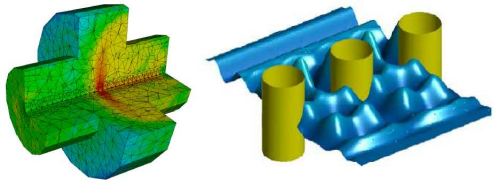
Lecture - Introduction

Ph.D. Course:
Nodal DG-FEM for solving partial differential equations

Allan P. Engsig-Karup
Scientific Computing Section
DTU Informatics
Technical University of Denmark

August 6, 2012

Today!



- Presentation and practical details
- Introduction to DG-FEM methods
- Getting setup for hands-on exercises

Course content

This course is organized and taught by

- Assoc. Prof. Allan Peter Engsig-Karup
  Building 321, r. 016
  DTU Informatics, Scientific Computing Section, DTU, Denmark.
- Prof. Jan Hesthaven
  Building 321, r. 009
  Division of Applied Mathematics, Brown University, USA.

Teaching assistance during afternoons

- Post Doc Ole Lindberg
  DTU Informatics, Scientific Computing Section, DTU, Denmark.

Course content

This course is an activity of GPUlab, DTU Informatics

GPUlab
DTU Informatics

The course is sponsored by the PhD school at Technical University of Denmark

- DTU Informatics Graduate School ITMAN

and organized with support from

- The Danish Center for Applied Mathematics and Mechanics, DCAMM

Course content

The following topics are covered in the course

1. Introduction & DG-FEM in one spatial dimension
2. Implementation and numerical aspects (1D)
3. Insight through theory
4. Nonlinear problems
5. Extensions to two spatial dimensions
6. Introduction to mesh generation
7. Higher-order operators
8. Problem with three spatial dimensions and other advanced topics

Course structure

Week 1:

Time	Monday	Thuesday	Wednesday	Thursday	Friday
08.30-09.00	Breakfast				
09.00-11.30	1	2	3	4	Project work
12.30-16.00	Hands-on	Hands-on	Hands-on	Hands-on	Project work

Week 2:

Time	Monday	Thuesday	Wednesday	Thursday	Friday
08.30-09.00	Breakfast				
09.00-11.30	5	5+6	7	8	Project work
12.30-16.00	Hands-on	Hands-on	Hands-on	Hands-on	Project work

- Lectures: approx. 2.5 h/day, including 15 mins review + 15 mins break.
- Hand-on exercises: approx. 3.5 h/day.

Learning objectives

A student who has met the objectives of the course will be able to:

- Apply the basic ideas underlying discontinuous Galerkin methods.
- Apply the building blocks of DG-FEM methods for the simulation of phenomena described by partial differential equations.
- Identify and exploit the properties and structure of the underlying problem.
- Be able to complete basic analysis to formulate a suitable scheme for a new problem.
- Implement such methods and extensions in Matlab using the provided Matlab based toolbox.
- Skillfully perform numerical experiments.
- Analyse and explain the observed behavior of the methods based on a basic theoretical insight.
- Apply important principles underlying the use of modern numerical methods in selected applications.

Coursework and assessment

This 2-week course has approx. 70 scheduled hours

- Breakfast and coffee/tee (0.5 hours/day)
- Lectures (approx. 2 hours/day)
- Discussions (as needed)
- Hands-on computer exercises (approx. 4 hours/day)
- Lunch (1 hours/day)

To pass the course and get a diploma the requirements are

- Completing a written report for assessment of work
- Satisfactory completion of assignment problems (approx. 40 hours)
- Short notes, produced codes and relevant results from successfully completed exercises during afternoons sessions

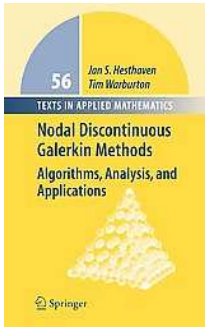
The assignment is divided in two parts

- Each part will be available tuesday morning of each week
- It is highly recommended that you Initiate your work on the assignments after completion of exercises

Practical details

- Background
  - What is your background? Experiences?
  - Why are you here?
- Access to the databar terminals, software and Internet
- Access to Matlab codes, `http://www.nudg.org`
- Access to hands-on exercises/slides/ect., `http://www2.imm.dtu.dk/~apek/DGFEMCourse/`
- General information

Practical details



- Course material:
  Nodal Discontinuous Galerkin Methods - Algorithms, Analysis, and Applications
  By J. S. Hesthaven & T. Warburton (2008), Springer.

