Discontinuous Galerkin Methods

Vishal Hariharan [1] Dr. Aditya Konduri [2]

[1]Indian Institute of Technology Madras [2]Indian Institute of Science

Introduction

Consider the one-dimensional scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = S(x), \quad x \in \Omega, \tag{1}$$

subject to an appropriate set of initial and boundary conditions

- We will need to decide how we solve this system, and we will need to discretize in space and time
- How does DG compare with already established methods such as FEM, FVM ?

Weak Form

- The weak form of the PDE is the integral (over the global domain Ω) of our solution times some test function ϕ:

$$\int_{\Omega} \frac{\partial u}{\partial t} \phi \, dx + \int_{\Omega} \frac{\partial f(u)}{\partial x} \phi \, dx = 0 \tag{2}$$

- In DG the global domain is split into K elements, where the local solution is defined for a particular element

- The global solution is then the direct sum of each of these local polynomials, a piece-wise polynomial

Domain Decomposition

- Integration by-parts to reduce smoothness requirements

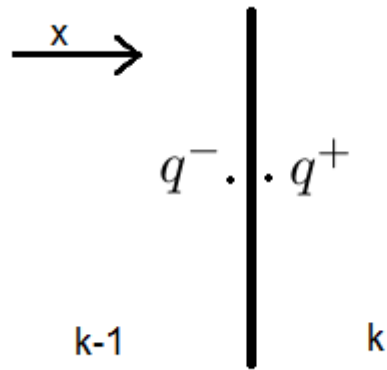
$$\int_k \frac{\partial u}{\partial t}\,\phi\,dx + [f(u)\phi]\Big|_{x_L}^{x_R} - \int_k f(u)\,\frac{d\phi}{dx}\,dx = 0 \qquad (3)$$

- The integration domain is now over the element instead of the whole domain such that $\{x|x \in k, x_L \leq x \leq x_R\}$

- This is similar to a FEM approach, however continuity is not enforced across elements

Flux function

- Many choices: Lax-Friedrichs, Richtmyer, Godunov, Osher, etc
- In the linear case $f(q) = cq$, all the above numerical fluxes coincide with the so-called *upwind* flux
- Flux is very important in wave dominated problems



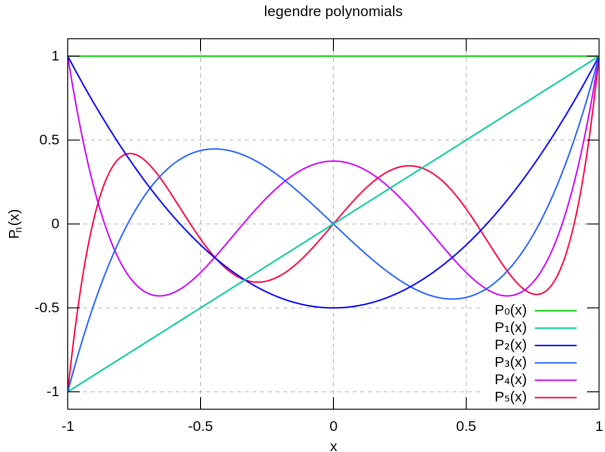
Upwind flux

Soln approximation

- We need a suitable basis set for our solution approximation
- To get a flexible method, we'd like to be able to generate an arbitrary order solution approximation. How do we do this in code?
- We can think in terms of nodal or modal

$$u(x) = \sum_{i=0}^{M} a_i \psi_i(x) \tag{4}$$

Legendre Polynomials



https://en.wikipedia.org/wiki/Legendrepolynomials

Basis functions

Let us begin by introducing the affine mapping

$$x \in \mathbf{D}^k : \; x(r) = x_l^k + \frac{1+r}{2} h^k, \; h^k = x_r^k - x_l^k \tag{5}$$

with the reference variable $r \in \mathbf{I} = [-1, 1]$. We consider local polynomial approximations of the form

$$x \in \mathbf{D}^k : \; u_h^k(x(r), t) = \sum_{n=1}^{N_p} \hat{u}_n^k \psi_n(r) = \sum_{n=1}^{N_p} u_h^k(x_i, t) l_i^k(r) \tag{6}$$

Basis functions (contd...)

$$\psi_n(r) = \widetilde{P}_{n-1}(r) = \frac{P_{n-1}(r)}{\gamma_{n-1}}$$

where $P_n(r)$ are the classic Legendre polynomials of order n and

$$\gamma_n = \frac{2}{2n+1}$$

is the normalization.

