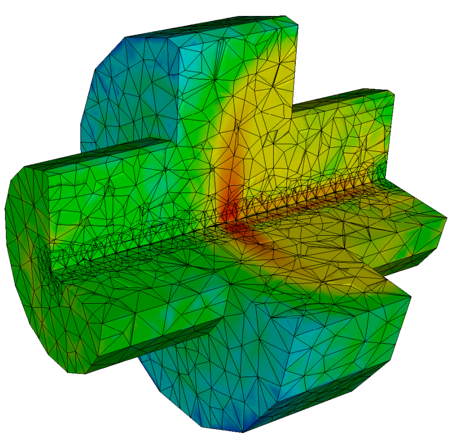
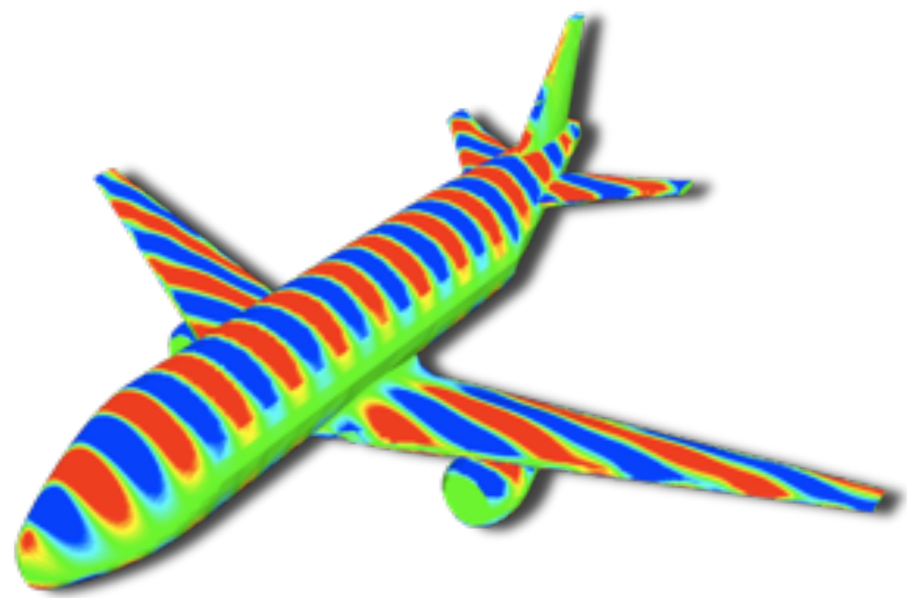
DG-FEM for PDE's
Lecture 7

Jan S Hesthaven
Brown University
Jan.Hesthaven@Brown.edu





A brief overview of what's to come

- Lecture 1: Introduction and DG-FEM in 1D
- Lecture 2: Implementation and numerical aspects
- Lecture 3: Insight through theory
- Lecture 4: Nonlinear problems
- Lecture 5: Extension to two spatial dimensions
- Lecture 6: Introduction to mesh generation
- Lecture 7: Higher order/Global problems
- Lecture 8: 3D and advanced topics

Lecture 7

- ✓ Let's briefly recall what we know
- ✓ Brief overview of multi-D analysis
- ✓ Part I: Time-dependent problems
  - ✓ Heat equations
  - ✓ Extensions to higher order problems
- ✓ Part II: Elliptic problems
  - ✓ Different formulations
  - ✓ Stabilization
  - ✓ Solvers and application examples

Lets summarize

We have a thorough understanding of 1st order problems

- ✓ For the linear problem, the error analysis and convergence theory is essentially complete.
- ✓ The theoretical support for DG for conservation laws is very solid.
- ✓ Limiting is perhaps the most pressing open problem
- ✓ The extension to 2D is fairly straightforward
- ✓ .... and we have a nice and flexible way to implement it all

Time to move beyond the 1st order problem

Brief overview of multi-D analysis

In 1D we discussed that

$$\|u - u_h\|_{\Omega,h} \leq Ch^{N+1}\|u\|_{\Omega,N+2,h},$$

.. but this was a somewhat special case.

Question is -- is it possible in multi-D ?

Answer - No

$$\|u - u_h\|_{\Omega,h} \leq Ch^{N+1/2}\|u\|_{\Omega,N+1,h},$$

... but the optimal rate is often observed as initial error dominates over the accumulated error

The heat equation

Let us consider the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, 2\pi], \qquad\qquad u(x,t) = e^{-t}\sin(x).$$

We can be tempted to write this as

$$\frac{\partial u}{\partial t} - \frac{\partial}{\partial x} u_x = 0,$$

and then just use our standard approach

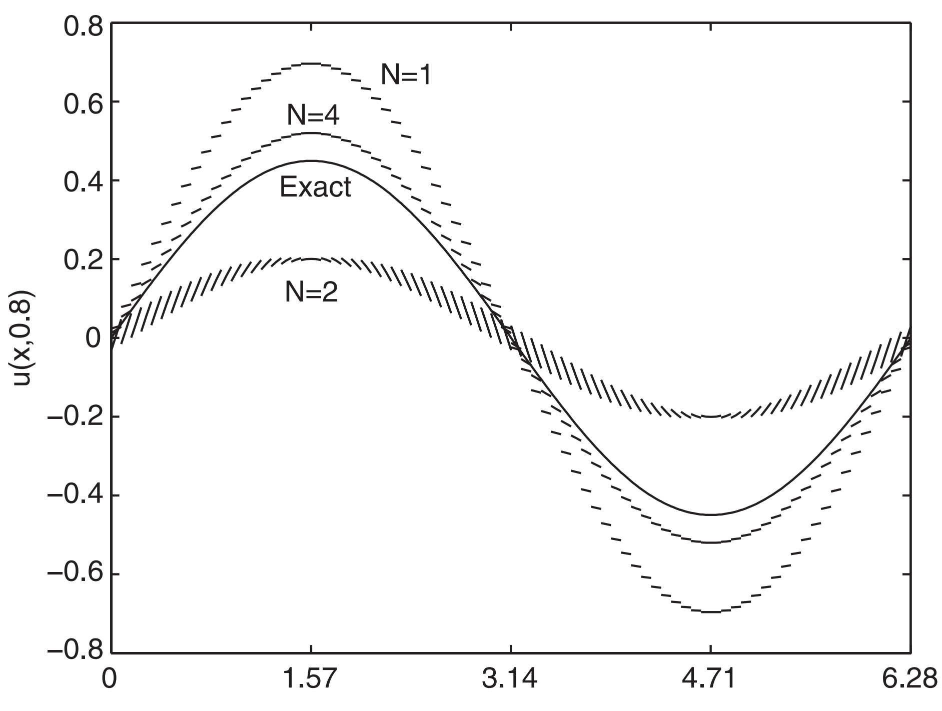
$$\boldsymbol{v}_h^k = \mathcal{D}_r \boldsymbol{u}_h^k, \quad \mathcal{M}^k \frac{d\boldsymbol{u}_h^k}{dt} - \mathcal{S}\boldsymbol{v}_h^k = -\int_{\partial \mathsf{D}^k} \hat{\boldsymbol{n}} \cdot \left(v_h^k - v^*\right) \boldsymbol{\ell}^k(x)\, dx,$$

Given the nature of the problem, a central flux seems reasonable $\qquad v^* = \{\{v_h\}\}$

The heat equation

Lets see what happens when we run it

$N\backslash K$	10	20	40	80	160
1	4.27E-1	4.34E-1	4.37E-1	4.38E-1	4.39E-1
2	5.00E-1	4.58E-1	4.46E-1	4.43E-1	4.42E-1
4	1.68E-1	1.37E-1	1.28E-1	1.26E-1	–
8	7.46E-3	8.60E-3	–	–	–



It does not work!

It is weakly unstable

The heat equation

We need a new idea -- consider

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} a(x) \frac{\partial u}{\partial x},$$

We know that DG is good for 1st order systems.

Since a(x)>0 we can write this as

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \sqrt{a} q, \quad q = \sqrt{a} \frac{\partial u}{\partial x},$$

Now follow our standard approach

$$\begin{bmatrix} u(x,t) \\ q(x,t) \end{bmatrix} \simeq \begin{bmatrix} u_h(x,t) \\ q_h(x,t) \end{bmatrix} = \bigoplus_{k=1}^{K} \begin{bmatrix} u_h^k(x,t) \\ q_h^k(x,t) \end{bmatrix} = \bigoplus_{k=1}^{K} \sum_{i=1}^{N_p} \begin{bmatrix} u_h^k(x_i,t) \\ q_h^k(x_i,t) \end{bmatrix} \ell_i^k(x),$$

The heat equation

Treating this as a 1st order system we have

$$\mathcal{M}^k \frac{d\boldsymbol{u}_h^k}{dt} = \tilde{\mathcal{S}}^{\sqrt{a}} \boldsymbol{q}_h^k - \int_{\partial \mathsf{D}^k} \hat{\boldsymbol{n}} \cdot \left((\sqrt{a} q_h^k) - (\sqrt{a} q_h^k)^* \right) \boldsymbol{\ell}^k(x)\, dx,$$

$$\mathcal{M}^k \boldsymbol{q}_h^k = \mathcal{S}^{\sqrt{a}} \boldsymbol{u}_h^k - \int_{\partial \mathsf{D}^k} \hat{\boldsymbol{n}} \cdot \left(\sqrt{a} u_h^k - (\sqrt{a} u_h^k)^* \right) \boldsymbol{\ell}^k(x)\, dx,$$

or the corresponding weak form

$$\mathcal{M}^k \frac{d\boldsymbol{u}_h^k}{dt} = -(\mathcal{S}^{\sqrt{a}})^T \boldsymbol{q}_h^k + \int_{\partial \mathsf{D}^k} \hat{\boldsymbol{n}} \cdot (\sqrt{a} q_h^k)^* \boldsymbol{\ell}^k(x)\, dx$$

$$\mathcal{M}^k \boldsymbol{q}_h^k = -(\tilde{\mathcal{S}}^{\sqrt{a}})^T \boldsymbol{u}_h^k + \int_{\partial \mathsf{D}^k} \hat{\boldsymbol{n}} \cdot (\sqrt{a} u_h^k)^* \boldsymbol{\ell}(x)\, dx.$$

Here

$$\tilde{\mathcal{S}}_{ij}^{\sqrt{a}} = \int_{\mathsf{D}^k} \ell_i^k(x) \frac{d\sqrt{a(x)}\ell_j^k(x)}{dx}\, dx, \quad \mathcal{S}_{ij}^{\sqrt{a}} = \int_{\mathsf{D}^k} \sqrt{a(x)} \ell_i^k(x) \frac{d\ell_j^k(x)}{dx}\, dx.$$

The heat equation

How do we choose the fluxes?

$$(\sqrt{a}q_h)^* = f((\sqrt{a}q_h)^-, (\sqrt{a}q_h)^+, (\sqrt{a}u_h)^-, (\sqrt{a}u_h)^+),$$
$$(\sqrt{a}u_h)^* = g((\sqrt{a}q_h)^-, (\sqrt{a}q_h)^+, (\sqrt{a}u_h)^-, (\sqrt{a}u_h)^+).$$

$$\mathcal{M}^k \frac{d\boldsymbol{u}_h^k}{dt} = \tilde{\mathcal{S}}^{\sqrt{a}} \boldsymbol{q}_h^k - \int_{\partial \mathsf{D}^k} \hat{\boldsymbol{n}} \cdot \left((\sqrt{a}q_h^k) - (\sqrt{a}q_h^k)^* \right) \boldsymbol{\ell}^k(x)\, dx,$$
$$\mathcal{M}^k \boldsymbol{q}_h^k = \mathcal{S}^{\sqrt{a}} \boldsymbol{u}_h^k - \int_{\partial \mathsf{D}^k} \hat{\boldsymbol{n}} \cdot \left(\sqrt{a}u_h^k - (\sqrt{a}u_h^k)^* \right) \boldsymbol{\ell}^k(x)\, dx,$$

Problem: Everything couples -- loss of locality

However, if we restrict it as

$$(\sqrt{a}q_h)^* = f((\sqrt{a}q_h)^-, (\sqrt{a}q_h)^+, (\sqrt{a}u_h)^-, (\sqrt{a}u_h)^+),$$
$$(\sqrt{a}u_h)^* = g((\sqrt{a}u_h)^-, (\sqrt{a}u_h)^+),$$

we can eliminate q-variable locally

The heat equation

Given the nature of the heat-equation, a natural flux could be central fluxes

$$(\sqrt{a}q_h)^* = \{\{\sqrt{a}q_h\}\}, \;\; (\sqrt{a}u_h)^* = \{\{\sqrt{a}u_h\}\}.$$

But is it stable ?

