Received: 29 June 2023 | Accepted: 22 August 2023

DOI: 10.1002/pamm.202300077

RESEARCH ARTICLE

Coupled clustering strategies for hierarchical matrix preconditioners in saddle point problems

Jonas Grams | **Sabine Le Borne**

Hamburg University of Technology,
Institute of Mathematics, Hamburg,
Germany

Correspondence
Jonas Grams, Hamburg University of
Technology, Institute of Mathematics, Am
Schwarzenberg-Campus 3, 21073
Hamburg.
Email: jonas.grams@tuhh.de

Abstract

Fluid flow problems can be modeled by the Navier–Stokes or, after linearization, by the Oseen equations. Their discretization results in discrete saddle point problems. These systems of equations are typically very large and need to be solved iteratively. Standard (block-) preconditioning techniques for saddle point problems rely on an approximation of the Schur complement. Such an approximation can be obtained by a hierarchical ($\mathcal{H}$-) matrix LU-decomposition, which first approximates the Schur complement explicitly. The computational complexity of this computation depends, among other things, on the hierarchical block structure of the involved matrices. However, widely used techniques do not consider the connection between the discretization grids for the velocity field and the pressure, respectively. Here, we present a hierarchical block structure for the finite element discretization of the gradient operator that is improved by considering the connection between the two involved grids. Numerical results imply that the improved block structure allows for a faster computation of the Schur complement, which is the bottleneck for the set-up of the $\mathcal{H}$-matrix LU-decomposition.

1 | INTRODUCTION

The Navier–Stokes and Oseen equations are common models for fluid flow problems, and the discretization of these equations typically leads to large-scale saddle point problems. These linear systems are typically not only large but also sparse, non-symmetric, and indefinite. Thus, Krylov space methods are often used to solve these systems. In order to achieve a fast convergence, a preconditioner is required. Here, we will pursue a preconditioning approach based on an $\mathcal{H}$-LU decomposition of the stiffness matrix, which requires the computation and $\mathcal{H}$-LU decomposition of the pressure Schur complement.

Hierarchical ($\mathcal{H}$-) matrices were first introduced 1999 [1] and provide accurate approximations of matrix arithmetic (as shown, e.g., in [2, 3] for finite element matrices) with a log-linear complexity for memory and matrix-vector operations. The log-linear complexity is achieved by a hierarchical block structure with dense as well as low-rank matrix blocks. This block structure is typically built upon partitions of the involved index sets obtained from cluster trees, see Section 3 for details.

The discretization of the Oseen problem typically incorporates two index sets, one for the velocity field and one for the pressure. In previous work [4, 5], these two index sets have been clustered independently of each other. In Section 4, based upon observations regarding the sparsity structure of the off-diagonal blocks of the saddle point system describing the coupling between velocity and pressure, we will introduce a new cluster strategy that couples the clustering of the velocity index set with the clustering of the pressure index set. This new coupled clustering will create larger zero subblocks accelerating the subsequent $\mathcal{H}$-matrix–matrix multiplications in the computation of the Schur complement and its $\mathcal{H}$-LU factorization.

In Section 5, we will then provide numerical results for the performance of $\mathcal{H}$-LU preconditioners resulting from both uncoupled and coupled clustering strategies.

2 | MODEL PROBLEM

We consider a three-dimensional Oseen problem with Dirichlet boundary conditions: find functions $\mathbf{u} : \Omega \to \mathbb{R}^3$ and $p : \Omega \to \mathbb{R}$ with

$$\Delta\mathbf{u} + \mathbf{b}\cdot\nabla\mathbf{u} + \nabla p = \mathbf{f} \quad \text{on } \Omega := (-1,1)^3, \quad \nabla\cdot\mathbf{u} = \mathbf{0} \quad \text{on } \Omega, \quad \mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega. \tag{1}$$

A finite element discretization of this problem with Taylor–Hood elements and piecewise-linear basis functions is based on two grids: a coarse grid $\mathcal{T}_h$ for the pressure discretization, and a refined grid $\mathcal{T}_{h/2}$ for the velocity field discretization with respective bases $(\Psi_1, \dots, \Psi_M)$ and $(\varphi_1, \dots, \varphi_n)$. The discretization leads to a saddle point problem of the form

$$\begin{pmatrix} \mathcal{F} & \mathcal{B}^\top \\ \mathcal{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{F} & & & \mathbf{B}_1^\top \\ & \mathbf{F} & & \mathbf{B}_2^\top \\ & & \mathbf{F} & \mathbf{B}_3^\top \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{0} \end{pmatrix}, \tag{2}$$

