DG-FEM for PDE’s
Lecture 1

Jan S Hesthaven
Brown University
Jan.Hesthaven@Brown.edu
Start by saying thank you

- **The main organizers**
  - Dr Miguel Mendez
  - Dr Jeronimo Rodriguez Garcia

- **Funding agencies to help enable this**
  - NSF, AFOSR, Sloan, Minister of Science (ES)

- **Collaborators over time**
  - T. Warburton, D. Gottlieb, J. Rodriguez ..

You for taking the time!
A brief overview of what’s to come

- Lecture 1: Introduction and DG in 1D
- Lecture 2: Implementation and numerical aspects
- Lecture 3: Insight through theory
- Lecture 4: Nonlinear problems
- Lecture 5: Mesh generation and 2D DG-FEM
- Lecture 6: Implementation and applications
- Lecture 7: Higher order/Global problems
- Lecture 8: 3D and advanced topics
A little self promotion

Lectures and exercises are based on text

All software and codes are available at www.nudg.org
Moore’s law and the performance gap

Raw performance largely driven by innovation in hardware has so far followed Moore’s law: ‘Bang-per-buck’ doubles every 24 months

Current predictions in required performance needs show a large and growing gap, known as the performance gap.
What is causing this growing gap?

Our way of doing science is being transformed as simulation science matures, enabling entirely new ways of addressing problems of national interest.

Solving ‘real’ problems introduce new challenges:

- Modeling of very large scale complex systems, incl unsteady, multi-physics, multi-scale problems
- Treatment of uncertainties and stochastic effects
- Interaction/assimilation with experimental data
- Very large scale data manipulation and visualization ‘from data to knowledge’
- Very large scale optimization, estimation, design
Can we close the gap?

Which algorithms are needed for the next generation of scientific problems? They should address the statistical challenges of massive dimensions, and the heterogeneous multiscale nature of the science driver problems.

Infrastructure Issues

Databases of software should be made available to practitioners. Automatic assistance needs to be provided to help locate the most appropriate algorithm for a given problem and data. Problems such as how do you communicate the meta-assumptions of your data need to be resolved. We need to be able to assure quality whenever possible, but not set standards prematurely or too strictly as to inhibit development of new tools and technologies.

RECOMMENDATIONS

Software Infrastructure Support

To develop, manage, leverage and maintain the software infrastructure needed for the next generation of science, sustained infrastructure and facility support will be required. Science needs to be the driver for software investments. At the same time, software and algorithms which have proved successful at a research-group level for scientific applications should be extended and made available to the wider scientific community. The latter is not only by developing more advanced algorithms and by paying careful attention to the dirty details. ‘Parallel scaling is not all’.

.. only by developing more advanced algorithms

.. and by paying careful attention to the dirty details

‘Parallel scaling is not all’
Can we close the gap?

It is the algorithms, their analysis, and implementation that are the hard components of this quest toward real simulations:

- Complex to develop and implement
- Very labor intensive (manpower, hero program.)
- Long lived (20-30 years vs 3-5 for hardware)
- Costly!

Question:
Can we hope to mimic the success of standard packages for linear algebra and ODE’s on PDE’s?
What do we need?

For the application of numerical methods we want

- Accuracy at minimal effort
- Flexibility to solve different types of problems
- ‘Easy and quick’ prototyping
- Solid theoretical backing
- Good performance on advanced computers

But why something new?
Conservation laws

Conservation principles are what we (often) use to derived physical model so our world

General **differential** form

\[
\partial_t u + \nabla \cdot F(u) = S(u)
\]

or

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

where

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

Conserved variables \hspace{2cm} Flux \hspace{2cm} Source
Conservation laws

- **Euler equations of compressible gas dynamics (1D)**

\[
\begin{align*}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} &= 0 \quad \text{(Mass)} \\
\frac{\partial \rho u}{\partial t} + \frac{\partial (\rho u^2 + p)}{\partial x} &= 0 \quad \text{(Momentum)} \\
\frac{\partial E}{\partial t} + \frac{\partial (E + p) u}{\partial x} &= 0 \quad \text{(Energy)}
\end{align*}
\]

\[ p = (\gamma - 1) \left( E - \frac{1}{2} \rho u^2 \right), \quad c = \sqrt{\frac{\gamma p}{\rho}} \quad \text{(Ideal gas low)} \]

- **Nonlinear shallow water equations (1D)**

\[
\begin{align*}
\frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} &= 0 \quad \text{(Mass)} \\
\frac{\partial hu}{\partial t} + \frac{\partial (hu^2 + \frac{1}{2} gh^2)}{\partial x} &= 0 \quad \text{(Momentum)}
\end{align*}
\]

- and many many more...
How do we solve this?