Computing the local energy in a single element yields

$$\frac{1}{2}\frac{d}{dt}\|u_h\|^2_{\mathsf{D}} + \|q_h\|^2_{\mathsf{D}} + \Theta_r - \Theta_l = 0,$$

$$\Theta = \sqrt{a}u_h q_h - (\sqrt{a}q_h)^* u_h - (\sqrt{a}u_h)^* q_h.$$

$$(\sqrt{a}q_h)^* = \sqrt{a}\{\{q_h\}\}, \;\; (\sqrt{a}u_h)^* = \sqrt{a}\{\{u_h\}\}. \quad \Rightarrow \quad \Theta_r = -\frac{\sqrt{a}}{2}\left(u_h^- q_h^+ + u_h^+ q_h^-\right).$$

$$\Rightarrow \quad \frac{1}{2}\frac{d}{dt}\|u_h\|^2_{\Omega,h} + \|q_h\|^2_{\Omega,h} = 0, \quad \Rightarrow \quad \text{Stability}$$

The heat equation

So this is stable!

How about boundary conditions

Dirichlet $$u_h^+ = -u_h^-, \;\; q_h^+ = q_h^- \quad \Rightarrow \quad \begin{cases} \{\{u_h\}\} = 0, & [\![u_h]\!] = 2\hat{\boldsymbol{n}}^- u_h^- \\ \{\{q_h\}\} = q_h^-, & [\![q_h]\!] = 0. \end{cases}$$

Neumann $$u_h^+ = u_h^-, \;\; q_h^+ = -q_h^- \quad \Rightarrow \quad \begin{cases} \{\{u_h\}\} = u_h^-, & [\![u_h]\!] = 0 \\ \{\{q_h\}\} = 0, & [\![q_h]\!] = 2\hat{\boldsymbol{n}}^- q_h^-. \end{cases}$$

Inhomogeneous BC

$$u_h^+ = -u_h^- + 2f(t), \;\; q_h^+ = q_h^-,$$

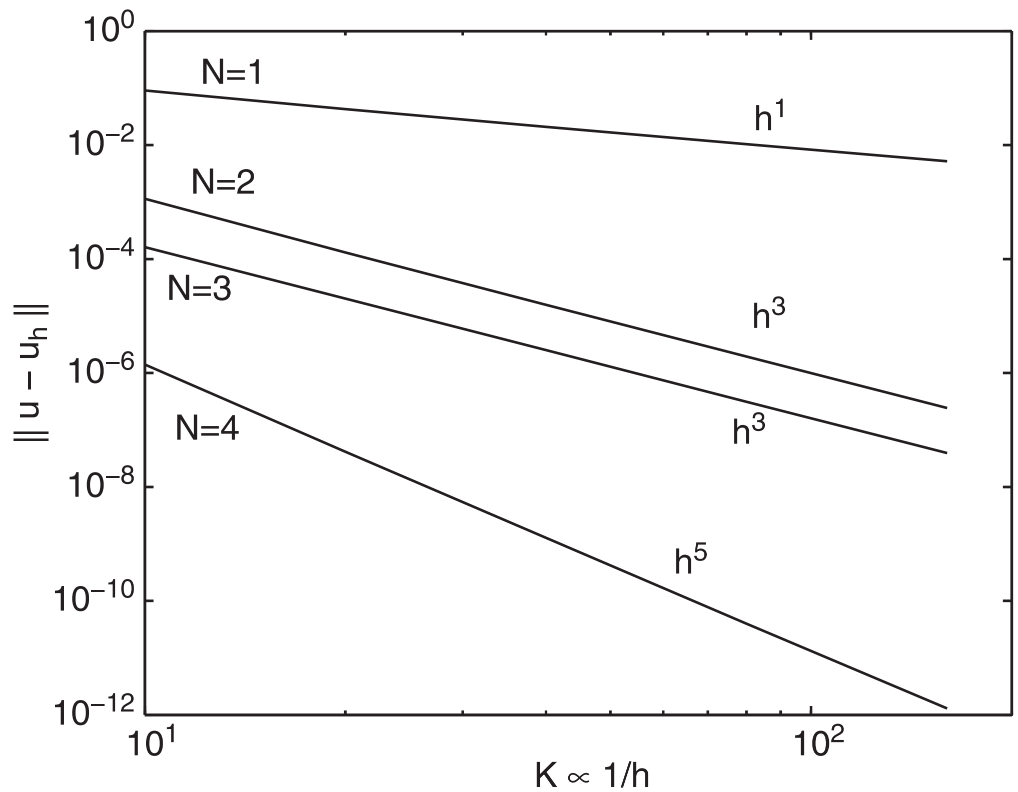
... and likewise for Neumann

The heat equation

Back to the example

Looks good -

.. but an even/odd pattern



Theorem 7.3. *Let $\varepsilon_u = u_h - u$ and $\varepsilon_q = q_h - q$ signify the pointwise errors for the heat equation with periodic boundaries and a constant coefficient $a(x)$, computed with Eq. (7.1) and central fluxes. Then*

$$\|\varepsilon_u(T)\|^2_{\Omega,h} + \int_0^T \|\varepsilon_q(s)\|^2_{\Omega,h}\, ds \le Ch^{2N},$$

where C depends on the regularity of u, T, and N. For N even, C is $\mathcal{O}(h^2)$.

The heat equation

Can we do anything to improve on this?

Recall the stability condition

$$\frac{1}{2}\frac{d}{dt}\|u_h\|_{\mathsf{D}}^2 + \|q_h\|_{\mathsf{D}}^2 + \Theta_r - \Theta_l = 0,$$

$$\Theta_r^- - \Theta_l^+ \geq 0$$

$$\Theta = \sqrt{a}u_h q_h - (\sqrt{a}q_h)^* u_h - (\sqrt{a}u_h)^* q_h.$$

Stable choices

$$(\sqrt{a}u_h)^* = \{\{\sqrt{a}\}\}u_h^+, \;\; (\sqrt{a}q_h)^* = \sqrt{a^-}q_h^-.$$

$$(\sqrt{a}u_h)^* = \sqrt{a^-}u_h^-, \;\; (\sqrt{a}q_h)^* = \{\{\sqrt{a}\}\}q_h^+,$$

$$\{\{\sqrt{a}u_h\}\} + \hat{\boldsymbol{\beta}} \cdot [\![\sqrt{a}u_h]\!], \;\; (\sqrt{a}q_h)^* = \{\{\sqrt{a}q_h\}\} - \hat{\boldsymbol{\beta}} \cdot [\![\sqrt{a}q_h]\!],$$

Upwind/downwind - LDG flux

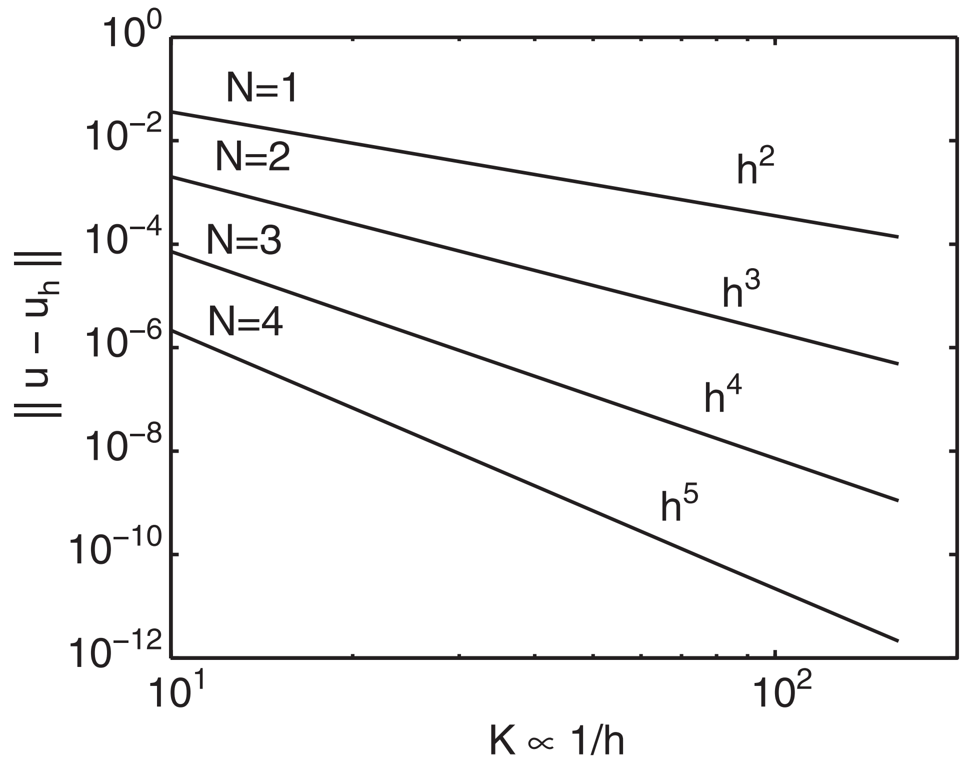
$$\hat{\beta} = \hat{n}$$

The heat equation

Back to the example

Looks good -

.. full order restored



Theorem 7.4. *Let $\varepsilon_u = u - u_h$ and $\varepsilon_q = q - q_h$ signify the pointwise errors for the heat equation with periodic boundaries and a constant coefficient $a(x)$, computed with Eq. (7.1) and LDG fluxes. Then*

$$\|\varepsilon_u(T)\|^2_{\Omega,h} + \int_0^T \|\varepsilon_q(s)\|^2_{\Omega,h}\, ds \leq Ch^{2N+2},$$

where C depends on the regularity of u, T, and N.