with $\mathbf{F} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ and $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3 \in \mathbb{R}^{\mathcal{J}\times\mathcal{I}}$ where $\mathcal{I} = \{1, \dots, n\}$ and $\mathcal{J} = \{1, \dots, M\}$ denote the velocity and pressure index sets, respectively. A popular approach for preconditioning of such saddle point problems is derived from the block factorization

$$\begin{pmatrix} \mathcal{F} & \mathcal{B}^\top \\ \mathcal{B} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \\ \mathcal{B}\mathcal{F}^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathcal{F} & \\ & \mathbf{S} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathcal{F}^{-1}\mathcal{B}^\top \\ & \mathbf{I} \end{pmatrix} \approx \begin{pmatrix} \mathcal{F} & \\ & \mathbf{S} \end{pmatrix} \approx \begin{pmatrix} \widetilde{\mathcal{F}} & \\ & \widetilde{\mathbf{S}} \end{pmatrix} = \mathcal{P}$$

with the Schur complement $\mathbf{S} = -\mathcal{B}\mathcal{F}^{-1}\mathcal{B}^\top \in \mathbb{R}^{\mathcal{J}\times\mathcal{J}}$. While block triangular or block-LDU preconditioners are also possible [6], here, we will consider the block diagonal preconditioner $\mathcal{P}$, where $\widetilde{\mathcal{F}}, \widetilde{\mathbf{S}}$ are easily invertible approximations to $\mathcal{F}, \mathbf{S}$ in the form of $\mathcal{H}$-LU decompositions. In the following, we will describe the necessary operations (steps) for the $\mathcal{H}$-LU factorization of the Schur complement and will later (in Section 5) analyze their respective computational costs for different clustering strategies.

$$\begin{aligned}
\mathbf{S} = -\mathcal{B}\mathcal{F}^{-1}\mathcal{B}^\top = -\sum_{i=1}^{3} \mathbf{B}_i\mathbf{F}^{-1}\mathbf{B}_i^\top & \\
\approx -\sum_{i=1}^{3} \mathbf{B}_i(\mathbf{L}_\mathbf{F}\mathbf{U}_\mathbf{F})_\mathcal{H}^{-1}\mathbf{B}_i^\top & \qquad \text{①} \quad \mathcal{H}-\mathrm{LUdecomposition}\mathbf{F} \approx \mathbf{L}_\mathbf{F}\mathbf{U}_\mathbf{F} \\
\approx -\sum_{i=1}^{3} \left(\mathbf{B}_i\mathbf{U}_\mathbf{F}^{-1}\right)_\mathcal{H}\left(\mathbf{L}_\mathbf{F}^{-1}\mathbf{B}_i^\top\right)_\mathcal{H} & \qquad \text{②, ③} \quad \mathrm{Compute}\mathbf{W}_\mathrm{i} = \mathbf{L}_\mathbf{F}^{-1}\mathbf{B}_\mathrm{i}^\top, \mathrm{and}\mathbf{V}_\mathrm{i} = \mathbf{B}_\mathrm{i}\mathbf{U}_\mathbf{F}^{-1}, \mathrm{i} = 1, 2, 3 \\
\approx -\sum_{i=1}^{3} \left(\mathbf{B}_i\mathbf{U}_\mathbf{F}^{-1}\right)_\mathcal{H} \odot_\mathcal{H} \left(\mathbf{L}_\mathbf{F}^{-1}\mathbf{B}_i^\top\right)_\mathcal{H} & \qquad \text{④} \quad \mathrm{Computeandsumuptheproducts}\widetilde{\mathbf{S}} = \sum_{\mathrm{i}=1}^{3} \mathbf{V}_\mathrm{i}\mathbf{W}_\mathrm{i} \\
\approx (\mathbf{L}_\mathbf{S}\mathbf{U}_\mathbf{S})_\mathcal{H} & \qquad \text{⑤}\mathcal{H}-\mathrm{LUdecomposition}\widetilde{\mathbf{S}} \approx \mathbf{L}_\mathbf{S}\mathbf{U}_\mathbf{S}
\end{aligned}$$

3 | CLUSTER- AND BLOCK CLUSTER TREES

For now let $\mathcal{I}$ and $\mathcal{J}$ be arbitrary index sets. Then, the block structure of a hierarchical matrix $\mathbf{H} \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ is based on (hierarchies of) partitions of $\mathcal{I}$ and $\mathcal{J}$ is obtained by clustering an index set via a tree structure as described, for example, in [7].

