Discontinuous Galerkin Finite Element Method for the Diffusion-Advection Equation

Florian Pöppl

02689 Advanced Numerical Methods for Differential Equations

Introduction

DG-FEM is a variation of classical FEM Methods where elements are only weakly coupled. The test functions are polynomial functions on their respective element and vanish everwhere else. The global solution is therefore not necessarily continuous. Coupling is achieved using a numerical flux between element borders.

This poster aims to present the results of an exploration into the implementation and properties of DG-FEM methods. A DG-FEM method for the 1D advection equation was implemented and tested. Higher dimensional and more general diffusion-advection problems were solved with the help of the Netgen/NGSolve software.

Advection Equation in 1D

Consider the linear advection equation in 1D with periodic boundary conditions

$$\begin{aligned} u_t + f(u) &= 0, \quad x \in [0,1], \\ u(-1,t) &= u(1,t), \\ u(x,0) &= \sin(\pi x), \end{aligned}$$

where $f = au_x$ the exact solution is given by $u(x,t) = sin(\pi x - a\pi t)$. The domain $[-1,1]$ is divided into K consecutive sub-intervals (elements) of size $h := \frac{2}{K}$. On each element k, we discretize the differential equation using a nodal spectral galerkin method with $N+1$ Gauss-Lobatto nodes. The elements are coupled by introducing a numerical flux f^*, resulting in a discrete operator $\mathcal{L}_h$

$$u_t^k = -\frac{2}{h}\mathcal{D}_x(a^k) + \frac{2}{h}\mathcal{M}^{-1}[(au^k - (au)^*)L_i^k]_{x_l^k}^{x_r^k} =: \mathcal{L}_h(u),$$

where L_i^k are the Lagrange polynomials of element k and $\mathcal{M}$, $\mathcal{D}_x$ are the corresponding mass and differentiation matrices.

With the average operator $\{u\} := \frac{u^- + u^+}{2}$ and the jump operator $[u] := u^- + u^+$ we can define the numerical flux

$$f^* = (au)^* := \{au\} + |a|\frac{1-\alpha}{2}[nu]$$

with a parameter $\alpha > 0$. Here, $+$ refers to the exterior of an element, $-$ to the interior and $n^\pm$ to the corresponding normal vector. This is called *upwind flux* for $\alpha = 0$ and *central flux* for $\alpha = 1$.

In order to solve this semi-discrete problem, we have to decide on a time-stepping method. Figure 1 shows the eigenvalues of L_h for $N = 16$. For a central flux, the eigenvalues are purely imaginary (which also means there is no dissipation). For an upwind flux, the eigenvalues also have a real part. It can also be shown that the magnitude of the largest eigenvalue behaves asymptotically like $\mathcal{O}(N^2)$ meaning that heuristically, the time-step Δt has to be chosen so that $\Delta t \leq \frac{C}{N^2}$, where the constant C depends on the time-integration method used. Here, a $RK4$ method with a suitably small time-step is chosen.

Figure 2 shows that this method converges with order h^{N+1}.

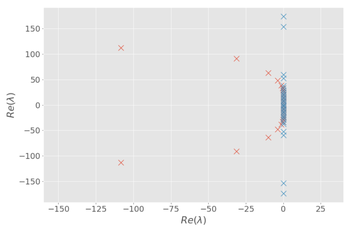


Figure: Eigenvalues of L_h for $N = 16$.

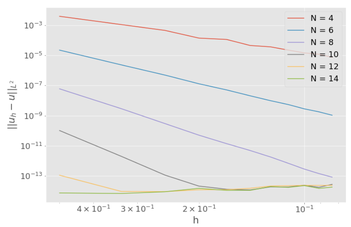


Figure: Errors for different N over h.

Netgen/NGSolve

NGSolve is a high performance multiphysics finite element software. It is widely used to analyze models from solid mechanics, fluid dynamics and electromagnetics. Due to its flexible Python interface new physical equations and solution algorithms can be implemented easily.

NGSolve takes care of meshing, building the finite element space and the discrete operators and can efficiently solve the resulting linear systems. Users have to

- Define the problem geometry and boundary conditions.
- Specify a suitable variational formulation of the PDE.
- Time-integrate the resulting discretizations.

In the case of DG methods, NGSolve works with an L^2 finite element space V_h consisting of L^2-orthogonal element-wise polynomial functions of arbitrary order.

Stationary Diffusion-Advection Equation

Consider now the more general case, $D \in \mathbb{R}$, $b \in C(\Omega, \mathbb{R}^d)$ and $f \in L^2(\Omega)$:

$$\begin{aligned} -D\Delta u + \nabla \cdot (bu) &= f, \quad x \in \Omega \subset \mathbb{R}^d, \\ u &= 0, \quad x \in \partial\Omega. \end{aligned}$$

For $u, v \in H_0^1$, this can be written in variational formulation as

$$A(u,v) := \int D\nabla u \nabla v - \int b \cdot u \nabla v = \int fv =: l(v).$$

The element-wise DG formulation $A^{DG}(u,v) = l(v)$ for $u, v \in L^2$ is then given by

$$\begin{aligned} A^{DG}(u,v) &= A^{dif}(u,v) + A^{adv}(u,v), \\ A^{dif}(u,v) &:= \sum_K \int_K \nabla u \nabla v - \sum_F \int_F \{n\nabla u\}[v] - \sum_F \int_F \{n\nabla v\}[u] + \frac{\beta}{h}\sum_F \int_F [u][v], \\ A^{adv}(u,v) &:= -\sum_K \int_K bu\nabla v + \sum_F \int_F b \cdot n\, u^* v. \end{aligned}$$

with A^{dif} and A^{adv} corresponding to the diffusion and advection term respectively and a user-definable parameter $\beta > 0$. Here, K denotes the elements of the mesh (e.g. triangle, tetrahedon) and F the faces of those elements.

