An Introduction to Discontinuous Galerkin Methods

Module 1: What is DG?

J. Bevan

Department of Mechanical Engineering, Grad Student
University of Massachusetts at Lowell

Module 1: What is DG?

Overall Content Structure: Assumed Prerequisite Knowledge

- It is assumed the interested viewer is an advanced undergrad or graduate student with the typical STEM background of Calculus, Linear Algebra, and ODEs/PDEs.
- Additionally, it is assumed the viewer has at least a basic background in a programming language of their choice (Matlab etc.)
- Finally it is assumed the viewer has taken a general Numerical Methods course as well a Solution of PDEs course.
- Not intended to teach common underlying techniques (interpolation etc.), but we may [Recall] important features of them.

Numerical Methods Prerequisites

- Linear algebra
  - Vector spaces, bases, properties, etc.
  - Orthogonality
  - maybe some useful spaces (Hilbert, square integrable, etc)?
- Polynomial interpolation(1 and 2D)
  - Lagrange, Hermite
  - monomial basis (and ill-conditioned nature)
  - Orthogonal basis (Legendre, Chebyshev)
  - L2 projection
  - choice of interpolation points (equispaced, GL, LGL, etc)
  - Runge phenomenon
  - Vandermonde matrix (transformation from modal to nodal spaces)
- Quadrature (1 and 2D)
  - Newton-Cotes
  - Gauss/Hermite(Legendre)
  - relation to interpolation
- Solution of ODEs
  - Forward Euler
  - RK4
  - Implicit schemes (e.g. Backward Euler)
  - Stability, Convergence

Solution of PDEs Prerequisites

- Domain representation
  - meshing
  - BCs (Neumann and Dirichlet)
- Finite difference methods (FDM)
  - Pointwise spatial derivatives
  - Computational vs Physical domains
  - Basic mapping (bilinear)
- Finite volume methods (FVM)
  - Flux functions
  - Artificial viscosity
  - Linear vs nonlinear fluxes
- Finite element methods (FEM)
  - Weak and strong form formulation
  - Piecewise linear solution approximation
  - Galerkin style test functions
  - local support

Lecture Goals

- Understand DG spatial discretization (advective)
  - DG weak form (test function to minimize residual or test function orthogonal)
  - solution approximation (and initial conditions)
  - mapping physical to computation domain (for curvilinear domains)
  - DG Galerkin formulation
  - Integration by parts $\rightarrow$ flux functions (solution smoothness requirements): differences from FEM
  - linear vs non-linear flux: ramifications for semi-discrete system
  - hyperbolic vs parabolic
  - applying BCs (include periodic BCs)
- Understand time discretization
  - Method of lines style semi-discrete form
  - Types: e.g. Forward Euler, RK4
  - CFL condition and stability
- Learn how to apply DG to arbitrary PDEs and realm of applicability
  - intuitive understanding of methodology
  - conceptualization of process (not tied down to specific examples)
  - understand pros/cons
  - understand how DG "simplifies" to FVM and FEM
- Generate runnable code of your own
  - Self-contained set of knowledge and algorithms to be able to write a full solver

Topics Layout

Module 1: What is DG?
DG motivation (why vs FEM, FVM, FDM)
Scalar conservation law (linear) PDE
Weak form derivation
Global domain vs local element
Multiple-valued element boundaries
Recall: Flux functions

Topics Layout(cont.)

Module 2: A Simple 1D DG Solver

Linear solution approximation
Test function choice (Galerkin)
Upwind flux
Mass Matrix
Stiffness Matrix
Putting it all together (linear system)
Semi-discrete system
Forward Euler
Investigate h-convergence
Investigate t-convergence
Investigate stability (CFL)

Topics Layout(cont.)

Module 3: To Higher-Orders (nodal)
3A: Sol'n Approximation

Revisit weak form
-Approx. space
-L2 Projection minimizes residual norm
-Test space $\rightarrow$ orthogonal
Monomial basis?
Ill-conditioning of monomials
Recall: Lagrange interpolation (code)
Derive Lagrange spatial approximation
Equispaced interp points?
Runge phenomenon
Why: Bernstein/Markov inequality
Roots of Leg instead

Topics Layout(cont.)

