Discontinuous Galerkin Finite Element Methods for Compressible Flows

Jaap van der Vegt
Numerical Analysis and Computational Mechanics Group
Department of Applied Mathematics
University of Twente

Joint Work with
Sander Rhebergen, Onno Bokhove, Christiaan Klaij, Fedderik van der Bos

Toulouse, December 8, 2009
Introduction

Motivation of research:

- Many important flow phenomena are modelled by nonconservative (hyperbolic) partial differential equations, e.g. dispersed multiphase flows.
- Our aim is to develop space-time discontinuous Galerkin discretizations which are suitable for both conservative and nonconservative partial differential equations.
Introduction

Motivation of research:

- Nonconservative hyperbolic partial differential equations contain nonconservative products

\[ \partial_x u + A(u) \partial_x u = 0 \]

- The essential feature of nonconservative products is that \( A \neq Df \), hence \( A \) is not the Jacobian matrix of a flux function \( f \).

- This causes problems once the solution becomes discontinuous, because the weak solution in the classical sense of distributions then does not exist.

- This also complicates the derivation of discontinuous Galerkin discretizations since there is no direct link with a Riemann problem.
- **Alternative:** use the theory for nonconservative products from Dal Maso, LeFloch and Murat (DLM)
Overview of Presentation

- Overview of main results of the theory of Dal Maso, LeFloch and Murat for nonconservative products
- Space-time DG discretization of nonconservative hyperbolic partial differential equations
- Extension to the compressible Navier-Stokes equations
- Multigrid techniques for space-time DG discretizations
- Conclusions
Nonconservative Products

• Consider the function $u(x)$

$$u(x) = u_L + \mathcal{H}(x - x_d)(u_R - u_L), \quad x, x_d \in ]a, b[,$$

with $\mathcal{H} : \mathbb{R} \to \mathbb{R}$ the Heaviside function.

• For any smooth function $g : \mathbb{R}^m \to \mathbb{R}^m$ the product $g(u)\partial_x u$ is not defined at $x = x_d$ since here $|\partial_x u| \to \infty$.

• Introduce a smooth regularization $u^\varepsilon$ of $u$. If the total variation of $u^\varepsilon$ remains uniformly bounded with respect to $\varepsilon$ then Dal Maso, LeFloch and Murat (DLM) showed that

$$g(u)\frac{du}{dx} \equiv \lim_{\varepsilon \to 0} g(u^\varepsilon)\frac{du^\varepsilon}{dx}$$

gives a sense to the nonconservative product as a bounded measure.
Effect of Path on Nonconservative Product

The limit of the regularized nonconservative product depends in general on the path used in the regularization.

- Introduce a Lipschitz continuous path $\phi : [0, 1] \rightarrow \mathbb{R}^m$, satisfying $\phi(0) = u_L$ and $\phi(1) = u_R$, connecting $u_L$ and $u_R$ in $\mathbb{R}^m$.

- The following regularization $u^\varepsilon$ for $u$ then emerges:

$$u^\varepsilon(x) = \begin{cases} 
  u_L, & \text{if } x \in ]a, x_d - \varepsilon[, \\
  \phi\left(\frac{x - x_d + \varepsilon}{2\varepsilon}\right), & \text{if } x \in ]x_d - \varepsilon, x_d + \varepsilon[, \\
  u_R, & \text{if } x \in ]x_d + \varepsilon, b[
\end{cases}$$
• When $\varepsilon$ tends to zero, then:

$$g(u^\varepsilon) \frac{du^\varepsilon}{dx} \rightharpoonup C \delta_{x_d}, \text{ with } C = \int_0^1 g(\phi(\tau)) \frac{d\phi}{d\tau}(\tau) \, d\tau,$$

weakly in the sense of measures on $]a, b[$, where $\delta_{x_d}$ is the Dirac measure at $x_d$.

• The limit of $g(u^\varepsilon) \partial_x u^\varepsilon$ depends on the path $\phi$.

• There is one exception, namely if an $q : \mathbb{R}^m \to \mathbb{R}$ exists with $g = \partial_u q$. In this case $C' = q(u_R) - q(u_L)$. 
Dal Maso, LeFloch and Murat provided a general theory for nonconservative hyperbolic pde's.

- Introduce the Lipschitz continuous maps \( \phi : [0, 1] \times \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^m \) which satisfy the following properties:

  \((H1)\) \( \phi(0; u_L, u_R) = u_L, \phi(1; u_L, u_R) = u_R, \)

  \((H2)\) \( \phi(\tau; u_L, u_L) = u_L, \)

  \((H3)\) \( |\frac{\partial \phi}{\partial \tau}(\tau; u_L, u_R)| \leq K |u_L - u_R|, \text{ a.e. in } [0, 1]. \)
Theorem (DLM). Let $u : ]a, b[ \to \mathbb{R}^m$ be a function of bounded variation and $g : \mathbb{R}^m \to \mathbb{R}^m$ a continuous function. Then, there exists a unique real-valued bounded Borel measure $\mu$ on $]a, b[$ with:

1. If $u$ is continuous on a Borel set $B \subset ]a, b[$, then
   $$\mu(B) = \int_B g(u) \frac{du}{dx}$$

2. If $u$ is discontinuous at a point $x_d$ of $]a, b[$, then
   $$\mu(\{x_d\}) = \int_0^1 g(\varphi(\tau; u_L, u_R)) \frac{\partial \phi}{\partial \tau}(\tau; u_L, u_R) \, d\tau.$$

By definition, this measure $\mu$ is the nonconservative product of $g(u)$ by $\partial_x u$ and denoted by $\mu = \left[ g(u) \frac{du}{dx} \right] \phi.$
Rankine-Hugoniot Relations

- For conservative hyperbolic system of pde’s, \( \partial_x u + \partial_x f(u) = 0 \) the Rankine-Hugoniot relations across a jump with \( u^L \) and \( u^R \) and velocity \( v \) are equal to
  \[
  -v(u^R - u^L) + f(u^R) - f(u^L) = 0.
  \]

- For a nonconservative hyperbolic pde \( \partial_x u + A(u) \partial_x u = 0 \) the Rankine-Hugoniot relations in the DLM theory are equal to
  \[
  -v(u^R - u^L) + \int_0^1 A(\phi_D(s; u^L, u^R)) \partial_s \phi_D(s; u^L, u^R) \, ds = 0
  \]
  with \( \phi_D \) a Lipschitz continuous path satisfying \( \phi_D(0; u_L, u_R) = u_L \) and \( \phi_D(1; u_L, u_R) = u_R \).

