two-level method.

We chose the polynomial degree of the coarse space on element $\mathcal{K} \in \mathcal{T}_H$ as $q_{\mathcal{K}} = \min_{\forall K \subset \mathcal{K}}(p_K)$.

By $\overline{K}$ we denote the reference fine element in the middle of the coarse element $\mathcal{K}$.

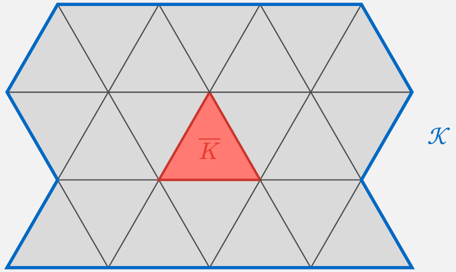
The basis functions Φ_j of the coarse space S_{Hq} defined by

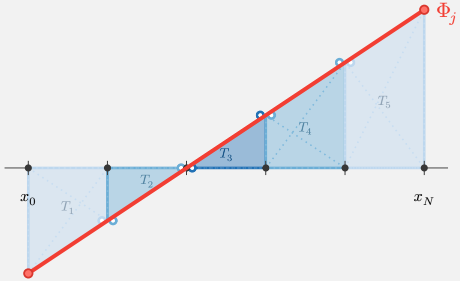
$$\Phi_j\,|_K = \sum_{i=1}^{n} \mathbf{R}_0(i,j)\varphi_i^K, \quad \forall j = 1, \ldots, n_0,$$

where φ_i^K are the basis functions of the fine DG space S_{hp} on element K.

The coefficients $\mathbf{R}_0(i,j)$ are determined by a **cell-local L^2 projection** of the coarse polynomial modes $\varphi_j^{\overline{K}}$ onto the fine DG basis functions φ_i^K on each fine cell $K \subset \mathcal{K}$:

The coarse space satisfies $S_{Hq} \subset S_{hp}$.





Parallel computing: Weak and Strong Scaling

When dealing with parallel computations we often speak about **strong** and **weak** scaling.

Definition 4 (Weak scaling). Weak scaling refers to how the computation time varies with the number of processors when the **problem size per processor is fixed**.

Ideal: time = const.

Note: To this end, we assume that the coarse space is small enough so that we can neglect the cost and communication of the coarse solve. In practice, this is often not the case and the coarse solve can become a bottleneck for scalability.

- *Motivation & Background*
- *Scalability of Preconditioners*
- ***Hybrid Two-Level Schwarz***
- *Conclusions*

Hybrid Two-Level Schwarz

Hybrid Two-Level Schwarz

The (symmetrised) **hybrid** variant applies the coarse correction in a multiplicative manner with respect to the fine-level smoother.

Definition 5 (Hybrid two-level preconditioner).

$$\mathbf{M}_{\mathrm{hyb}}^{-1} = \mathbf{R}_0\mathbf{A}_0^{-1}\mathbf{R}_0^T + (\mathrm{I} - \mathbf{R}_0\mathbf{A}_0^{-1}\mathbf{R}_0^T\mathbf{A})\mathbf{M}_{\mathrm{add},1}^{-1}(\mathrm{I} - \mathbf{R}_0\mathbf{A}_0^{-1}\mathbf{R}_0^T\mathbf{A})^T$$

Additive $(\mathbf{M}_{\mathrm{add},2}^{-1})$

$$\kappa\left(\mathbf{M}_{\mathrm{add},2}^{-1}\mathbf{A}\right) \leq C\frac{H}{h}\frac{p^2}{q}(N_S + 2)$$

- All corrections applied simultaneously

Hybrid $(\mathbf{M}_{\mathrm{hyb}}^{-1})$

$$\kappa\left(\mathbf{M}_{\mathrm{hyb}}^{-1}\mathbf{A}\right) \leq C\frac{H}{h}\frac{p^2}{q}(N_S + 1)$$

- Need to apply the coarse correction first and then the fine-level smoother

Only slightly more expensive than $\mathbf{M}_{\mathrm{add},2}^{-1}$ but with a slightly **tighter condition number bound**.

