

Discontinuous Galerkin methods

Alan Dodgson
Heriot-Watt University

We will discuss a category of finite element methods known as 'Discontinuous Galerkin' (DG) methods.

These slides are based on "Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations", by Béatrice Rivière.

Table of Contents

- 1D Elliptic problems
- Elliptic problems in higher dimensions
- Comparison to other methods

Part I) 1D Elliptic problems

Model problem

We begin by considering the following model problem:

$$\begin{cases} -(K(x)y'(x))' = f(x) & \text{in } (0,1) \\ y(0) = 1 \\ y(1) = 0 \end{cases}$$

If we set $K(x) \equiv 1$ here, we get Poisson's equation on $(0,1)$ with boundary conditions. If we also set $f(x) \equiv 0$, we get the Laplace equation.

We assume that $f \in C^0(0,1)$, $K \in C^1(0,1)$, and $p \in C^2(0,1)$. For the sake of this model problem, we also assume that on $(0,1)$, $0 < K_0 \leq K(x) \leq K_1$.

Derivation of the DG method

Setting up the DG method is very similar to setting up any other finite element method: the very first thing to do is split up the domain into elements (or, in 1D, intervals). For our 1D case, we create a partition

$$\mathcal{E} = \bigcup_{n=0}^{N-1} I_n = \bigcup_{n=0}^{N-1} (x_n, x_{n+1}) \quad (1)$$

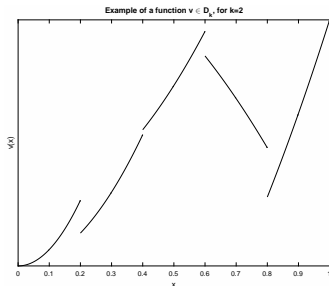
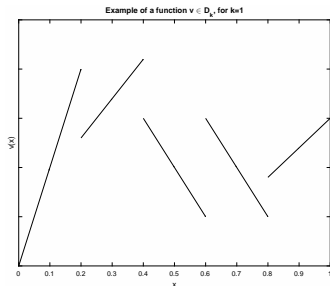
of $(0,1)$.

Discontinuities

The main difference to traditional finite element methods is that, rather than being continuous, the solution obtained by our DG method will belong to the space

$$\mathcal{D}_k(\mathcal{E}) = \{v : (0,1) \rightarrow \mathbb{R} \mid v|_{I_n} \in \mathcal{P}_k(I_n)\}.$$

In other words, over each interval I_n , p will be a continuous polynomial of degree k .



Weak formulation

We need to account for any discontinuities in the solution. We must consider *jump* and *average* terms:

$$[v(x_n)] = v(x_n^+) - v(x_n^-), \quad \{v(x_n)\} = \left(\frac{v(x_n^+) + v(x_n^-)}{2} \right).$$

These are only necessary at points where two or more elements meet.

We set up our DG method using two main steps. We first set up the *weak formulation* of our problem. To do this, we multiply by a test function $v \in \mathcal{D}_k(\mathcal{E})$ and integrate by parts over some interval I_n .

Weak formulation

$$-(K(x)y'(x))' = f(x)$$

$$\implies -(K(x)y'(x))'v(x) = f(x)v(x)$$

$$\implies -\int_{I_n} (K(x)y'(x))'v(x) dx = \int_{I_n} f(x)v(x) dx$$

$$\implies -[K(x)y'(x)v(x)]_{\partial I_n} + \int_{I_n} K(x)y'(x)v'(x) dx = \int_{I_n} f(x)v(x) dx$$

Summing over all intervals I_n gives us

$$\sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} K(x)y'(x)v'(x) dx - \sum_{n=0}^N [K(x_n)y'(x_n)v(x_n)] = \int_0^1 f(x)v(x) dx.$$

Additional terms

Next, we introduce some terms to this equation:

- ▶ The term $\epsilon \sum_{n=0}^N \{K(x_n)v'(x_n)\} [y(x_n)]$ allows some customisation in the method, with the choice of $\epsilon \in \mathbb{R}$. Common choices are $\epsilon = 0, 1$ or -1 .
- ▶ The terms

$$J_0(a, b) = \sum_{n=0}^N \frac{\sigma^0}{h_{n-1,n}} [a(x_n)] [b(x_n)],$$
$$J_1(a, b) = \sum_{n=1}^{N-1} \frac{\sigma^1}{h_{n-1,n}} [a'(x_n)] [b'(x_n)]$$

are known as *penalty* terms; we say they *penalise* the approximate solution for having many or large discontinuities. σ^0 and σ^1 are also parameters that can be chosen, and the h terms are worked out from the lengths of the relevant intervals.

