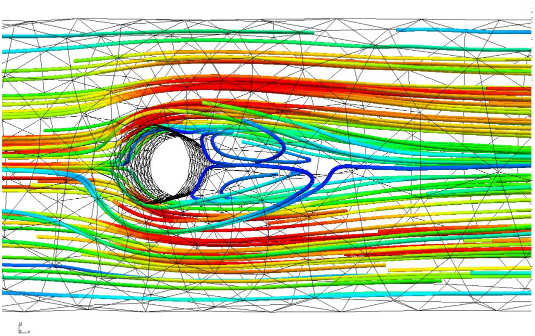
An operator splitted Hybrid DG - DG method for solving incompressible Navier Stokes equations

Christoph Lehrenfeld, Joachim Schöberl

Vienna, April 14th, 2011



Overview

Overview

Operator splitting for conv. diff. problems: Motivation I

Convection Diffusion type problems

We consider problems of the form

$$\left(\frac{\partial}{\partial t} + A + C(u)\right) u = f$$

- with A a linear elliptic operator ("stiff"), e.g. $-\Delta u$
- with C a (nonlinear) hyperbolic operator ("non-stiff"), e.g. $\operatorname{div}(b\ u)$

CFL-type restriction

Applying standard discretization techniques (FD,FV,FEM,DG) in space combined with an *explicit time integration method* typically results in time step restrictions of the form (h the resolution length):

$$\Delta t \overset{!}{\leq} \min(C_A h^2, C_C(u) h)$$

Operator splitting for conv. diff. problems: Motivation II

The stiff part

For the purely elliptic problem $\left(\frac{\partial}{\partial t} + A\right) u = f$ the time step restriction $\Delta t^A \overset{!}{\leq} C_A h^2$ is typically very strong. $\Rightarrow$ Many time steps

The non-stiff part

For the purely hyperbolic problem $\left(\frac{\partial}{\partial t} + C(u)\right) u = f$ the time step restriction $\Delta t^C \overset{!}{\leq} C_C(u) h$ is typically less severe. $\Rightarrow$ Few time steps

The reality

- for moderate h and/or small C_A time step restriction might not be serious
- for small h and/or large $C_C(u)$ time step restriction might already be very serious
- no uniform grid, s.t. time step restrictions are not uniform in space (operator splitting space, local time stepping, etc.)

Operator splitting for conv. diff. problems: Motivation III

Choice of Time Integration method

- implicit time integration:
  - linearization schemes necessary (for $C(u) \neq C$)
  - large linear systems have to be solved
  - robust
  - typically unconditionally stable:
    $\Rightarrow \Delta t \leq \Delta t_{accuracy}$
- explicit time integration:
  - no large (non)linear systems to solve for $A + C(u)$
  - (eventually) solutions with mass matrix necessary
  - only conditionally stable
    $\Rightarrow \Delta t \leq \min(\Delta t_{accuracy}, \Delta t_{stability})$

Explicit approaches work fine as long as

$$\Delta t_{accuracy} \leq \min(\Delta t^{A}_{stability}, \Delta t^{C}_{stability})$$

Operator splitting for conv. diff. problems: Motivation III

Choice of Time Integration method

- implicit time integration:
  - linearization schemes necessary (for $C(u) \neq C$)
  - large linear systems have to be solved
  - robust
  - typically unconditionally stable:
    $\Rightarrow \Delta t \leq \Delta t_{accuracy}$
- explicit time integration:
  - no large (non)linear systems to solve for $A + C(u)$
  - (eventually) solutions with mass matrix necessary
  - only conditionally stable
    $\Rightarrow \Delta t \leq \min(\Delta t_{accuracy}, \Delta t_{stability})$

Explicit approaches work fine as long as

$$\Delta t_{accuracy} \leq \frac{\texttt{time for expl. step}}{\texttt{time for impl. step}} \min(\Delta t^{A}_{stability}, \Delta t^{C}_{stability})$$

Operator splitting for conv. diff. problems: Motivation IV

Operator splitting idea:

Can't I integrate some operator explicitely and the other operator implicitely?

- use implicit time integration where necessary (e.g. constraints) or more efficient (strong CFL-restrictions)
- use explicit time integration where it is more efficient

Remark:

The distinction between explicit and implicit integration is not the only interesting splitting!