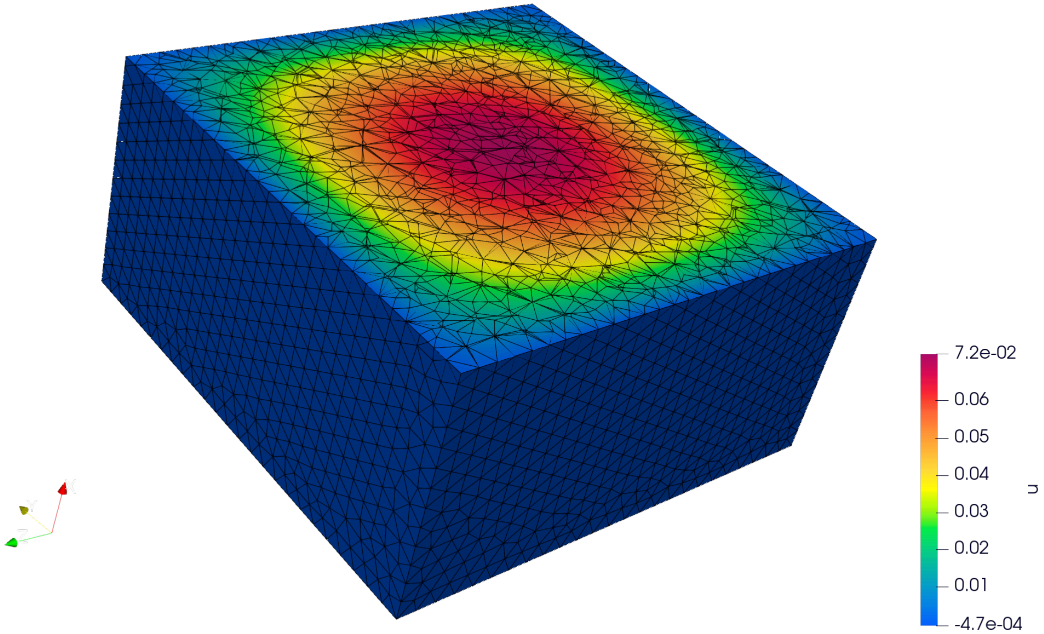
Our main goal is to demonstrate the **scalability** and **computational advantages** of the hybrid method compared to the additive preconditioner.

Weak Scaling Results (2D)

					$p = 1$			$p = 2$			$p = 3$		
	$\#\mathcal{T}_h$	N	$\#\mathcal{T}_{h,i}$	$\#\mathcal{T}_H$	**iter**	t_{solve} [s]	t_{comm} [s]	**iter**	t_{solve} [s]	t_{comm} [s]	**iter**	t_{solve} [s]	t_{comm} [s]
$\mathbf{M}_{\text{add},2}^{-1}$	3872	4	968	16	**83**	9.547e-02	9.240e-04	**128**	6.310e-01	1.502e-03	**157**	1.826e+00	1.968e-03
	7938	8	992	32	**94**	1.046e-01	1.465e-03	**143**	8.454e-01	2.400e-03	**179**	2.694e+00	3.291e-03
	16200	17	952	68	**99**	1.678e-01	2.125e-03	**142**	8.493e-01	3.411e-03	**171**	1.999e+00	4.690e-03
	31752	33	962	132	**105**	1.659e-01	2.943e-03	**153**	7.311e-01	5.048e-03	**178**	2.432e+00	7.049e-03
	63368	65	974	260	**109**	1.792e-01	4.053e-03	**152**	8.552e-01	7.136e-03	**172**	2.388e+00	1.031e-02
	119072	123	968	492	**106**	1.651e-01	5.302e-03	**142**	7.005e-01	9.727e-03	–	–	–
	239432	247	969	988	**108**	3.221e-01	7.904e-03	–	–	–	–	–	–
$\mathbf{M}_{\text{hyb}}^{-1}$	3872	4	968	16	**63**	5.611e-02	8.109e-04	**97**	2.773e-01	1.313e-03	**119**	9.481e-01	1.716e-03
	7938	8	992	32	**67**	5.927e-02	1.206e-03	**107**	4.094e-01	2.067e-03	**134**	1.025e+00	2.825e-03
	16200	17	952	68	**73**	7.513e-02	1.806e-03	**106**	3.557e-01	2.921e-03	**128**	1.045e+00	4.008e-03
	31752	33	962	132	**76**	1.067e-01	2.448e-03	**110**	4.871e-01	4.145e-03	**129**	1.398e+00	5.801e-03
	63368	65	974	260	**76**	1.766e-01	3.234e-03	**105**	7.535e-01	5.598e-03	**119**	2.076e+00	8.052e-03
	119072	123	968	492	**76**	2.550e-01	4.327e-03	**105**	1.242e+00	8.117e-03	–	–	–
	239432	247	969	988	**73**	4.887e-01	6.039e-03	–	–	–	–	–	–

Table 1: 2D two-level additive preconditioner $\mathbf{M}_{\text{add},2}^{-1}$ (upper) and 2D hybrid preconditioner $\mathbf{M}_{\text{hyb}}^{-1}$ (lower)

3D Test Problem



Solution of a 3D domain decomposition problem with the hybrid two-level Schwarz preconditioner.