- The Rankine-Hugoniot relations are essential for the definition of the NCP flux used in the DG discretization.
A time-dependent problem is considered directly in four dimensional space, with time as the fourth dimension.
Key Features of Space-Time Discontinuous Galerkin Methods

- Simultaneous discretization in space and time: time is considered as a fourth dimension.

- Discontinuous basis functions, both in space and time, with only a weak coupling across element faces resulting in an extremely local, element based discretization.

- The space-time DG method is closely related to the Arbitrary Lagrangian Eulerian (ALE) method.
Benefits of Discontinuous Galerkin Methods

- Due to the extremely local discretization DG methods provide optimal flexibility for
  - achieving higher order accuracy on unstructured meshes
  - $hp$-mesh adaptation
  - unstructured meshes containing different types of elements, such as tetrahedra, hexahedra and prisms
  - parallel computing
Benefits of Space-Time Discontinuous Galerkin Methods

- A conservative discretization is obtained on moving and deforming meshes.
- No data interpolation or extrapolation is necessary on dynamic meshes, at free boundaries and after mesh adaptation.
Disadvantages of Space-(Time) Discontinuous Galerkin Methods

- Algorithms are generally rather complicated, in particular for elliptic and parabolic partial differential equations.
- On structured meshes DG methods are computationally more expensive than finite difference and finite volume methods.
Space-Time Domain

- Consider an open domain: $\mathcal{E} \subset \mathbb{R}^d$.
- The flow domain $\Omega(t)$ at time $t$ is defined as:
  \[ \Omega(t) := \{ x \in \mathcal{E} \mid x_0 = t, \ t_0 < t < T \} \]
- The space-time domain boundary $\partial \mathcal{E}$ consists of the hypersurfaces:
  \[ \Omega(t_0) := \{ x \in \partial \mathcal{E} \mid x_0 = t_0 \}, \]
  \[ \Omega(T) := \{ x \in \partial \mathcal{E} \mid x_0 = T \}, \]
  \[ Q := \{ x \in \partial \mathcal{E} \mid t_0 < x_0 < T \}. \]
- The space-time domain is covered with a tessellation $\mathcal{T}_h$ consisting of space-time elements $\mathcal{K}$. 
**Discontinuous Finite Element Approximation**

- The finite element space associated with the tessellation $T_h$ is given by:

$$W_h := \left\{ W \in (L^2(\mathcal{E}_h))^m : W|_K \circ G_K \in (P^k(\hat{K}))^m, \quad \forall K \in T_h \right\}$$

- The jump of $f$ at an internal face $S \in S^n_I$ in the direction $k$ of a Cartesian coordinate system is defined as:

$$
\left[ f \right]_k = f_L \bar{n}_L^k + f_R \bar{n}_R^k,
$$

with $\bar{n}_R^k = -\bar{n}_L^k$.

- The average of $f$ at $S \in S^n_I$ is defined as:

$$\left\{ f \right\} = \frac{1}{2}(f_L + f_R).$$
Space-Time DG Formulation of Nonconservative Hyperbolic PDE’s

• Consider the nonlinear hyperbolic system of partial differential equations in nonconservative form in multi-dimensions:

\[
\frac{\partial U_i}{\partial t} + \frac{\partial F_{ik}}{\partial x_k} + G_{ikr} \frac{\partial U_r}{\partial x_k} = 0, \quad \bar{x} \in \Omega \subset \mathbb{R}^q, \ t > 0,
\]

with \( U \in \mathbb{R}^m, \ F \in \mathbb{R}^m \times \mathbb{R}^q, \ G \in \mathbb{R}^m \times \mathbb{R}^q \times \mathbb{R}^m \)

• These equations model for instance bubbly flows, granular flows, shallow water equations and many other physical systems.
Weak formulation for nonconservative hyperbolic system:

Find a $U \in V_h$, such that for $V \in V_h$ the following relation is satisfied

$$
\sum_{K \in T_h} \int_K V_i (U_{i,0} + F_{ik,k} + G_{ikr} U_{r,k}) \, dK \\
+ \sum_{K \in T_h} \left( \int_{K(t_{n+1}^-)} \hat{V}_i (U_i^R - U_i^L) \, dK - \int_{K(t_{n+1}^+)} \hat{V}_i (U_i^R - U_i^L) \, dK \right) \\
+ \sum_{S \in S_I} \int_S \hat{V}_i \left( \int_0^1 G_{ikr} (\phi(\tau; U_L^i, U_R^i)) \frac{\partial \phi_r}{\partial \tau}(\tau; U_L^i, U_R^i) \, d\tau \right) \, dS \\
- \sum_{S \in S_I} \int_S \hat{V}_i [F_{ik} - v_k U_i]_k \, dS = 0
$$
Relation with Space-Time DG Formulation of Conservative Hyperbolic PDE’s

