

# HDG Assignment #7

Arthur Lustman

June 5, 2019

## Exercise 1

The local strong form of the problem is, find  $\mathbf{q}_i$  and  $u_i$  for each  $i = 1, \dots, n_{el}$  such that

$$\begin{cases} \nabla \cdot \mathbf{q}_i = s & \text{in } \Omega_i, \\ \mathbf{q}_i + \nabla \cdot (\kappa \nabla u_i) = 0 & \text{in } \Omega, \\ u_i = u_D & \text{on } \partial\Omega_i \cap \Gamma_D, \\ u_i = \hat{u} & \text{on } \partial\Omega_i \cap (\Gamma_N \cup \Gamma_R) \end{cases}$$

The global strong form of the problem is to solve for the hybrid variable  $\hat{u}$

$$\begin{cases} \llbracket u\mathbf{n} \rrbracket = \mathbf{0} & \text{on } \Gamma, \\ \llbracket \mathbf{n} \cdot \mathbf{q} \rrbracket = 0 & \text{on } \Gamma, \\ \mathbf{n} \cdot \mathbf{q} = -t & \text{on } \Gamma_N, \\ -\mathbf{n} \cdot \mathbf{q} + \gamma u = g & \text{on } \Gamma_R \end{cases}$$

The definition of the numerical flux is useful for the weak form

$$\mathbf{n}_i \cdot \hat{\mathbf{q}}_i := \begin{cases} \mathbf{n}_i \cdot \mathbf{q}_i + \tau_i(u_i - u_D) & \text{on } \partial\Omega_i \cap \Gamma_D, \\ \mathbf{n}_i \cdot \mathbf{q}_i + \tau_i(u_i - \hat{u}) & \text{elsewhere} \end{cases}$$

The weak form is derived from the strong form, after integrating by parts and using the numerical flux. Although the local weak form has exactly the same formulae as for the only dirichlet boundary problem on the slides of the class (09-HDG-Wide, slide 12) except that the viscosity parameter  $\kappa$  is present

$$-(\mathbf{w}, \mathbf{q}_e^h)_{\Omega_e} + \kappa(\nabla \cdot \mathbf{w}, u_e^h)_{\Omega_e} = \kappa \langle \mathbf{n}_e \cdot \mathbf{w}, u_D \rangle_{\partial\Omega_e \setminus \Gamma_D} + \kappa \langle \mathbf{n}_e \cdot \mathbf{w}, \hat{u}^h \rangle_{\partial\Omega_e \setminus \Gamma_D} \quad (1)$$

The global weak form of the problem is written

$$\sum_{e=1}^{n_{el}} \left( \langle \hat{v}, \mathbf{n}_e \cdot \mathbf{q}_e^h \rangle_{\partial\Omega_e \setminus \Gamma_D} + \langle \hat{v}, \tau_e u_e^h \rangle_{\partial\Omega_e \setminus \Gamma_D} - \langle \hat{v}, \tau_e \hat{u}^h \rangle_{\partial\Omega_e \setminus \Gamma_D} \right) = - \sum_{e=1}^{n_{el}} \left( \langle \hat{v}, t \rangle_{\partial\Omega_e \cap \Gamma_N} + \langle \hat{v}, g + \gamma \hat{u}_e^h \rangle_{\partial\Omega_e \cap \Gamma_R} \right) \quad (2)$$

The part  $\langle \hat{v}, \gamma \hat{u}_e^h \rangle_{\partial\Omega_e \cap \Gamma_R}$  should be on the left hand side due to the fact that is is expressed by the term  $\hat{u}_e^h$ . Taking the matrix notation introduced in the slides 13,15 and 16 of the course, the previous equation could be expressed in matrix form

$$\sum_{i=1}^{n_{el}} \left\{ [A_{u\hat{u}}^\top \quad A_{u\hat{u}}^\top]_i \begin{pmatrix} \mathbf{u}_i \\ \mathbf{q}_i \end{pmatrix} + [A_{\hat{u}\hat{u}}^\top]_i \hat{\mathbf{u}}_i \right\} = \sum_{i=1}^{n_{el}} [\mathbf{f}_{\hat{u}}]_i \quad (3)$$

The terms  $[A_{\hat{u}\hat{u}}^\top]_i$  and  $[\mathbf{f}_{\hat{u}}]_i$  are slightly different. The first one is subject to the Robin boundary condition and the second is subject to Robin and Neumann boundary condition.

## Exercise 2

In order to implement Neumann and Robin boundary condition in the code, there are multiple sides to take care of.

