Space-Time Discontinuous Galerkin Finite Element Methods

J.J.W. van der Vegt
University of Twente, Department of Applied Mathematics
P.O. Box 217, 7500 AE, Enschede, The Netherlands
email: j.j.w.vandervegt@math.utwente.nl

Abstract

In these notes an introduction is given to space-time discontinuous Galerkin (DG) finite element methods for hyperbolic and parabolic conservation laws on time dependent domains. The space-time DG discretization is explained in detail, including the definition of the numerical fluxes and stabilization operators necessary to maintain stable and non-oscillatory solutions. In addition, a pseudo-time integration method for the solution of the algebraic equations resulting from the DG discretization and the relation between the space-time DG method and an arbitrary Lagrangian Eulerian approach are discussed. Finally, a brief overview of some applications to aerodynamics is given.

Keywords: discontinuous Galerkin finite element methods, space-time finite element methods, hyperbolic and parabolic conservation laws, upwind schemes, pseudo-time integration methods, local mesh refinement, compressible gas dynamics, dynamic grid motion, Arbitrary Lagrangian Eulerian (ALE) methods.

Subject classifications: 65P25, 76N15.

1 Introduction

Many applications in fluid dynamics have time-dependent boundaries where the boundary movement is either prescribed or part of the solution. Examples are fluid-structure interaction, two-phase flows with free surfaces, Stefan problems and water waves. In all of these problems the computational mesh has to follow the boundary movement, which requires the interior mesh points to move also in order to maintain a consistent mesh without grid folding. This mesh movement imposes, however, additional complications for the numerical discretization. In particular, ensuring that the numerical discretization is conservative on time-dependent meshes is non-trivial. This is important because the equations of fluid dynamics express conservation of mass, momentum and energy, which should also be satisfied at the discrete level.

A very natural way to derive numerical discretizations for problems which require deforming and moving meshes is to use the space-time approach, in particular the space-time discontinuous Galerkin finite element method. In this technique time is considered as an extra dimension and treated in the same way as the spatial coordinates. Space-time

DG methods combine well known benefits of a DG method, such as optimal flexibility for local mesh refinement (h-adaptation), adjustment of the polynomial order in each element (p-adaptation) and excellent performance on parallel computers, with a fully conservative Arbitrary Lagrangian Eulerian (ALE) approach to deal with deforming meshes.

Discontinuous Galerkin methods have recently received significant attention, both for hyperbolic and (incompletely) parabolic problems. For surveys of DG methods to discretize elliptic, parabolic and hyperbolic partial differential equations in space we refer to [4; 16; 19; 21]. In these notes we will focus on the main aspects of the space-time discontinuous Galerkin finite element method. We will use scalar hyperbolic and parabolic conservation laws as an example since they are well suited to highlight the main concepts.

The extension of the techniques described in these notes to inviscid compressible flows described by the Euler equations of gas dynamics, including local mesh refinement and a multigrid accelerated pseudo-time integration method to solve the non-linear algebraic equations resulting from the DG discretization, can be found in van der Vegt and van der Ven [42], whereas quadrature techniques to improve the efficiency of the algorithm are discussed and analyzed in van der Ven and van der Vegt [44]. The space-time DG method discussed in [42; 44] has been recently extended to the compressible Navier-Stokes equations in Klaij, van der Vegt and van der Ven [26; 27; 28] and applied to a number of (unsteady) viscous aerodynamic flows. Applications of the space-time DG method to rotorcraft can be found in Boelens and van der Ven [11; 45] and for deforming wings in van der Ven, van der Vegt and Bouwman [46]. A detailed error and stability analysis of the space-time DG method for the advection-diffusion equation is given in Sudirham, van der Vegt and van Damme [35] and the Oseen equations in van der Vegt and Sudirham [41].