- **Theorem 2.** If the numerical flux \( \hat{V} \) for the test function \( V \) is defined as:

\[
\hat{V} = \begin{cases} 
\{ \{ V \} \} & \text{at } S \in S_I, \\
0 & \text{at } K(t_n) \subset \Omega_{h(t_n)} \quad \forall n \geq 0,
\end{cases}
\]

then the DG formulation will reduce to the conservative space-time DG formulation when there exists a \( Q \), such that \( G_{ikr} = \partial Q_{ik}/\partial U_r \).
After the introduction of the numerical flux $\hat{V}$ we obtain the weak formulation:

$$
\sum_{K \in T_h} \int_K \left( - V_{i,0} U_i - V_{i,k} F_{ik} + V_i G_{ikr} U_{r,k} \right) dK \\
+ \sum_{K \in T_h} \left( \int_{K(t_{n+1})^-} V_i^L U_i^L dK - \int_{K(t_n^+)} V_i^L U_i^L dK \right) \\
+ \sum_{S \in S_I} \int_S (V_i^L - V_i^R) \{ \{ F_{ik} - v_k U_i \} \} \bar{n}_k^L dS \\
+ \sum_{S \in S_B} \int_S V_i^L (F_{ik}^L - v_k U_i^L) \bar{n}_k^L dS \\
+ \sum_{S \in S_I} \int_S \{ \{ V_i \} \} \left( \int_0^1 G_{ikr}(\phi(\tau; U^L, U^R)) \frac{\partial \phi}{\partial \tau}(\tau; U^L, U^R) d\tau \right) \bar{n}_k^L dS = 0
$$
Numerical Fluxes

- The fluxes at the element faces do not contain any stabilizing terms yet, both for the conservative and nonconservative part.

- At the time faces, the numerical flux is selected such that causality in time is ensured:
  \[
  \hat{U} = \begin{cases} 
  U_L & \text{at } K(t_{n+1}^-) \\
  U_R & \text{at } K(t_n^+) 
  \end{cases}
  \]

- The space-time DG formulation is stabilized using the NCP (Non-Conservative Product) flux:
  \[
  \hat{P}_{i}^{nc} = (\{ F_{ik} - \nu_k U_i \} + P_{ik}) \vec{n}_k^L
  \]
Nonconservative Product Flux

Wave pattern of the solution for the Riemann problem
NCP Flux

Main steps in derivation of NCP flux:

• Consider the nonconservative hyperbolic system:
\[
\partial_t U + \partial_x F(U) + G(U) \partial_x U = 0,
\]

• Introduce the averaged exact solution \( \bar{U}_{LR}^*(T) \) as:
\[
\bar{U}_{LR}^*(T) = \frac{1}{T(S_R - S_L)} \int_{T_{SL}}^{T_{SR}} U(x, T) \, dx.
\]

• Apply the Gauss theorem over each subdomain \( \Omega_1, \cdots, \Omega_4 \) and connect each subdomain using the generalized Rankine-Hugoniot relations.
• The NCP-flux is then given by:

\[
\hat{P}_{nc}^i(U_L, U_R, v, \bar{n}^L) = \begin{cases} 
F_{ik}^L \bar{n}_k^L - \frac{1}{2} \int_0^1 G_{ikr}(\bar{\phi}(\tau; U_L, U_R)) \frac{\partial \bar{\phi}_r}{\partial \tau}(\tau; U_L, U_R) d\tau \bar{n}_k^L \\
\{ F_{ik} \} \bar{n}_k^L + \frac{1}{2} ((S_R - v)\bar{U}_i^* + (S_L - v)\bar{U}_i^* - S_L U_i^L - S_R U_i^R) \\
F_{ik}^R \bar{n}_k^L + \frac{1}{2} \int_0^1 G_{ikr}(\bar{\phi}(\tau; U_L, U_R)) \frac{\partial \bar{\phi}_r}{\partial \tau}(\tau; U_L, U_R) d\tau \bar{n}_k^L 
\end{cases}
\]

if \( S_L > v \),

if \( S_L < v < S_R \),

if \( S_R < v \),

• Note, if \( G \) is the Jacobian of some flux function \( Q \), then \( \hat{P}_{nc}^i(U_L, U_R, v, \bar{n}^L) \) is exactly the HLL flux derived for moving grids in van der Vegt and van der Ven (2002).
Efficient Solution of Nonlinear Algebraic System

- The space-time DG discretization results in a large system of nonlinear algebraic equations:

\[ \mathcal{L}(\hat{U}^n; \hat{U}^{n-1}) = 0 \]

- This system is solved by marching to steady state using pseudo-time integration and multigrid techniques:

\[ \frac{\partial \hat{U}}{\partial \tau} = -\frac{1}{\Delta t} \mathcal{L}(\hat{U}; \hat{U}^{n-1}) \]
The dimensionless depth-averaged two fluid model of Pitman and Le, ignoring source terms for simplicity, can be written as:

$$\partial_t U + \partial_x F + G \partial_x U = 0,$$

where:

$$U = \begin{bmatrix} h(1 - \alpha) \\ h \alpha \\ h \alpha v \\ hu(1 - \alpha) \\ b \end{bmatrix}, \quad F = \begin{bmatrix} h(1 - \alpha)u \\ h \alpha v \\ h \alpha v^2 + \frac{1}{2} \varepsilon (1 - \rho) \alpha_x gh^2 \alpha \\ hu^2 + \frac{1}{2} \varepsilon gh^2 \\ 0 \end{bmatrix},$$

$$G(U) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \varepsilon (1 - \rho) \alpha_x gh \alpha + \varepsilon \rho \alpha gh \\ \varepsilon \rho \alpha gh & \varepsilon \rho \alpha gh & 0 & 0 & 0 & 0 \\ 2u - \alpha u^2 - \varepsilon gh \alpha & -\varepsilon gh \alpha - \alpha u^2 & u(\alpha - 1) & u\alpha - \frac{2u\alpha}{1 - \alpha} & (1 - \alpha) \varepsilon gh & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \end{bmatrix}.$$
Steady-state solution for a subcritical two-phase flow (320 cells).

Total flow height $h + b$, flow height due to the fluid phase $h(1 - \alpha)$, flow height due to solids phase $h\alpha$ and the topography $b$. 
Error in \( h(1 - \alpha) + b \), \( h\alpha + b \), \( hu(1 - \alpha) \) and \( hv\alpha \) for subcritical flow over a bump.
Two-phase dam break problem

Two-phase dam break problem at time $t = 0.175$; mesh with 128 elements compared to mesh with 10000 elements.
Effect of Path

(d) The solution on the whole domain. (e) The solution zoomed in on the left shock wave.