Module 3: To Higher-Orders (nodal)
3B: Discrete System
Numerical Quadrature (Gauss)
Hermite interpolation (2N+1 quad)
Truncation error/exact quadrature
GL Lagrange orthogonality
Local Mapping Fun
Mass Integral -¿diagonal/inversion
Log differentiation
Flux interpolation
Stiffness Integral
Numerical Flux (interpolated)
Assembly of system
RK4 time discretization
Investigate p-convergence (smoothness reqs)

A Pedagogical Comment

- Take advantage of format: replay, pause, speed up, slow down
- Each section may have subsections, but the overall section is intended to be a self-contained concept. The first slide of a new section has the title format **Section**: Subsection
- Easy to "zone-out", before the start of a new section try and put what you learned into action. Make a code snippet to test your understanding or verify a claimed result etc.
- Each Module has a larger self-contained concept. You should be able to put together a script that accomplishes something substantial.

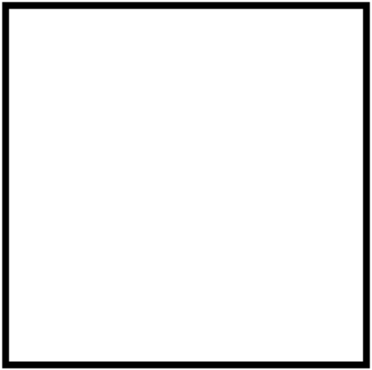
DG Motivation: Why DG?

- Overall purpose: PDE models physical system, solve PDE numerically
- Compare to common techniques FDM, FEM, FVM:
- Increasing order of solution can be more efficient, smooth functions especially. Most comm. packages low order (LO)
- FDM not explicitly conservative, extended stencil for high order (HO) a problem for unstructured grids
- FEM not good for hyperbolic problems, discontinuities in sol'n troublesome
- FVM constant solution of volume necessitates extended stencil for HO, ruins flexibility for unstructured grids
- DG is explicitly conservative, well-suited for hyperbolic problems, able to handle discontinuities, and can still use unstructured grids at HO
- Local nature of solution permits good parallelizability
- Can use numerical flux functions to capture physical behavior

Example PDE: Scalar Conservation Law

A brief motivating example:

Imagine some scalar quantity that is subject to a conservation law



Example PDE: Scalar Conservation Law

- As worked out, PDE for system in 1D is

$$\frac{\partial q}{\partial t} + \frac{\partial f(q)}{\partial x} = S(x) \tag{1}$$

- We will need to decide how we solve this system, and we will need to discretize in space and time
- The first term will be discretized with a Forward Euler approach, the second term will be handled by DG
- For simplicity we assume no sources, and a linear flux with an externally prescribed velocity field

The Weak Form of the PDE

- We will discuss the finer points of this in Module 3, for now permit the following; the weak form solution of the PDE is the integral (over the global domain Ω) of our solution times some test function ϕ:

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

- In DG the global domain is split into K elements, where the local sol'n is defined for a particular element. The global sol'n is then the direct sum of each of these local polynomials, a piecewise polynomial
- This is similar to a FEM approach, however continuity is not enforced across elements

Domain Decomposition: Global vs Local

- The integration domain is now over the element instead of the whole domain such that $\{x|x \in k, x_L \leq x \leq x_R\}$
- To reduce smoothness requirements on the flux we integrate the second term by parts to get

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

- Notice that we include the endpoints of our domain k
- Endpoints of neighboring elements are coincident, so q is multiply defined, what ramifications does this have? Also, we now have K independent local solutions, how do we recover the global solution?

Element Boundaries: Multiply Defined?

- If q is multiply defined at endpoints, what should $f(q)$ be?
- In order to ensure conservation flux between elements should be equal $-f_{k-1}(q(x_R)) = f_k(q(x_L))$
- Take a FVM approach and permit a *numerical flux* denoted $\hat{f}(q)$
- The numerical flux function permits communication between elements, allowing recovery of global sol'n

[Recall] Flux Functions

- Flux functions describe the "flow" of some quantity depending on the quantity itself and possibly other factors
- Permits insertion of problem specific knowledge into an otherwise agnostic PDE
- Many choices, Lax-Friedrichs, Richtmyer, Godunov, Osher, Roe, etc.
- For simplicity we will use the upwind flux where $\hat{f}(q^+, q^-) = cq^-$ if $c > 0$ and cq^+ if $c < 0$