Course work

The work in the course should be carried out in teams

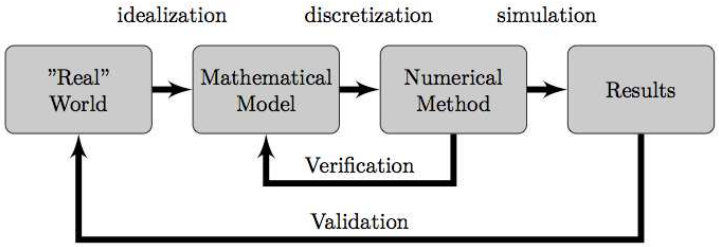
- Two persons per team
- Hands-on exercises and assignment work is made by the team

Everyone is encouraged to take the opportunity to

- Interact!
- Get to know each other!
- Discuss the work!
- Share experiences!

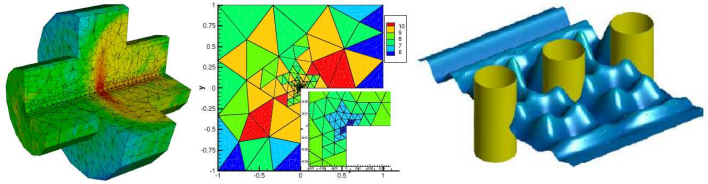
Introduction
- discussion of numerical schemes and properties

Computational Science and Engineering



- Scientific computing has become indispensable for progress and breakthroughs in science through (cost-efficient) numerical experiments and validation against real experiments.
- Numerical simulation using modern computers indispensable today for engineering analysis and prediction of physical events.
- Interplay between experiments, simulation and theory.

Our goals



For the application of numerical methods we want

- accuracy at minimal effort
- flexibility to solve classes of problems with same code
- easy problem prototyping and code maintenance (avoid adhoc solutions)
- ensure that numerical results can be thrusted (verification and validation)

Numerical solution of PDEs

To construct a numerical method for solving PDEs we need to consider

- How to represent the solution $u(x,t)$ by an approximate solution $u_h(x,t)$?
- In which sense will the approximate solution $u_h(x,t)$ satisfy the PDE?

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

Bottom line is that we need ways to

- Generate a (coupled) system of algebraic equations from the well-posed PDE and incorporate boundary conditions
- Solve the system and equations while minimizing unavoidable errors that are introduced in the process

Conservation laws

Conservation laws appear in many brances of computational science and engineering and are typically derived from physical conservation principles, e.g. conservation of energy, momentum and mass.

A general nonlinear conservation law (3D) can be stated in differential form as

$$\partial_t u + \nabla \cdot \mathbf{F}(u) = S(u)$$

or

$$\partial_t u + \partial_x F(u) + \partial_y G(u) + \partial_z H(u) = S(u)$$

where

$$u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \; F(u) = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}, \; G(u) = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix}, \; H(u) = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_m \end{bmatrix}, \; S(u) = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix}$$

$u(\mathbf{x},t)$ is a vector of conserved variables and F, G, H are flux vectors. S is a source vector.

Conservation laws

Examples of conservation laws

- Euler equations of compressible gas dynamics (1D)

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0 && \text{(Mass)} \\ &\frac{\partial \rho u}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} = 0 && \text{(Momentum)} \\ &\frac{\partial E}{\partial t} + \frac{\partial (E+p)u}{\partial x} = 0 && \text{(Energy)} \\ &p = (\gamma - 1)\left(E - \tfrac{1}{2}\rho u^2\right), c = \sqrt{\frac{\gamma p}{\rho}} && \text{(Ideal gas low)} \end{aligned}$$

- Nonlinear shallow water equations (1D)

$$\begin{aligned} &\frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} = 0 && \text{(Mass)} \\ &\frac{\partial hu}{\partial t} + \frac{\partial (hu^2 + \frac{1}{2}gh^2)}{\partial x} = 0 && \text{(Momentum)} \end{aligned}$$

- and many many more...

Conservation laws

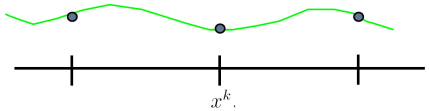
For now, we restrict ourselves to consider the one-dimensional scalar conservation law

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

where $f(u)$ is the flux function, $g(x,t)$ is a source function.

Let's discuss basic ideas, advantages and disadvantages of different classical numerical methods for solving this PDE...