Applications to nonconservative hyperbolic partial differential equations, in particular models for dispersed multiphase flows, can be found in Rhebergen, Bokhove and van der Vegt [33]. The use of entropy and pressure-primitive variables in DG discretizations of the compressible Navier-Stokes equations to deal with (nearly) incompressible flows and general equations of states is discussed in [32]. Another important field of applications of space-(time) DG methods is the simulation of nonlinear water waves, see e.g. Bokhove [12], Ambati and Bokhove [1; 2], van der Vegt and Tomar [40], van der Vegt and Xu [43], Tomar and van der Vegt [38]. Shallow flows, including sediment transport, can be found in Tassi et al. [36; 37]. An overview of a software toolkit for DG finite element methods is given in [31].

The outline of these notes is as follows. In Section 2 we start with an introduction to space-time DG methods for scalar hyperbolic conservation laws in one space dimension. This section serves to explain the basic aspects of space-time DG methods. First, the space-time formulation is introduced and the geometry of the space-time domain and elements is defined in Section 2.1. Next, we discuss the space-time discontinuous Galerkin discretization in Section 2.2. An important aspect for any numerical discretization of hyperbolic partial differential equations is to ensure that it does not introduce numerical oscillations around discontinuities. For this purpose we introduce a stabilization operator in Section 2.3. The space-time DG discretization results in a large set of coupled nonlinear equations which need to be solved each time step. This is done efficiently with a pseudo-time integration method, which is discussed in Section 2.4. The stability analysis of the pseudo-time integration method is discussed in Section 2.5.

In Section 3 we present the extension of the space-time DG method to multiple dimensions and to parabolic scalar conservation laws. After a brief introduction in Section 3.1, the space-time formulation is discussed in Section 3.2, the finite element spaces, trace and lifting operators in Section 3.3, and the space-time DG discretization is derived in Section 3.4. For this purpose the parabolic scalar conservation law is rewritten as a first order system by introducing auxiliary variables. This system is then subsequently discretized, after which the auxiliary variables are eliminated using lifting operators. The relation of the space-time formulation with the ALE method is also explained in this section. Finally, in Section 4 a brief overview of the extension to the compressible Navier-Stokes equations and some advanced applications in aerodynamics are discussed.

2 Space-time methods for scalar hyperbolic conservation laws in one space dimension

2.1 Space-time formulation

Consider a scalar conservation law in a time dependent flow domain $\Omega(t) \subseteq \mathbb{R}^d$ with boundary $\partial\Omega$:

$$\frac{\partial u}{\partial t} + \mathrm{div} f(u) = 0, \qquad \bar{x} \in \Omega(t),\ t \in (t_0, T), \tag{1}$$

with $u : \Omega \to \mathbb{R}$ the conserved quantity, $f : \mathbb{R} \to \mathbb{R}^d$ the flux vector and boundary conditions:

$$u(t, \bar{x}) = \mathcal{B}(u, u_w), \qquad \bar{x} \in \partial\Omega(t),\ t \in (t_0, T), \tag{2}$$

and initial condition:

$$u(0, \bar{x}) = u_0(\bar{x}), \qquad \bar{x} \in \Omega(t_0). \tag{3}$$

Here, u denotes a scalar quantity, t represents time, with t_0 the initial and T the final time of the time evolution. The boundary operator is denoted as $\mathcal{B}(u, u_w)$, with u_w prescribed data at the boundary, and defines which type of boundary conditions are imposed at $\partial\Omega$. Examples are Dirichlet boundary conditions, with $u = u_w$, or Neumann boundary conditions with $\frac{\partial u}{\partial \bar{n}} = u_w$, where $\bar{n}$ denotes the unit outward normal vector at $\partial\Omega$. An example of a time-dependent domain, resembling a piston moving into and out of a cylinder is given in Figure 1.

If we would directly discretize (1)-(3) with a finite element or finite volume method then at each instant of time the mesh points have to move in order to account for the boundary movement. At their new position we generally do not have data points available and we need to interpolate or extrapolate the data from the old mesh to the new mesh. This interpolation process is generally non-conservative and can introduce substantial errors. Also, one has to be very careful in defining the proper mesh velocities. For a detailed discussion see Lesoinne and Farhat [29].