Definition 3.1 (Cluster tree). Let $\mathcal{I}$ be an index set and $\mathcal{T} = (V, E)$ a tree with $V \subseteq \mathcal{P}(\mathcal{I})$. Then, $\mathcal{T}$ is called a cluster tree for the index set $\mathcal{I}$, if

1. the root of $\mathcal{T}$ is given by $\mathrm{root}(\mathcal{T}) = \mathcal{I}$,
2. for each node $t \in V$ with sons $\mathrm{sons}(t)$ there holds $t = \bigcup_{t' \in \mathrm{sons}(t)} t'$ and all sons are pairwise disjoint.

We denote cluster trees for an index set $\mathcal{I}$ as $\mathcal{T}_{\mathcal{I}}$ and the set of leafs of $\mathcal{T}_{\mathcal{I}}$ (i.e., the nodes $t \in \mathcal{T}_{\mathcal{I}}$ with $\mathrm{sons}(t) = \emptyset$) as $\mathcal{L}_{\mathcal{I}}$.

The set of leafs $\mathcal{L}_{\mathcal{I}}$ yields a partition of the index set $\mathcal{K}$ (and $\mathcal{L}_{\mathcal{J}}$ for $\mathcal{J}$, respectively). In Section 4.1, we will provide a sketch for two popular techniques to construct a cluster tree.

Two cluster trees $\mathcal{T}_{\mathcal{I}}, \mathcal{T}_{\mathcal{J}}$ can be used to construct a block cluster tree whose leaves provide a partition of $\mathcal{I} \times \mathcal{J}$.

Definition 3.2 (Block tree). Let $\mathcal{I}, \mathcal{J}$ be index sets, $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$ corresponding cluster trees, and $\mathcal{T} = (V, E)$ a tree with $V \subseteq \mathcal{P}(\mathcal{I} \times \mathcal{J})$. Then, $\mathcal{T}$ is called a block tree for the cluster trees $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$ if

1. all $b \in \mathcal{T}$ are of the form $b = t \times s$ for $t \in \mathcal{T}_{\mathcal{I}}$ and $s \in \mathcal{T}_{\mathcal{J}}$,
2. the root of $\mathcal{T}$ is $\mathrm{root}(\mathcal{T}) := \mathcal{I} \times \mathcal{J}$
3. if $b = t \times s \in \mathcal{T}$ has sons, then there holds $\mathrm{sons}(b) = \{t' \times s' : t' \in \mathrm{sons}(t), s' \in \mathrm{sons}(s)\}$.

In the following, we denote a block tree for cluster trees $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$ by $\mathcal{T}_{\mathcal{I}\times\mathcal{J}}$ and the set of its leaves by $\mathcal{L}_{\mathcal{I}\times\mathcal{J}}$.

The set of leaves $\mathcal{L}_{\mathcal{I}\times\mathcal{J}}$ yields a partition of the product index set $\mathcal{I} \times \mathcal{J}$. In order to distinguish between dense and low-rank matrix blocks, an admissibility condition $\mathrm{adm} : \mathcal{T}_{\mathcal{I}} \times \mathcal{T}_{\mathcal{J}} \to \{\mathrm{true}, \mathrm{false}\}$ is used. Admissible blocks $\mathbf{H}|_{t\times s}$ are represented as low rank blocks, whereas inadmissible matrix blocks $\mathbf{H}|_{t\times s}$ are represented as dense matrix blocks.

4 | (UN-)COUPLED CLUSTERING STRATEGIES

In order to compute the $\mathcal{H}$-LU preconditioner as described in Section 2, we need three block cluster trees. One for the matrix $\mathbf{F} \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$, one for all of the matrices $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3 \in \mathbb{R}^{\mathcal{J}\times\mathcal{I}}$, and one for the Schur complement $\mathbf{S} = -\sum_{i=1}^{3} \mathbf{B}_i \mathbf{F}^{-1} \mathbf{B}_i^{\mathsf{T}} \in \mathbb{R}^{\mathcal{J}\times\mathcal{J}}$. These block cluster trees $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}, \mathcal{T}_{\mathcal{J}\times\mathcal{I}}, \mathcal{T}_{\mathcal{J}\times\mathcal{J}}$ are based upon two cluster trees $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$ for the velocity and pressure index sets $\mathcal{I}$ and $\mathcal{J}$.

In this section, we will review and analyze a first approach for the construction of these cluster trees as it was previously described in [4]. Here, the cluster trees $\mathcal{T}_{\mathcal{I}}$ are constructed independently of each other. Based on this we will then introduce a cluster strategy that couples the clustering of the velocity and pressure index sets, leading to modified (block-) cluster trees $\mathcal{T}_{\mathcal{I}}$, $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$, and $\mathcal{T}_{\mathcal{J}\times\mathcal{I}}$.