Higher order and mixed problems

We can now mix and match what we know

Consider

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} f(u) = \frac{\partial}{\partial x} a(x) \frac{\partial u}{\partial x},$$

and rewrite as

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(f(u) - \sqrt{a}q\right) = 0, \longrightarrow (f(u) - \sqrt{a}q)^*$$

$$q = \sqrt{a}\frac{\partial u}{\partial x}, \longrightarrow (\sqrt{a}u_h)^*$$

Now choose fluxes as we know how

$$f(u)^* = \{\{f(u)\}\} + \frac{C}{2}[\![u]\!], \quad C \geq \max|f'(u)|.$$

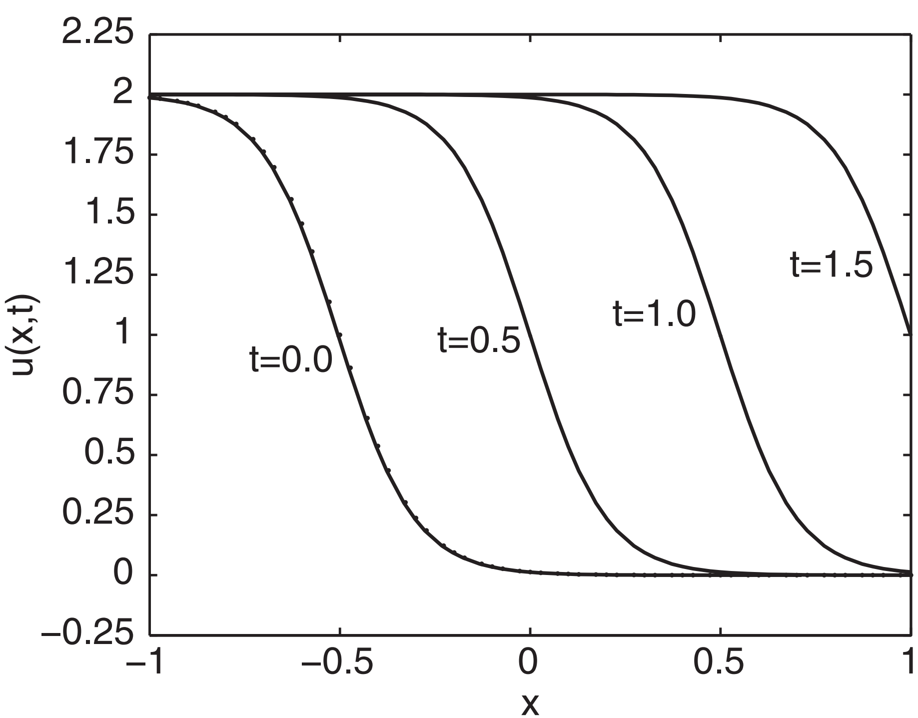
$$(\sqrt{a}u_h)^* = \{\{\sqrt{a}\}\}u_h^+, \quad (\sqrt{a}q_h)^* = \sqrt{a^-}q_h^-.$$

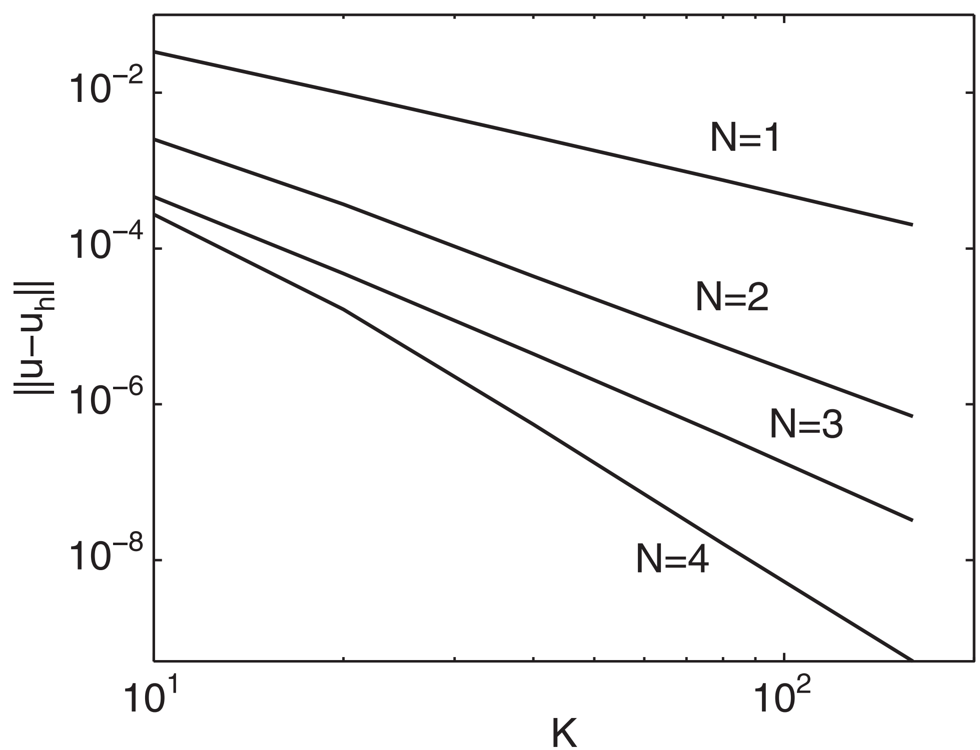
Higher order and mixed problems

Consider viscous Burgers equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = \varepsilon \frac{\partial^2 u}{\partial x^2}, \quad x \in [-1, 1],$$

$$u(x,t) = -\tanh\left(\frac{x + 0.5 - t}{2\varepsilon}\right) + 1.$$





Higher order and mixed problems

Consider the 3rd order dispersive equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3}.$$

Which boundary conditions do we need?

$$\frac{1}{2}\frac{d}{dt}\|u\|_{\Omega}^2 = \left[u\frac{\partial^2 u}{\partial x^2} - \frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2 \right]_{x_l}^{x_r},$$ **must be bounded**

$$x = x_l : \quad \text{On } u \text{ or } \frac{\partial^2 u}{\partial x^2} \text{ and } \frac{\partial u}{\partial x},$$

$$x = x_r : \quad \text{On } u \text{ or } \frac{\partial^2 u}{\partial x^2}.$$

Higher order and mixed problems

Write it as a 1st order system

$$\frac{\partial u}{\partial t} = \frac{\partial q}{\partial x}, \quad q = \frac{\partial p}{\partial x}, \quad p = \frac{\partial u}{\partial x}.$$

To choose the fluxes, we consider the energy

$$\frac{1}{2}\frac{d}{dt}\|u_h\|^2_{\mathsf{D}^k} = \Theta_r - \Theta_l, \qquad \Theta = \frac{p_h^2}{2} - u_h q_h + u_h (q_h)^* + q_h (u_h)^* - p_h (p_h)^*.$$

Central fluxes yields

$$\Theta = \frac{1}{2}\left(u_h^+ q_h^- + u_h^- q_h^+ - p_h^- p_h^+\right), \quad \Longrightarrow \quad \frac{1}{2}\frac{d}{dt}\|u_h\|^2_{\mathsf{D}^k} = 0$$

Alternative LDG-flux

$$(u_h)^* = u_h^-, \quad (q_h)^* = q_h^+, \quad (p_h)^* = p_h^-,$$

$$(u_h)^* = u_h^+, \quad (q_h)^* = q_h^-, \quad (p_h)^* = p_h^-.$$

Higher order and mixed problems

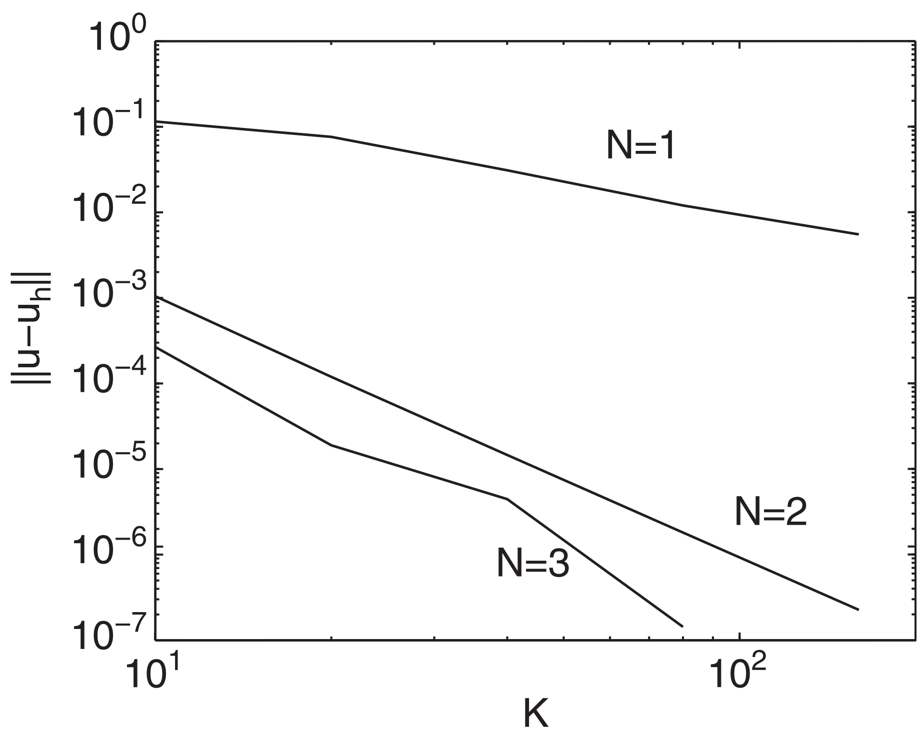
Consider

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3}, \quad x \in [-1, 1],$$

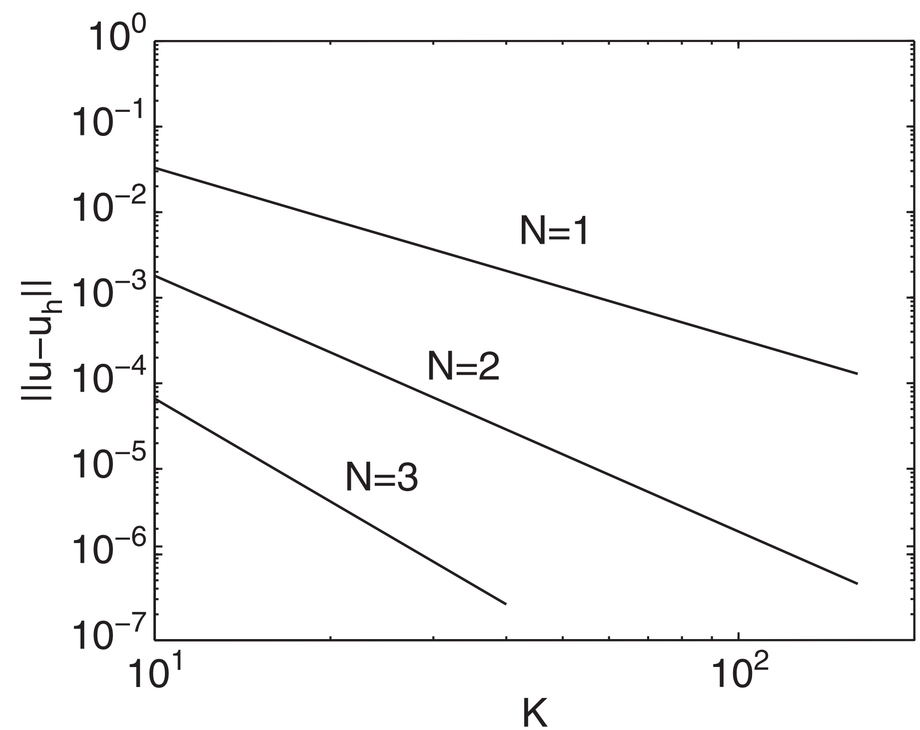
$$u(x,t) = \cos(\pi^3 t + \pi x).$$

Convergence behavior exactly as for the 2nd order problem

Central flux



LDG flux



Higher order and mixed problems

Few comments

- ✓ The reformulation to a system of 1st order problems is entirely general for any order operator
- ✓ When combined with other operators, one chooses fluxes for each operator according to the analysis.
- ✓ The biggest problem is cost -- a 2nd order operator require two derivates rather than one.
- ✓ There are alternative 'direct' ways but they tend to be problem specific

What about the time step ?