Operator splitting: Additive splitting methods I

Example

Forward-Backward / Semi-Implicit Euler:

$$\left(\frac{1}{\Delta t} + A\right) u^{n+1} = f^{n+1} + \left(\frac{1}{\Delta t} + C(u^n)\right) u^n$$

Structure

- Evaluate explicit and implicit parts at different time stages
- Evaluate explicit part only at old (known) time stages

Generalizations

- Use partitioned Runge-Kutta methods, i.e. two butcher tableaus with identical time stages.
- Multistep methods, e.g. SBDF, i.e. BDF methods for implicit part combined with AB for explicit

Operator splitting: Additive splitting methods I

Example

BDF / Adams-Bashforth:

$$\left(\frac{a_0}{\Delta t} + A\right) u^{n+1} = f^{n+1} + \frac{1}{\Delta t}\sum_{i=1}^{k} a_i u^{n+1-i} + \sum_{i=1}^{k} b_i C(u^{n+1-i}) u^{n+1-i}$$

Structure

- Evaluate explicit and implicit parts at different time stages
- Evaluate explicit part only at old (known) time stages

Generalizations

- Use partitioned Runge-Kutta methods, i.e. two butcher tableaus with identical time stages.
- Multistep methods, e.g. SBDF, i.e. BDF methods for implicit part combined with AB for explicit

Operator splitting: Additive splitting methods I

Example

partitioned Runge-Kutta:

$$\begin{array}{c|ccc|cc} 0 & 0 & 0 & 0 & & \\ \gamma & \gamma & 0 & 0 & \gamma & 0 \\ 1 & \delta & 1-\delta & 0 & 1-\gamma & \gamma \\ \hline & \delta & 1-\delta & 0 & 1-\gamma & \gamma \end{array}$$

Structure

- Evaluate explicit and implicit parts at different time stages
- Evaluate explicit part only at old (known) time stages

Generalizations

- Use partitioned Runge-Kutta methods, i.e. two butcher tableaus with identical time stages.
- Multistep methods, e.g. SBDF, i.e. BDF methods for implicit part combined with AB for explicit

Operator splitting: Additive splitting methods II

Names/Aliases

- IMEX (implicit-explicit) [Ascher, Ruuth, Wetton, '95]
- ARK (additive Runge-Kutta) [Carpenter, Kennedy, '01]
- Semi-Implicit (Euler/BDF/...)
- partitioned Runge-Kutta methods [also appear in other splitting approaches]

Dis-/Advantages of additive splitting

(+) avoids implicit solutions with $C(u)$

(+) fairly simple to implement

(+) consistent

(-) time steps for explicit and implicit are not decoupled
$\Rightarrow \Delta t^A = \Delta t^C \leq \Delta t^C_{stability}$

Operator splitting: Multiplicative splitting methods I

Idea:

Can't we decompose the problem into subproblems of the following form?

$$\left(\frac{\partial}{\partial t} + A\right) u = \tilde{f} \quad \text{and} \quad \left(\frac{\partial}{\partial t} + C(u)\right) u = \tilde{f}$$

Operator-Integration-Factor Splitting

Rewrite original problem to

$$\frac{\partial}{\partial t}(Q^{t \to t^*} u) = Q^{t \to t^*}(f - Au)$$

with Q the *propagation operator*, s.t. $Q^{t_1 \to t_2} u_1 = v(t_2)$ with v the solution of the explicit *propagation problem*:

$$\frac{\partial}{\partial s} v = -C(v) v \quad v(t_1) = u_1$$

[Maday, Patera, Rønquist, '90]

Operator splitting: Multiplicative splitting methods II

Operator-Integration-Factor Splitting

Rewrite original problem to

$$\frac{\partial}{\partial t}(Q^{t\to t^*}u) = Q^{t\to t^*}(f - Au)$$

with Q the *propagation operator*, s.t. $Q^{t_1\to t_2}u_1 = v(t_2)$ with v the solution of the explicit *propagation problem*:

$$\frac{\partial}{\partial s}v = -C(v)v \quad v(t_1) = u_1$$

[Maday, Patera, Rønquist, '90]