What are the benefits of using Legendre functions?

Initial approximation

Equating inner products of each of the test functions with our approximation, and with that of initial condition $u_0(x)$,

(Eq 2.2) Cockburn, Bernardo and Shu, Chi-Wang
Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems
In : *Journal of Scientific Computing 16/3 (2001), 173-261.*

$$\hat{u}_n \simeq \sum_{i=1}^{N_p} u(r_i)\widetilde{P}_{n-1}(r)w_i \tag{7}$$

where r_i are the quadrature points and w_i are the weights.

Lagrange interpolation

The nodal approach is convenient to generalize problems to higher dimensions. Therefore, if

$$u(r) \simeq u_h(r) = \sum_{n=1}^{N_p} \hat{u}_n \widetilde{P}_{n-1}(r),$$

is an interpolant (i.e., $u(\varepsilon_i) = u_h(\varepsilon_i)$ at the grid points, ε_i,

$$u_h(r) = \sum_{n=1}^{N_p} u(\varepsilon_i) l_i(r) \quad \text{where} \quad l_i(r) = \prod_{n=1}^{Np} \frac{r - \varepsilon_j}{\varepsilon_i - \varepsilon_j}$$

1D advection-diffusion eqn

- We have done what seems like quite a bit of tangential work, but it now pays off. Let us illustrate through an example

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \alpha \frac{\partial^2 u}{\partial x^2} \quad \text{in} \quad (0,1) \times (0,T) \tag{8}$$

$$u(x,0) = \sin(2\pi x), \quad u(0,t) = u(1,t) \;\; \& \;\; u_x(0,t) = u_x(1,t)$$

- We will now substitute the expression for our solution approximation and arrive at linear systems describing evolution of the variable $u(x,t)$

Semi-discrete system

Let us revisit the weak form

$$\int_k \frac{\partial u}{\partial t}\,\phi\,dx + \left[(cu - \alpha u_x)\phi\right]\Big|_{x_L}^{x_R} - \int_k (cu - \alpha u_x)\,\frac{d\phi}{dx}\,dx = 0 \qquad (9)$$

Plugging in $u(x,t) = \sum_{i=1}^{N} a_i(t)\psi_i(x)$,

$$\int_k \frac{\partial u}{\partial t}\,\phi_j\,dx = \sum_{i=1}^{N} \frac{da_i}{dt}\int_k \psi_i(x)\phi_j(x)\,dx = \sum_{i=1}^{N} \frac{da_i}{dt}\int_I l_i(r)l_j(r)\frac{h^k}{2}dr$$

Semi-discrete system (contd...)

$$\int_k \sum_{i=1}^{N} \left[c a_i \psi_i(x) \phi_j'(x) - \alpha a_i \psi_i'(x) \phi_j'(x) \right] dx = \sum_{i=1}^{N} c a_i \int_I l_i(r) l_j'(r) dr - \sum_{i=1}^{N} \alpha a_i \int_I l_i'(r) l_j'(r) dr$$

Upwind : $$[u\phi]\Big|_{x_L}^{x_R} = u_K(x_R)\phi_j(x_R) - u_{K-1}(x_R)\phi_j(x_L)$$

Downwind : $$[u_x\phi]\Big|_{x_L}^{x_R} = \{u_x\}_{K+1}(x_L)\phi_j(x_R) - \{u_x\}_K(x_L)\phi_j(x_L)$$

Semi-discrete system (contd...)

$$\text{Mass matrix} \quad \mathcal{M}^k \doteq \frac{h^k}{2} \int_I l_i(r) l_j(r) dr$$

$$\text{Stiffness matrix} \quad \mathcal{S}^k \doteq c \int_I l_i(r) l_j'(r) dr - \alpha \int_I l_i'(r) l_j'(r) dr$$

Since the Legendre functions form a complete set, we can represent Lagrange interpolants through Legendre basis to simplify above expressions