4.1 | Uncoupled clustering

Since $\mathbf{F}$ is a sparse finite element stiffness matrix, a domain decomposition based clustering as described in [7, 8] is an adequate approach for the cluster tree $\mathcal{T}_{\mathcal{I}}$. Since the Schur complement is dense, a geometry-based bisection clustering is an appropriate choice for the cluster tree $\mathcal{T}_{\mathcal{J}}$. Figure 1 illustrates the bisection clustering of the pressure grid and the domain decomposition clustering for the velocity grid for 2D example grids.

On each level of clustering, the index sets are reordered. In the first step, the pressure index set is subdivided into the two clusters $\mathcal{J}_0^1$, $\mathcal{J}_1^1$ containing all nodes on the left and right half of the domain, respectively. The velocity index set is

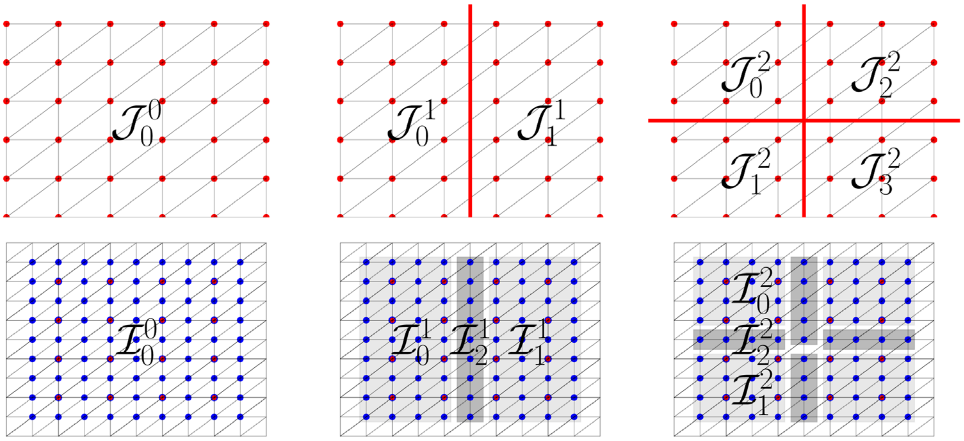

FIGURE 1 2D example of the bisection clustering of the pressure discretization index set (top) and the domain decomposition based clustering for the velocity discretization index set (bottom).

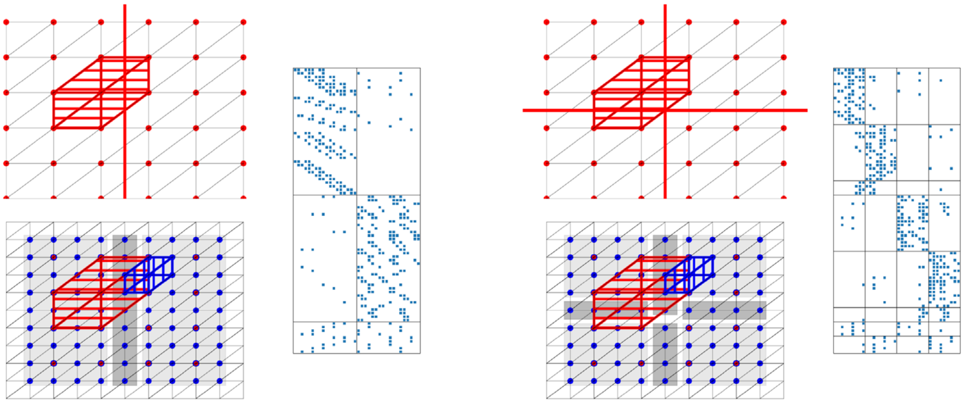

FIGURE 2 From left to right: pressure and velocity grid on the first level of clustering with the supports of a node from the left pressure cluster (red) and a node from the right domain cluster (blue), sparsity pattern of $\mathbf{B}_i (i = 1, 2)$ reordered according to the first level of clustering, pressure and velocity grid on the second level of clustering with the same supports as before, and the sparsity pattern of $\mathbf{B}_i (i = 1, 2)$ reordered according to the second level of clustering.

subdivided into three clusters $\mathcal{I}_0^1, \mathcal{I}_1^1, \mathcal{I}_2^1$, one interface cluster $\mathcal{I}_2^1$ separating the two domain clusters $\mathcal{I}_0^1, \mathcal{I}_1^1$ containing nodes from the left and right half of the domain, respectively. We denote the set of domain clusters with

$$C_{\text{dom}}(\mathcal{T}_\mathcal{I}) := \{s \in \mathcal{T}_\mathcal{I} \ : \ s \text{ is a domain cluster}\}.$$