First order example: Implicit Euler for implicit problem

Choose $t^* = t^{n+1}$ and replace $\frac{\partial}{\partial t}$ by a 1st order backward difference:

$$\frac{1}{\Delta t}(Q^{t^{n+1}\to t^{n+1}}u^{n+1} - Q^{t^{n}\to t^{n+1}}u^{n}) = Q^{t^{n+1}\to t^{n+1}}(f^{n+1} - Au^{n+1})$$

Operator splitting: Multiplicative splitting methods II

Operator-Integration-Factor Splitting

Rewrite original problem to

$$\frac{\partial}{\partial t}(Q^{t \rightarrow t^*} u) = Q^{t \rightarrow t^*}(f - Au)$$

with Q the *propagation operator*, s.t. $Q^{t_1 \rightarrow t_2} u_1 = v(t_2)$ with v the solution of the explicit *propagation problem*:

$$\frac{\partial}{\partial s} v = -C(v) v \quad v(t_1) = u_1$$

[Maday, Patera, Rønquist, '90]

First order example: Implicit Euler for implicit problem

Choose $t^* = t^{n+1}$ and replace $\frac{\partial}{\partial t}$ by a 1st order backward difference:

$$\frac{1}{\Delta t}(u^{n+1} - Q^{t^n \rightarrow t^{n+1}} u^n) = (f^{n+1} - Au^{n+1})$$

Operator splitting: Multiplicative splitting methods III

First order example: Implicit Euler for implicit problem

Choose $t^* = t^{n+1}$ and replace $\frac{\partial}{\partial t}$ by a 1st order backward difference:

$$\frac{1}{\Delta t}(u^{n+1} - Q^{t^n \to t^{n+1}} u^n) = (f^{n+1} - A u^{n+1})$$

Algorithm

1. Propagate

$$\bar{u}^n = w(t^{n+1}), \quad \frac{\partial}{\partial s} w = -C(w)w, \quad w(t^n) = u^n$$

2. Solve

$$(I + \Delta t A)u^{n+1} = \bar{u}^n + \Delta t f^{n+1}$$

Generalizations

This approach is applicable for other implicit time integration methods as well.

Operator splitting: Multiplicative splitting methods IV

Dis-/Advantages of multiplicative splitting

- (+) avoids implicit solutions with $C(u)$
- (+) time steps for implicit problem and explicit problem are decoupled
- (+) allows for discretizations which are separately taylored for the explicit and the implicit problem
- (+) as long as the explicit *propagation problem* is solved in a stable manner, the overall scheme is stable
- (-) introduces an additional consistency error (splitting error)

Pseudo-implicit schemes I

Idea:

Decompose the problem into one part with expl. A und impl. C and another with explicit C and implicit A. Then solve the equation which involves A implicitely by means of an iteration involving only explicit evaluations of A.

First order method

$$\begin{aligned}
\left(\frac{2}{\Delta t} + C(u^{n+\frac{1}{2}})\right) u^{n+\frac{1}{2}} &= f^{n+\frac{1}{2}} + \left(\frac{2}{\Delta t} - A\right) u^{n} =: g^{n+\frac{1}{2}} \\
\left(\frac{2}{\Delta t} + A\right) u^{n+1} &= f^{n+1} + \left(\frac{2}{\Delta t} - C(u^{n+\frac{1}{2}})\right) u^{n+\frac{1}{2}}
\end{aligned}$$

where the first equation is solved by pseudo-time-stepping methods (or iterative methods):

$$\frac{\partial}{\partial t} w = g^{n+\frac{1}{2}} - \left(\frac{2}{\Delta t} + C(w)\right) w$$

Pseudo-implicit schemes II

Dis-/Advantages of Pseudo-implicit schemes

(+) consistent

(+) time steps for implicit problem and explicit problem are decoupled

(+) allows for discretizations which are separately taylored for the explicit and the implicit problem

(-) indirect implicit solutions with $C(u)$ is typically more expensive as evaluation or propagation

Consequences for spatial discretization

Opportunities for and requirements on spatial discretization

- For additive splitting methods one common space discretization is appropriate as no "explicit time stepping" is done.
- For multiplicative and pseudo-implicit schemes separate spatial discretizations are possible as long as they provide reasonable translation operations from one setting to the other.

