Algebraic Hybridization for the Darcy Problem

Barry Fadness

Portland State University

MFEM Community workshop

September 10, 2025

We want to solve the Darcy Problem. The discrete finite element system is given by

$$\begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix} \begin{pmatrix} p \\ u \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

where u is velocity and p is pressure. The subvector g is boundary data. We approximate u in the Raviart-Thomas space where the flux is continuous across interior element facets. In hybridization we break the continuity and enforce it through a constraint. On each element we have

$$\begin{pmatrix} 0 & 0 & C \\ 0 & 0 & B \\ C^T & B^T & A \end{pmatrix} \begin{pmatrix} \lambda \\ p \\ u \end{pmatrix} = \begin{pmatrix} 0 \\ f \\ g \end{pmatrix}.$$

We abuse notation: The blocks are now element matrices.

Partition the block system:

$$\left(\begin{array}{c|cc} 0 & 0 & C \\ \hline 0 & 0 & B \\ C^T & B^T & A \end{array}\right) \left(\begin{array}{c} \lambda \\ \hline p \\ u \end{array}\right) = \left(\begin{array}{c} 0 \\ \hline f \\ g \end{array}\right).$$

we get two equations.

$$(0 \;\; C)\begin{pmatrix} p \\ u \end{pmatrix} = 0$$

$$\begin{pmatrix} 0 \\ C^T \end{pmatrix}\lambda + \begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}\begin{pmatrix} p \\ u \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

Matrix $\begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}$ is invertible, so solve for $\begin{pmatrix} p \\ u \end{pmatrix}$ and substitute it into the other equation.

$$\begin{pmatrix} p \\ u \end{pmatrix} = \begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}^{-1} \left[\begin{pmatrix} f \\ g \end{pmatrix} - \begin{pmatrix} 0 \\ C^T \end{pmatrix} \lambda \right]$$

$$\begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}^{-1} \begin{pmatrix} f \\ g \end{pmatrix} - \begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ C^T \end{pmatrix} \lambda$$

$$\begin{pmatrix} 0 & C \end{pmatrix} \begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ C^T \end{pmatrix} \lambda = \begin{pmatrix} 0 & C \end{pmatrix} \begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}^{-1} \begin{pmatrix} f \\ g \end{pmatrix}$$

Assemble the last equation into a global system for the multiplier λ.

Store the **vectors** and **matrices** in contiguous arrays.

But from where does the constraint matrix C come?

In the Hybridization class, the constraint matrix is computed using an integrator for the mixed bilinear form

$$c(\mu, v) = \sum_K \int_{\partial K} \mu \, v \cdot n$$

The multiplier variable μ belongs to a normal trace space on the mesh skeleton. The numerical integration takes most of the time to set up the reduced system

$$\begin{pmatrix} 0 & C \end{pmatrix} \begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ C^T \end{pmatrix} \lambda = \begin{pmatrix} 0 & C \end{pmatrix} \begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}^{-1} \begin{pmatrix} f \\ g \end{pmatrix}.$$

Instead we construct C algebraically to improve the setup time.

The setup time in my new code is roughly 50% faster than my old code.

The new code is based on an implementation in the library parELAG github.com/llnl/parelag

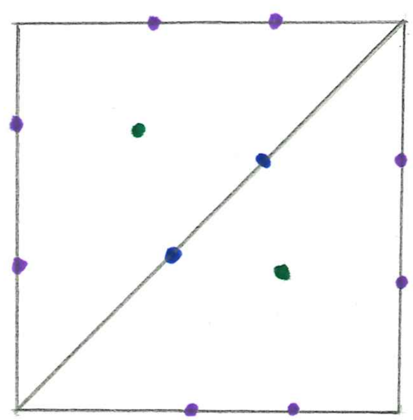
(src > amge > HybridHdivL2.cpp)

We rely on the connectivity tables in the MFEM Mesh and FiniteElementSpace classes.

multiplier degrees of freedom (dofs) which we define correspond to velocity dofs on interior facets.

The Mesh class has a list of boundary elements (facets that lie on the true boundary of the mesh). We use the list to obtain an interior facet marker:

1	0	0	0	0



The FiniteElementSpace has a facet (edge, face) to dof table. Ours for velocity dofs is

0	2	4	6	8	10

0	1	3	2	5	4	6	7	8	9

interior facet marker	1	0	0	0	0

facet to velocity dof table

0	2	4	6	8	10

0	1	3	2	5	4	6	7	8	9

Accumulate the dofs on interior facets into an array:

0	1

We define the array indices to be the multiplier dof indices, so the array gives a multiplier dof to velocity dof mapping.

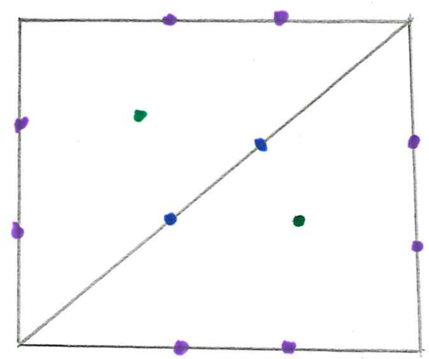
For each element, we assume we are given $\begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}$, so we must use the velocity dof order in the element to velocity dof table:

0	8	16

0	1	3	2	5	4	10	11	1	0	6	7	8	9	12	13

Remember that we have

$$\begin{pmatrix} 0 & C \end{pmatrix} \begin{pmatrix} 0 & B \\ B^T & A \end{pmatrix}^{-1}$$



in the reduced system. Therefore, we need a facet velocity dof to multiplier dof mapping

0	1	-1	-1	-1	-1	-1	-1	-1	-1

element to velocity dof table:

0	8	16

0	1	3	2	5	4	10	11	1	0	6	7	8	9	12	13

facet velocity dof to multiplier dof mapping

0	1	-1	-1	-1	-1	-1	-1	-1	-1

Let us compute C for element number 2:

$$\begin{array}{c c} & \begin{array}{cccccc} 1 & 0 & 6 & 7 & 8 & 9 \end{array} \\ \begin{array}{c} 1 \\ 0 \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

We choose the multiplier dof order, unlike in the Hybridization class. The number of element facet dofs can be computed with tables. What if there are interior velocity dofs?

We statically condense the element interior velocity dofs.

Because interior (i) dofs are numbered last, the upside down block structure makes elimination convenient:

$$\left(\begin{array}{c|ccc} 0 & 0 & C_f & 0 \\ \hline 0 & 0 & B_f & B_i \\ C_f^T & B_f^T & A_{ff} & A_{fi} \\ 0 & B_i^T & A_{if} & A_{ii} \end{array}\right)$$

We end up with facet dofs (f) only, and the Schur complement S of A_{ii} is given by

$$S = \begin{pmatrix} 0 & B_f \\ B_f^T & A_{ff} \end{pmatrix} - \begin{pmatrix} B_i \\ A_{fi} \end{pmatrix} A_{ii}^{-1} \left(B_i^T A_{fi} \right).$$