Solution of $h(1 - \alpha)$, $h\alpha$, $b$ and $h$ at time $t = 0.175$ calculated on a mesh with 1024 elements using the paths defined by Toumi.
Flow Through a Contraction

Flow depth $h$ of water-sand mixture in a contraction $h/L = 0.01$, $\rho_f/\rho_s = 0.5$, slope $10^\circ$. 
Compressible Navier-Stokes Equations

- Compressible Navier-Stokes equations in space-time domain $\mathcal{E}$:
  \[
  \frac{\partial U_i}{\partial x_0} + \frac{\partial F_k^e(U)}{\partial x_k} - \frac{\partial F_k^e(U, \nabla U)}{\partial x_k} = 0
  \]

- Conservative variables $U \in \mathbb{R}^5$ and inviscid fluxes $F_k^e \in \mathbb{R}^{5 \times 3}$
  \[
  U = \begin{bmatrix} \rho \\ \rho u_j \\ \rho E \end{bmatrix}, \quad F_k^e = \begin{bmatrix} \rho u_k \\ \rho u_j u_k + p\delta_{jk} \\ \rho hu_k \end{bmatrix}
  \]
Compressible Navier-Stokes Equations

- Viscous flux \( F^v \in \mathbb{R}^{5 \times 3} \)

\[
F^v_k = \begin{bmatrix}
0 \\
\tau_{j,k} \\
\tau_{k,j}u_j - q_k
\end{bmatrix}
\]

with the total stress tensor \( \tau \) is defined as:

\[
\tau_{j,k} = \lambda \frac{\partial u_i}{\partial x_i} \delta_{j,k} + \mu \left( \frac{\partial u_j}{\partial x_k} + \frac{\partial u_k}{\partial x_j} \right)
\]

and heat flux vector \( \vec{q} \) is defined as:

\[
q_k = -\kappa \frac{\partial T}{\partial x_k}
\]
Compressible Navier-Stokes Equations

- The viscous flux $F^v$ is homogeneous with respect to the gradient of the conservative variables $\nabla U$:

$$F_{ik}^v(U, \nabla U) = A_{ikrs}(U) \frac{\partial U_r}{\partial x_s}$$

with the homogeneity tensor $A \in \mathbb{R}^{5 \times 3 \times 5 \times 3}$ defined as:

$$A_{ikrs}(U) := \frac{\partial F_{ik}^v(U, \nabla U)}{\partial (\nabla U)}$$

- The system is closed using the equations of state for an ideal gas.
First Order System

- Rewrite the compressible Navier-Stokes equations as a first-order system using the auxiliary variable $\Theta$:

$$
\frac{\partial U_i}{\partial x_0} + \frac{\partial F^e_{ik}(U)}{\partial x_k} - \frac{\partial \Theta_{ik}(U)}{\partial x_k} = 0,
$$

$$
\Theta_{ik}(U) - A_{ikrs}(U) \frac{\partial U_r}{\partial x_s} = 0.
$$
Weak Formulation

- Weak formulation for the compressible Navier-Stokes equations

Find a \( U \in W_h, \Theta \in V_h \), such that for all \( W \in W_h \) and \( V \in V_h \), the following holds:

\[
- \sum_{K \in T_h} \int_K \left( \frac{\partial W_i}{\partial x_0} U_i + \frac{\partial W_i}{\partial x_k} (F_{ik}^e - \Theta_{ik}) \right) dK
\]

\[
+ \sum_{K \in T_h} \int_{\partial K} W_i^L (\hat{U}_i + \hat{F}_{ik}^e - \hat{\Theta}_{ik}) n_k^L d(\partial K) = 0,
\]

\[
\sum_{K \in T_h} \int_K V_{ik} \Theta_{ik} dK = \sum_{K \in T_h} \int_K V_{ik} A_{ikrs} \frac{\partial U_r}{\partial x_s} dK
\]

\[
+ \sum_{K \in T_h} \int_Q V_{ik} A_{ikrs}^L (\hat{U}_r - U_r^L) \bar{n}_s^L dQ
\]
Transformation to Arbitrary Lagrangian Eulerian form

- The space-time normal vector on a grid moving with velocity $\vec{v}$ is:

$$n = \begin{cases} 
(1, 0, 0, 0)^T & \text{at } K(t_{n+1}^-), \\
(-1, 0, 0, 0)^T & \text{at } K(t_{n}^+), \\
(-v_k\vec{n}_k, \vec{n})^T & \text{at } Q^n.
\end{cases}$$

- The boundary integral then transforms into:

$$\sum_{K \in T_h} \int_{\partial K} W_i^L (\hat{U}_i + \hat{F}_{ik}^e - \hat{\Theta}_{ik}) n_k^L \, d(\partial K)$$

$$= \sum_{K \in T_h} \left( \int_{K(t_{n+1}^-)} W_i^L \hat{U}_i \, dK + \int_{K(t_{n}^+)} W_i^L \hat{U}_i \, dK \right)$$

$$+ \sum_{K \in T_h} \int_Q W_i^L (\hat{F}_{ik}^e - \hat{U}_i v_k - \hat{\Theta}_{ik}) n_k^L \, dQ$$
Numerical Fluxes

- The numerical flux $\hat{U}$ at $K(t_{n+1}^-)$ and $K(t_n^+)$ is defined as an upwind flux to ensure causality in time:

$$\hat{U} = \begin{cases} U_L & \text{at } K(t_{n+1}^-), \\ U_R & \text{at } K(t_n^+), \end{cases}$$

- At the space-time faces $Q$ we introduce the HLLC approximate Riemann solver as numerical flux:

$$\bar{n}_k(F_{ik}^e - \hat{U}_i v_k)(U_L, U_R) = H_{i}^{\text{HLLC}}(U_L, U_R, v, \bar{n})$$
ALE Weak Formulation