Applying the ideas to a scalar convection diffusion equation

Operator splitting for a scalar convection diffusion equation

$$\frac{\partial}{\partial t} u + \operatorname{div}\left(-\alpha \nabla u + b u\right) = f \quad + \quad b.c., \quad \operatorname{div} b = 0$$

Discretization implicit part (diffusion)

- Hybrid Discontinuous Galerkin Interior Penalty formulation

Discretization explicit part (convection)

- Discontinuous finite elements
- Discontinuous Galerkin Upwind formulation

Applying the ideas to incompressible Navier Stokes

Operator splitting for incompressible Navier Stokes (DAE)

$$\begin{array}{rcl} \frac{\partial}{\partial t} u + \operatorname{div}(-\nu \nabla u + u \otimes u + pI) & = & f \\ \operatorname{div}\ u & = & 0 \end{array} \quad + \quad b.c.$$

Note that there is no away around treating the algebraic constraint implicitely (in our operator splitting context)!

Discretization implicit part (viscosity,pressure, incompressibility)

- $H(\text{div})$-conforming Hybrid Discontinuous Galerkin IP formulation

Discretization explicit part (convection)

- Discontinuous Galerkin Upwind formulation
- Take extrapolated velocity from implicit stages (linearization argument of the explicit operator $C(\cdot)$ no longer depends on the explicit problem but is known) $\Rightarrow$ ensures divergencefree-constraint for the convective velocity

Boussinesq-Approximation

Boussinesq's assumptions

changes in density are small:

- incompressibility model is still acceptable
- changes in density just cause some buoyancy forces

$$f = g \quad \rightarrow \quad (1 - \beta(T - T_0))g \qquad \beta: \text{ heat expansion coefficient}$$

Operator splitting for Boussinesq-Equation

$$\begin{aligned} \frac{\partial}{\partial t} u + \operatorname{div}(-\nu \nabla u + u \otimes u + pI) + \beta \rho T g &= (1 + \beta \rho T_0) g \\ \operatorname{div} u &= 0 \qquad\qquad + b.c. \& i.c. \\ \frac{\partial}{\partial t} T + \operatorname{div}(-\alpha \nabla T + u T) &= 0 \end{aligned}$$

- weak coupling
- Note that for the implicit problem Stokes and temperature part decouple.
- Discretization: Navier Stokes + scalar convection diffusion equation

Overview

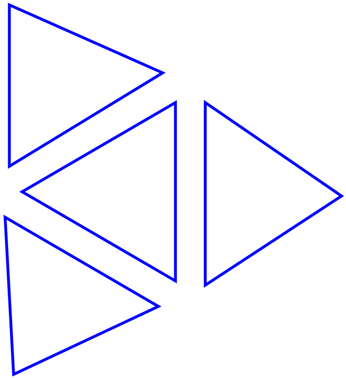
Discretization space for explicit problem

Trial functions are

- (element-)piecewise polynomials
- discontinuous

With appropriate formulations we get

- couplings only between neighbouring (in the sense of shared facets) elements
- one elements contribution just need the information of 4 (2D), 5 (3D) elements, easy to parallize.



DG formulation for the operator div(b u) (div(b) = 0)

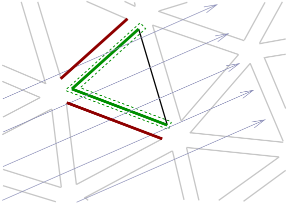
Numerical flux: Upwind

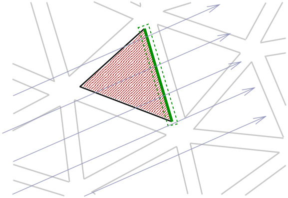
Testing against v and doing partial integration on each element results in

$$\int_\Omega \operatorname{div}(bu)\ v\ dx = \sum_T \left\{ -\int_T bu\ \nabla v\ dx + \int_{\partial T} b_n u^? v\ ds \right\}$$

DG flexibility: we use upwinding to replace $u^?$ by

$$u^{up} = \begin{cases} u_{neighbour} & \text{if } b_n \leq 0 \\ u & \text{if } b_n > 0 \end{cases}$$