Weak Scaling Results (3D)

					$p = 1$			$p = 2$		
	$\#\mathcal{T}_h$	N	$\#\mathcal{T}_{h,i}$	$\#\mathcal{T}_H$	**iter**	t_{solve} [s]	t_{comm} [s]	**iter**	t_{solve} [s]	t_{comm} [s]
$\mathbf{M}_{\text{add},2}^{-1}$	4636	9	515	36	**85**	1.709e-01	1.434e-03	**137**	2.100e+00	2.682e-03
	15923	31	513	124	**97**	2.112e-01	2.802e-03	**153**	1.819e+00	5.845e-03
	36644	71	516	284	**102**	3.817e-01	4.305e-03	**159**	2.235e+00	1.010e-02
	71703	139	515	556	**105**	1.003e+00	6.319e-03	–	–	–
	178569	347	514	1388	**109**	2.920e+00	1.175e-02	–	–	–
$\mathbf{M}_{\text{hyb}}^{-1}$	4636	9	515	36	**62**	7.048e-02	1.206e-03	**103**	9.421e-01	2.311e-03
	15923	31	513	124	**70**	2.182e-01	2.319e-03	**116**	1.955e+00	5.035e-03
	36644	71	516	284	**73**	5.811e-01	3.513e-03	**118**	3.739e+00	8.453e-03
	71703	139	515	556	**76**	1.316e+00	5.182e-03	–	–	–
	178569	347	514	1388	**77**	3.647e+00	9.318e-03	–	–	–

Table 2: 3D two-level additive preconditioner $\mathbf{M}_{\text{add},2}^{-1}$ (upper) and 3D hybrid preconditioner $\mathbf{M}_{\text{hyb}}^{-1}$ (lower)

Non-symmetric Problems

Symmetric (SPD) — e.g., pure diffusion

$$-\nabla \cdot (a\nabla u) = f$$

- $\mathbf{A}_i$ is SPD → direct Cholesky solve
- CG applicable
- Condition number bounds well-established

Non-symmetric — e.g., convection-diffusion

$$-\nabla \cdot (a\nabla u) + \boldsymbol{b} \cdot \nabla u = f$$

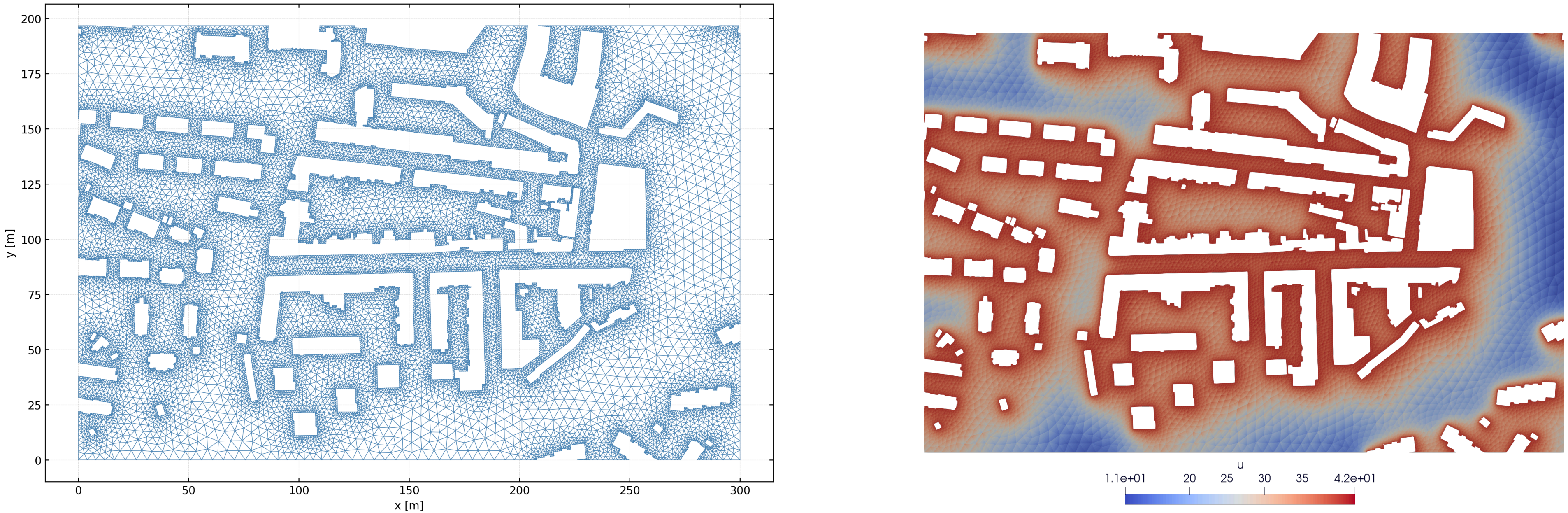
- $\mathbf{A}_i$ no longer SPD → LU locally
- GMRES required globally
- Condition number bounds more complex, often involving **field of values** or **generalised stability conditions**.