As in the 1D case we choose an upwind numerical flux $u^* := b \cdot n\{u\} + \frac{1}{2}|b \cdot n|[u]$.

Given A^{DG} NGSolve constructs a discrete operator A_h and right-hand side l_h such that the modal coefficients u_h of the solution are given by $A_h u_h = l_h$.

Instationary Diffusion-Advection Equation

Consider the (bi)linear forms A^{DG} and l from above and the time-dependent problem

$$\begin{aligned} \partial_t u &= D\Delta u + \nabla \cdot (bu) + f, \quad x \in \Omega, \\ u &= 0, \quad x \in \partial\Omega, \end{aligned}$$

with starting condition $u(x,0) = u_0(x)$. It's discontinuous Galerkin formulation is

$$\partial_t \underbrace{\int uv}_{=:M(u,v)} = A^{DG}(u,v) + l(v)$$

NGSolve provides the discretization of M, the mass matrix M_h, so that the solution u_h is given by the ODE $M_h \partial_t u_h = A_h^{DG} u_h + l_h$.

Solving the Poisson Equation

Using the formulations from before, solving the Poisson equation with NGSolve

$$\begin{aligned} -\Delta u &= 2y(1-y) + x(1-x), \quad (x,y) \in [0,1]^2, \\ u(x,y) &= 0, \quad (x,y) \in \partial[0,1]^2 \end{aligned}$$

requires specifying a mesh, the linear forms A^{DG} and l and then solving the resulting linear system. The exact solution is $u = x(1-x)y(1-y)$).

The choice of the parameter β is important because if it is chosen too small, A^{DG} is not coercive and the system therefore not solvable. A larger β also results in a smoother solution (Figures 3 and 4).

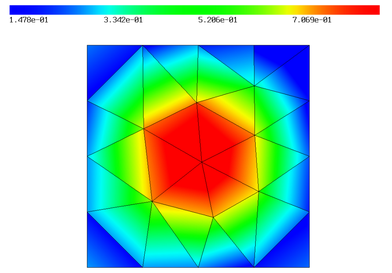


Figure: Plot of u_h with $\beta = 2.5$, $N = 1$, $h_{max} = 0.3$.

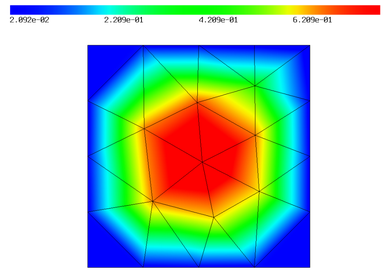


Figure: Plot of u_h with $\beta = 25$, $N = 1$, $h_{max} = 0.3$.

For polygonal geometries the method converges with $\mathcal{O}(h_{max}^{N+1})$. For other geometries, e.g. circles/spheres a geometric error of magnitude $\mathcal{O}(h)$ is introduced.

Solving the Time-Dependent Diffusion Advection Equation

Given an initial condition u_0 we have to advance the semi-discrete system

$$M_h \partial_t u_h = A_h^{DG} u_h + l_h$$

in time by choosing an appropriate time-stepping method.

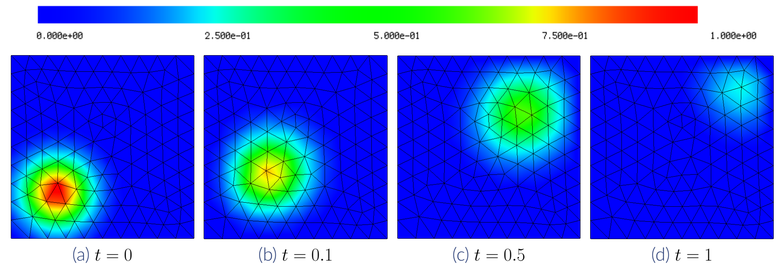
For $D > 0$ and $\beta >> 0$ the ODE is stiff, meaning implicit methods must be used. One implicit Euler step for this problem is given by

$$(M_h + \Delta t A_h^{DG}) u_h^{n+1} = M_h u_h^n + l_h^n,$$

which can be solved efficiently since both M_h and A_h are sparse matrices.

In a similar fashion, higher-order time-integration methods can be constructed from singly diagonally implicit Runge-Kutta methods.

Figures 5a-5d show the time-evolution of the diffusion-advection equation for $D = 0.05$ and $b(x,y) = (1-x, 1-y)$ with $u_0 = e^{-32((x-0.1)^2+(y-0.1)^2}$.



(a) $t = 0$ (b) $t = 0.1$ (c) $t = 0.5$ (d) $t = 1$

Figure: Time evolution with $\Delta t = 0.01$, $N = 3$, $h_{max} = 0.1$.