Definition of the DG method

Our numerical method is as follows: find $Y \in \mathcal{D}_k(\mathcal{E})$ such that, for all test functions $v \in \mathcal{D}_k(\mathcal{E})$,

$$a_\epsilon(Y, v) = L(v), \text{ where}$$

$$a_\epsilon(Y, v) = \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} K(x) Y'(x) v'(x) dx - \sum_{n=0}^N \{K(x_n) Y'(x_n)\} [v(x_n)] \\ + \epsilon \sum_{n=0}^N \{K(x_n) v'(x_n)\} [Y(x_n)] + J_0(Y, v) + J_1(Y, v),$$

$$L(v) = \int_0^1 f(x) v(x) dx - \epsilon K(x_0) v'(x_0) + \frac{\sigma_0}{h_{0,1}} v(x_0).$$

Linear system

Our problem

$$a_\epsilon(Y, v) = L(v)$$

can be expressed in the form of a linear system of equations $A\mathbf{c} = \mathbf{b}$ of $N(k+1)$ unknowns. We first choose a set of k basis functions $\{\phi_i^n\}_{i=0}^k$ for $\mathcal{P}_k(I_n)$, for each interval $I_n \in \mathcal{E}$. For example, choosing quadratic monomial functions could lead to

$$\phi_0^n(x) = 1, \quad \phi_1^n(x) = 2 \frac{x - x_{n+\frac{1}{2}}}{x_{n+1} - x_n}, \quad \phi_2^n = 4 \left(\frac{x - x_{n+\frac{1}{2}}}{x_{n+1} - x_n} \right)^2.$$

We can then extend these basis functions to \mathcal{E} as

$$\Phi_i^n(x) = \begin{cases} \phi_i^n(x) & \text{if } x \in I_n \\ 0 & \text{if } x \notin I_n. \end{cases}$$

We can express our approximate solution as

$$Y(x) = \sum_{n=0}^{N-1} \sum_{j=0}^2 c_j^n \Phi_n^j(x), \text{ and we must now solve for } \mathbf{c}.$$

Summary of DG method in 1D

To summarise the method we have derived:

- ▶ We began with a model problem: $-(K(x)y'(x))' = f(x)$, over $(0,1)$.
- ▶ We found the weak formulation of this problem and added some addition terms. This resulted in the new problem $a_\epsilon(Y, v) = L(v)$.
- ▶ This problem was then changed to the solving of a linear system $A\mathbf{c} = \mathbf{b}$. The approximate solution is then a linear combination of basis functions with coefficients from the vector \mathbf{c} .

We will look at a numerical example now.

Numerical experiment

Consider the case where $K(x) \equiv 1$ and $f(x) \equiv 2$ over $(0,1)$; the problem is then

$$\begin{cases} -y''(x) = 2 & \text{in } (0,1) \\ y(0) = 1 \\ y(1) = 0. \end{cases}$$

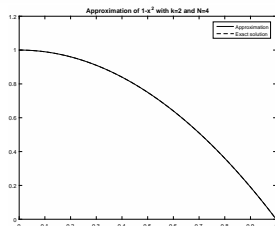
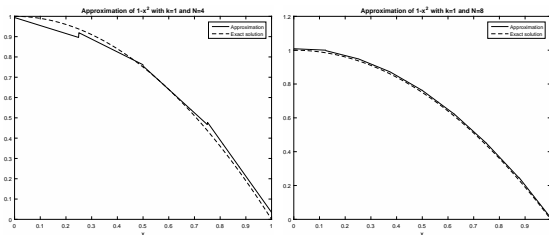
Using the boundary conditions, we can easily work out the solution to be $y(x) = 1 - x^2$; let's check how well the method works. The DG method was used three times to solve this, taking

- ▶ $k = 1, N = 4$
- ▶ $k = 1, N = 8$
- ▶ $k = 2, N = 4$.

In each case, we must compute the left-hand matrix A and the right-hand vector \mathbf{b} , and solve for 8, 16 and 12 unknown coefficients respectively.

Numerical experiment

The entries of the matrix A depend on the product of the various basis functions. The components of the right-hand vector \mathbf{b} involve an integral, and must be approximated using a quadrature rule. After these computations, the following three results were obtained:



Part II) Elliptic problems in higher dimensions

Model problem

For the higher-dimensional elliptic case, we consider the model problem

$$\begin{cases} -\nabla \cdot (K \nabla y) + \alpha y = f & \text{in } \Omega \\ y = g_D & \text{on } \Gamma_D \\ K \nabla y \cdot \mathbf{n} = g_N & \text{on } \Gamma_N \end{cases}$$

with K a matrix-valued function, $f \in L^2(\Omega)$, $g_D \in H^{1/2}(\Omega)$ and $g_N \in L^2(\Omega)$. A strong solution would satisfy $y \in L^2(\bar{\Omega})$.