Let us consider a few well known schemes and their basic properties to understand what is needed.

Consider the basic equation

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

All schemes involve two choices

- In which way does one approximate the solution?
- In which way should the approximation satisfy the PDE?
Finite difference methods

The local approximation is a 1D polynomial
The equation is satisfied in a pointwise manner

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

\[ \frac{du_h(x^k, t)}{dt} + \frac{f_h(x^{k+1}, t) - f_h(x^{k-1}, t)}{h^k + h^{k-1}} = g(x^k, t), \]
Finite difference schemes

Main benefits
- Simple to implement and fast
- High-order is feasible
- Local and explicit in time
- Direction can be exploited - upwind
- Extensive body of theory

Main problem
- Simple local approximation and geometric flexibility are not agreeable
- Grid smoothness requirements
Finite volume methods

The local approximation is a cell average

\[
\int_{x^{k-1/2}}^{x^{k+1/2}} u_h(x) \, dx = h^k \bar{u}^k,
\]

The equation is satisfied on conservation form

\[
h^k \frac{d\bar{u}^k}{dt} + f^{k+1/2} - f^{k-1/2} = h^k \bar{g}^k,
\]
Finite volume methods

The key challenge is one of reconstruction

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

\[ f(x^{k-1/2}, t) = F(\bar{u}^{k-1}, \bar{u}^k) \]

\[ f(x^{k+1/2}, t) = F(\bar{u}^k, \bar{u}^{k+1}) \]

- **Main benefit**
  - Robust and fast due to locality
  - Complex geometries
  - Well suited for conservation laws
  - Local and explicit in time
  - Solid theoretical framework

- **Main problem**
  - Inability to achieve high-order on general grids due to extended stencils
  - Grid smoothness requirements
Finite element methods

We begin by splitting the solution into elements as

- The solution is defined in a nonlocal manner
  \[ u_h(x) = \sum_{k=1}^{K} u(x_k, t) N^k(x), \quad N^i(x_j) = \delta_{ij} \]
- The equation is satisfied globally
  \[ \int_{\Omega} \left( \frac{\partial u_h}{\partial t} + \frac{\partial f_h}{\partial x} - g_h \right) N^j(x) \, dx = 0, \]
Finite element methods

This yields the global equation

$$\mathcal{M} \frac{d\mathbf{u}_h}{dt} + \mathbf{S} f_h = \mathcal{M} \mathbf{g}_h,$$

$$\mathcal{M}_{ij} = \int_{\Omega} N^i(x)N^j(x) \, dx, \quad S_{ij} = \int_{\Omega} N^i(x)\frac{dN^j}{dx} \, dx,$$

- **Main benefits**
  - High-order accuracy and complex geometries can be combined
  - Extensive theoretical setting

- **Main problems**
  - Implicit in time
  - Not well suited for problems with direction
Let's summarize the observations

<table>
<thead>
<tr>
<th></th>
<th>Complex geometries</th>
<th>High-order accuracy and (hp)-adaptivity</th>
<th>Explicit semi-discrete form</th>
<th>Conservation laws</th>
<th>Elliptic problems</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDM</td>
<td>(\times)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>FVM</td>
<td>(\checkmark)</td>
<td>(\times)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>((\checkmark))</td>
</tr>
<tr>
<td>FEM</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>(\times)</td>
<td>((\checkmark))</td>
<td>(\checkmark)</td>
</tr>
<tr>
<td>DG-FEM</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>(\checkmark)</td>
<td>((\checkmark))</td>
</tr>
</tbody>
</table>

What we need is a scheme that combines
- The local high-order/flexible element of FEM
- The local statement on the equation for FVM

These are exactly the components of the **Discontinuous Galerkin Finite Element Method**
So let's see how we can achieve this

It is the multi-element component of FEM/FVM which gives the geometric flexibility