- the boundary condition has to be represented in the code in form of a list containing the sides for Robin and Neumann. The structure of the variable `extFaces` has been recycled. After passing a test checking that coordinates of the nodes of the faces are on the Neumann or Robin boundary, they are saved in the respective variable `extNFaces` or `extRFaces`.
- the files `analyticalPoisson` and `sourcePoisson` had to be modified to stick with the function given. Two similar files for the Robin and Neumann boundary conditions were added, the expressions of the functions are given in the third exercise.
- the Neumann and Robin contribution had to be added in the elemental matrices computed inside the function `hdgMatrixPoisson`. Their contribution is added accordingly to the expression of the weak form equation (2).
- the viscosity parameter  $\kappa$  needs to be implemented in the function `HDGpostprocess` to compute the value of  $u^*$ . The  $\mathcal{L}_2$ -norm of the  $\mathbf{q}$  needs to be added in a similar way than the error of  $u$  is computed. Since the variable is a vector, there are two components per node, the error is computed using the same principles as for the error of  $u$ .

## Exercise 3

Using the Matlab tools for symbolic calculus, considering the function  $u(x, y)$  given, the analytic expression of  $u_D$ ,  $t$  and  $g$  were computed using the equations of the description of the problems. They are expressed in as the following

$$u(x, y) = \log(a \cos^2(b\pi x) + \kappa) + \gamma(x + c)^3 \quad (4)$$

$$g(x, y) = \gamma(\log(\kappa + a \cos^2(b\pi x)) + \gamma(c + x)^3) \quad (5)$$

$$t(x, y) = -\kappa \left( 3\gamma(c + x)^2 - \frac{2a b\pi \cos(b\pi x) \sin(b\pi x)}{\kappa + a \cos^2(b\pi x)} \right) \quad (6)$$

## Exercise 4

The final modified code to run the solution of the HDG problem can be done with multiple parameters such as the refinement of the mesh and the degree of the element. The refinement of the results can be seen on the following figure showing the solution for degree 2 element with configuration mesh 1 and 3.

The solutions for the third mesh is much more delicate. Due to very coarse mesh, the values of the solution are much higher than for the finer mesh. Discontinuities are always apparent in the solution  $U$  due to the discontinuous nature of HDG but they tend to disappear with the solution  $U^*$  with finer meshes.

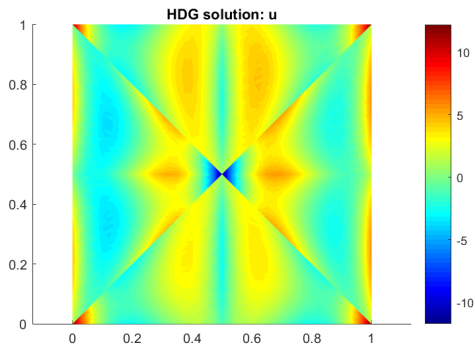


Figure 1: Mesh 1 solution  $U$

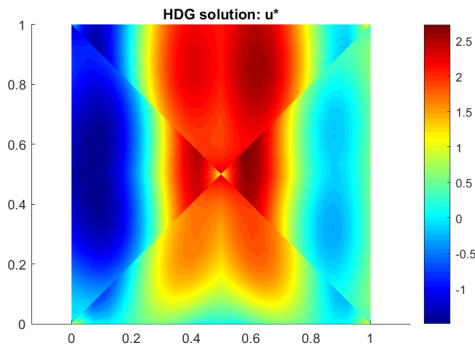


Figure 2: Mesh 1 solution  $U^*$

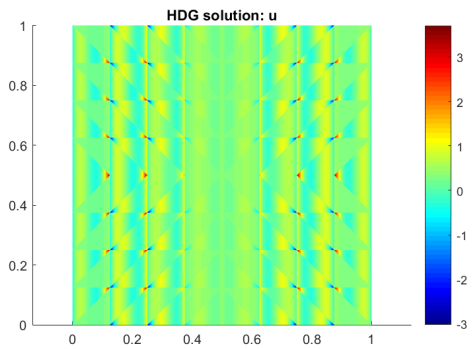


Figure 3: Mesh 3 solution  $U$

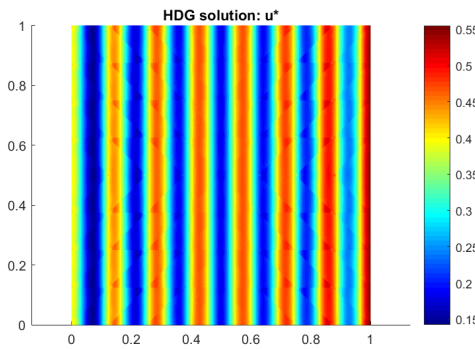


Figure 4: Mesh 3 solution  $U^*$

The correctness of the method is performed by checking the convergence rate and comparing them between the variables of  $u$  and  $u^*$ , as done in the last exercise. Although, the good functioning of the code can be tested by checking the convergence of nodes towards the exact values of the solution. The values of the solution  $u$  has been extracted for each mesh and degrees for the coordinate  $(0,0)$ , i.e. node  $(1,1)$  in each mesh. The exact value for the solution is  $u = 0.395$ . The values of  $u$  is oscillating towards the exact value.