The clustering implies the following ordering of the index sets: indices in $\mathcal{J}_0^1, \mathcal{I}_0^1$ are ordered first, followed by indices in $\mathcal{J}_1^1, \mathcal{I}_1^1$ whereas indices in the interface cluster $\mathcal{I}_2^1$ are ordered last. Although these cluster strategies are a good choice for the matrix $\mathbf{F}$ and the Schur complement $\mathbf{S}$, they are not chosen with the sparsity structure of the (sparse) matrices $\mathbf{B}_i$ in mind. The blocks $\mathbf{B}_i|_{\mathcal{J}_0^1 \times \mathcal{I}_1^1}$ and $\mathbf{B}_i|_{\mathcal{J}_1^1 \times \mathcal{I}_1^1}$ typically have a few non-zero entries and might be subdivided further. As illustrated in Figure 2, these non-zero entries are generated by overlaps of supports of pressure basis functions from $\mathcal{J}_0^1$ and supports of velocity basis function from $\mathcal{I}_1^1$ (or supports of pressure basis functions from $\mathcal{J}_1^1$ and supports of velocity basis functions from $\mathcal{I}_0^1$, respectively).

4.2 | Coupled clustering

In the previous section, we have shown that the uncoupled clustering produces blocks with only a few non-zero entries, which are produced by overlapping supports of pressure and velocity basis functions. We now propose a new strategy for the clustering of the velocity grid.

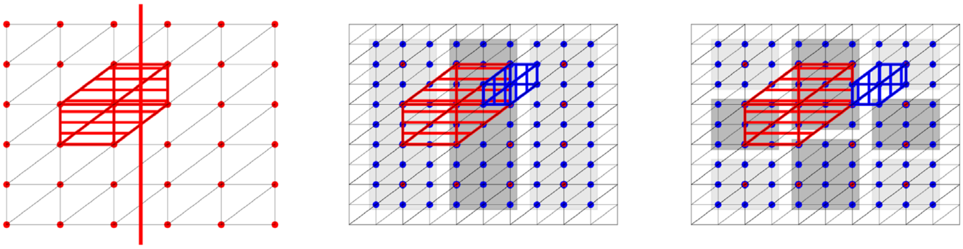

FIGURE 3 Left: pressure grid and support of a node of the left cluster. Middle: velocity grid, support of the node from the pressure grid and one node of the interface that was in the right domain for the (uncoupled) domain decomposition clustering. Right: velocity grid, support of the node from the pressure grid and the support of a node from the (new, smaller) right domain cluster.

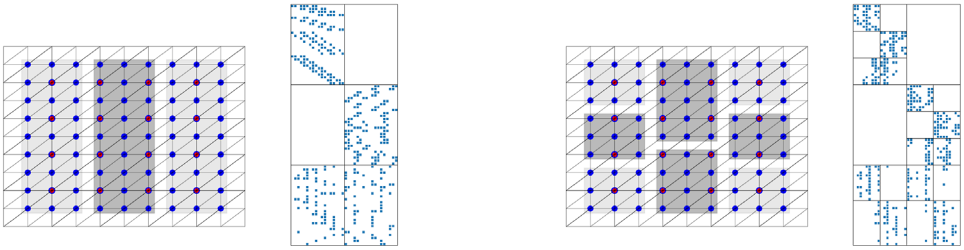

FIGURE 4 Coupled clustering of the 2D example velocity grid on the first level (left) and the second level (right), and the respective sparsity pattern of the (reordered) matrices $\mathbf{B}_i, (i = 1, 2)$.

The pressure index set $\mathcal{J}$ is subdivided into two clusters $\mathcal{J}_0^1, \mathcal{J}_0^1$ as before while the velocity index set $\mathcal{I}$ is subdivided into two smaller domain clusters $\mathcal{I}_0^1, \mathcal{I}_1^1$, and a larger interface cluster $\mathcal{I}_2^1$. The few nodes from the velocity grid, which yield these non-zero entries are shifted to the interface cluster as shown in Figure 3.

As a result, the blocks $\mathbf{B}_i|_{\mathcal{J}_0^1 \times \mathcal{I}_1^1}$ and $\mathbf{B}_i|_{\mathcal{J}_1^1 \times \mathcal{I}_0^1}$ are zero blocks since there are no overlapping supports between the respective basis functions. This clustering yields the sparsity structure shown in Figure 4 for the blocks $\mathbf{B}_i$ in the 2D example.