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

and the solution is represented as

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

$$x \in D^k : u_h^k(x, t) = \sum_{n=1}^{N_p} \hat{u}_n^k(t) \psi_n(x) = \sum_{i=1}^{N_p} u_h^k(x_i^k, t) \ell_i^k(x).$$

with a high-order local basis as in FEM:

- on modal form
- on nodal form
The first DG schemes

So let us consider the scalar problem

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

We form the local residual

\[ x \in D^k : \mathcal{R}_h(x, t) = \frac{\partial u_h^k}{\partial t} + \frac{\partial f_h^k}{\partial x}. \]

and require this to vanish \textit{locally} in a Galerkin sense

\[ \int_{D^k} \mathcal{R}_h(x, t) \ell_j^k(x) \, dx = 0, \]

The \textbf{problem} is that all elements are now disconnected due to the \textbf{local} statement on the residual!
The first DG schemes

Let us apply Gauss’s theorem

\[
\int_{D^k} \frac{\partial u_h^k}{\partial t} \ell_j^k - f_h^k \frac{d \ell_j^k}{dx} - g \ell_j^k \, dx = - [f_h^k \ell_j^k]_{x^k}^{x^{k+1}}.
\]

We have multiple solutions and we need to pick one!
The first DG schemes

The lack of solution uniqueness at the interface is addressed as in FVM by a numerical flux

\[ f^* = f^*(u_h^-, u_h^+) , \]

\[ \int_{D^k} \frac{\partial u_h^k}{\partial t} \ell_j^k - f_h^k \frac{d\ell_j^k}{dx} - g\ell_j^k \, dx = - \left[ f_h^k \ell_j^k \right] x^{k+1} . \]

with the corresponding strong form being

\[ \int_{D^k} \mathcal{R}_h(x, t) \ell_j^k(x) \, dx = \left[ (f_h^k - f^*) \ell_j^k \right] x^{k+1} . \]

Naturally, the choice of the flux is important!
The basics of DG-FEM

To simplify the notation, introduce

\[
\mathcal{M}_{ij}^k = \int_{D^k} \ell_i^k(x)\ell_j^k(x)\,dx, \quad S_{ij}^k = \int_{D^k} \ell_i^k(x)\frac{d\ell_j^k}{dx}\,dx.
\]


to obtain the two basic forms of DG-FEM

**Weak:**

\[
\mathcal{M}^k \frac{du_h^k}{dt} - (S^k)^T f_h^k - \mathcal{M}^k g_h^k = -f^*(x^{k+1})\ell^k(x^{k+1}) + f^*(x^k)\ell^k(x^k)
\]

**Strong:**

\[
\mathcal{M}^k \frac{du_h^k}{dt} + S^k f_h^k - \mathcal{M}^k g_h^k = (f_h^k(x^{k+1}) - f^*(x^{k+1}))\ell^k(x^{k+1}) - (f_h^k(x^k) - f^*(x^k))\ell^k(x^k).
\]
A first example

Let us consider a simple example

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

<table>
<thead>
<tr>
<th>N\ K</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>Convergence rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>–</td>
<td>4.0E-01</td>
<td>9.1E-02</td>
<td>2.3E-02</td>
<td>5.7E-03</td>
<td>1.4E-03</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>2.0E-01</td>
<td>4.3E-02</td>
<td>6.3E-03</td>
<td>8.0E-04</td>
<td>1.0E-04</td>
<td>1.3E-05</td>
<td>3.0</td>
</tr>
<tr>
<td>4</td>
<td>3.3E-03</td>
<td>3.1E-04</td>
<td>9.9E-06</td>
<td>3.2E-07</td>
<td>1.0E-08</td>
<td>3.3E-10</td>
<td>5.0</td>
</tr>
<tr>
<td>8</td>
<td>2.1E-07</td>
<td>2.5E-09</td>
<td>4.8E-12</td>
<td>2.2E-13</td>
<td>5.0E-13</td>
<td>6.6E-13</td>
<td>≈ 9.0</td>
</tr>
</tbody>
</table>

The error clearly behaves as

\[ \| u - u_h \|_{\Omega, h} \leq C h^{N+1}. \]
Shallow water equations

Consider 1D shallow water equations in 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

\[
\left\| u - u_h \right\|_{\Omega,h} \leq Ch^{N+1}
\]
Any advantage to high-order?

**Observation**: Significant potential for savings without impacting accuracy by using high-order elements over long times of integration.

