

Discontinuous Galerkin Methods

Li, Yang

Discontinuous Galerkin methods (DG methods) in mathematics form a class of numerical methods for solving partial differential equations. They combine features of the finite element and the finite volume framework and have been successfully applied to hyperbolic, elliptic and parabolic problems arising from a wide range of applications. DG methods have in particular received considerable interest for problems with a dominant first-order part, e.g. in electrodynamics and fluid mechanics.

Discontinuous Galerkin methods were first proposed and analyzed in the early 1970's as a technique to numerically solve partial differential equations. In 1973 Reed and Hill introduced a DG method to solve the hyperbolic neutron transport equation. The origin of the DG method for elliptic problems cannot be traced back to a single publication as features such as jump penalization in the modern sense were developed gradually. However, among the early influential contributors were Babuška, J.-L. Lions, Nitsche and Zlamal. A more complete account of the historical development and an introduction to DG methods for elliptic problems is given in a publication by Arnold, Brezzi, Cockburn and Marini.

1. Methods of solving PDEs

As we know, there are lots of numerical approaches to solve partial differential equations. Among these methods are widely used finite difference, finite element, and finite volume methods, which are all techniques used to derive discrete representations of the spatial derivative operators. In this chapter, we will mainly discuss about how these three methods work and what are the differences between them in general. These will help us to understand DG methods in a better way.

For instance, we consider the one-dimensional scalar conservation law for the unknown solution $u(x, t)$

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = g, x \in \Omega$$

subject to an appropriate set of initial conditions and boundary conditions on the boundary, $\partial\Omega$. Here $f(u)$ is the flux, and $g(x, t)$ is some prescribed forcing function.

The construction of any numerical methods for solving a partial differential equation requires one to consider the two choices:

How to represent the solution $u(x, t)$ by an approximate solution $u_h(x, t)$?

In which sense will this approximation solution $u_h(x, t)$ satisfy the partial differential equation?

The two choices above separate the different methods and define the properties of the methods.

1.1 Finite Difference Methods

In this approach, a grid, $x^k, k = 1 \dots N_p$, is laid down in space and spatial derivatives are approximated by difference methods. The conservation law is approximated as:

$$\frac{du_h(x^k, t)}{dt} + \frac{f_h(x^{k+1}, t) - f_h(x^{k-1}, t)}{h^k + h^{k-1}} = g(x^k, t)$$

where u_h and f_h are the numerical approximation to the solution and the flux, respectively, and $h^k = x^{k+1} - x^k$ is the local grid size. The reconstruction scheme is assumed by:

$$x \in [x^{k-1}, x^{k+1}]: u_h(x, t) = \sum_{i=0}^2 a_i(t)(x - x^k)^i, f_h(x, t) = \sum_{i=0}^2 b_i(t)(x - x^k)^i$$

where the coefficients $a_i(t)$ and $b_i(t)$ are found by requiring that the approximate function interpolates at the grid points, x^k . Inserting these local approximations into the conservation law equation, we can get the residual:

$$R_h(x, t) = \frac{\partial u_h}{\partial t} + \frac{\partial f_h}{\partial x} - g(x, t)$$

We can see that FDM is simple to implement. However, the reliance on the local one-dimensional polynomial approximation enforces a dimension-by-dimension structure in higher dimensions. Additional complications caused by the simple underlying structure are introduced around boundaries and discontinuous internal layers. This makes FDM ill-suited to deal with complex geometries. These lead us to know that to ensure geometric flexibility, one should abandon the simple one-dimensional approximation and we consider about an element-based discretization.

1.2 Finite Volume Methods

A method closely related to the finite difference method, but with added geometric

flexibility, is the finite volume method. The solution $u(x,t)$ is approximated on the element by a constant, $\bar{u}^k(t)$, at the center, x^k , of the element. The residual becomes:

$$x \in D^k : R_h(x,t) = \frac{\partial \bar{u}^k}{\partial t} + \frac{\partial f(\bar{u}^k)}{\partial x} - g(x,t)$$

where the element is defined as $D^k = [x^{k-1/2}, x^{k+1/2}]$ with $x^{k+1/2} = \frac{1}{2}(x^k + x^{k+1})$. Use the Divergence Theorem then leads to the scheme:

$$h^k \frac{d\bar{u}^k}{dt} + f^{k+1/2} - f^{k-1/2} = h^k \bar{g}^k$$

To reconstruct the solution, we assume:

$$u_h(x) = \sum_{i=0}^p a_i (x - x^k)^i$$

To find the $p+1$ unknown coefficient, we need information from at least $p+1$ cells. In the simple one-dimensional case, this will be done straightforwardly, as for the finite difference scheme. However, the need for a high-order reconstruction reintroduces the need for a particular grid structure and thus destroys the geometric flexibility of the finite volume method in higher dimensions.