To shift the nodes producing overlapping supports, we can use the sparsity structure of one of the blocks $\mathbf{B}_i$, for example, $\mathbf{B}_1$. Additionally, we assume that there already is a (bisection) cluster tree $\mathcal{T}_\mathcal{J}$ for the pressure index set $\mathcal{J}$. Then, each domain cluster of $\mathcal{T}_\mathcal{I}$ is subdivided with respect to an associated cluster in $\mathcal{T}_\mathcal{J}$, where we start clustering $\mathcal{I}$ with respect to the root $\mathcal{J}$ of the pressure cluster tree. A domain cluster $s \subseteq \mathcal{I}$ is subdivided with respect to its associated cluster $t \in \mathcal{T}_\mathcal{J}$ as follows ($\mathrm{nnz}(\mathbf{B}_1)$ denotes the index set of nonzero entries in $\mathbf{B}_1$):

- if t has no sons, s is not subdivided either,
- if t has two sons t_1 and t_2, let

$$v_1 := \{i \in s : (j, i) \in \mathrm{nnz}(\mathbf{B}_1) \text{ for a } j \in t_1\},$$

$$v_2 := \{i \in s : (j, i) \in \mathrm{nnz}(\mathbf{B}_1) \text{ for a } j \in t_2\}.$$

Then, the sons $\mathrm{sons}(s) := \{s_1, s_2, s_3\}$ of s are given by

$$s_1 := v_1 \setminus v_2,$$

$$s_2 := v_2 \setminus v_1,$$

$$s_3 := v_1 \cap v_2.$$

The domain clusters s_1 and s_2 will be further subdivided with respect to associated clusters t_1 and t_2, respectively. The interface cluster s_3 is subdivided with bisection independently of the pressure cluster tree $\mathcal{T}_\mathcal{J}$ (Algorithm 1).

ALGORITHM 1 Coupled clustering algorithm

```
Input: Domain cluster $s \subseteq \mathcal{I}$ with associated cluster $t \in \mathcal{T}_{\mathcal{J}}$, (sparse) matrix $\mathbf{B} \in \mathbb{R}^{\mathcal{J}\times\mathcal{I}}$
  procedure BuildCoupledCluster$s, t, \mathbf{B}$
    if sons$(t) = \emptyset$ then
      sons$(s) \leftarrow \emptyset$
    else
      $\{t_1, t_2\} \leftarrow$ sons$(t)$
      $v_1 \leftarrow \{i \in s : (j,i) \in \mathrm{nnz}(\mathbf{B})$ for a $j \in t_1\}$
      $v_2 \leftarrow \{i \in s : (j,i) \in \mathrm{nnz}(\mathbf{B})$ for a $j \in t_2\}$
      $s_1 \leftarrow v_1 \setminus v_2$
      $s_2 \leftarrow v_2 \setminus v_1$
      $s_3 \leftarrow v_1 \cap v_2$
      BuildCoupledCluster$(s_1, t_1, \mathbf{B})$
      BuildCoupledCluster$(s_2, t_2, \mathbf{B})$
      BuildInterfaceCluster$(s_3)$
      sons$(s) \leftarrow \{s_1, s_2, s_3\}$
    end if
end procedure
```

5 | NUMERICAL RESULTS

In the following, we will compare numerical results obtained when using the uncoupled clustering introduced in Section 4.1 and those with the coupled clustering. For this we use the viscosity parameter $\nu = \frac{1}{100}$ and the recirculating convection

$$\mathbf{b}(\mathbf{x}) = \begin{pmatrix} -\sin(\pi x_1)(\cos(\pi x_2)\sin(\pi x_1) + \sin(\pi x_2)\cos(\pi x_3)) \\ \sin(\pi x_2)(\cos(\pi x_1)\sin(\pi x_3) - \sin(\pi x_1)\cos(\pi x_3)) \\ \sin(\pi x_3)(\cos(\pi x_1)\sin(\pi x_2) + \sin(\pi x_1)\cos(\pi x_3)) \end{pmatrix}$$

in our model Oseen problem (1). The right-hand side for the linear system (2) is chosen such that $\mathbf{e} = (1, \dots, 1)^{\mathsf{T}}$ is the exact solution of the saddle point problem.

We use the η-admissibility condition from [7] with $\eta = 16$ for the construction of the block tree $\mathcal{T}_{\mathcal{J}\times\mathcal{J}}$:

$$\mathrm{adm}_{\mathcal{J}\times\mathcal{J}} := \mathrm{adm}_{\eta}(t,s) = \begin{cases} \text{true}, & \text{if } \min(\mathrm{diam}(t), \mathrm{diam}(s)) \leq \mathrm{dist}(t,s), \\ \text{false}, & \text{else.} \end{cases}$$

Since the cluster tree $\mathcal{T}_{\mathcal{I}}$ is constructed with a domain decomposition clustering, we use

$$\mathrm{adm}_{\mathcal{I}\times\mathcal{I}}(t,s) = \begin{cases} \text{true}, & \text{if } t \neq s \text{ and } t, s \in C_{\mathrm{dom}}(\mathcal{T}_{\mathcal{I}}), \\ \mathrm{adm}_{\eta}(t,s), & \text{else} \end{cases}$$

as an admissibility condition for the block tree $\mathcal{T}_{\mathcal{I}\times\mathcal{I}}$ (for both the uncoupled and coupled clustering of $\mathcal{T}_{\mathcal{I}}$).