For 1st order problems we know

$$\Delta t \leq C \frac{h}{aN^2}$$

Explicit
time-stepping

This gets worse -

$$\Delta t \leq C \left(\frac{h}{N^2} \right)^p$$

p = order of operator

Options :

✓ Local time stepping
✓ Implicit time stepping

Lecture 7

- ✓ Let's briefly recall what we know
- ✓ Brief overview of multi-D analysis
- ✓ Part I: Time-dependent problems
    - ✓ Heat equations
    - ✓ Extensions to higher order problems
- ✓ Part II: Elliptic problems
    - ✓ Different formulations
    - ✓ Stabilization
    - ✓ Solvers and application examples

Elliptic problems

Now we could consider solving a problem like

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - f(x),$$

However, if we are interested in the steady state we may be better off considering

$$\frac{\partial^2 u}{\partial x^2} = f(x),$$

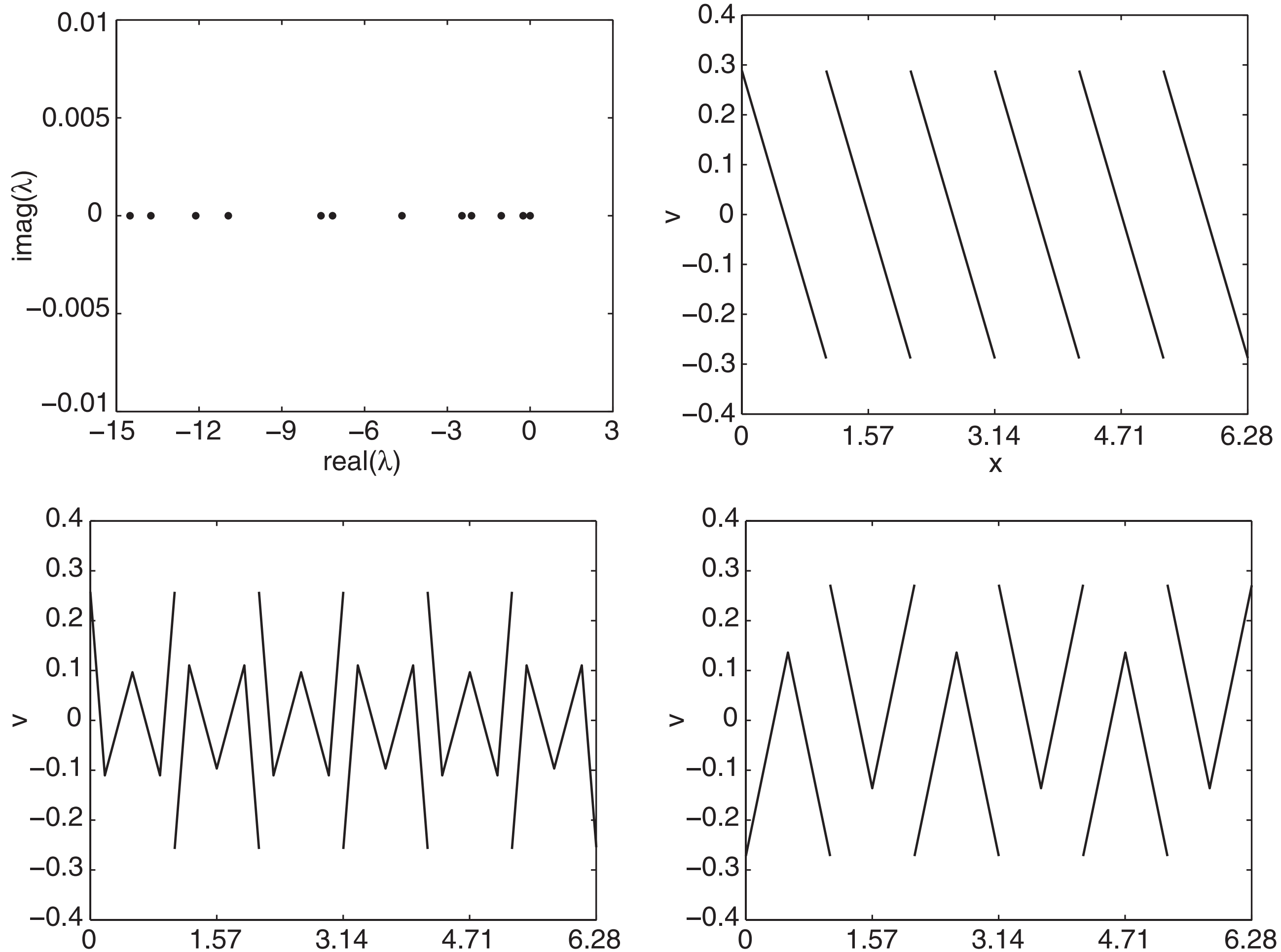
We can use any of the methods we just derived to obtain the linear system

$$A\boldsymbol{u}_h = \boldsymbol{f}_h,$$

Elliptic problems

Assume we use a central flux.

When we try to solve we discover that A is singular!



N=1

K=6

N=4

K=6

N=2

K=6

Elliptic problems

What is happening?

The discontinuous basis is too rich -- it allows one extra null vector:

A local null vector with {{u}}=0

What can we do ?

Change the flux by *penalizing* this mode

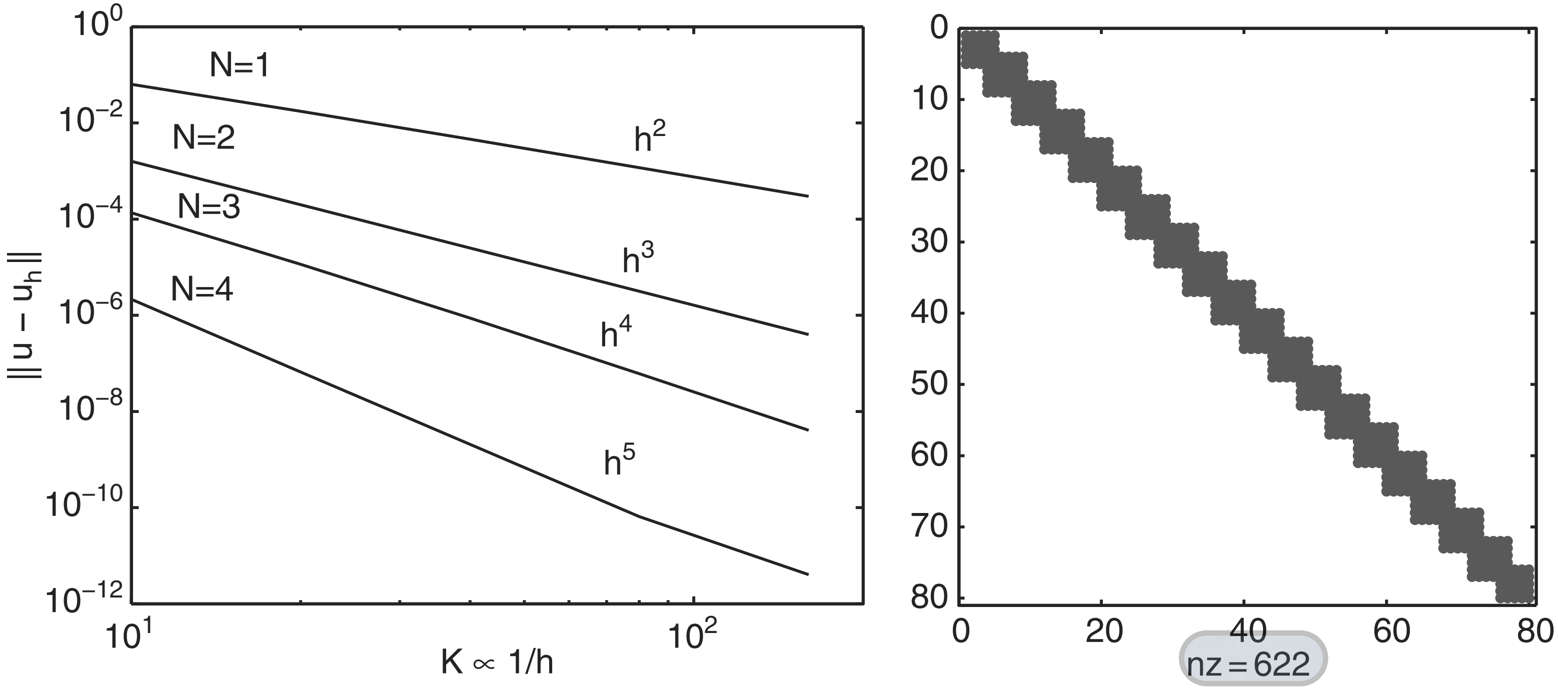
$$q^* = \{\{q\}\} - \tau [\![u]\!], \;\; u^* = \{\{u\}\}.$$

The flexibility of DG shows its strength!

Elliptic problems

Does it work?

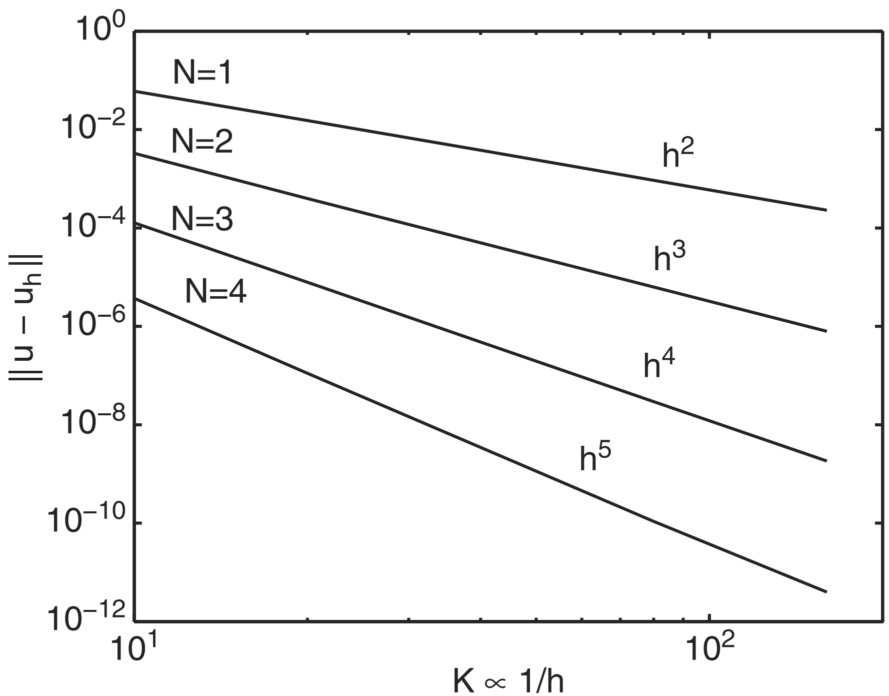
$$\frac{d^2u}{dx^2} = -\sin(x), \quad x \in [0, 2\pi], \qquad u(0) = u(2\pi) = 0.$$

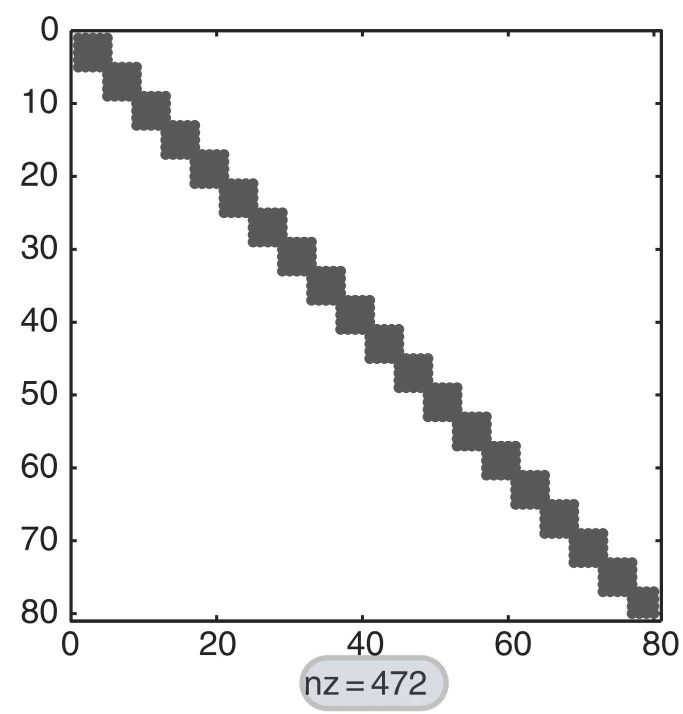


What about the other flux - the LDG flux?

Elliptic problems

Consider the stabilized LDG flux





Works fine as expected - but we also note that A is much more sparse!

Elliptic problems

Why is one more sparse than the other?