- The ALE flux formulation of the compressible Navier-Stokes equations transforms now into:

Find a $U \in W_h$, such that for all $W \in W_h$, the following holds:

$$
- \sum_{K \in T^n_h} \int_K \left( \frac{\partial W_i}{\partial x_0} U_i + \frac{\partial W_i}{\partial x_k} (F^e_{ik} - \Theta_{ik}) \right) dK \\
+ \sum_{K \in T^n_h} \left( \int_{K(t_{n+1}^-)} W^L_i U^L_i dK - \int_{K(t_{n+1}^+)} W^L_i U^R_i dK \right) \\
+ \sum_{K \in T^n_h} \int_Q W^L_i (H^{HLLC}_i (U^L, U^R, v, \bar{n}) - \hat{\Theta}_{ik} \bar{n}^L_k) dQ = 0.
$$
Lifting Operator

• Introduce the global lifting operator $\mathcal{R} \in \mathbb{R}^{5 \times 3}$, defined in a weak sense as:
  
  Find an $\mathcal{R} \in V_h$, such that for all $V \in V_h$:
  
  $$
  \sum_{K \in T_h} \int_K V_{ik} \mathcal{R}_{ik} \, dK = \sum_{S \in S_I^n} \int_S \{\{ V_{ik} A_{ikrs} \}\{ U_r \}_s dS \\
  + \sum_{S \in S_B^n} \int_S V_{ik}^L A_{ikrs}^L (U_r^L - U_r^b) \tilde{n}_s^L dS.
  $$

• The weak formulation for the auxiliary variable is now transformed into
  
  $$
  \sum_{K \in T_h} \int_K V_{ik} \Theta_{ik} \, dK = \sum_{K \in T_h} \int_K V_{ik} \left( A_{ikrs} \frac{\partial U_r}{\partial x_s} - \mathcal{R}_{ik} \right) \, dK, \quad \forall V \in V_h.
  $$
Equation

- The primal formulation can be obtained by eliminating the auxiliary variable $\Theta$ using

$$\Theta_{ik} = A_{ikrs} \frac{\partial U_r}{\partial x_s} - R_{ik}, \quad \text{a.e. in } \mathcal{E}_h^n$$

- Note, this is possible since $\nabla_h W_h \subset V_h$
ALE Weak Formulation for Primal Variables

- Recall the ALE flux formulation of the compressible Navier-Stokes equations:

Find a $U \in W_h$, such that for all $W \in W_h$, the following holds:

$$
- \sum_{K \in T_h^n} \int_K \left( \frac{\partial W_i}{\partial x_0} U_i + \frac{\partial W_i}{\partial x_k} (F_{ik}^e - \Theta_{ik}) \right) dK
+ \sum_{K \in T_h^n} \left( \int_{K(t_{n+1}^-)} W_i^L U_i^L dK - \int_{K(t_{n+1}^+)} W_i^L U_i^R dK \right)
+ \sum_{K \in T_h^n} \int_Q W_i^L \left( H_{i}^{\text{HLLC}}(U_i^L, U_i^R, v, \bar{n}) - \hat{\Theta}_{ik} \bar{n}_k^L \right) dQ = 0.
$$
Numerical Fluxes for $\Theta$

- The numerical flux $\hat{\Theta}$ in the primary equation is defined following Brezzi as a central flux $\hat{\Theta} = \{ \{ \Theta \} \}$:

$$\hat{\Theta}_{ik}(U^L, U^R) = \begin{cases} \{ A_{ikrs} \frac{\partial U_r}{\partial x_s} - \eta R^S_{ik} \} & \text{for internal faces}, \\ A^b_{ikrs} \frac{\partial U^b_r}{\partial x_s} - \eta R^S_{ik} & \text{for boundary faces}, \end{cases}$$

- The local lifting operator $R^S \in \mathbb{R}^{5 \times 3}$ is defined as follows: Find an $R^S \in V_h$, such that for all $V \in V_h$:

$$\sum_{K \in T_h} \int_K V_i R^S_{ik} \, dK = \begin{cases} \int_S \{ V_i A_{ikrs} \} [U_r]_s \, dS & \text{for internal faces}, \\ \int_S V_i A^L_{ikrs} (U^L_r - U^b_r) \overline{n}_s \, dS & \text{for external faces}. \end{cases}$$
Delta Wing Simulations

- Simulations of viscous flow about a delta wing with $85^\circ$ sweep angle.

- Conditions

  - Mach number $M = 0.3$
  - Reynolds number $Re = 40.000$
  - Angle of attack $\alpha = 12.5^\circ$.
  - Fine grid mesh 1,600,000 elements, 40,000,000 degrees of freedom
  - Adapted mesh, initial mesh 208,896 elements, after four adaptations 286,416 elements
Delta Wing Simulations

Streaklines and vorticity contours in various cross-sections
Delta Wing Simulations

Impression of the vorticity based mesh adaptation
Delta Wing Simulations

Large eddy simulation of turbulent flow about a delta wing
Computational Efficiency

- Computational efficiency is the key factor limiting industrial applications of higher order accurate discontinuous Galerkin methods in computational fluid dynamics.

- Multigrid methods are good candidates to increase computational efficiency, but need significant improvements for higher order accurate DG discretizations.
Objectives

- Perform a theoretical analysis of multigrid performance for advection dominated flows, in particular for higher order accurate DG discretizations.

- Improve multigrid performance using theoretical analysis tools.

- Test multigrid performance on realistic problems.
Main Components of $h$-Multigrid Algorithm

- Consider a finite sequence $N_c$ of increasingly coarser meshes $\mathcal{M}_{nh}$, $n \in \{1, \cdots, N_c\}$

- Define operators to connect data on the different meshes:
  - restriction operators
    \[
    R_{nh}^{mh} : \mathcal{M}_{nh} \rightarrow \mathcal{M}_{mh}, \quad 1 \leq n < m \leq N_c,
    \]
  - prolongation operators
    \[
    P_{mh}^{nh} : \mathcal{M}_{mh} \rightarrow \mathcal{M}_{nh}, \quad 1 \leq n < m \leq N_c.
    \]
• Use iterative solvers $S_{nh}$ to approximately solve the system of algebraic equations on the various grid levels

$$L_{nh}v_h = f_{nh} \quad \text{on } \mathcal{M}_{nh}$$

• Since, the main effect of the multigrid algorithm should be the damping of high frequency error components $S_{nh}$ is called a smoothing operator.