With the coupled clustering described in Section 4.2, each (velocity) domain cluster $s \in \mathcal{T}_{\mathcal{I}}$ has an associated pressure cluster $t_s \in \mathcal{T}_{\mathcal{J}}$. Thus, we use

$$\mathrm{adm}_{\mathcal{J}\times\mathcal{I}}(t,s) = \begin{cases} \text{true}, & \text{if } t \neq t_s, \text{ where } t_s \in \mathcal{T}_{\mathcal{J}} \text{ is the cluster associated with } s \in C_{\mathrm{dom}}(\mathcal{T}_{\mathcal{I}}), \\ \mathrm{adm}_{\eta}(t,s), & \text{else} \end{cases} \tag{3}$$

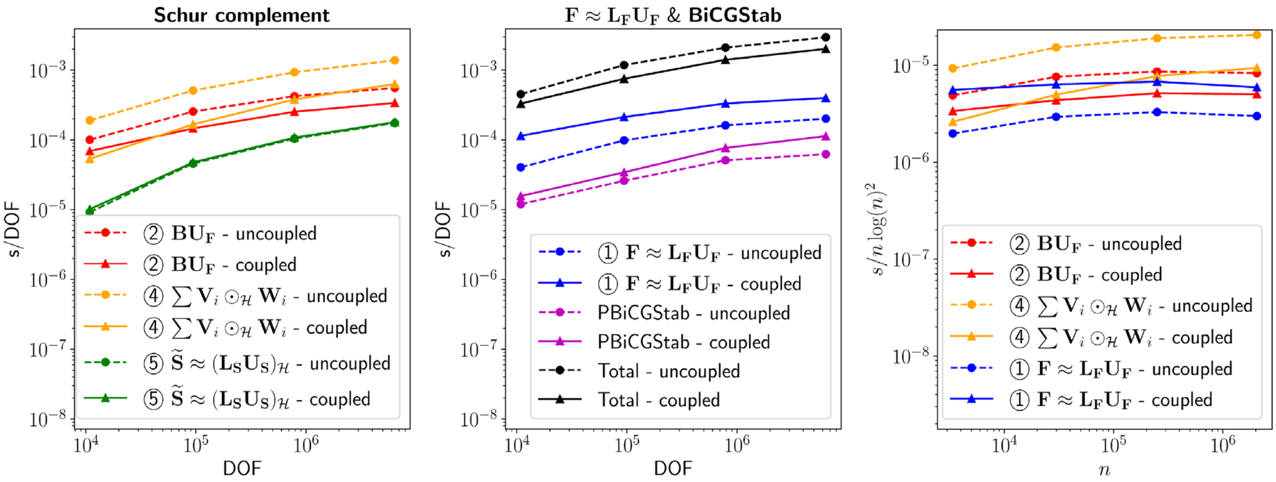

FIGURE 5 Computation times per degree of freedom for the uncoupled (dashed lines) and coupled clustering (solid lines). Left: times for the computation of the Schur complement. Middle: times for the $\mathcal{H}$-LU decomposition of the block $\mathbf{F}$, the solve with BiCGStab, and the total time per degree of freedom. Right: time/$n\log(n)^2$ for the computation of the Schur complement and the $\mathcal{H}$-LU decomposition of $\mathbf{F}$.

TABLE 1 Absolute computation times for the computation steps of the preconditioner set-up and the solve with BiCGStab. Top: uncoupled clustering. Bottom: coupled clustering.

Total #DOF	M	①	②	③	④	⑤	②–⑤	BiCGStab	Total
94,286	4,913	9 s	23 s	24 s	48 s	4 s	99 s	3 s	111 s
786,078	35,937	127 s	336 s	333 s	734 s	82 s	1485 s	40 s	1652 s
6,419,774	274,625	1294 s	3730 s	3579 s	8894 s	1122 s	17325 s	401 s	19020s
94,286	4,913	20 s	14 s	14 s	16 s	4 s	48 s	3 s	71 s
786,078	35,937	260 s	203 s	199 s	299 s	85 s	786 s	60 s	1106 s
6,419,774	274,625	2564 s	2271 s	2177 s	4071 s	1151 s	9670 s	728 s	12962 s