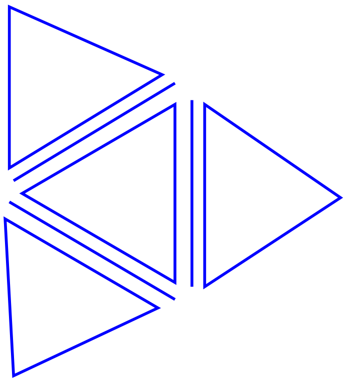




Discretization space for implicit problem

Trial functions are

- discontinuous
- piecewise polynomials
- on each facet ($O(p^{d-1})$ `dof`) and element ($O(p^d)$ `dof`)



With appropriate formulations we get

- more unknowns but typically less matrix entries
- implementation fits into standard element-based assembling
- structure allows for static condensation of element unknowns

HDG formulation for $-\Delta u$

Derivation

Integrating against v

$$\int_\Omega (-\Delta u) v \; dx \quad = \quad \sum_T \int_T \nabla u \; \nabla v \; dx - \int_{\partial T} \frac{\partial u}{\partial n} \; v \; ds$$

HDG formulation for $-\Delta u$

Derivation

manipulating by adding a consistent term

$$\int_\Omega (-\Delta u) v \; dx \quad \to \quad \sum_T \int_T \nabla u \; \nabla v \; dx - \int_{\partial T} \frac{\partial u}{\partial n} \; (v - v_F) \; ds$$

where we use

$$\int_{\partial T^+} \frac{\partial u}{\partial n} v_F \; ds + \int_{\partial T^-} \frac{\partial u}{\partial n} v_F \; ds = 0$$

for the exact solution u on inner facets.

HDG formulation for $-\Delta u$

Derivation

symmetrizing

$$\int_\Omega (-\Delta u) v \, dx \quad \rightarrow \quad \sum_T \int_T \nabla u \, \nabla v \, dx - \int_{\partial T} \frac{\partial u}{\partial n} \, (v - v_F) \, ds$$
$$- \int_{\partial T} \frac{\partial v}{\partial n} \, (u - u_F) \, ds$$

where we use $$u - u_F = 0$$

for the exact solution u on facets.

HDG formulation for $-\Delta u$

Derivation

stabilizing

$$\int_\Omega (-\Delta u) v \, dx \quad \rightarrow \quad \sum_T \int_T \nabla u \, \nabla v \, dx - \int_{\partial T} \frac{\partial u}{\partial n} (v - v_F) \, ds$$

$$- \int_{\partial T} \frac{\partial v}{\partial n} (u - u_F) \, ds + \int_{\partial T} \tau_h (u - u_F) \, (v - v_F) \, ds$$

where we use

$$u - u_F = 0$$

for the exact solution u on facets.

The stabilization parameter τ_h has to scale correctly, i.e. $\tau_h \sim \frac{p^2}{h}$

HDG formulation for $-\Delta u$

Derivation

manipulating by adding a consistent term, symmetrizing, stabilizing

$$\int_\Omega (-\Delta u) v \; dx \quad \rightarrow \quad \sum_T \int_T \nabla u \; \nabla v \; dx - \int_{\partial T} \frac{\partial u}{\partial n} \; (v - v_F) \; ds$$

$$- \int_{\partial T} \frac{\partial v}{\partial n} \; (u - u_F) \; ds + \int_{\partial T} \tau_h (u - u_F) \; (v - v_F) \; ds$$

where we use

$$u - u_F = 0$$

for the exact solution u on facets.

The stabilization parameter τ_h has to scale correctly, i.e. $\tau_h \sim \frac{p^2}{h}$

Properties

The formulation is consistent, conservative, stable and optimally convergent.

This and other *hybridizations* of CG, mixed and DG methods were discussed in [Cockburn+Gopalakrishnan+Lazarov,'08]

HDG formulation for $-\Delta u$

Derivation

$$\int_\Omega (-\Delta u) v \; dx \quad \rightarrow \quad \sum_T \int_T \nabla u \; \nabla v \; dx - \int_{\partial T} \frac{\partial u}{\partial n} \, (v - v_F) \; ds - \int_{\partial T} \frac{\partial v}{\partial n} \, (u - u_F) \; ds + \int_{\partial T} \tau_h (u - u_F) \, (v - v_F) \; ds$$

where we use $u - u_F = 0$

for the exact solution u on facets.