Finite Difference Method



- Domain is represented by a set of collocation points
- Solution is represented locally as a polynomial

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

- PDE is satisfied in a point-wise manner

$$\mathcal{R}_h(x^k) = \frac{du_h(x^k,t)}{dt} + \frac{f_h(x^{k+1},t) - f_h(x^{k-1},t)}{h^k + h^{k-1}} - g(x^k,t) = 0$$

- Local smoothness requirement pose a problem for resolving complex geometries, internal discontinuities and overall grid structure.

Finite Difference Method

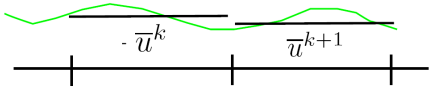
Main benefits

- Simple to understand
- Straightforward implementation on structured meshes
- High-order acurate approximations feasible
- Method is local and can be made explicit in time
- Simple techniques for local adaptivity (upwinding)
- Extensive body of theoretical and practical work on these methods since 1960's

Main problems

- Implementation complexity increases if geometric flexibility is needed
- Less well-suited for problems with discontinuities
- Grid smoothness requirements

Finite Volume Method



- Domain is represented by non-overlapping cells
- Solution is represented locally as a cell average

$$\bar{u}^k \equiv \frac{1}{h^k} \int_{\Omega^k} u^k dx^k$$

- PDE is satisfied on conservation form

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

- The flux function needs to be reconstructed on cell interfaces $x^{k\pm 1/2}$

$$f(x^{k-1/2}, t) = F(\bar{u}^{k-1}, \bar{u}^k), \quad f(x^{k+1/2}, t) = F(\bar{u}^k, \bar{u}^{k+1})$$

Finite Volume Method

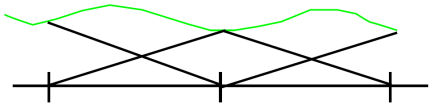
Main benefits

- Robust
- Support resolution of complex geometries
- Well-suited for hyperbolic conservation laws (local upwinding)
- Method is local and can be made explicit in time
- Method is locally conservative (due to telescopic property)
- Extensive theoretical framework since 1970's

Main problems

- Inability to achieve high-order accuracy in a straightforward way on general grids due to requirement for extended stencils (flux reconstruction problem)
- Grid smoothness requirements

Finite Element Method



- Domain is represented by non-overlapping elements
- Solution is represented globally using piecewise continuous polynomials

$$u_h(x) = \sum_{k=1}^{K} u(x_k, t) N^k(x), \quad N^j(x_j) = \delta_{ij}$$

- PDE is satisfied in a global manner

$$\int_{\Omega_h} \left(\frac{\partial u_h}{\partial t} + \frac{\partial f_h}{\partial x} - g_h \right) N^j(x) dx = 0, \quad j = 1, ..., K \quad \Rightarrow \mathcal{M} \frac{du_h}{dt} + \mathcal{S} f_h = \mathcal{M} g_h$$

- The semi-discrete scheme is implicit by construction and reduces overall efficiency for explicit time-integration

Finite Element Method

Main benefits

- Robust
- Systematic implementation on unstructured meshes
- High-order accuracy can be combined with complex geometries
- Well-suited for elliptic problems (global statement)
- Extensive theoretical framework since 1970's

Main problems

- Not well-suited for problems with direction (global statement)
- Implicit in time reduces overall efficiency

Properties of numerical methods

Numerical methods for solving PDEs can in general be characterized by the properties

- Accuracy
  Can we reduce the error? and how fast?
- Flexibility
  What is the range of problems that can be solved using the chosen method?
- Robustness
  Can we always expect a solution from our numerical model?
- Efficiency
  How long does it take to compute our solution?

Note: Very often it is difficult to achieve all properties at once!

⇒ Thus, we need to prioritize!

- Choice is often dictated by domain complexity and required levels of accuracy.

General properties of the numerical methods

Assesment of general properties of some classical numerical methods

	Complex geometries	High-order accuracy and *hp*-adaptivity	Explicit semi-discrete form	Conservation laws	Elliptic problems
FDM	×	✓	✓	✓	✓
FVM	✓	×	✓	✓	(✓)
FEM	✓	✓	×	(✓)	✓
DG-FEM	✓	✓	✓	✓	(✓)

We want a scheme which have the properties

- The local high-order elements of FEM.
- The geometric flexbility of FEM and FVM.
- The local statement of the FVM.

