Project Report - PyDG solver

Computational Acoustics

Thomas Brunner (12018550)

SS23

23.05.2024

1 Theory

I chose the following scalar elliptic partial differential equation as a model problem

$$-\nabla u = f \text{ in } \Omega, \quad u = 0 \text{ on } \partial\Omega \tag{1}$$

To solve this using a Discontinous Galerkin method I chose a *symmetric interior penalty method (SIPG)*, which uses the continuity of the analytical solution to ensure a symmetric bilinear form. I employ the following discrete flux

$$n \cdot \nabla u \to \langle n \cdot \nabla u \rangle - \frac{\beta}{h}[u] \tag{2}$$

with the interior penalty parameter β and the element size h. To find more information on DG methods I highly recommend the amazing textbook [1]. The main derivation steps can be found in the attached pdf-file (*PyDG_MainDerivation.pdf*).

2 Implementation

I implemented the following classes and test-scripts in Python, as Python is well known to be a slow object-oriented language I did not even try to write speed-optimised object-oriented code. I treated this more as a way to understand DG (SIPG) methods and not to get a fast solver. For a fast solver I would never start in Python but use for example C++.

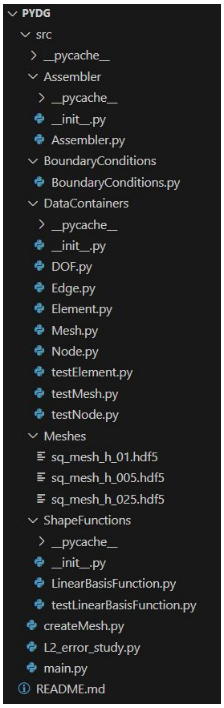

Figure 1: Basic Class Structure of PyDG.

3 Validation

To have a analytical solution at hand to which I could compare the solution I created a unit square mesh and used

$$f = 2\pi^2 \sin(\pi x)\sin(\pi y) \tag{3}$$

as my right hand side forcing. This gives the analytical solution

$$u_{anal} = \sin(\pi x)\sin(\pi y) \tag{4}$$

As a mesher I used MeshPy which gives Python interfaces for the meshers GMSH, TetGen and Triangle. For validation I used three unit square meshes (see figure 2) with element sizes $h = [0.25, 0.1, 0.05]$.

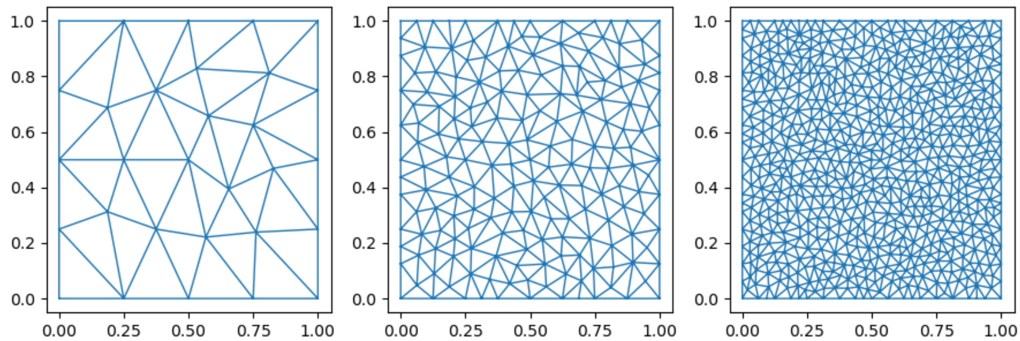

Figure 2: Used meshes.

To study the impact of the penalty parameter β I did a parameter study using $\beta = (4, 10, 20, 40, 80)$. In figure 3 I show the solution of the solver using the different mesh sizes and β values.