**Test**: Time to compute solution at 5% error

![Graph showing CPU time vs. integration time for different versions.](image)

**Observation**: Significant potential for savings without impacting accuracy by using high-order elements over long times of integration.
A brief history

- DG-FEM was first proposed by Reed/Hill in 1973

\[ \sigma u + \nabla \cdot (au) = f, \]

- First analysis (Lesaint, Raviart, 1974) showing in \( O(h^N) \) general and optimal \( O(h^{N+1}) \) for special grids.

- Sharp analysis (Johnson 1986) showed \( O(h^{N+1/2}) \)

- However the schemes did not enjoy much use
A brief history

- Extension from scalar conservation laws in late 1980’s to system in late 1990’s (Cockburn/Shu)
- Development of limiters and RKDG
- Nodes, modes and large codes (H, Warburton, Karniadakis etc etc - from 1995)
- Maxwell’s equations, MHD, water waves, elasticity etc -- last decade has seen explosion
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, etc
A brief overview - and what now?

- Local flexibility to achieve high-order and geometric flexibility in the spirit of FEM
- Explicit scheme and ‘problem control’ in the spirit of FVM

... but many answers remain unanswered

- How do we achieve high-order accuracy?
- How do we choose the numerical flux?
- Is the scheme stable?
- What is the price?
A bit of notation

It is the multi-element component of FEM/FVM which gives the geometric flexibility

\[ \Omega \simeq \Omega_h = \bigcup_{k=1}^{K} D^k, \]

and the solution is approximated as

\[ u(x, t) \simeq u_h(x, t) = \bigoplus_{k=1}^{K} u_h^k(x, t), \]

Local norms

\[ (u, v)_{D^k} = \int_{D^k} uv \, dx, \quad \|u\|_{D^k}^2 = (u, u)_{D^k}, \]
A bit of notation

Global norms

\[(u, v)_{\Omega, h} = \sum_{k=1}^{K} (u, v)_{D^k}, \quad \|u\|_{\Omega, h}^2 = (u, u)_{\Omega, h}.\]

To deal with the solutions across an interface, we define

The jump:

\[[u] = \hat{n}^- u^- + \hat{n}^+ u^+,\]

The average:

\[
\{u\} = \frac{u^- + u^+}{2},
\]

\[u^- = \text{local/interior solution}\]

\[u^+ = \text{neighbor/exterior solution}\]

outward pointing normal
The basics of DG-FEM

To understand the role of the different choices, let us consider an example

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

First recall

\[
\frac{d}{dt} \|u_h\|_{\Omega,h}^2 = -a (u^2(R) - u^2(L)),
\]

Leading to suitable boundary conditions as

\[
\begin{align*}
    u(L, t) &= g(t) & \text{if} & \quad a \geq 0, \\
    u(R, t) &= g(t) & \text{if} & \quad a \leq 0.
\end{align*}
\]

... and energy conservation when

\[
u(R) = u(L)\]
Let us now assume that the local solution is

\[ x \in D^k : \; u^k_h(x, t) = \sum_{n=1}^{N_p} \hat{u}_n^k(t) \psi_n(x) = \sum_{i=1}^{N_p} u^k_h(x^k_i, t) \ell^k_i(x). \]

Where we have

- Some local modal basis \( \psi_n(x) \)
- A local nodal basis \( \ell^k_i(x) \)

We form the local residual

\[ x \in D^k : \; \mathcal{R}_h(x, t) = \frac{\partial u^k_h}{\partial t} + \frac{\partial f^k_h}{\partial x}. \]

and require this to vanish \textit{locally} in a Galerkin sense

\[ \int_{D^k} \mathcal{R}_h(x, t) \ell^k_j(x) \, dx = 0, \]
The basics of DG-FEM

The lack of solution uniqueness at the interface is addressed as in FVM by a numerical flux

\[ f^* = f^*(u_h^-, u_h^+), \]

\[ \int_{D^k} \frac{\partial u_h^k}{\partial t} \ell_j^k - f_h^k \frac{d\ell_j^k}{dx} - g\ell_j^k \, dx = - \left[ f_h^k \ell_j^k \right] x^{k+1}. \]

with the corresponding strong form being

\[ \int_{D^k} R_h(x, t) \ell_j^k(x) \, dx = \left[ (f_h^k - f^*) \ell_j^k \right] x^{k+1}. \]
The basics of DG-FEM