These are exactly the components of the

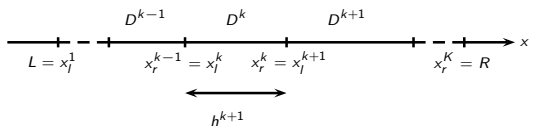
Discontinuous Galerkin Method Finite Element Method

A first look at DGFEM

Formulating a DG-FEM scheme

By subdividing the domain $\Omega \in [L, R]$ similar to FVM/FEM into a union of non-overlapping elements D^k

$$\Omega \cong \Omega_h = \bigcup_{k=1}^{K} D^k$$



we have the basis for geometric flexibility (any type of grid).

Generating the actual grid is a separate problem.

Formulating a DG-FEM scheme

We seek to represent the global solution using local high-order polynomial approximations similar to FEM

$$u(x,t) \cong u_h(x,t) = \bigoplus_{k=1}^{K} u_h^k(x,t),$$

$$u_h^k(x,t) = \sum_{j=1}^{N_p} \hat{u}_j^k(t)\psi_j(x) = \sum_{j=1}^{N_p} u_h^k(x_j^k,t) l_j(x)$$

using either a modal or nodal form.

This is the basis for arbitrary high-order accurate approximations.

Note: both low and high-order approximations then an option in the scheme.

A high-order accurate method has asymptotic behavior $\mathcal{O}(h^p)$ of truncation error for $h \to 0$ with $p > 2$.

Formulating a DG-FEM scheme

We want to find an approximation u_h to the solution u of the general scalar conservation law

$$\partial_t u + \partial_x f(u) = g(x,t), \quad x \in \Omega$$

To do this, we form the local residual on the $k = 1, .., K$ elements

$$x \in \mathcal{D}^k : \mathcal{R}_h^k(x,t) = \partial_t u_h^k + \partial_x f_h^k - g_h^k$$

and require this to vanish locally in a Galerkin sense

$$\int_{\mathcal{D}^k} \mathcal{R}_h^k(x,t) l_i^k(x) dx = 0, \quad i = 1, ..., N_p, \quad k = 1, ..., K$$

This is the basis for a nodal DG-FEM scheme.

However, we are not done yet... all elements are disconnected due to the local statement on the residual.

Formulating a DG-FEM scheme

To connect elements, we apply Gauss's Theorem

$$\int_{\mathcal{D}^k} \mathcal{R}_h^k(x,t) l_i^k(x) dx = \int_{\mathcal{D}^k} \left[\partial_t u_h^k l_j^k + \partial_x u_h^k l_j^k - g_h^k l_j^k \right] dx = 0$$

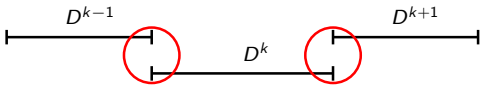
to convert the term with a spatial derivative such that

$$\int_{\mathcal{D}^k} \left[\partial_t u_h^k l_j^k - u_h^k \partial_x l_j^k - g_h^k l_j^k \right] dx = - \oint_{\partial \mathcal{D}^k} \hat{n} \cdot f_h^k l_j^k dx$$

where the boundary integral in 1D takes the form

$$\oint_{\partial \mathcal{D}^k} \hat{n} \cdot f_h^k l_j^k dx = \left[f_h^k l_j^k \right]_{x_r^k}^{x_l^k} = f_h^k(x_r^k)\delta_{N_p j} - f_h^k(x_l^k)\delta_{1j}$$

The solution is not unique at interfaces between adjacent elements.



We have multiple solutions! How can we address this problem?

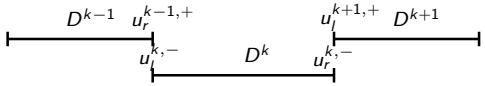
Formulating a DG-FEM scheme

Similar to FVM, we could introduce a numerical flux f^* which approximate the physical flux, i.e.

$$\hat{n} \cdot f^* \cong \hat{n} \cdot f_h^k$$

to address the lack of solution uniqueness at the interfaces. We require that the numerical flux is somehow defined in terms of interior (-) and exterior (+) interface states

$$f^* = f^*(u_h^-, u_h^+)$$



Clearly, the choice of the numerical flux must be important!