• Choose a cycling strategy between the different meshes, e.g. V- or W-cycle.
Multigrid error transformation operator for linear problems

- In order to understand the performance of the multigrid algorithm we need to investigate the multigrid error transformation operator.

- The multigrid error transformation operator shows how much the error in the iterative solution of the algebraic system is reduced by one full multigrid cycle.

- We analyze three-level multigrid algorithms for 2D problems to obtain better estimates for the convergence rate.
• Given an initial error $e^A_h$, the error $e^D_h$ after one full multigrid cycle with three grid levels is given by the relation
\[ e^D_h = M^{3g}_h e^A_h \]
with
\[ M^{3g}_h = S^{\nu_2}_h (I_h - P^{2h}_h (I_{2h} - M^{\gamma e}_{2h}) L^{-1}_{2h} R^{2h}_h L_h) S^{\nu_1}_h \]
and
\[ M_{2h} = S^{\nu_4}_{2h} (I_{2h} - P^{2h}_{4h} L^{-1}_{4h} R^{4h}_{2h} L_{2h}) S^{\nu_3}_{2h}. \]

• The properties of the multigrid error transformation operator are analyzed using discrete Fourier analysis.
Three-grid Fourier analysis

Low, medium and high frequencies Fourier modes in three-level multigrid
Multigrid Error Transformation Operator

- The discrete Fourier transform of the error transformation operator for a three-level multigrid cycle \( \hat{M}_h^3(\theta) \in \mathbb{C}^{16m \times 16m} \) which is equal to

\[
\hat{M}_h^3(\theta) = (\hat{S}_h^3(\theta))^{v_2} \left( I^{3g} - \hat{P}_h^3(\theta) \hat{U}^{3g}(\theta; \gamma_c) \hat{Q}_h^3(\theta) \hat{R}_h^3(\theta) \hat{L}_h^3(\theta) \right) (\hat{S}_h^3(\theta))^{v_1} \\
\forall \theta \in \Theta_{4h} \setminus \Psi_{3g}
\]

The matrix \( \hat{U}^{3g}(\theta; \gamma_c) \) is equal to

\[
\hat{U}^{3g}(\theta; \gamma_c) = I^{2g} - (\hat{M}_{2h}^{2g}(2\theta_\beta))^{\gamma_c}.
\]

with

\[
\hat{M}_{2h}^{2g}(2\theta_\beta) = (\hat{S}_{2h}^{2g}(2\theta_\beta))^{v_4} \left( I^{2g} - \hat{P}_{2h}^{2g}(2\theta_\beta) \hat{L}_{4h}^{-1}(4\theta_0^0) \hat{R}_{2h}^{2g}(2\theta_\beta) \hat{L}_{2h}^{2g}(2\theta_\beta) \right) (\hat{S}_{2h}^{2g}(2\theta_\beta))^{v_3}
\]
Asymptotic Convergence Rate

The asymptotic convergence factor per cycle is defined as

$$\mu = \lim_{m \to \infty} \left( \sup_{e_h^{(0)} \neq 0} \frac{\|e_h^{(m)}\|_{\ell^2(G_h)}}{\|e_h^{(0)}\|_{\ell^2(G_h)}} \right)^{\frac{1}{m}}$$

and can be rewritten into

$$\mu = \rho(M_h^{ng}).$$
Optimization of multigrid performance

- The spectral radius of the error transformation operator can be used to optimize the multigrid algorithm.

- In particular, the smoother is a good candidate for optimization since in general it contains a number of free parameters.

- We will consider a Runge-Kutta type smoother.
Analysis and Optimization of Runge-Kutta Smoothers for the Advection-Diffusion Equation

- The advection-diffusion equation in the domain $\Omega$ is defined as

$$\frac{\partial u}{\partial t} + a \cdot \nabla u = \nabla \cdot (A \nabla u)$$

- The equations are discretized with a higher order accurate space-time discontinuous Galerkin finite element method.

- The space-time discretizations are solved using a multigrid algorithm with a Runge-Kutta type smoother.
Pseudo-time Runge-Kutta smoothers

- Let the system of algebraic equations be denoted as
  
  \[ \mathcal{L}(\hat{u}^n, \hat{u}^{n-1}) = 0. \]

- A pseudo time derivative is added to the system and integrated to steady-state in pseudo-time
  
  \[ \mathcal{M} \frac{\partial \hat{u}^*}{\partial \tau} = -\mathcal{L}(\hat{u}^*, \hat{u}^{n-1}), \]

- At steady state \( \hat{u}^n = \hat{u}^* \).

- For the pseudo-time integration we consider 4- and 5-stage Runge-Kutta methods.
A four-stage Runge-Kutta scheme is given by:

\[
(1 + \beta_1 \lambda I) \hat{V}^1 = \hat{V}^0 - \frac{1}{\Delta x \Delta y} \alpha_{21} \lambda \bar{M}^{-1} \mathcal{L}(\hat{V}^0) + \beta_1 \lambda \hat{V}^0
\]

\[
(1 + \beta_2 \lambda I) \hat{V}^2 = \hat{V}^0 - \frac{\lambda \bar{M}^{-1}}{\Delta x \Delta y} (\alpha_{31} \mathcal{L}(\hat{V}^0) + \alpha_{32} \mathcal{L}(\hat{V}^1)) + \beta_2 \lambda \hat{V}^1
\]

\[
(1 + \beta_3 \lambda I) \hat{V}^3 = \hat{V}^0 - \frac{\lambda \bar{M}^{-1}}{\Delta x \Delta y} (\alpha_{41} \mathcal{L}(\hat{V}^0) + \alpha_{42} \mathcal{L}(\hat{V}^1) + \alpha_{43} \mathcal{L}(\hat{V}^2)) + \beta_3 \lambda \hat{V}^2
\]

\[
(1 + \beta_4 \lambda I) \hat{V}^4 = \hat{V}^0 - \frac{\lambda \bar{M}^{-1}}{\Delta x \Delta y} (\alpha_{51} \mathcal{L}(\hat{V}^0) + \alpha_{52} \mathcal{L}(\hat{V}^1) + \alpha_{53} \mathcal{L}(\hat{V}^2) + \alpha_{54} \mathcal{L}(\hat{V}^3)) + \beta_4 \lambda \hat{V}^3
\]