Consider the N=0 case

$$q_h^*(q_h^k, q_h^{k+1}, u_h^k, u_h^{k+1}) - q_h^*(q_h^k, q_h^{k-1}, u_h^k, u_h^{k-1}) = hf_h^k,$$
$$u_h^*(u_h^k, u_h^{k+1}) - u_h^*(u_h^k, u_h^{k-1}) = hg_h^k.$$

Using the central flux yields

$$q_h^*(q_h^-, q_h^+, u_h^-, u_h^+) = \{\{q_h\}\} - \tau[\![u_h]\!], \quad u_h^*(u_h^-, u_h^+) = \{\{u_h\}\},$$

$$\frac{u_h^{k+1} - 2u_h^k + u_h^{k-1}}{h^2} + \tau \frac{u_h^{k+1} - u_h^{k-1}}{h} = f_h^k.$$

Wide

Using the LDG flux yields

$$q_h^*(q_h^-, q_h^+, u_h^-, u_h^+) = q_h^- - \tau[\![u_h]\!], \quad u_h^*(u_h^-, u_h^+) = u_h^+,$$

$$\frac{u_h^{k+1} - 2u_h^k + u_h^{k-1}}{h^2} + \tau \frac{u_h^{k+1} - u_h^{k-1}}{h} = f_h^k.$$

Elliptic problems

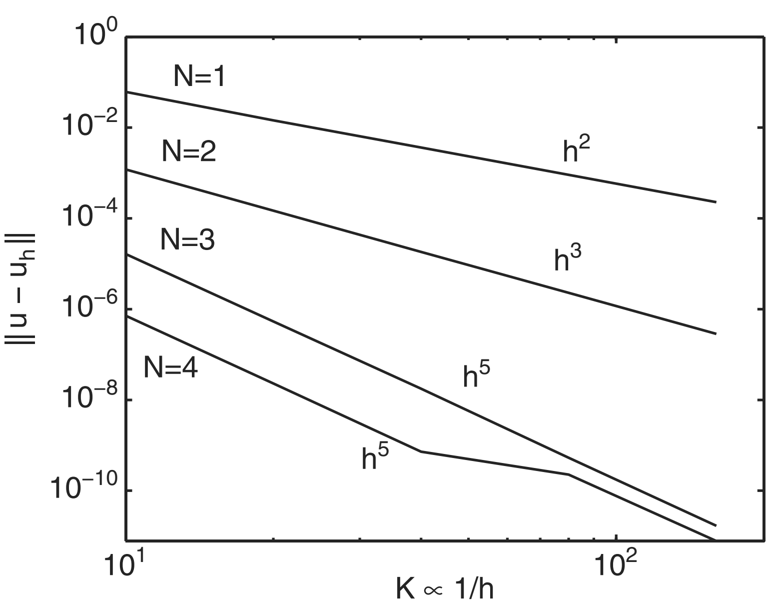
The sparsity is a good thing -- but it comes at a price

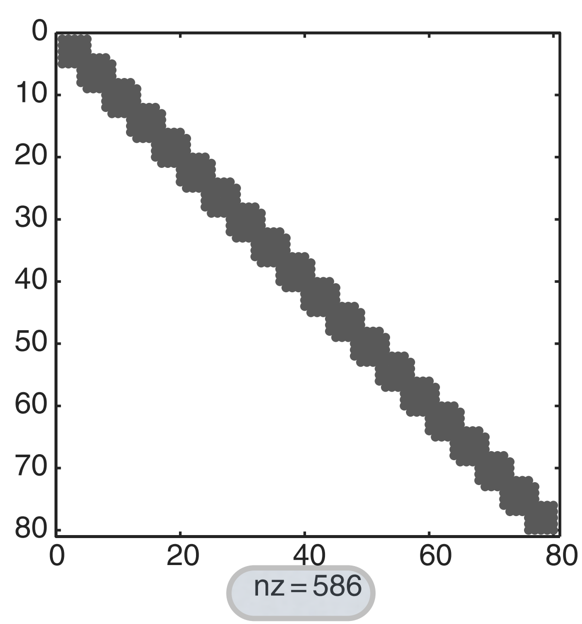
$$\kappa(\mathcal{A}_{LDG}) \simeq 2\kappa(\mathcal{A}_C);$$

We seek a flux balancing sparsity and conditioning?

$$q_h^* = \{\{(u_h)_x\}\} - \tau[\![u_h]\!], \quad u_h^* = \{\{u_h\}\}.$$

Internal penalty flux





$$\kappa(\mathcal{A}_C) \simeq \kappa(\mathcal{A}_{IP});$$

Mission accomplished

Elliptic problems

Remaining question: How do you choose τ ?

The analysis shows that:

- ✓ For the central flux, $\tau > 0$ suffices
- ✓ For the LDG flux, $\tau > 0$ suffices
- ✓ For the IP flux, one must require

$$\tau \geq C\frac{(N+1)^2}{h}, \quad C \geq 1,$$

These suffices to guarantee stability,
but they may not give the best accuracy

Generally, a good choice is $\tau \geq C\dfrac{(N+1)^2}{h}, \quad C \geq 1,$

Elliptic problems

What can we say more generally?

Consider

$$-\nabla^2 u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega,$$

Discretized as $\quad -\nabla \cdot \boldsymbol{q} = f, \quad \boldsymbol{q} = \nabla u.$

$$\left(\boldsymbol{q}_h, \nabla \phi_h\right)_{\Omega,h} - \sum_{k=1}^{K} \left(\hat{\boldsymbol{n}} \cdot \boldsymbol{q}_h^*, \phi_h\right)_{\partial \mathsf{D}^k} = \left(f, \phi_h\right)_{\Omega,h},$$

$$\left(\boldsymbol{q}_h, \boldsymbol{\pi}_h\right)_{\Omega,h} = \sum_{k=1}^{K} \left(u_h^*, \hat{\boldsymbol{n}} \cdot \boldsymbol{\pi}_h\right)_{\partial \mathsf{D}^k} - \left(u_h, \nabla \cdot \boldsymbol{\pi}_h\right)_{\Omega,h}$$

Using one of the fluxes

	u_h^*	$\boldsymbol{q}_h^*$
Central flux	$\{\{u_h\}\}$	$\{\{\boldsymbol{q}_h\}\} - \tau [\![u_h]\!]$
Local DG flux (LDG)	$\{\{u_h\}\} + \boldsymbol{\beta} \cdot [\![u_h]\!]$	$\{\{\boldsymbol{q}_h\}\} - \boldsymbol{\beta}[\![\boldsymbol{q}_h]\!] - \tau [\![u_h]\!]$
Internal penalty flux (IP)	$\{\{u_h\}\}$	$\{\{\nabla u_h\}\} - \tau [\![u_h]\!]$

Elliptic problems

For the 3 discrete systems, one can prove (see text)

- ✓ They are all symmetric for any N
- ✓ The are all invertible provided stabilization is used
- ✓ The discretization is consistent
- ✓ The adjoint problem is consistent
- ✓ They have optimal convergence in L2

Many of these results can be extended to more general problems (saddle-point, non-coercive etc)

There are other less used fluxes also

Solving the systems

After things are discretized, we end up with

$$\mathcal{A}\boldsymbol{u}_h = \boldsymbol{f}_h$$

We can solve this in two different ways

- ✓ Direct methods
- ✓ Iterative methods

The 'right' choice depends on things such as size, speed, sparsity etc

Solving the systems

Direct methods are 'LU' based

$$>> [\mathsf{L},\ \mathsf{U}] = \mathsf{lu(A)};$$
$$>> \mathsf{u} = \mathsf{U}\backslash(\mathsf{L}\backslash\mathsf{f});$$

Example:

$$\nabla^2 u = f(x,y) = \left(\left(16-n^2\right)r^2 + \left(n^2-36\right)r^4\right)\sin\left(n\theta\right),\ \ x^2+y^2 \le 1,$$

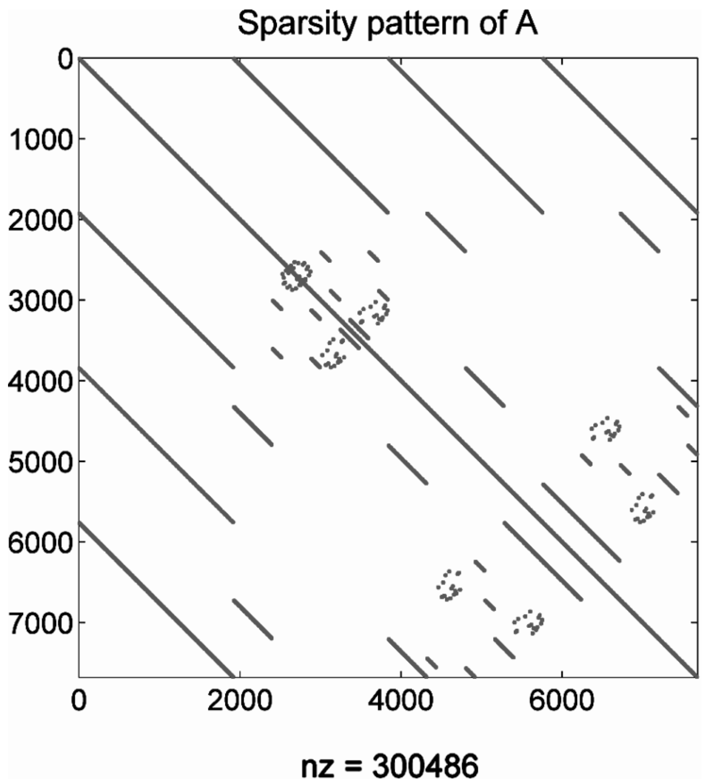
$$n = 12,\ r = \sqrt{x^2+y^2}, \theta = \arctan\left(y,x\right)$$

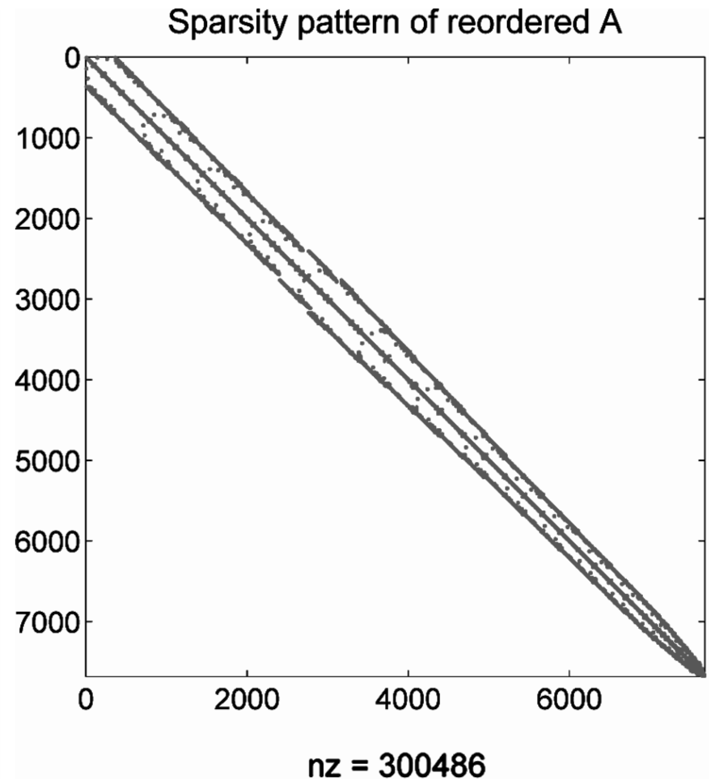
K=512

N=4

7680 DoF

Solving the systems





8,7m extra non-zero entries in (L,U)

Reordering is needed !

Cuthill-McKee ordering

3,7m extra non-zero entries in (L,U)

Solving the systems

Re-ordering:

```
>> P = symrcm(A);
>> A = A(P,P);
>> rhs = rhs(P);
>> [L,U] = lu(A);
>> u = U\(L\f);
>> u(P) = u;
```

.. but A is SPD: $\mathcal{A} = \mathcal{C}^T \mathcal{C}$ Cholesky decomp

```
>> C = chol(A);
>> u = C\(C'\f);
```

1,9m extra non-zero entries in C

Solving the systems

If the problem is too large, iterative methods are the only choice

```
>> ittol = 1e-8; maxit = 1000;
>> u = pcg(A, f, ittol, maxit);
```

Example requires 818 iterations - 100 times slower than LU !