Key challenge

Loss of symmetry breaks standard coercivity estimates — the Schwarz framework requires **generalised stability conditions** on the local solvers and coarse space.

Note: Despite the lack of symmetry, numerical experiments confirm that both $\mathbf{M}_{\mathrm{add},2}^{-1}$ and $\mathbf{M}_{\mathrm{hyb}}^{-1}$ retain good scalability — iteration counts grow only mildly as N increases.

Motivating Example: Urban Domain



Mesh of an urban domain — triangulation $\mathscr{T}_h$ (left) and solution of a steady-state convection-diffusion-reaction equation (right).

Non-symmetric scalability

					$p = 1$			$p = 2$		
	$\#\mathcal{T}_h$	N	$\#\mathcal{T}_{h,i}$	$\#\mathcal{T}_H$	**iter**	t_{solve} [s]	t_{comm} [s]	**iter**	t_{solve} [s]	t_{comm} [s]
$\mathbf{M}_{\text{add},2}^{-1}$	6335	18	351	108	**60**	1.702e-02	1.399e-03	**106**	1.294e-01	2.902e-03
	18079	40	451	240	**85**	3.279e-02	2.832e-03	**130**	2.327e-01	5.503e-03
	36952	80	461	480	**95**	6.188e-02	4.444e-03	**140**	4.224e-01	9.073e-03
	61822	140	441	840	**97**	1.143e-01	6.201e-03	**143**	9.101e-01	1.365e-02
	108966	240	454	1440	**102**	2.359e-01	9.177e-03	–	–	–
$\mathbf{M}_{\text{hyb}}^{-1}$	6335	18	351	108	**47**	2.164e-02	1.260e-03	**81**	1.450e-01	2.534e-03
	18079	40	451	240	**65**	5.835e-02	2.477e-03	**102**	3.187e-01	4.900e-03
	36952	80	461	480	**71**	1.086e-01	3.778e-03	**110**	7.935e-01	8.039e-03
	61822	140	441	840	**73**	1.904e-01	5.279e-03	**110**	1.517e+00	1.178e-02
	108966	240	454	1440	**77**	3.980e-01	7.794e-03	–	–	–

Table 3: Urban non-symmetric results for $\mathbf{M}_{\text{add},2}^{-1}$ and $\mathbf{M}_{\text{hyb}}^{-1}$

- *Motivation & Background*
- *Scalability of Preconditioners*
- *Hybrid Two-Level Schwarz*
- ***Conclusions***

Conclusions

Summary and Outlook

Summary

- DG via SIPG for symmetric elliptic PDEs — fully discontinuous spaces, yielding SPD linear systems.
- DD as a preconditioner: one-level ASM (parallel, not scalable) → two-level (scalable).
- Hybrid $\mathbf{P}_{\mathrm{hyb}}$: interleaves coarse and fine corrections, achieving faster convergence.
- Scalability demonstrated in 2D and 3D symmetric, and 2D non-symmetric problems.

Outlook

- Run time performance and parallel efficiency of $\mathbf{M}_{\mathrm{hyb}}^{-1}$ vs. $\mathbf{M}_{\mathrm{add},2}^{-1}$ for large 3D problems on cluster environments.
- Different coarse space constructions and their impact on scalability.
- Adaptive subdomain construction and coarse mesh construction.

Thank You! Questions & Discussion