To simplify the notation, introduce

\[ \mathcal{M}^k_{ij} = \int_{D^k} \ell^k_i(x) \ell^k_j(x) \, dx, \quad S^k_{ij} = \int_{D^k} \ell^k_i(x) \frac{d\ell^k_j}{dx} \, dx. \]

to obtain the two basic forms of DG-FEM

**Weak:**

\[ \mathcal{M}^k \frac{du^k_h}{dt} - (S^k)^T f^k_h - \mathcal{M}^k g^k_h = -f^*(x^{k+1}) \ell^k(x^{k+1}) + f^*(x^k) \ell^k(x^k) \]

**Strong:**

\[ \mathcal{M}^k \frac{du^k_h}{dt} + S^k f^k_h - \mathcal{M}^k g^k_h = (f^k_h(x^{k+1}) - f^*(x^{k+1})) \ell^k(x^{k+1}) \]
\[ - (f^k_h(x^k) - f^*(x^k)) \ell^k(x^k). \]
The basics of DG-FEM

Let us consider the strong form

\[ \mathcal{M}^k \frac{d}{dt} u_h^k + \mathcal{S}^k (au_h^k) = \left[ \ell^k(x)(au_h^k - (au_h)^*) \right]^{x_r^k}_{x_i^k}. \]

First note that

\[ u_h^T M^k u_h = \int_{D^k} \sum_{i=1}^{N_p} u_h^k(x_i^k) \ell_i^k(x) \sum_{j=1}^{N_p} u_h^k(x_j^k) \ell_j^k(x) \, dx = \|u_h^k\|^2_{D^k}: \]

Then also note that

\[ u_h^T S^k u_h = \int_{D^k} \sum_{i=1}^{N_p} u_h^k(x_i^k) \ell_i^k(x) \sum_{j=1}^{N_p} u_h^k(x_j^k) \frac{d\ell_j^k}{dx} \, dx = \int_{D^k} u_h^k(x)(u_h^k(x))' \, dx \]

\[ = \frac{1}{2} [(u_h^k)^2]^{x_r^k}_{x_i^k}. \]
The basics of DG-FEM

Combining this, we achieve from

\[ \mathcal{M}_h^k \frac{d}{dt} u_h^k + S_h^k (au_h^k) = \left[ \ell_h^k (x) (au_h^k - (au)^*) \right]_{x_r^k}. \]

the local energy estimate

\[ \frac{d}{dt} \| u_h^k \|_{D^k}^2 = -a[(u_h^k)^2]_{x_r^k, x_l^k} + 2 [u_h^k (au_h^k - (au)^*)]_{x_r^k, x_l^k}. \]

which should behave in such a way that

\[ \sum_{k=1}^{K} \frac{d}{dt} \| u_h^k \|_{D^k}^2 = \frac{d}{dt} \| u_h \|_{\Omega, h}^2 \leq 0. \quad \text{Stability} \]
The basics of DG-FEM

Let us consider the contribution from one interface

\[
\hat{n}^-(au_h^2(x^-) - 2u_h(x^-)(au)^*(x^-)) + \hat{n}^+(au_h^2(x^+) - 2u_h(x^+)(au)^*(x^+)) \leq 0
\]

We have the freedom to choose numerical flux to guarantee this.

Consider

\[
f^* = (au)^* = \{\{au\}\} + |a| \frac{1 - \alpha}{2} [u].
\]

- \(\alpha = 0\) - upwind flux
- \(\alpha = \frac{1}{2}\) - central/average flux

This yields the local term from each interface

\[-|a|(1 - \alpha)^2[u_h]^2, \quad \leq 0, \text{ for } 0 \leq \alpha \leq 1\]
The basics of DG-FEM

What about the boundary conditions?

\[ a > 0 : u(L, t) = g(t) \]

**Approach #1:**

\[ f_L = a g(t), \quad f_R = a u_h^k(x^K_r) \]

**Approach #2:**

\[ f_L = -a u_h^k(x^L_l) + 2g(t), \quad f_R = a u_h^k(x^K_r) \]
The basics of DG-FEM

Approach #1:

\[
\frac{d}{dt} \|u_h\|_{\Omega,h}^2 = -|a|(1 - \alpha) \sum_{k=1}^{K-1} [u_h^k(x_r^k)]^2 - (1 - \alpha) a(u_h^1(x_l^1))^2 - a(u_h^K(x_r^K))^2.
\]