The stabilization parameter τ_h has to scale correctly, i.e. $\tau_h \sim \frac{p^2}{h}$

Hybrid upwinding

Hybrid upwinding (for implicit handling), see [Egger+Schöberl, '09]

Projected jumps

Suboptimality

W.r.t. the facet approximation the solution quality of u is suboptimal. Hybrid mixed methods achieve order $k+1$ approximations in the volume while using order k approximations on the facet.

Modifying the formulation

If we denote the L^2-projection on the polynomial space of degree $k-1$ on a facet as Π_F you can also use:

$$\int_\Omega (-\Delta u) v \; dx \quad \rightarrow \quad \sum_T \int_T \nabla u \; \nabla v \; dx - \int_{\partial T} \frac{\partial u}{\partial n} \; \Pi_F(v - v_F) \; ds$$
$$- \int_{\partial T} \frac{\partial v}{\partial n} \; \Pi_F(u - u_F) \; ds$$
$$+ \int_{\partial T} \tau_h \Pi_F(u - u_F) \; \Pi_F(v - v_F) \; ds$$

$\Rightarrow$ Now only $\Pi_F v_F$ is involved instead of v_F, s.t. we can reduce the polynomial degree of the facet functions.

Overview

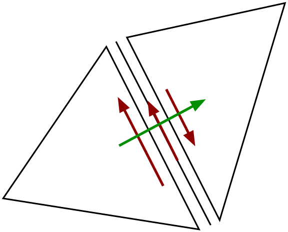
H(div)-conforming elements for (Navier) Stokes

[Cockburn, Kanschat, Schötzau, 2005]

DG methods for the incompressible Navier-Stokes equations cannot be both *locally conservative* as well as *energy-stable* unless the approximation to the convective velocity is exactly divergence-free.

Trial functions

- normal-continuous, tangential-discontinuous velocity element functions, piecewise polynomial (degree k)
- facet velocity functions for the tangential component only, piecewise polynomial (degree k / degree $k-1$)
- discontinuous element pressure functions, piecewise polynomial (degree $k-1$)



Hybrid DG - DG Navier Stokes bilinearforms

- unknowns u in elements ($H(\text{div})$-conforming)
- unknowns for the tangential component u_F^t on facets
- unknowns for pressure p on each element

Viscosity: $[\![v^t]\!] := v^t - v_F^t$

$$A\big((u,u_F),(v,v_F)\big) = \sum_T \Big\{ \int_T \nu \, \nabla u : \nabla v \; dx - \int_{\partial T} \nu \, \frac{\partial u}{\partial n} \, \Pi_F [\![v^t]\!] \; ds - \int_{\partial T} \nu \, \frac{\partial v}{\partial n} \, \Pi_F [\![u^t]\!] \; ds + \int_{\partial T} \nu \tau_h \Pi_F [\![u^t]\!] \cdot \Pi_F [\![v^t]\!] \; ds \Big\}$$

Convection:

$$C\big(w; u, v\big) = \sum_T \Big\{ \int_T u \otimes w : \nabla v \; dx - \int_{\partial T} w_n u^{up} v \; ds \Big\}$$

pressure / incompressibility constraint:

$$D\big((u,u_F),q\big) = \sum_T \int_T \text{div}(u) q \; dx$$

weak incompressibility ($+$ $H(\text{div})$-conformity)

$\Rightarrow$ exactly divergence-free solutions

Comparison Std. DG, NodalDG, HDG, p. HDG

Test problem

$H(\mathrm{div})$-conforming finite element space, vector-valued $[H^1]^d$ problem:

$$-\Delta u + u = f$$

$$A(u,v) + M(u,v) \; = \; (f,v)_{L_2}$$

with $A(\cdot,\cdot)$ a Std. DG, a HDG or a p. HDG formulation, considering a modal and a special nodal DG basis.
System matrix B is s.p.d.. Use sparse direct solver to get decomposition