Formulating a DG-FEM scheme

So, after having applied Gauss's Theorem we found

$$\int_{\mathcal{D}^k} \left[\partial_t u_h^k l_j^k - u_h^k \partial_x l_j^k - g_h^k l_j^k\right] dx = -\oint_{\partial\mathcal{D}^k} \hat{n} \cdot f_h^k l_j^k dx$$

With the introduction of a numerical flux f^*, the local scheme in the weak form then becomes

$$\int_{\mathcal{D}^k} \left[\partial_t u_h^k l_j^k - u_h^k \partial_x l_j^k - g_h^k l_j^k\right] dx = -\oint_{\partial\mathcal{D}^k} \hat{n} \cdot f^* l_j^k dx$$

Formulating a DG-FEM scheme

From the weak form

$$\int_{\mathcal{D}^k} \left[\partial_t u_h^k l_j^k - u_h^k \partial_x l_j^k - g_h^k l_j^k\right] dx = -\oint_{\partial\mathcal{D}^k} \hat{n} \cdot f^* l_j^k dx$$

we can generate a local linear system by inserting the polynomial approximation u_h^k arriving at the compact scheme

$$\mathcal{M}^k \frac{du_h^k}{dt} - (\mathcal{S}^k)^T f_h^k - \mathcal{M}^k g_h^k = -f^* \delta_{1j} + f^* \delta_{N_p j}$$

where δ_{ij} is Kronecker's delta and the element mass and stiffness[1] matrices have been introduced. These are defined from

$$\mathcal{M}_{ij}^k = \int_{\mathcal{D}^k} l_i^k(x) l_j^k(x) dx, \quad \mathcal{S}_{ij}^k = \int_{\mathcal{D}^k} l_i^k(x) \frac{dl_j^k}{dx} dx$$

[1] In classical finite element terminology, the discrete operator approximating the first derivative is called a convection/advection matrix.

Formulating a DG-FEM scheme

It is also possible to derive yet another scheme from the weak form

$$\int_{\mathcal{D}^k} \left[\partial_t u_h^k l_j^k - u_h^k \partial_x l_j^k - g_h^k l_j^k\right] dx = -\oint_{\partial\mathcal{D}^k} \hat{n} \cdot f^* l_j^k dx$$

by applying Gauss's Theorem once more

$$\int_{\mathcal{D}^k} \left[\partial_t u_h^k l_j^k + \partial_x u_h^k l_j^k - g_h^k l_j^k\right] dx = \oint_{\partial\mathcal{D}^k} \hat{n} \cdot (f_h^k - f^*) l_j^k dx$$

This is the so-called strong form.

We now have two basic DG-FEM schemes. How will they perform?

Formulating a DG-FEM scheme

Consider the strong scheme

$$\int_{\mathcal{D}^k} \left[\partial_t u_h^k l_j^k + \partial_x u_h^k l_j^k - g_h^k l_j^k\right] dx = \oint_{\partial\mathcal{D}^k} \hat{n} \cdot (f_h^k - f^*) l_j^k dx$$

From this we can generate a local linear system of the form

$$\mathcal{M}^k \frac{du_h^k}{dt} + \mathcal{S}^k f_h^k - \mathcal{M}^k g_h^k = (f_h^k - f^*)\delta_{1j} - (f_h^k - f^*)\delta_{N_p j}$$

which is clearly a semi-discrete system of the form

$$\frac{du_h^k}{dt} = -(\mathcal{M}^k)^{-1} \left(\mathcal{S}^k f_h^k - \mathcal{M}^k g_h^k = (f_h^k - f^*)\delta_{1j} - (f_h^k - f^*)\delta_{N_p j}\right)$$

which can be solved by an appropriate explicit/implicit ODE solver.

The first examples...

Examples: error behavior

Consider the simple advection equation on a periodic domain

$$\partial_t u - 2\pi \partial_x u = 0, \quad x \in [0, 2\pi], \quad u(x, 0) = \sin(lx), \quad l = \frac{2\pi}{\lambda}$$

Exact solution is then $u(x, t) = \sin(l(x - 2\pi t))$.

Errors at final time $T = \pi$.