1.3 Finite Element Methods

Define the element D^k at the interval bounded by the grid points $[x^k, x^{k+1}]$ and with a total of K elements where, in one spatial dimension, $K = N_p - 1$. Inside this element, we assume that the local solution is expressed in the form:

$$x \in D^k : u_h(x) = \sum_{n=1}^{N_p} b_n \psi_n(x)$$

where we have introduced the use of a locally defined basis function, $\psi_n(x)$. With this local element-based model, each element shares the nodes with one other element. We have a global representation of u_h as:

$$u_h(x) = \sum_{k=1}^K u(x^k) N^k(x)$$

where the piecewise linear shape function, $N^i(x_j) = \delta_{ij}$ is the basis function. Define

a space of test functions, V_h , and require that the residual is orthogonal to all test functions in this space:

$$\int_{\Omega} \left(\frac{\partial u_h}{\partial t} + \frac{\partial f_h}{\partial x} - g_h \right) \phi_h(x) dx = 0, \forall \phi_h \in V_h$$

Different scheme is determined by how the space of test functions is defined. The classical choice is the spaces spanned by the basis functions and test functions are the same. This will lead to:

$$\int_{\Omega} \left(\frac{\partial u_h}{\partial t} + \frac{\partial f_h}{\partial x} - g_h \right) N^j(x) dx = 0,$$

We will get the scheme:

$$\mathbf{M} \frac{d\mathbf{u}_h}{dt} + \mathbf{S}\mathbf{f}_h = \mathbf{M}\mathbf{g}_h$$

where $\mathbf{M}_{ij} = \int_{\Omega} N^i(x)N^j(x)dx$, $\mathbf{S}_{ij} = \int_{\Omega} N^i(x) \frac{dN^j}{dx} dx$, reflects the globally defined mass matrix and stiffness matrix respectively. This classical finite element method is easy to extend to high-order approximation by adding additional degrees of freedom to the element. However, the problems are the semi-discrete scheme becomes implicit and \mathbf{M} must be inverted. Compared to FDM and FVM, FEM is not a good choice to deal with time dependent problem.

2. Introduction of Discontinuous Galerkin Methods

The Discontinuous Galerkin method is somewhere between a finite element and a finite volume method and has many good features of both, utilizing a space of basis and test functions that mimics the finite element method but satisfying the equation in a sense closer to the finite volume method. It provides a practical framework for the development of high-order accurate methods using unstructured grids. The method is well suited for large-scale time-dependent computations in which high accuracy is required. An important distinction between the DG method and the usual finite-element method is that in the DG method the resulting equations are local to the generating element. The solution within each element is not reconstructed by looking to neighboring elements. Its compact formulation can be applied near boundaries without special treatment, which greatly increases the robustness and accuracy of any boundary condition implementation.

2.1 From FEM and FVM to DG-FEM

We maintain the definition of elements as in the finite element scheme such that $D^k = [x^k, x^{k+1}]$. However, to ensure the locality of the scheme, we duplicate the variables located at the nodes x^k , and the vector of unknowns is defined as:

$$\mathbf{u}_h = [u^1, u^2, u^2, u^3, \dots, u^{K-1}, u^K, u^K, u^{K+1}]^T$$

Assume that the local solution in each element is:

$$x \in D^k : u_h^k(x) = u^k \frac{x - x^{k+1}}{x^k - x^{k+1}} + u^{k+1} \frac{x - x^k}{x^{k+1} - x^k} = \sum_{i=0}^1 u^{k+i} l_i^k(x)$$

and likewise for the flux, f_h^k . The space of basis functions is defined as

$V_h = \bigoplus_{k=1}^K \{l_i^k\}_{i=0}^1$. The local residual is

$$x \in D^k : R_h(x, t) = \frac{\partial u_h^k}{\partial t} + \frac{\partial f_h^k}{\partial x} - g(x, t),$$