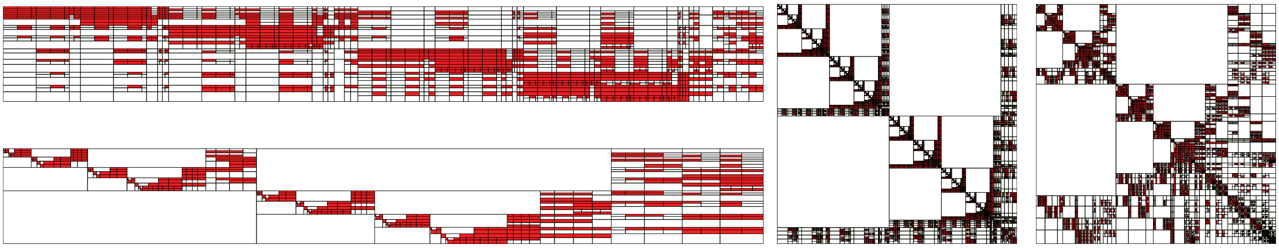

FIGURE 6 Block structure of the blocks $\mathbf{B}_i \in \mathbb{R}^{729\times3375}$ and $\mathbf{F} \in \mathbb{R}^{3375\times3375}$ for the uncoupled clustering (left top and second from right, respectively), and the coupled clustering (left bottom and right, respectively). The low rank blocks are shown in white, the dense matrix blocks are shaded.

as an admissibility condition for the block tree $\mathcal{T}_{\mathcal{J}\times\mathcal{I}}$ for the coupled clustering. For the uncoupled clustering, we used $\mathrm{adm}_{\mathcal{J}\times\mathcal{I}} = \mathrm{adm}_\eta$ as an admissibility condition for block tree $\mathcal{T}_{\mathcal{J}\times\mathcal{I}}$.

In order to preserve low ranks, the arithmetic operations of $\mathcal{H}$-matrices include the rank truncation of the sum of two low-rank matrices. This can either be done with a fixed rank or adaptively with a given (relative) accuracy. We chose a (relative) truncation accuracy of $\delta_{\mathcal{H}} = 10^{-1}$ for all arithmetic operations.

Figure 5 shows the computation times per degree of freedom, while Table 1 shows the absolute computation times for the operations to as well as solving with BiCGStab. As stopping criterion for the BiCGStab method, a relative residual norm of $\varepsilon = 10^{-3}$ was used.

Due to the different block structure of the matrices $\mathbf{B}_i$ (Figure 6), the computation of the Schur complement is about 50% faster with the coupled clustering compared to the uncoupled clustering. But this speed-up comes at a cost. The interface clusters got three times larger. Thus, all operations involving the block **F**, or its LU factors, are slower for the coupled clustering than for the uncoupled clustering. But despite this, the uncoupled clustering still yields a speed-up of about 30%.

ACKNOWLEDGMENTS

Open access funding enabled and organized by Projekt DEAL.

ORCID

Jonas Grams https://orcid.org/0009-0009-3521-9030

REFERENCES

1. Hackbusch, W. (1999). A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I. Introduction to $\mathcal{H}$-matrices. *Computing*, *62*(2), 89–108.
2. Bebendorf, M., & Hackbusch, W. (2003). Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients. *Numerical Mathematics*, *95*(1), 1–28.
3. Faustmann, M., Melenk, J. M., & Praetorius, D. (2015). $\mathcal{H}$-matrix approximability of the inverses of FEM matrices. *Numerical Mathematics*, *131*(4), 615–642.
4. Le Borne, S. (2008). Hierarchical matrix preconditioners for the Oseen equations. *Computing and Visualization in Science*, *11*(3), 147–157.
5. Le Borne, S. (2009). Preconditioned nullspace method for the two-dimensional Oseen problem. *SIAM Journal on Scientific Computing*, *31*(4), 2494–2509.
6. Benzi, M., Golub, G. H., & Liesen, J. (2005). Numerical solution of saddle point problems. *Acta Numerica*, *14*, 1–137.
7. Hackbusch, W. (2015). *Hierarchical Matrices: Algorithms and Analysis* (Vol. 49). Springer Ser. Comput. Math., Springer.
8. Grasedyck, L., Kriemann, R., & Le Borne, S. (2009). Domain decomposition based $\mathcal{H}$-LU preconditioning. *Numerical Mathematics*, *112*(4), 565–600.

How to cite this article: Grams, J., & Le Borne, S. (2023). Coupled clustering strategies for hierarchical matrix preconditioners in saddle point problems. *Proceedings in Applied Mathematics and Mechanics*, *23,* e202300077. https://doi.org/10.1002/pamm.202300077