In the 1D case, we:

- ▶ Found the weak formulation by adding a test function and integrating by parts
- ▶ Obtained the variational formulation by introducing some new terms

We will follow these same steps here.

Preliminaries

We need to introduce a couple of function space-related ideas here which were not needed in our 1D case:

- ▶ For this case we must use Sobolev spaces:

$$H^s(\Omega) = \{v \in L^2(\Omega) \mid D^\alpha v \in L^2(\Omega) \text{ if } |\alpha| \leq s\}$$

- ▶ In particular, we need *broken* Sobolev spaces

$$H^s(\mathcal{E}) = \{v \in L^2(\Omega) \mid v|_E \in H^s(\mathcal{E}) \text{ for all } E \in \mathcal{E}\}$$

and *fractional* Sobolev spaces $H^{s+1/2}(\Omega)$.

- ▶ We also need to use the *trace theorems*, which allow us to extend functions and their derivatives to the boundary of our domain, provided $s > \frac{3}{2}$.

Weak formulation

Like the one-dimensional case, to solve this problem using a numerical DG method we must first obtain this problem in weak formulation. We need to use a generalized version of Green's first identity:

$$\int_{\Omega} \left(K \nabla a \cdot \nabla b + (\nabla \cdot (K \nabla a)) b \right) dx = \int_{\partial\Omega} K (\nabla a \cdot \mathbf{n}) b dx$$

This time, we split up the solution y using the trace theorem: set $y = y_D + w$ where $y_D = g_D$ on $\partial\Omega$ and $w \in H_0^1(\Omega)$. We once again multiply by a test function $v \in H_0^1$ and integrate by parts:

Weak formulation

$$\begin{aligned} -\nabla \cdot (K\nabla y) + \alpha y &= f \\ \implies -\nabla \cdot (K\nabla y)v + \alpha yv &= fv \\ \implies -\int_{\Omega} \nabla \cdot (K\nabla y)v + \int_{\Omega} \alpha yv &= \int_{\Omega} fv \\ \implies \int_{\Omega} \left(K\nabla y \cdot \nabla v + \alpha yv \right) &= \int_{\Omega} fv \end{aligned}$$

Now splitting up $y = y_D + w$ gives us

$$\int_{\Omega} \left(K\nabla y_D \cdot \nabla v + \alpha y_D v \right) = \int_{\Omega} fv - \int_{\Omega} \left(K\nabla w \cdot \nabla v + \alpha wv \right).$$

Variational formulation

In the 1D case, we introduced penalty terms. We do the same here:

$$J_0^{\sigma_0, \beta_0}(v, w) = \sum_e \frac{\sigma_e^0}{|e|^{\beta_0}} \int_e [v][w],$$

$$J_1^{\sigma_1, \beta_1}(v, w) = \sum_e \frac{\sigma_e^1}{|e|^{\beta_1}} \int_e [K \nabla v \cdot \mathbf{n}_e][K \nabla w \cdot \mathbf{n}_e]$$

Variational formulation

The variational formulation of the DG method for this problem is:
find $y \in H^s(\mathcal{E})$ ($s > \frac{3}{2}$) such that $a_\epsilon(y, v) = L(v)$ for all
 $v \in H^s(\mathcal{E})$, where

$$a_\epsilon(y, v) = \sum_E \int_E K \nabla y \cdot \nabla v + \int_\Omega \alpha y v - \sum_e \int_e \{K \nabla y \cdot \mathbf{n}_e\} [v] + \\ \epsilon \sum_e \int_e \{K \nabla v \cdot \mathbf{n}_e\} [y] + J_0^{\sigma_0, \beta_0}(y, v) + J_1^{\sigma_1, \beta_1}(y, v),$$

$$L(v) = \int_\Omega f v + \epsilon \sum_e \int_e \left(K \nabla y \cdot \mathbf{n}_e + \frac{\sigma^0}{|e|^{\beta_0}} v \right) g_D + \sum_e \int_e y g_N.$$

Reference elements

When it comes to running this method on a computer, rather than computing integrals over each 'physical' element, it is much easier to map each one to a 'reference' element.

Part III) Comparison to other methods

I'll finish by highlighting a few of the differences between the DG method and other finite element methods which give a continuous approximation to a problem.

- ▶ On a small rectangular mesh, the DG method is more cost-efficient if high-order polynomials are being used. For triangular meshes, the DG method is more costly.
- ▶ It is very easy when using DG to implement refinement and derefinement, and have different orders of polynomials on different elements. Due to lack of continuity restraints, we can introduce as many new nodes as we like.
- ▶ The DG method can be reduced to a balance equation on each element. This gives us local mass conservation, able to track the amount of mass passing through the boundary to other elements. Other finite element methods give us only global mass balance.

Thanks for listening