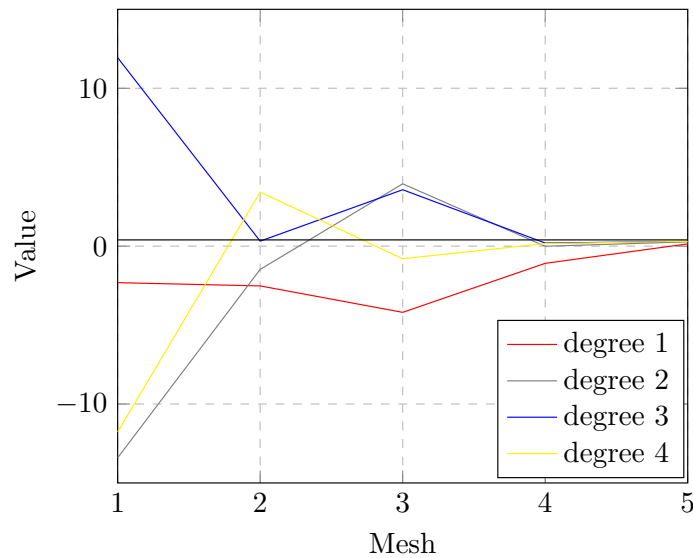


Figure 5: Value of  $U$  for node 1

### Exercise 5

The log of the  $\mathcal{L}_2$  error is given in the following figure. The flux variable  $\mathbf{q}$  has been added, although it has the same convergence rate as  $u$ , as it can be seen on the figure. Convergence of order  $p+1$  for  $q$  and  $u$  is achieved and super-convergence  $p+2$  is obtained for the solution  $u^*$ . = Rate of converge  $p+1$  for  $u$  and  $\mathbf{q}$  and  $p+2$  for the variable  $u^*$ .

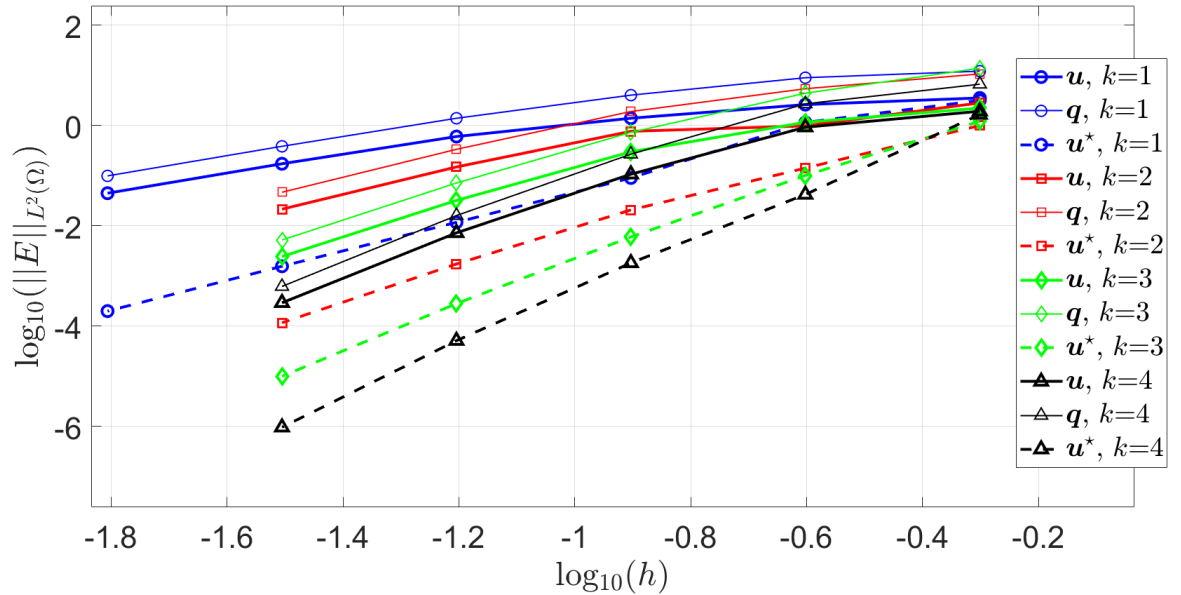


Figure 6: Convergence over mesh parameters