Solution: Preconditioning

$$\mathcal{C}^{-1}\mathcal{A}\boldsymbol{u}_h = \mathcal{C}^{-1}\boldsymbol{f}_h,$$

Solving the systems

How to choose the preconditioning ?

.. more an art than a science !

Example - Incomplete Cholesky Preconditioning

```
>> ittol = 1e-8; maxit = 1000;
>> Cinc = cholinc(OP, '0')
>> u = pcg(A, f, ittol, maxit, Cinc', Cinc);
```

Sparsity preserving

138 iterations - but still 50 times slower

```
>> ittol = 1e-8; maxit = 1000; droptol = 1e-4;
>> Cinc = cholinc(A, droptol);
>> u = pcg(A, b, ittol, maxit, Cinc', Cinc);
```

Drop tolerance

17 iterations - only 2 times slower

Solving the systems

Choosing fast and efficient linear solvers is not easy -- but there are many options

- ✓ **Direct solvers**
  - ✓ MUMPS (multi-frontal parallel solver)
  - ✓ SuperLU (fast parallel direct solver)

- ✓ **Iterative solvers**
  - ✓ Trilinos (large solver/precon library)
  - ✓ PETSc (large solver/precon library)

Very often you have to try several options and combinations to find the most efficient and robust one(s)

A few examples

So far we have seen lots of theory and “homework” problems.

To see that it also works for more complex problems - but still 2D - let us look at a few examples

- ✓ Incompressible Navier-Stokes
- ✓ Boussinesq problems
- ✓ Compressible Euler equations

Incompressible fluid flow

Time-dependent Navier-Stokes equations

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\, \boldsymbol{u} = -\nabla p + \nu \nabla^2 \boldsymbol{u}, \quad \boldsymbol{x} \in \Omega,$$
$$\nabla \cdot \boldsymbol{u} = 0,$$

- Water
- Low speed
- Bioflows
- etc

Written on conservation form

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \mathcal{F} = -\nabla p + \nu \nabla^2 \boldsymbol{u},$$
$$\nabla \cdot \boldsymbol{u} = 0,$$

$$\mathcal{F} = [\boldsymbol{F}_1, \boldsymbol{F}_2] = \begin{bmatrix} u^2 & uv \\ uv & v^2 \end{bmatrix}.$$

Solved by stiffly stable time-splitting and pressure projection

Incompressible fluid flow

The basics are

$$N_x(\boldsymbol{u}) = \nabla \cdot \boldsymbol{F}_1 = \frac{\partial\left(u^2\right)}{\partial x} + \frac{\partial(uv)}{\partial y}, \quad N_y(\boldsymbol{u}) = \nabla \cdot \boldsymbol{F}_2 = \frac{\partial(uv)}{\partial x} + \frac{\partial\left(v^2\right)}{\partial y}$$

.. and then take an inviscous time step

$$\frac{\gamma_0 \tilde{\boldsymbol{u}} - \alpha_0 \boldsymbol{u}^n - \alpha_1 \boldsymbol{u}^{n-1}}{\Delta t} = -\beta_0 \mathcal{N}(\boldsymbol{u}^n) - \beta_1 \mathcal{N}(\boldsymbol{u}^{n-1}) .$$

The pressure is computed to ensure incompressibility

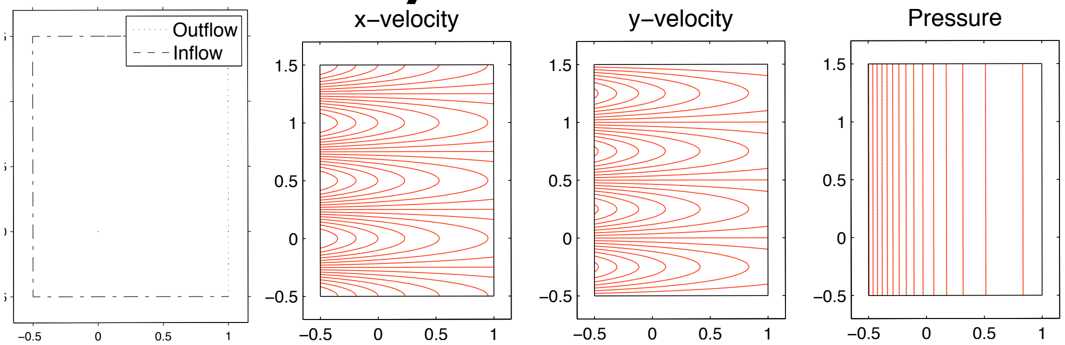
$$\gamma_0 \frac{\tilde{\tilde{\boldsymbol{u}}} - \tilde{\boldsymbol{u}}}{\Delta t} = -\nabla \bar{p}^{n+1}. \quad -\nabla^2 \bar{p}^{n+1} = -\frac{\gamma_0}{\Delta t} \nabla \cdot \tilde{\boldsymbol{u}}. \qquad \tilde{\tilde{\boldsymbol{u}}} = \tilde{\boldsymbol{u}} - \frac{\Delta t}{\gamma_0} \nabla \bar{p}^{n+1}.$$

.. and the viscous part is updated

$$\gamma_0 \left(\frac{\boldsymbol{u}^{n+1} - \tilde{\tilde{\boldsymbol{u}}}}{\Delta t} \right) = \nu \nabla^2 \boldsymbol{u}^{n+1},$$

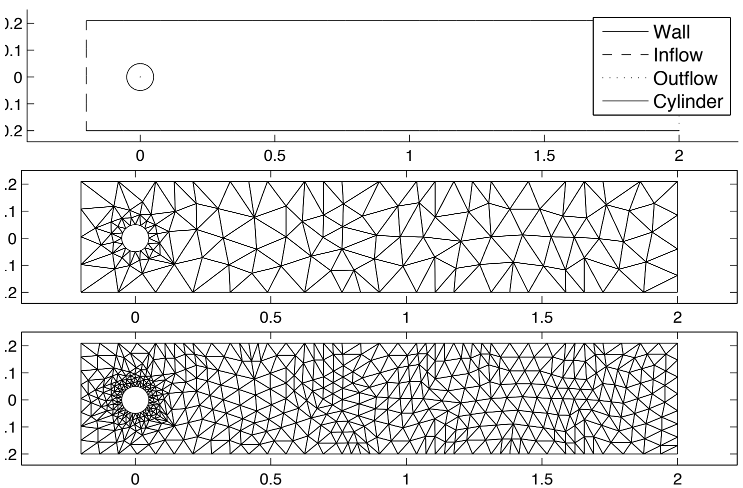
Incompressible fluid flow

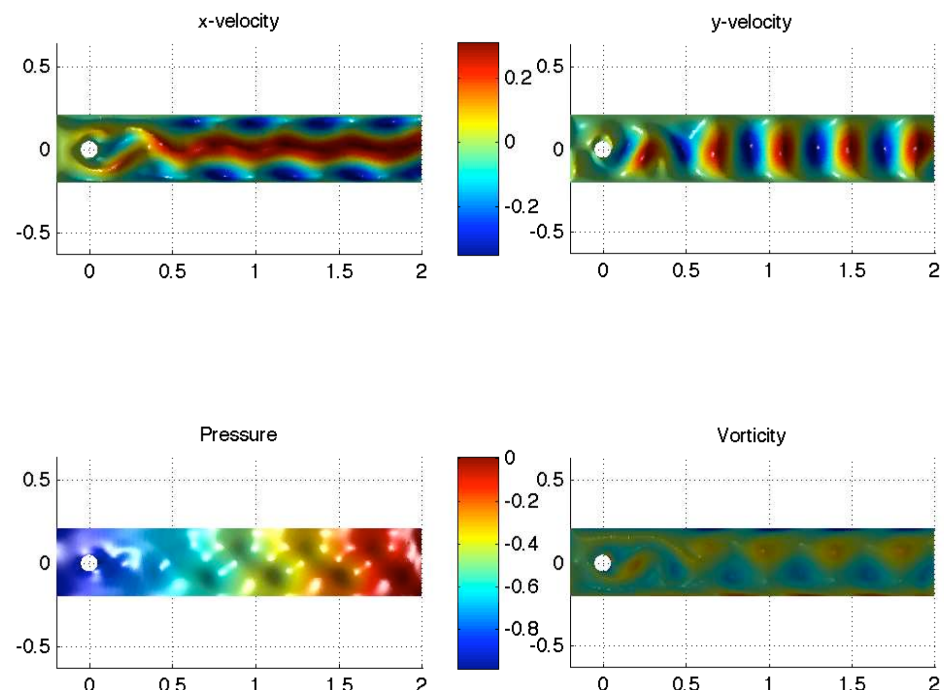
Kovasznay solution



N	Error in u: h	$h/2$	$h/4$	Rate	Error in p: h	$h/2$	$h/4$	Rate
1	1.32E+00	7.05E-01	1.23E-01	1.71	3.13E+00	1.53E+00	3.48E-01	1.59
2	5.01E-01	9.67E-02	1.45E-02	2.55	1.47E+00	2.54E-01	2.08E-02	3.07
3	2.41E-01	2.74E-02	1.89E-03	3.49	5.02E-01	2.79E-02	2.42E-03	3.85
4	6.40E-02	3.34E-03	1.00E-04	4.66	9.31E-02	8.87E-03	2.02E-04	4.42
5	1.87E-02	6.92E-04	8.96E-06	5.51	6.15E-02	1.02E-03	1.89E-05	5.84
6	9.07E-03	5.41E-05	6.93E-07	6.84	1.23E-02	2.17E-04	2.63E-06	6.09
7	1.43E-03	1.37E-05	4.03E-08	7.56	4.73E-03	4.67E-05	1.29E-07	7.58
8	5.91E-04	1.02E-06	7.17E-09	8.16	1.52E-03	3.54E-06	1.97E-08	8.12