We require from the 4- and 5-stage Runge-Kutta schemes that they are second order accurate.
Optimized Runge-Kutta smoothers for space-time DGFEM

- Optimization of the smoother coefficients for three-level multigrid algorithms
- Space-time DG discretization of the 2D advection-diffusion equation using quadratic basis-functions
- For the optimization process, we fix the Reynolds numbers $Re_x = Re_y = 100$, the flow angle $\gamma_{flow} = \pi/4$ and the aspect ratio $AR = 1$.
- For steady flows we fix $CFL^\Delta t = 100$, while for unsteady flows $CFL^\Delta t = 1$. 
• The optimization process has a big impact on the Runge-Kutta coefficients which greatly differ per case.

• The Runge-Kutta coefficients are very different from commonly used schemes since they are optimized for fast multigrid convergence and not for time accuracy.

• Finding a good balance between optimization and more generally applicable algorithms is still an open question.

• In practice, the Runge-Kutta smoothers use local time stepping and this gives the opportunity to apply locally the best smoother for the actual flow state.
Optimized coefficients for dRK5 and fRK5 smoothers for 3-level multigrid (steady flow).

<table>
<thead>
<tr>
<th></th>
<th>dRK5 p = 1</th>
<th>fRK5 p = 1</th>
<th>dRK5 p = 2</th>
<th>fRK5 p = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>α21</td>
<td>0.05768995298</td>
<td>0.0578331573</td>
<td>0.04865009589</td>
<td>0.04877436325</td>
</tr>
<tr>
<td>α31</td>
<td>-</td>
<td>-0.00020515554736</td>
<td>-</td>
<td>-0.0002188348438</td>
</tr>
<tr>
<td>α32</td>
<td>0.1405960888</td>
<td>0.1403808301</td>
<td>0.130316854</td>
<td>0.1300906122</td>
</tr>
<tr>
<td>α41</td>
<td>-</td>
<td>0.0003953470071</td>
<td>-</td>
<td>2.608884832e-05</td>
</tr>
<tr>
<td>α42</td>
<td>-</td>
<td>-0.001195029164</td>
<td>-</td>
<td>2.444376496e-05</td>
</tr>
<tr>
<td>α43</td>
<td>0.267958213</td>
<td>0.2681810517</td>
<td>0.2729621396</td>
<td>0.2734805705</td>
</tr>
<tr>
<td>α51</td>
<td>-</td>
<td>0.0001441249202</td>
<td>-</td>
<td>-0.001250385487</td>
</tr>
<tr>
<td>α52</td>
<td>-</td>
<td>-0.0002608610327</td>
<td>-</td>
<td>-0.000738720635</td>
</tr>
<tr>
<td>α53</td>
<td>-</td>
<td>-0.0003368070181</td>
<td>-</td>
<td>-0.0004890887712</td>
</tr>
<tr>
<td>α54</td>
<td>0.5</td>
<td>0.8473374098</td>
<td>0.5</td>
<td>4.412139367</td>
</tr>
<tr>
<td>α55</td>
<td>-</td>
<td>0.4115573097</td>
<td>-</td>
<td>0.8097217358</td>
</tr>
<tr>
<td>α62</td>
<td>-</td>
<td>-0.003144851878</td>
<td>-</td>
<td>0.08435089009</td>
</tr>
<tr>
<td>α63</td>
<td>-</td>
<td>-0.000109645683</td>
<td>-</td>
<td>-0.01986799007</td>
</tr>
<tr>
<td>α64</td>
<td>-</td>
<td>0.001555741114</td>
<td>-</td>
<td>0.01359815476</td>
</tr>
<tr>
<td>α65</td>
<td>1.0</td>
<td>0.5901414466</td>
<td>1.0</td>
<td>0.1121972094</td>
</tr>
<tr>
<td>CFL^τ</td>
<td>0.8</td>
<td>0.8</td>
<td>0.4</td>
<td>0.4</td>
</tr>
<tr>
<td>ρ_S</td>
<td>0.98812</td>
<td>0.98914</td>
<td>0.98974</td>
<td>0.9896</td>
</tr>
<tr>
<td>ρ_MG</td>
<td>0.89151</td>
<td>0.81762</td>
<td>0.90049</td>
<td>0.89903</td>
</tr>
<tr>
<td>ρ^{EXT-MG}</td>
<td>167.06</td>
<td>-</td>
<td>124.02</td>
<td>-</td>
</tr>
</tbody>
</table>
Stability domain of dRK5 smoother and spectrum of a space-time DG discretization of the 2D advection-diffusion equation using quadratic basis functions.
Stability domain of the fRK5 smoother and spectrum of a space-time DG discretization of the 2D advection-diffusion equation using quadratic basis functions.
Eigenvalue spectra for space-time DG discretizations of the 2D advection-diffusion equation using quadratic basis functions.
Testing multigrid performance

- In order to demonstrate the performance of the optimized algorithms we consider the 2D advection-diffusion equation on a square.

- The exact steady state solution is

\[
    u(x_1, x_2) = \frac{1}{2} \left( \frac{\exp(a_1/\nu_1) - \exp(a_1 x_1/\nu_1)}{\exp(a_1/\nu_1) - 1} + \frac{\exp(a_2/\nu_2) - \exp(a_2 x_2/\nu_2)}{\exp(a_2/\nu_2) - 1} \right).
\]
• The advection-diffusion equation is discretized using the space-time discontinuous Galerkin discretization with quadratic basis functions.