Approach #2:

\[
\frac{d}{dt} \|u_h\|_{\Omega,h}^2 = -|a|(1 - \alpha) \sum_{k=1}^{K-1} [u_h^k(x_r^k)]^2 - a(u_h^1(x_l^1))^2 - a(u_h^K(x_r^K))^2.
\]

\[a \geq 0, 0 \leq \alpha \leq 1\] \hspace{2cm} \textbf{Stability}

- Energy dissipation for \( \alpha \neq 1 \)
- Energy conservation for \( \alpha = 1, u(L) = u(R) \)
The basics of DG-FEM

What did we learn from this?

- Stability is enforced through the flux choice.
- No restrictions on the local basis, e.g., it need not be polynomial, and is chosen to provide accuracy.
- The numerical solution is discontinuous between elements.
- Boundary conditions and interface conditions are imposed *weakly*.
- All operators are local.
- Due to the weak interface based coupling, there are no restrictions on element size and local approximation.
Back to the example

Consider again the simple example

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

<table>
<thead>
<tr>
<th>N \ K</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>Convergence rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>–</td>
<td>4.0E-01</td>
<td>9.1E-02</td>
<td>2.3E-02</td>
<td><strong>5.7E-03</strong></td>
<td>1.4E-03</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>2.0E-01</td>
<td>4.3E-02</td>
<td><strong>6.3E-03</strong></td>
<td>8.0E-04</td>
<td>1.0E-04</td>
<td>1.3E-05</td>
<td>3.0</td>
</tr>
<tr>
<td>4</td>
<td><strong>3.3E-03</strong></td>
<td>3.1E-04</td>
<td>9.9E-06</td>
<td>3.2E-07</td>
<td>1.0E-08</td>
<td>3.3E-10</td>
<td>5.0</td>
</tr>
<tr>
<td>8</td>
<td>2.1E-07</td>
<td>2.5E-09</td>
<td>4.8E-12</td>
<td>2.2E-13</td>
<td>5.0E-13</td>
<td>6.6E-13</td>
<td>( \simeq 9.0 )</td>
</tr>
</tbody>
</table>

The error clearly behaves as

\[
\| u - u_h \|_{\Omega,h} \leq Ch^{N+1}.
\]
Back to the example

What about time dependence

<table>
<thead>
<tr>
<th>Final time (T)</th>
<th>$\pi$</th>
<th>$10\pi$</th>
<th>$100\pi$</th>
<th>$1000\pi$</th>
<th>$2000\pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{N,K}=(2,4)$</td>
<td>4.3E-02</td>
<td>7.8E-02</td>
<td>5.6E-01</td>
<td>&gt;1</td>
<td>&gt;1</td>
</tr>
<tr>
<td>$\text{N,K}=(4,2)$</td>
<td>3.3E-03</td>
<td>4.4E-03</td>
<td>2.8E-02</td>
<td>2.6E-01</td>
<td>4.8E-01</td>
</tr>
<tr>
<td>$\text{N,K}=(4,4)$</td>
<td>3.1E-04</td>
<td>3.3E-04</td>
<td>3.4E-04</td>
<td>7.7E-04</td>
<td>1.4E-03</td>
</tr>
</tbody>
</table>

The error behaves as

$$\|u - u_h\|_{\Omega,h} \leq C(T)h^{N+1} \approx (c_1 + c_2T)h^{N+1},$$
Back to the example

Central flux

Upwind flux
Back to the example

What about cost?

<table>
<thead>
<tr>
<th>N \ K</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00</td>
<td>2.19</td>
<td>3.50</td>
<td>8.13</td>
<td>19.6</td>
<td>54.3</td>
</tr>
<tr>
<td>2</td>
<td>2.00</td>
<td>3.75</td>
<td>7.31</td>
<td>15.3</td>
<td>38.4</td>
<td>110.</td>
</tr>
<tr>
<td>4</td>
<td>4.88</td>
<td>8.94</td>
<td>20.0</td>
<td>45.0</td>
<td>115.</td>
<td>327.</td>
</tr>
<tr>
<td>8</td>
<td>15.1</td>
<td>32.0</td>
<td>68.3</td>
<td>163.</td>
<td>665.</td>
<td>1271.</td>
</tr>
<tr>
<td>16</td>
<td>57.8</td>
<td>121.</td>
<td>279.</td>
<td>664.</td>
<td>1958.</td>
<td>5256.</td>
</tr>
</tbody>
</table>

Time $\approx C(T)K(N + 1)^2$,

Higher order is cheaper
A few remarks

We know now

- It is the flux that gives stability
- It is the local basis that gives accuracy.