(2008) Making it work in one dimension In: Nodal Discontinuous Galerkin Methods : Texts in Applied Mathematics, vol 54. Springer, New York

$$[c\hat{u}\phi]\Big|_{x_L}^{x_R} = \begin{bmatrix} -cu_{K-1} & 0 & 0 & \dots & . & 0 & 0 & cu_K \end{bmatrix}^T$$

$$[\alpha\hat{u_x}\phi]\Big|_{x_L}^{x_R} = \begin{bmatrix} -\alpha\{u_x\}_K & 0 & 0 & \dots & . & 0 & 0 & \alpha\{u_x\}_{K+1} \end{bmatrix}^T$$

$$\text{Flux vector } \hat{f} = \{c[\hat{u}\phi]|_{x_L}^{x_R} - \alpha[\hat{u_x}\phi]|_{x_L}^{x_R}\}$$

Assembly of system

- We can now combine the mass, stiffness, and numerical flux terms

$$\mathcal{M}^k \frac{\partial \boldsymbol{u}_h^k}{\partial t} + \hat{f} - \mathcal{S}^T \boldsymbol{u}_h^k = 0 \tag{10}$$

- Solving for $\frac{\partial \boldsymbol{u}_h^k}{\partial t}$,

$$\frac{\partial \boldsymbol{u}_h^k}{\partial t} = \{\mathcal{M}^k\}^{-1} \left(\mathcal{S}^T \boldsymbol{u}_h^k - \hat{f} \right)$$

Time discretization

We obtain from the semi-discrete equation

$$\frac{\partial \boldsymbol{u}_h}{\partial t} = \boldsymbol{\mathcal{L}}_h(\boldsymbol{u}_h, t)$$

RK4 trial steps to determine $\boldsymbol{u}_h$, the vector of unknowns at next time step

$$\boldsymbol{k}^{(1)} = \boldsymbol{\mathcal{L}}_h(\boldsymbol{u}_h^n, t^n)$$

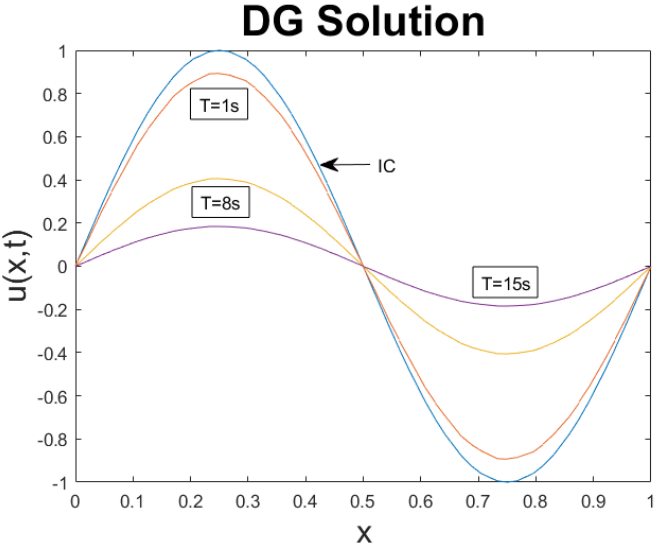
$$\boldsymbol{k}^{(2)} = \boldsymbol{\mathcal{L}}_h(\boldsymbol{u}_h^n + \frac{1}{2}\Delta t \boldsymbol{k}^{(1)}, t^n + \frac{1}{2}\Delta t)$$

$$\boldsymbol{k}^{(3)} = \boldsymbol{\mathcal{L}}_h(\boldsymbol{u}_h^n + \frac{1}{2}\Delta t \boldsymbol{k}^{(2)}, t^n + \frac{1}{2}\Delta t)$$

$$\boldsymbol{k}^{(4)} = \boldsymbol{\mathcal{L}}_h(\boldsymbol{u}_h^n + \Delta t \boldsymbol{k}^{(3)}, t^n + \Delta t)$$

$$\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n + \frac{1}{6}\Delta t\Big(\boldsymbol{k}^{(1)} + 2\boldsymbol{k}^{(2)} + 2\boldsymbol{k}^{(3)} + \boldsymbol{k}^{(4)}\Big)$$

Soln evolution



MATLAB simulation

Convergence & stability

- How does choice of initial parameters affect stability?
- Does the smoothness of the initial soln seem to effect the rate of convergence? (e.g. sin(x) vs gaussian curve)
- Does h or p refinement seem to be more efficient?

Reactive eqn

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \alpha\frac{\partial^2 u}{\partial x^2} + F(x,t)$$

$$F(x,t) = \begin{cases} Aexp(-B/u)\{1 - t/t_{end}\} & u \neq 0 \\ 0 & u = 0 \end{cases}$$

Parameters A and B impact the numerical stability