16.930 Advanced Topics in Numerical Methods for PDEs
Massachusetts Institute of Technology – Spring 2015

Version 1.0

Project 34.B – HDG for the convection-diffusion equation

Handed Out: March 30, 2015 *Due (on-line): either April 17 at 12pm or May 11 at 12pm*

In the first part of the project, you will be asked to answer some questions related to the lecture HDG for the Transport Equation. The questions are posed as follows:

Q1: In the lecture, we claim that we can choose the stabilization parameter τ of the HDG method to obtain a method which is exactly the original DG method. Determine τ that achieves this.

Q2: Please prove that the HDG method for the transport equation is well-defined (meaning that its solution exists and is unique) under the assumptions $\sigma + \frac{1}{2}\nabla \cdot \boldsymbol{c} \geq 0$ and $\tau > \frac{1}{2}\boldsymbol{c} \cdot \boldsymbol{n}$.

In the second part of this project, you will be asked to implement the HDG method to solve the following convection-diffusion equation

$$
\begin{aligned}
\boldsymbol{q} - \boldsymbol{\nabla} u &= \boldsymbol{0}, \quad \text{in } \Omega, \\
-\nabla \cdot (\kappa \boldsymbol{q}) + \nabla \cdot (\boldsymbol{c} u) &= f, \quad \text{in } \Omega, \\
u &= g, \quad \text{on } \partial\Omega.
\end{aligned}
$$

The code you'll write should be general to solve the convection-diffusion model problem on any mesh for given functions f and g, and diffusivity coefficient κ, and velocity field $\boldsymbol{c}$. However, for this project you will be asked to present results for the particular cases as described in **Your Tasks**.

You will be getting `hdg_cd.m` function that you can start with to develop your own HDG code. This function will call a function `hdg_solve.m` which is responsible for solving the convection-diffusion model problem and returning the numerical solution $(u_h, \boldsymbol{q}_h, \widehat{u}_h)$. In addition, the `hdg_cd.m` function will also call another function `hdg_postprocess.m` which is responsible for postprocessing the numerical solution to obtain a new approximation u_h^* of u. Your task is to complete `hdg_solve.m` and `hdg_postprocess.m` which have only the name and the header that describes their input and output arguments.

Your Tasks

- The first task is to write the function `hdg_solve.m` that implements the HDG method.
- The second task is to write the function `hdg_postprocess.m` that implements the local post-processing of the HDG solution.
- As the first example to test your code, consider the Poisson problem

$$
\begin{aligned}
\boldsymbol{q} - \boldsymbol{\nabla} u &= \boldsymbol{0}, \quad \text{in } \Omega, \\
-\nabla \cdot \boldsymbol{q} &= f, \quad \text{in } \Omega, \\
u &= 0, \quad \text{on } \partial\Omega.
\end{aligned}
$$

Here $\Omega = (0,1)^2$ is the unit square and the source term is given by $f = 2\pi^2 \sin(\pi x)\sin(\pi y)$. Note that the exact solution of this problem is $u = \sin(\pi x)\sin(\pi y)$. Your task is to solve this problem using your HDG code and verify the convergence of the numerical solution to the exact solution.

- As the next example to test your code, consider the convection-diffusion problem

$$\begin{aligned} \boldsymbol{q} - \boldsymbol{\nabla} u &= \boldsymbol{0}, \quad \text{in } \Omega, \\ -\nabla \cdot (\kappa \boldsymbol{q}) + \nabla \cdot (\boldsymbol{c} u) &= 1, \quad \text{in } \Omega, \\ u &= 0, \quad \text{on } \partial\Omega. \end{aligned}$$

Here $\Omega = (0,1)^2$ is the unit square. Your task is to solve this problem using your HDG code for $\boldsymbol{c} = (1,1)$ with three different values of diffusivity: $\kappa = 1$, 10^{-1}, and 10^{-2}.

Project deliverables

- Provide your answers to the questions **Q1** and **Q2**.
- Submit the functions `hdg_solve.m` and `hdg_postprocess.m`. Please also submit other (if any) functions which you use in `hdg_solve.m` and `hdg_postprocess.m`. Note: for this project, since the test cases only consider domains with straight boundaries, you do not need to use curved elements.
- For the Poisson example, perform a numerical h-convergence test for $p = 1, 2$ and 3 using three consecutive grids: a coarse **structured** mesh, then both a medium mesh and a fine mesh obtained by uniform refinement. Specifically, use `mkmesh_square.m` to obtain grids which discretize the unit square domain into $2n^2$ triangles. The three meshes correspond to $n = 8, 16$ and 32, respectively. Please plot the order of convergence in the L_2-norm for the approximate solution u_h, the approximate gradient $\boldsymbol{q}_h$, and the postprocessed solution u_h^*. Please do this convergence test for three different values of the stabilization parameter: $\tau = h$, $\tau = 1$, and $\tau = 1/h$. Here $h = 1/n$ is the mesh size.
- For the convection-diffusion example, we are interested in comparing the approximate solution u_h with the postprocessed solution u_h^* for the three cases above: $\kappa = 1$ (diffusion-dominated case), $\kappa = 10^{-1}$ (convection-diffusion case), and $\kappa = 10^{-2}$ (convection-dominated case). In each case you should perform the comparison for three different **unstructured** meshes and for various polynomial degrees. The second mesh is a uniform refinement of the first mesh and the third mesh is a uniform refinement of the second mesh. To generate these unstructured meshes, use `distmesh` (or a similar tool). A suggestion would be to modify `mkmesh_circle.m` which uses `distmesh` to generate an unstructured mesh for a circular domain. You might consider using a spacing field which will cluster the mesh towards the boundaries in which boundary layers will be present. You may use a different sequence of meshes for the three cases (since the boundary layers will be of different scales). Please present color plots of u_h and u_h^* for all three cases for $p = 1$ and $p = 3$ on all three meshes. For each case, please plot the $p = 1$ and $p = 3$ results side-by-side (3 rows by 2 columns), using the same color scale, to allow easier visual comparisons. In your submission, please describe the stabilization τ you choose for these convection-diffusion problems on unstructured meshes.