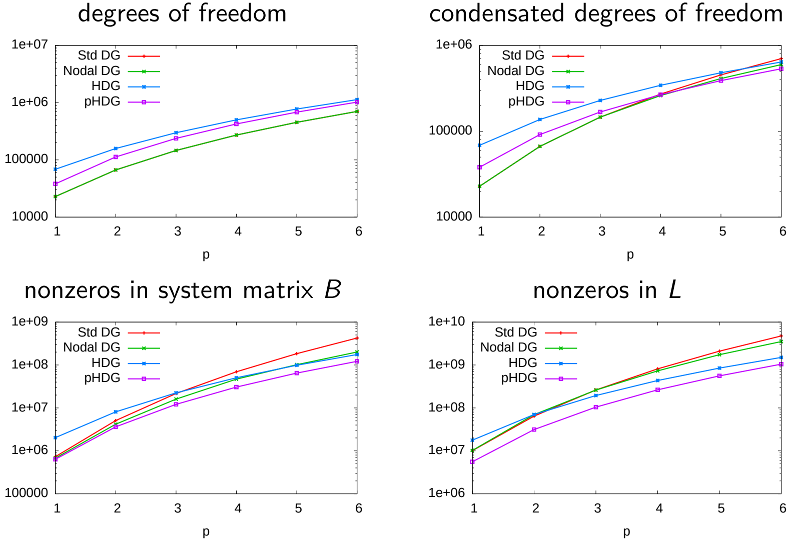
$$B = L^T L$$

Numbers of interest

- number of degrees of freedom
- statically condensated number of degrees of freedom
- nonzeros in system matrix B
- nonzeros in L

Comparison Std. DG, NodalDG, HDG, p. HDG

3D mesh, 2022 elements

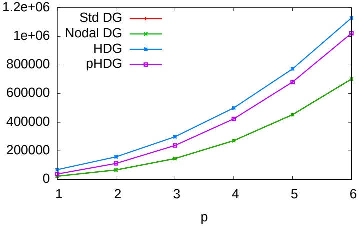


log-scaled

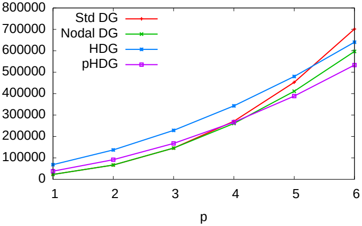
Comparison Std. DG, NodalDG, HDG, p. HDG

3D mesh, 2022 elements

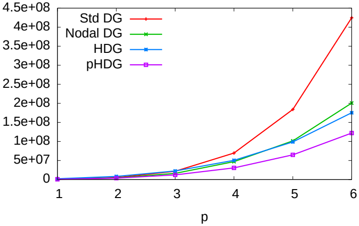
degrees of freedom



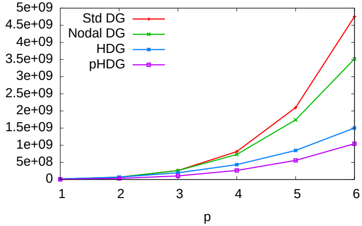
condensated degrees of freedom



nonzeros in system matrix B



nonzeros in L



Ingredients and properties of spat. disc.

Ingredients

We have presented a new Finite Element Method for Navier Stokes, with

- $H(\text{div})$-conformity resulting in exactly divergence-free solutions
- Hybrid Discontinuous Galerkin Method for viscous terms
- Upwind flux for the convection term

Properties

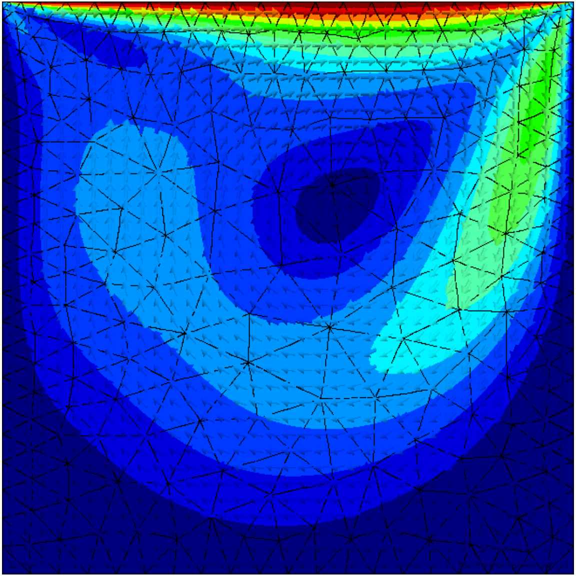
This discretization leads to solutions, which are