von Karman flow



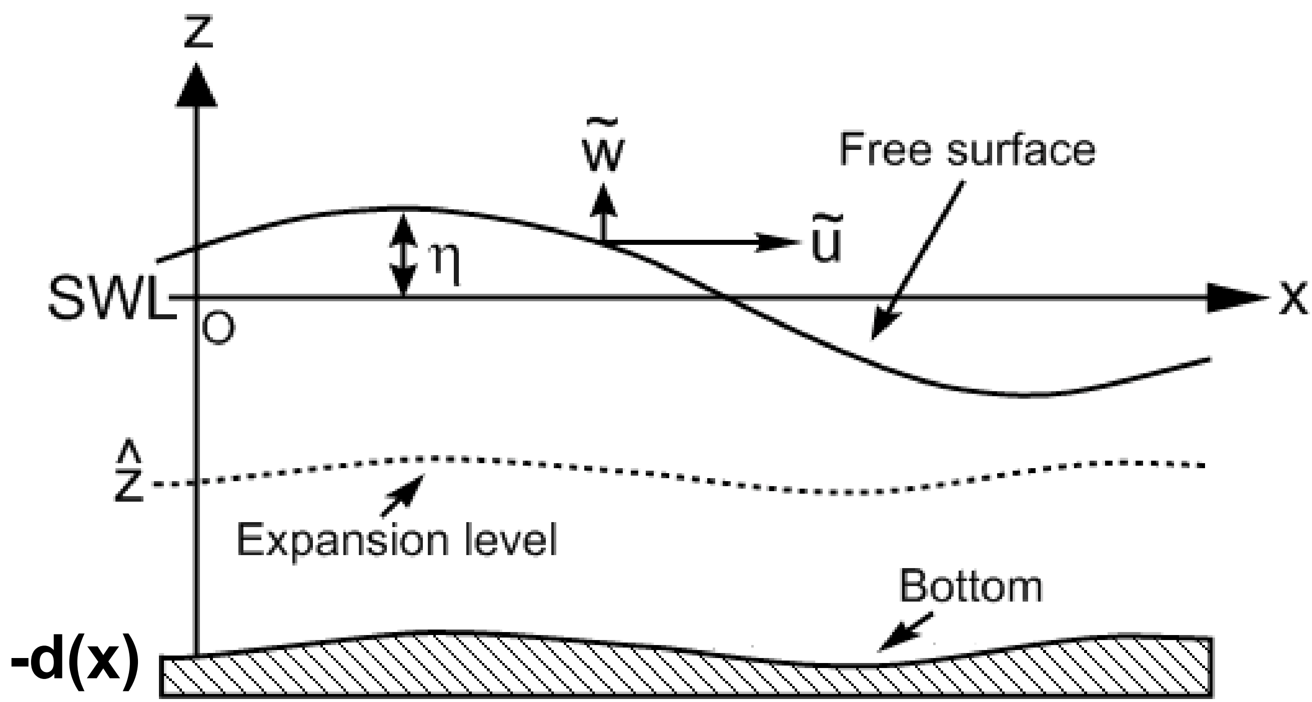


K	N	t_{C_d}	C_d	t_{C_l}	C_l	$\Delta p\,(t=8)$
115	6	3.9394263	2.9529140	5.6742937	0.4966074	-0.1095664
460	6	3.9363751	2.9509030	5.6930431	0.4778757	-0.1116310
236	8	3.9343595	2.9417190	5.6990205	0.4879853	-0.1119122
236	10	3.9370396	2.9545659	5.6927772	0.4789706	-0.1116177
[189]	N/A	3.93625	2.950921575	5.693125	0.47795	-0.1116

Fluid-structure interaction

Boussinesq modeling

The basis assumption of this approach is to approximate the vertical variation using an expansion in z.



Fluid-structure interaction

Under certain assumptions, the proper model (a high-order Boussinesq model) becomes

$$\partial_t \tilde{\boldsymbol{U}} + \boldsymbol{\nabla}\left(g\eta + \frac{1}{2}\left(\tilde{\boldsymbol{U}}\cdot\tilde{\boldsymbol{U}} - \tilde{w}^2(1+\boldsymbol{\nabla}\eta\cdot\boldsymbol{\nabla}\eta)\right)\right) = 0.$$

$$\partial_t\eta - \tilde{w} + \boldsymbol{\nabla}\eta\cdot\left(\tilde{\boldsymbol{U}} - \tilde{w}\boldsymbol{\nabla}\eta\right) = 0,$$

$$\begin{bmatrix} \tilde{U} \\ \tilde{V} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathcal{A}_1 - \partial_x\eta\cdot\mathcal{B}_{11} & -\partial_x\eta\cdot\mathcal{B}_{12} & \mathcal{B}_{11} + \partial_x\eta\cdot\mathcal{A}_1 \\ -\partial_y\eta\cdot\mathcal{B}_{11} & \mathcal{A}_1 - \partial_y\eta\cdot\mathcal{B}_{12} & \mathcal{B}_{12} + \partial_y\eta\cdot\mathcal{A}_1 \\ \mathcal{A}_{01} + \mathcal{S}_1 & \mathcal{A}_{02} + \mathcal{S}_2 & \mathcal{B}_0 + \mathcal{S}_{03} \end{bmatrix} \begin{bmatrix} \hat{u}^* \\ \hat{v}^* \\ \hat{w}^* \end{bmatrix},$$

$$\tilde{w} = -\mathcal{B}_{11}\hat{u}^* - \mathcal{B}_{12}\hat{v}^* + \mathcal{A}_1\hat{w}^*.$$

Fluid-structure interaction

Where we have high-order derivates since

$$\mathcal{A}_{01} = \lambda\partial_x + \gamma_3\lambda^3(\partial_{xxx} + \partial_{xyy}) + \gamma_5\lambda^5(\partial_{xxxxx} + 2\partial_{xxxyy} + \partial_{xyyyy}),$$
$$\mathcal{A}_{02} = \lambda\partial_y + \gamma_3\lambda^3(\partial_{xxy} + \partial_{yyy}) + \gamma_5\lambda^5(\partial_{xxxxy} + 2\partial_{xxyyy} + \partial_{yyyyy}),$$
$$\mathcal{A}_1 = 1 - \alpha_2(\partial_{xx} + \partial_{yy}) + \alpha_4(\partial_{xxxx} + 2\partial_{xxyy} + \partial_{yyyy}),$$

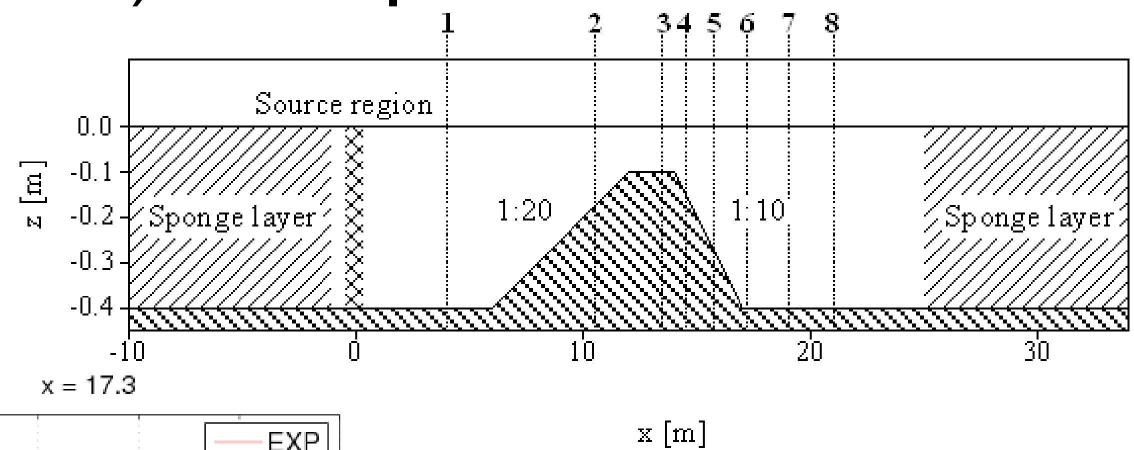
$$\mathcal{B}_0 = 1 + \gamma_2\lambda^2(\partial_{xx} + \partial_{yy}) + \gamma_4\lambda^4(\partial_{xxxx} + 2\partial_{xxyy} + \partial_{yyyy}),$$
$$\mathcal{B}_{11} = \beta_1\partial_x - \beta_3(\partial_{xxx} + \partial_{xyy}) + \beta_5(\partial_{xxxxx} + 2\partial_{xxxyy} + \partial_{xyyyy}),$$
$$\mathcal{B}_{12} = \beta_1\partial_y - \beta_3(\partial_{xxy} + \partial_{yyy}) + \beta_5(\partial_{xxxxy} + 2\partial_{xxyyy} + \partial_{yyyyy}),$$

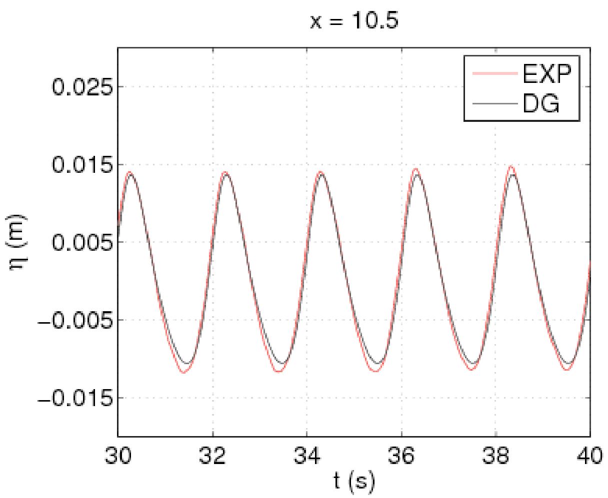
$$\mathcal{S}_1 = \partial_x d \cdot \mathcal{C}_1, \qquad \mathcal{C}_1 = 1 - c_2\lambda^2(\partial_{xx} + \partial_{yy}) + c_4\lambda^4(\partial_{xxxx} + 2\partial_{xxyy} + \partial_{yyyy}).$$
$$\mathcal{S}_2 = \partial_y d \cdot \mathcal{C}_1,$$

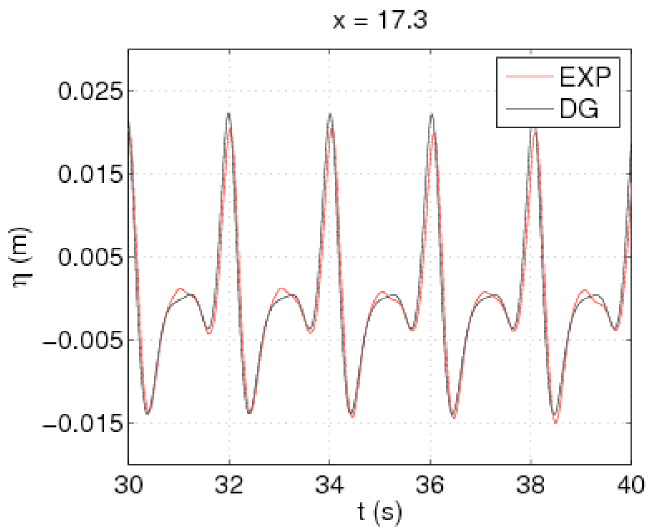
A bit on the complicated side !

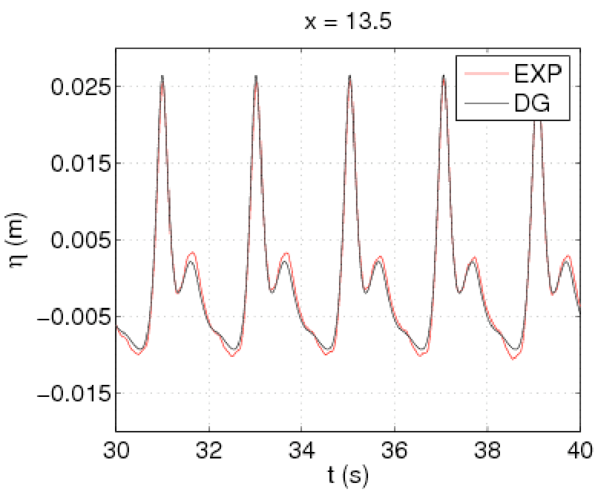
A couple of 2D(1D) tests

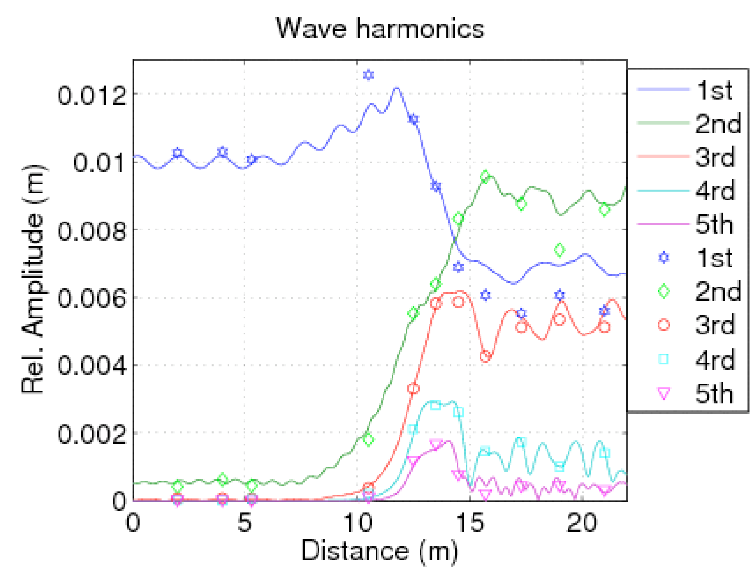
Submerged bar (K=110, P=8) - comparison with experimental data