N\ K	2	4	8	16	32	64	Convergence rate
1	-	4.0E-01	9.1E-02	2.3E-02	5.7E-03	1.4E-03	2.0
2	2.0E-01	4.3E-02	6.3E-03	8.0E-04	1.0E-04	1.3E-05	3.0
4	3.3E-03	3.1E-04	9.9E-06	3.2E-07	1.0E-08	3.3E-10	5.0
8	2.1E-07	2.5E-09	4.8E-12	2.2E-13	5.0E-13	6.6E-13	$\cong$ 9.0

Error is seen to behave as

$$||u - u_h||_{\Omega,h} \leq Ch^{N+1}$$

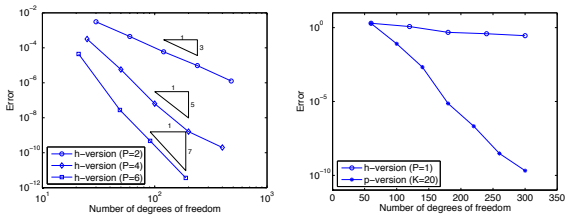
Clearly, paths to convergence are based on adjusting the size of elements (h-convergence), the polynomial order (p-convergence) or combinations hereof.

Examples: error behavior

Consider the linear shallow water equations in one horizontal dimension on a periodic domain

$$\frac{\partial}{\partial t}\begin{bmatrix} \eta \\ u \end{bmatrix} = \begin{bmatrix} 0 & -h \\ -g & 0 \end{bmatrix} \frac{\partial}{\partial x}\begin{bmatrix} \eta \\ u \end{bmatrix}$$

Tests of $h-$ and p-refinement



Again, the error behaves as

$$||u - u_h||_{\Omega,h} \leq Ch^{N+1}$$

Example - High-order makes the difference

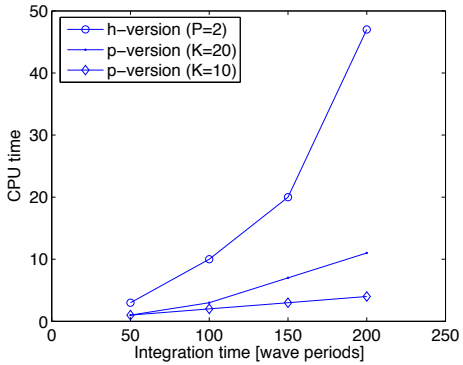


Figure: Optimized CPU-time vs. integration time for a fixed relative error in amplitude of 5% (assume satisfactory engineering accuracy).

- Conclusion: a significant improvement in performance can be achieved using high-order elements over long times of integration.

Numerical solution of PDEs

Important reasons for the interest in DG-FEM methods are

- Need for numerical methods of high accuracy in space and time
- Support for locally adaptive numerical solutions
  hp-adaptivity, meshes can be both non-conforming and unstructured.
- General and very flexible framework for solving large classes of PDEs
- Conceptually no difference between 1-D, 2-D or N-D
- The method is local (to the elements) which makes it highly suitable for parallel computations

A brief history

- DG-FEM was first proposed by Reed & Hill in 1973 for a neutron transport equation

$$\sigma u + \nabla \cdot (au) = f$$

- First analysis by Lesaint & Raviart (1974) showing in general $\mathcal{O}(h^N)$ and optimal $\mathcal{O}(h^{N+1})$ for special meshes.
- Sharp analysis by Johnson (1986) showed $\mathcal{O}(h^{N+1/2})$ for general meshes
- However, the schemes did not enjoy much use until further developments...

A brief history

- Extension from scalar conservation laws to systems
  1980s-late 1990s, Cockburn/Shu
- Development of limiters and RKDG for problems with discontinuities
  Late 1980s, Shu/Cockburn
- Nodes, modes and large codes
  from 1995, Warburton/Karniadakis
- Maxwell's eqations, MHD, water waves, elasticity, etc.
  - last decade has seen an explosion in development and applications
- Higher order problems
  - Interior-Penalty (IP), Arnold (1982)
  - Bassi-Rebay (BR), Bassi & Rebay (1997)
  - Local Discontinuous Galerkin (LDG), Cockburn & Shu (1998)

A brief history

The last decade has seen an explosion in activities

- Hamilton-Jacobi equations
- Non-coercive problems and spectral accuracy
- Adaptive solution techniques
- Improved solvers
- Advanced time-integration methods
- Large-scale production codes
- etc.

Brief summary

We have established a basic understanding of DG-FEM

- How to formulate DG-FEM schemes
- Local expansions to achieve a high-order accurate basis
- Geometric flexibility in the spirit of FEM/FVM
- Explicit scheme and 'problem control' in the spirit of FVM

However, many questions remains

- How do we choose the numerical flux?
- Is the scheme stable?
- How does the idea generalize to multi-dimensions?
- What is the price?
- etc...