- locally conservative (mass and momentum)
- energy-stable ($\frac{\partial}{\partial t}\|u\|_{L_2}^2 \leq \frac{C}{\nu}\|f\|_{L_2}^2$)
- exactly incompressible
- static condensation
- standard finite element assembly
- less matrix entries than for std. DG approaches
- (reduced basis possible)

Overview

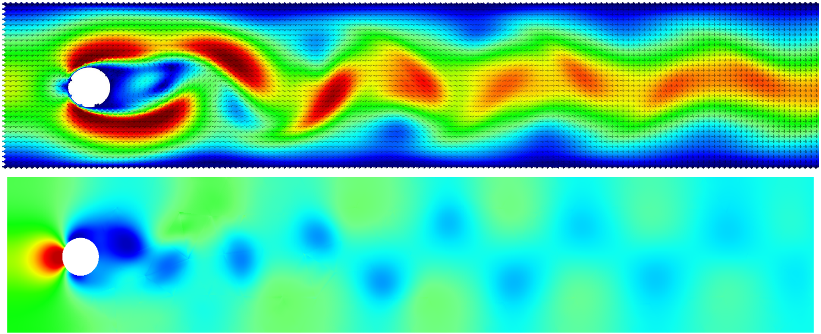
Examples: Steady Navier Stokes

2D Driven Cavity ($u_{top} = 0.25$, $\nu = 10^{-3}$)

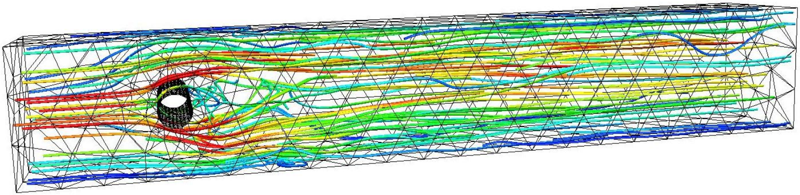


Examples: Unsteady Navier Stokes

2D laminar flow around a disk (Re=100):



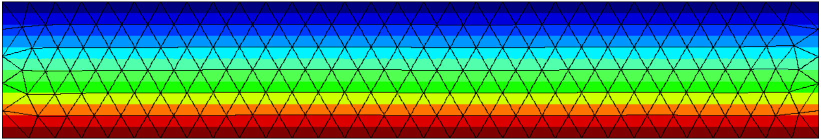
3D laminar flow around a cylinder (Re=100):



Examples: Heat driven flow

Benard-Rayleigh example:
Top temperature: constant $20°C$
Bottom temperature: constant $20.5°C$

Initial mesh and initial condition ($p = 5$):



Overview

Conclusion / Software

Discussed

- Time integration approaches suitable for incompressible Navier Stokes problems
- Spatial discretization (DG/HDG, $H(\mathrm{div})$-conforming) taylored for explicit/implicit problems

Software

- `Netgen`
- `NGSolve` (including the presented scalar HDG methods),
- `ngsflow` (including all other presented methods)

You can find us at sourceforge.

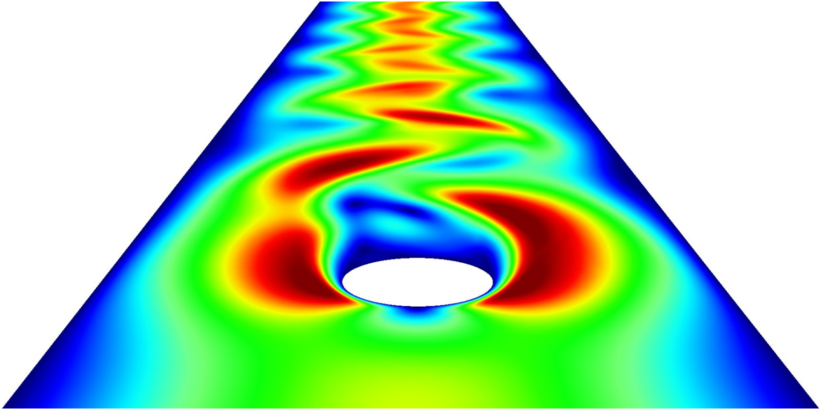
Ongoing work / work todo

explicit problem

- Google Summer of Code: explicit problems on GPU
- local time stepping
- (practically) reliable estimation of a stable time step for the explicit problem

implicit problem

- nice preconditioners (BDDC?) to go to large problems



Thank you for your attention!