Require that the residual is orthogonal to all test functions $\phi_h \in V_h$, leads to:

$$\int_{D^k} R_h(x, t) l_i^k(x) dx = 0$$

for all the test functions, $l_i^k(x)$. Similar to finite volume method, use Gauss' theorem

to obtain: $\int_{D^k} \frac{\partial u_h^k}{\partial t} l_j^k - f_h^k \frac{\partial l_j^k}{\partial x} - g l_j^k dx = -[f_h^k l_j^k]_{x^k}^{x^{k+1}}$. We want the right-hand-side to

connect elements, and both D^k and D^{k+1} depends on the flux evaluation at x^{k+1} .

So we introduce the numerical flux, f^* , as the unique value to be used at the interface and obtained by coming information from both elements. Then we recover the scheme:

$$\int_{D^k} \frac{\partial u_h^k}{\partial t} l_j^k - f_h^k \frac{\partial l_j^k}{\partial x} - g l_j^k dx = -[f^* l_j^k]_{x^k}^{x^{k+1}}$$

applying Gauss' theorem again:

$$\int_{D^k} R_h(x, t) l_j^k(x) dx = -[(f_h^k - f^*) l_j^k]_{x^k}^{x^{k+1}}$$

These two are the DG-FEM schemes for the scalar conservation law in weak and strong form, respectively. The choice of f^* leads to different scheme and can introduce the dynamics of the problem. We can also introduce the terminology like in FEM (e.g. stiffness and mass matrix).

2.2 General formulation

Consider the nonlinear, scalar, conservation law:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, x \in [L, R]$$

subject to appropriate initial conditions

$$u(x, 0) = u_0(x).$$

The boundary conditions are provided when the boundary is an inflow boundary:

$$u(L, t) = g_1(t) \quad \text{when} \quad f_h(u(L, t)) \geq 0,$$

$$u(R, t) = g_2(t) \quad \text{when} \quad f_h(u(R, t)) \leq 0,$$

We still assume that the global solution can be well approximated by a space of piecewise polynomial functions, defined on the union of D^k , and require the residual to be orthogonal to space of the test functions, $\phi_h = \bigoplus_{k=1}^K \phi_h^k \in V_h$, to recover the locally defined weak formulation:

$$\int_{D^k} \left(\frac{\partial u_h^k}{\partial t} \phi_h^k - f_h^k(u_h^k) \frac{d\phi_h^k}{dx} \right) dx = - \oint_{\partial D^k} \hat{\mathbf{n}} \cdot f^* \phi_h^k dx,$$

and the strong form:

$$\int_{D^k} \left(\frac{\partial u_h^k}{\partial t} + \frac{\partial f_h^k(u_h^k)}{\partial x} \right) \phi_h^k dx = \oint_{\partial D^k} \hat{\mathbf{n}} \cdot (f_h^k(u_h^k) - f^*) \phi_h^k dx,$$

for all test functions $\phi_h^k \in V_h^k$. Assume that all local test functions can be represented

by using a local polynomial basis, $\psi_n(x)$, as $x \in D^k : \phi_h^k(x) = \sum_{n=1}^{N_p} \hat{\phi}_n^k \psi_n(x)$, and leads

to N_p equations as:

$$\int_{D^k} \left(\frac{\partial u_h^k}{\partial t} \psi_n - f_h^k(u_h^k) \frac{d\psi_n}{dx} \right) dx = - \oint_{\partial D^k} \hat{\mathbf{n}} \cdot f^* \psi_n dx,$$

$$\int_{D^k} \left(\frac{\partial u_h^k}{\partial t} + \frac{\partial f_h^k(u_h^k)}{\partial x} \right) \psi_n dx = \oint_{\partial D^k} \hat{\mathbf{n}} \cdot (f_h^k(u_h^k) - f^*) \psi_n dx,$$

3. Works in one dimension

See the examples of the presentation.

4. Reference

1. Jan S Hesthaven, Tim Warburton: Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications, Springer
2. Cockburn B, Shu CW: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, MATHEMATICS OF COMPUTATION, v52 (1989), pp.411-435.
3. http://lsec.cc.ac.cn/lcfd/DGM_mem.html
4. <http://www.wikipedia.org/>
5. <http://www.nudg.org/>