The scheme is very (VERY) flexible.

BUT - we have doubled the number of degrees of freedom along the interfaces

In 1D not a big deal -- penalty is \( \frac{N+1}{N} \)
Does it generalize beyond linear 1D?

Let us first consider the scalar conservation law

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

where the boundary conditions are

\begin{align*}
u(L, t) &= g_1(t) \quad \text{when} \quad f_u(u(L, t)) \geq 0, \\
u(R, t) &= g_2(t) \quad \text{when} \quad f_u(u(R, t)) \leq 0.
\end{align*}

Assume as usual that

$$x \in D^k : u_h^k(x, t) = \sum_{n=1}^{N_p} \hat{u}_n^k(t) \psi_n(x) = \sum_{i=1}^{N_p} u_h^k(x_i^k, t) \ell_i^k(x).$$
Does it generalize beyond linear 1D?

From this we directly recover

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

and the corresponding strong form

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

Here we have the general local test functions

\[
\phi_h^k \in V_h^k, \quad x \in D^k : \quad \phi_h^k(x) = \sum_{n=1}^{N_p} \hat{\phi}_n^k \psi_n(x),
\]
Does it generalize beyond linear 1D?

This yields exactly $N_p$ equations for the local $N_p$ unknowns

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

and the corresponding strong form

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

Here we have also introduced

$$x \in D^k : f_h^k(u_h^k) = \sum_{n=1}^{N_p} \hat{f}_n \psi_n(x) = \sum_{i=1}^{N_p} f_h(x_i^k) \ell_i^k(x).$$

**Note:** Some delicate details here
Does it generalize beyond linear 1D?

The only thing that remains unknown is the flux

\[ f^* = f^*(u_h^-, u_h^+) \]

We rely on the hugely successful theory of finite volume monotone schemes

- The flux is consistent \( f(u_h) = f^*(u_h, u_h) \)
- The flux is monotone

\[ f^*(a, b) \begin{cases} \text{increasing in } a \\ \text{decreasing in } b \end{cases} \]
Does it generalize beyond linear 1D?

There are many choices for this

- The Lax-Friedrichs flux

\[ f^{LF}(a, b) = \frac{f(a) + f(b)}{2} + \frac{C}{2} \hat{n} \cdot (a - b), \]

- where the global LF flux is given by

\[ C \geq \max_{\inf u_h(x) \leq s \leq \sup u_h(x)} \left| f_u(s) \right| \]

- and the local LF flux is obtained for

\[ C \geq \max_{\min(a,b) \leq s \leq \max(a,b)} \left| f_u(s) \right|. \]

**NOTE:** for \( f = au \) we recover the flux seen earlier.
Does it generalize beyond linear 1D?

... but the FV literature is filled with alternatives

- Exact Riemann solvers
- Godunov fluxes
- Engquist-Osher fluxes
- Approximate Riemann fluxes (Roe, van Leer, HLLC etc)

To keep things simple we shall mainly focus on the LF flux which generally works very well, but is also the most dissipative flux.
Does it generalize beyond linear 1D?

Let us now consider the system of conservation laws

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

\[
u = [u_1(x, t), \ldots, u_m(x, t)]^T
\]

where the boundary conditions are

\[
B_L u(L, t) = g_1(t) \quad \text{at } x = L,
\]

\[
B_R u(R, t) = g_2(t) \quad \text{at } x = R,
\]

The only essential difference is that \( C \) in the LF flux depends on the eigenvalues of

\[
f_u = \frac{\partial f}{\partial u}.
\]
Let’s summarize

We already know a lot about the basic DG-FEM

- **Stability** is provided by carefully choosing the numerical flux.
- **Accuracy** appear to be given by the local solution representation.
- We can utilize major advances on **monotone schemes** to design fluxes.
- The scheme generalizes with very few changes to very general **systems of conservation laws**.

At least in principle -- but how we do it in practice?