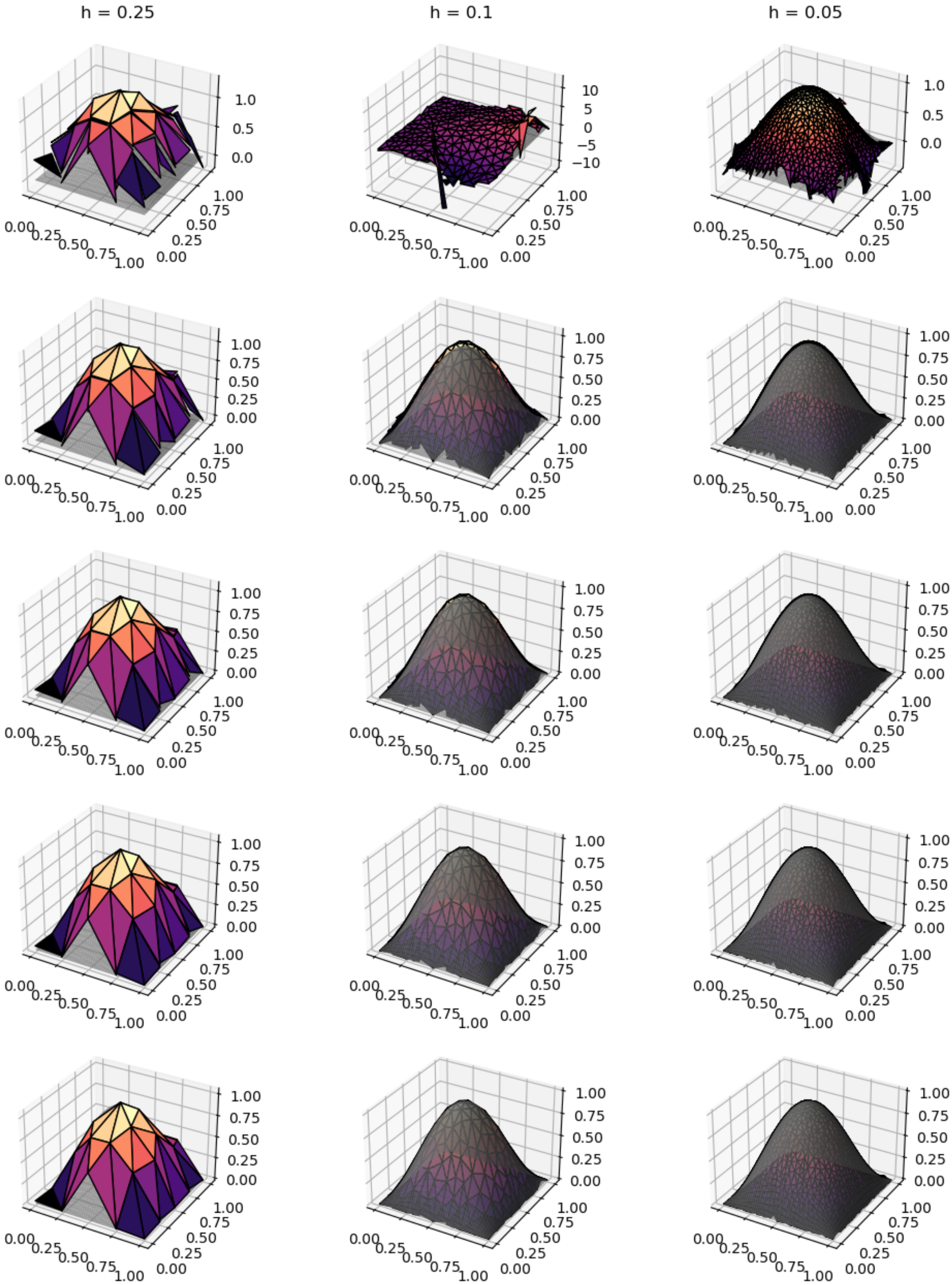

Figure 3: Grid size and penalty parameter study. The values of β from top to bottom are $\beta = [4, 10, 20, 40, 80]$. The analytical solution is shown in a transparent grey.

To show convergence of the solution at higher penalty parameters β I also calculated the L^2 error of the DG solution to the analytical one. The main deviations stem from the boundary values of the solution, this is due to the Dirichlet values being only penalised but not strongly enforced in the linear system. It is also interesting to see that the $h = 0.1$ mesh converges faster than the $h = 0.05$ mesh. I also attribute this to the weak handling of the boundary condition.

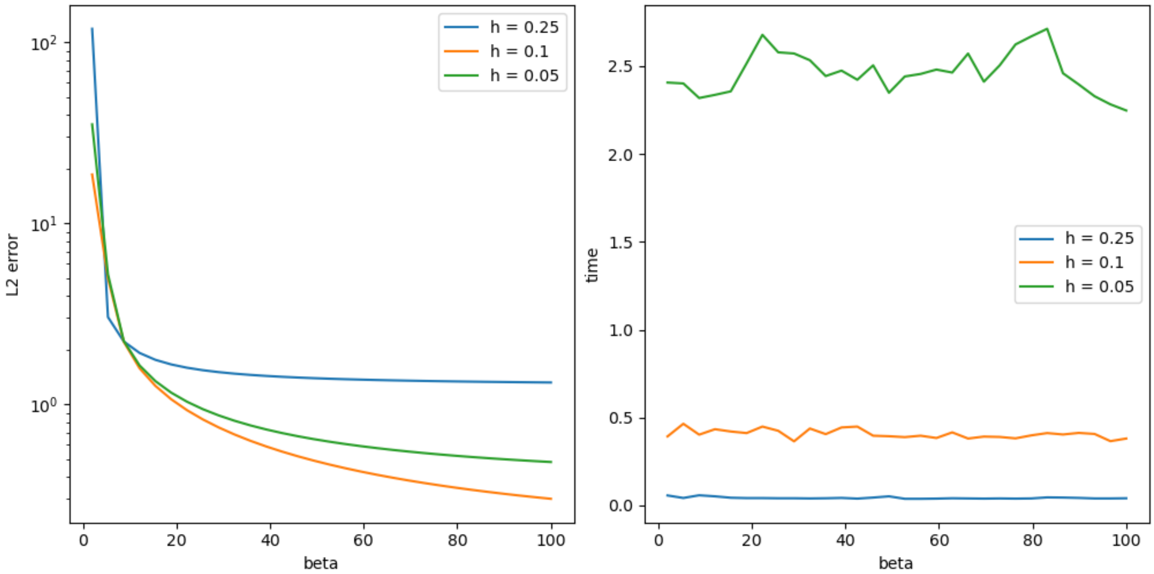

Figure 4: Convergence of the L^2-error and display of the required time t (s) to solve as a function of the penalty parameter β.

4 Bibliography

References

[1] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods - Algorithms, Analysis, and Applications*. Berlin Heidelberg: Springer Science Business Media, 2007. ISBN: 978-0-387-72067-8.