• In the discretization we use a Shishkin mesh which is suitable for dealing with boundary layers.

• In pseudo-time we apply a rescaling to reduce the effect of grid stretching.
- The parameters in the tests are:
  - $\Delta t = 100$, $a = \sqrt{2}$, $\nu_x = \nu_y = 0.01$, $N_1 = N_2 = 32$.
  - Flow angle $\gamma^{\text{flow}} = \pi/4$.
  - Depending on the stability of the smoother, we use different $CFL^\tau$ and $VN^\tau$ numbers.
  - The EXV scheme was used when $Re_i \leq 1$ and the EXI scheme was used otherwise.
  - For the multigrid computations, $\nu_1 = \nu_2 = \nu_3 = \nu_4 = 1$, $\nu_C = 4$ and $\gamma = 1$. 
Convergence of three-level multigrid with different Runge-Kutta smoothers for second order accurate space-time DG discretization.
Convergence of three level multigrid with different Runge-Kutta smoothers for third order accurate space-time DG discretization.
We can draw the following conclusions:

- In all cases using the optimized Runge-Kutta smoothers a big improvement is obtained over the original EXI-EXV smoother.

- For linear basis functions the number of multigrid work units to obtain 4 orders of reduction in the residual is reduced from 3300 to 380.

- For quadratic basis functions the number of multigrid work units reduces from 22000 to 184.

- For linear basis functions the difference in convergence rate between Runge-Kutta smoothers with only non-zero diagonal terms versus Runge-Kutta smoothers with all coefficients non-zero is negligible.
• For quadratic basis functions this difference is, however, significant.

• Using more Runge-Kutta coefficients enlarges the possibilities to optimize the smoother, but the optimization process requires a significantly larger computing time.

• In order to speed up the optimization process the coefficients of Runge-Kutta schemes with only non-zero diagonal terms are used as initial values.
• The effect of solving the equations on the coarsest mesh is very large.

• In particular, for nonlinear problems it is tempting to solve the algebraic system on the coarsest mesh only approximately, but the effect is non-negligible.

• The flow angle has a small effect on the convergence rate. In general if the flow angle is close to one of the mesh lines the convergence rate is the slowest.

• It is important not to extrapolate the results for the advection-diffusion equation directly to the Euler and Navier-Stokes equations.
Euler Equations

Amount of work per multigrid cycle differs single grid, $p$-, $h$-, and $hp$-multigrid:

\[
\text{work per cycle} = \begin{cases} 
    g^p b^p, & \text{SG,} \\
    (g^p b^p + g^{p-1} b^{p-1})(\nu_1^p + \nu_2^p) + g^{p-2} b^{p-2} \nu_C^p, & \text{pMG,} \\
    g^p b^p ((c^h + c^{h-1})(\nu_1^h + \nu_2^h) + c^{h-2} \nu_C^h), & \text{hMG,} \\
    c^h (g^p b^p + g^{p-1} b^{p-1})(\nu_1^p + \nu_2^p) + \\
    g^{p-2} b^{p-2} ((c^h + c^{h-1})(\nu_1^h + \nu_2^h) + c^{h-2} \nu_C^h), & \text{hpMG,} 
\end{cases}
\]

with $g^p$ and $b^p$, respectively, the number of Gauss quadrature points and basis functions in an element, and $c^h$ a weighting for the number of cells depending on the grid-level $h$.

Coefficients: $g^p = 9$, $g^{p-1} = 4$, $g^{p-2} = 1$, $b^p = 6$, $b^{p-1} = 4$, $b^{p-2} = 1$, $c^h = 1$, $c^{h-1} = 1/4$ and $c^{h-2} = 1/16$. 
<table>
<thead>
<tr>
<th>$p$-levels</th>
<th>$h$-levels</th>
<th>$CF^\tau$</th>
<th>$\rho$</th>
<th>work</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.0</td>
<td>0.99010</td>
<td>75006</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1.0</td>
<td>0.74318</td>
<td>33370</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1.0</td>
<td>0.73347</td>
<td>32091</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1.0</td>
<td>0.66350</td>
<td>24098</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.5</td>
<td>0.80364</td>
<td>45360</td>
</tr>
</tbody>
</table>

Spectral radii and multigrid work units of different multigrid strategies.
Convergence history of different multigrid techniques for space-time DG discretization for inviscid flow around a NACA0012 airfoil (MTC 1 test case, $\alpha = 2^\circ$, $Ma = 0.5$, $448 \times 64$ elements).
Conclusions

- A space-time DG discretization for nonconservative hyperbolic pde’s using the DLM theory and the compressible Navier-Stokes has been developed.

- A new numerical flux for nonconservative hyperbolic pde’s has been developed, which reduces to the HLLC flux for conservative pde’s.

- The effect of the choice of the path in phase space is in practice for nearly all cases negligible.

- The algorithm has been successfully tested on a depth averaged two-phase flow model and compressible flow simulations.
Conclusions

- A detailed three-level multigrid analysis has been conducted for \( h^- \), \( p^- \) and \( hp^- \)-multigrid algorithms for the advection-diffusion and the linearized Euler equations.

- The analysis provides the asymptotic convergence rate of the multigrid algorithms which is used to optimize the multigrid smoothers.

- For the advection-diffusion equation the optimization results in a significant improvement in the convergence rate in actual computations using the \( h^- \)-multigrid algorithm.
For the Euler equations the optimized Runge-Kutta smoothers for $h$-, $p$- and $hp$-multigrid show initially a significant improvement in convergence rate for inviscid flow about a NACA 0012 airfoil, but asymptotically most algorithms have the same convergence rate.

Using an exact solution of the algebraic system on the coarsest mesh has a major impact on the convergence rate for the advection-diffusion equation, but not for the Euler equations.