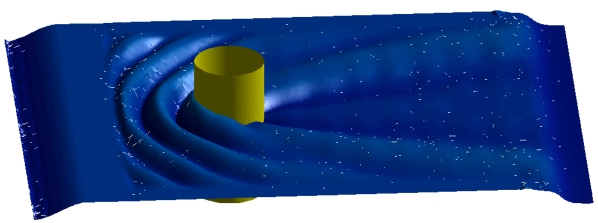


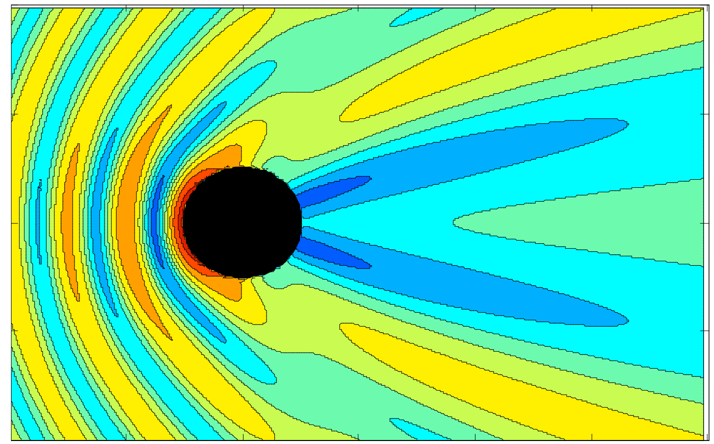


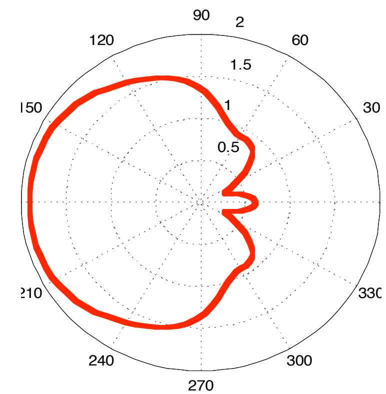


A couple of 3D(2D) tests

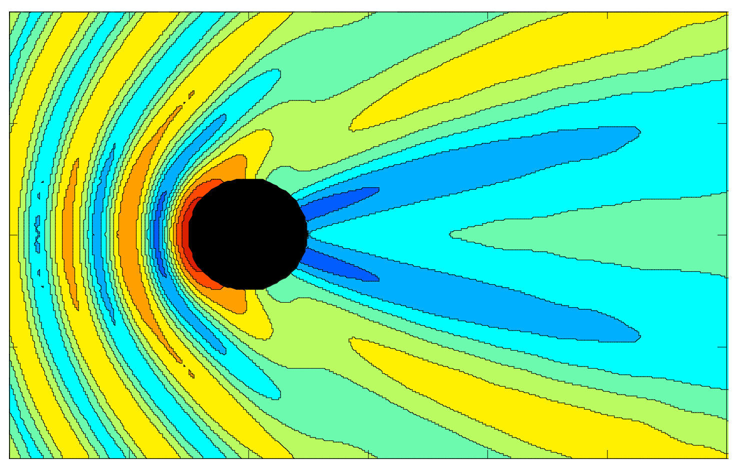
McCamy & Fuchs (1954)

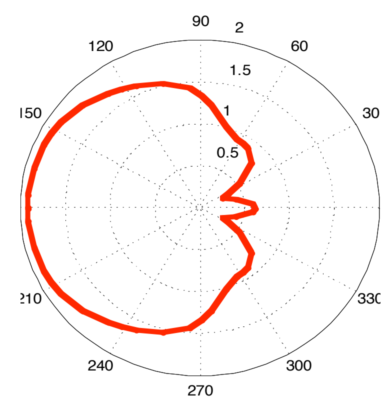






DG-FEM solution:
ka=pi, kd=1.0,
P=4, K=1261,
t=0.03s





Compressible fluid flow

Time-dependent Euler equations

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0,$$

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u\,(E+p) \end{pmatrix}, \mathbf{G} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v\,(E+p) \end{pmatrix},$$

- Gas
- High speed
- etc

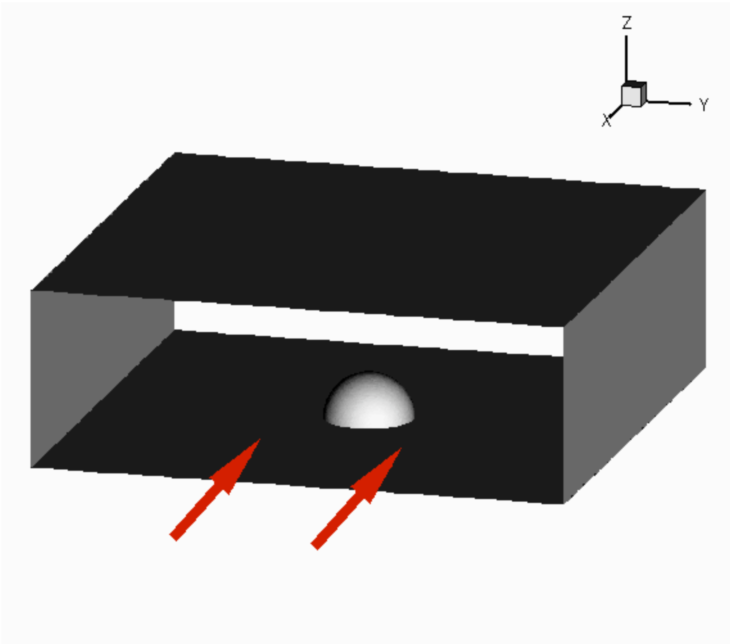
Formulation is straightforward

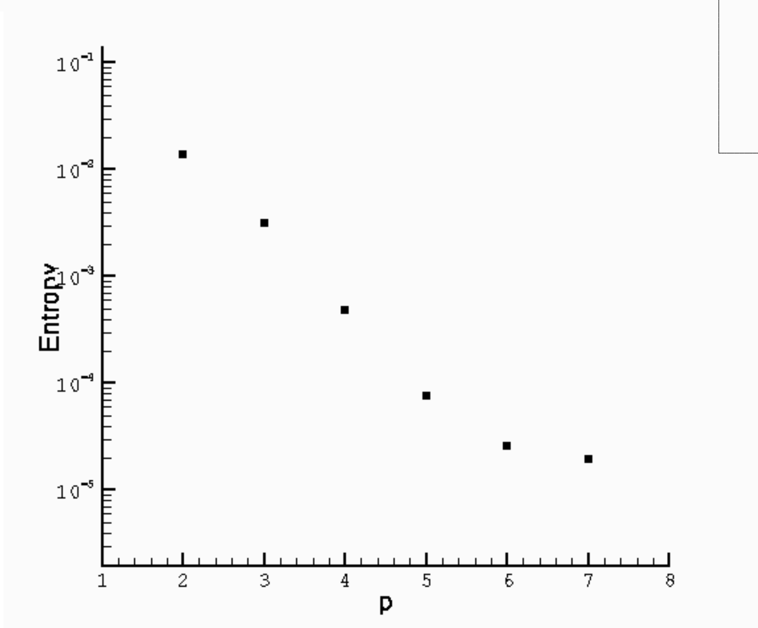
$$\int_{\mathsf{D}^k} \left(\frac{\partial \mathbf{q}_h}{\partial t}\phi_h - \mathbf{F}_h \frac{\partial \phi_h}{\partial x} - \mathbf{G}_h \frac{\partial \phi_h}{\partial y} \right) d\boldsymbol{x} + \oint_{\partial \mathsf{D}^k} \left(\hat{n}_x \mathbf{F}_h + \hat{n}_y \mathbf{G}_h \right)^* \phi_h d\boldsymbol{x} = 0.$$

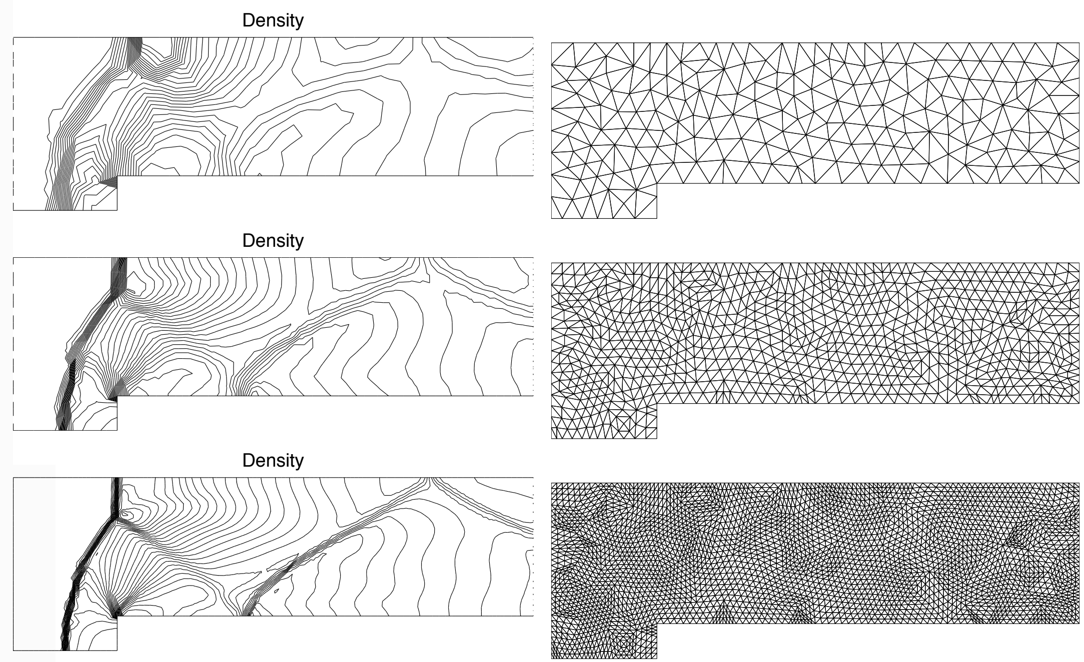
$$\left(\hat{n}_x \mathbf{F}_h + \hat{n}_y \mathbf{G}_h \right)^* = \hat{n}_x \{\{\mathbf{F}_h\}\} + \hat{n}_y \{\{\mathbf{G}_h\}\} + \frac{\lambda}{2} \cdot [\![\mathbf{q}_h]\!].$$

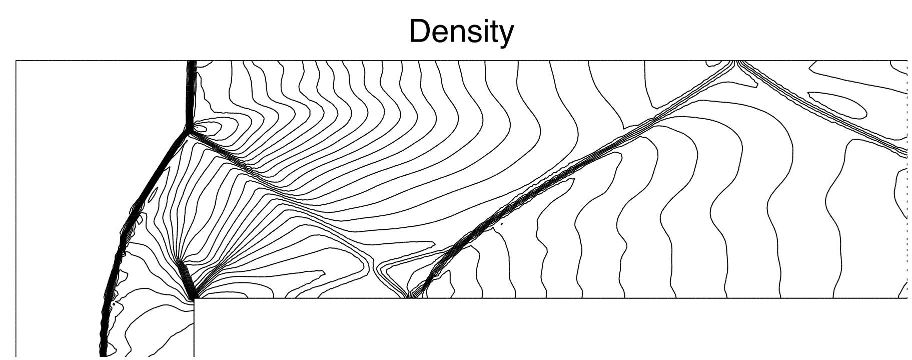
Challenge: Shocks -- this requires limiting/filtering

Compressible fluid flow









Remarks

We are done with all the basic now ! -- and we have started to see it work for us

What we need to worry about is:

- ✓ The need for 3D
- ✓ The need for speed
- ✓ Software beyond Matlab

Tomorrow !