An alternative approach is provided by the space-time discretization method. In a space-time discretization we directly consider the domain in $\mathbb{R}^{d+1}$. A point $x \in \mathbb{R}^{d+1}$ has coordinates $(x_0, \bar{x})$, with $x_0 = t$ representing time and $\bar{x} \in \mathbb{R}^d$ the spatial coordinate. We define the space-time domain as the open domain $\mathcal{E} \subset \mathbb{R}^{d+1}$, see Figure 2. The boundary $\partial\mathcal{E}$ of the space-time domain $\mathcal{E}$ consists of the hypersurfaces $\Omega(t_0) := \{x \in \partial\mathcal{E} \mid x_0 = t_0\}$,

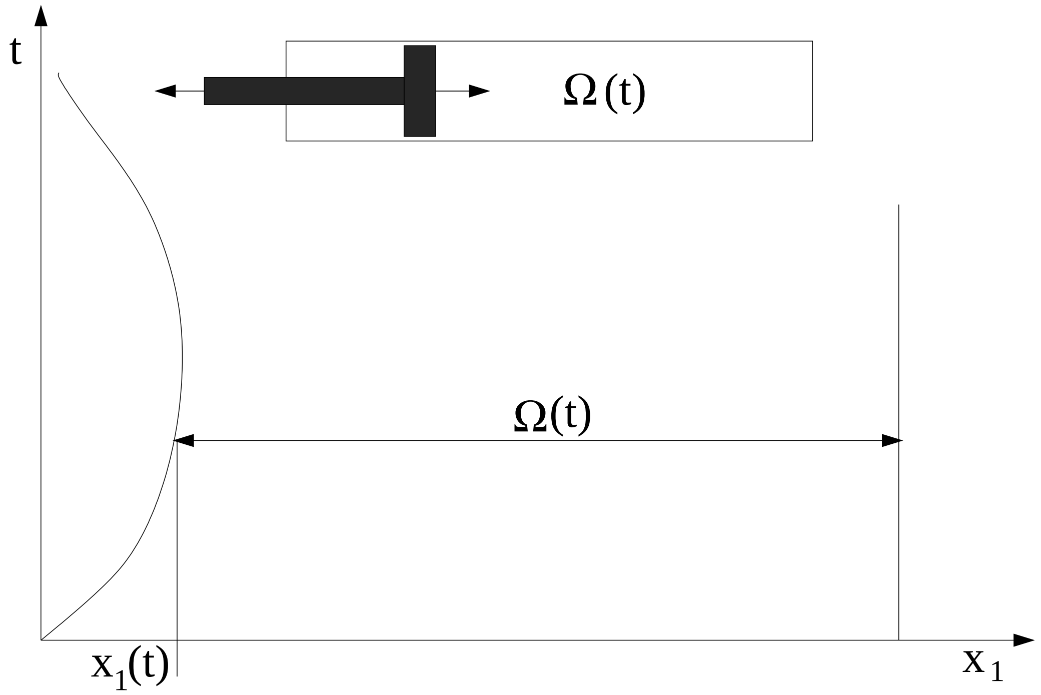


Figure 1: Example of a time dependent flow domain $\Omega(t)$.

$\Omega(T) := \{x \in \partial\mathcal{E} \,|\, x_0 = T\}$, and $\mathcal{Q} := \{x \in \partial\mathcal{E} \,|\, t_0 < x_0 < T\}$. The space-time domain boundary $\partial\mathcal{E}$ therefore is equal to $\partial\mathcal{E} = \Omega(t_0) \cup \mathcal{Q} \cup \Omega(T)$.

The scalar conservation law (1) can now be reformulated in the space-time frame work. The space-time formulation of the scalar conservation law (1) is obtained by introducing the space-time flux vector:

$$\mathcal{F}(u) := (u, f(u))^T.$$

The scalar conservation law (1) then can be written as:

$$\operatorname{div} \mathcal{F}(u(x)) = 0, \qquad x \in \mathcal{E}, \tag{4}$$

with boundary conditions:

$$u(x) = \mathcal{B}(u, u_w), \qquad x \in \mathcal{Q}, \tag{5}$$

and initial condition:

$$u(x) = u_0(x), \qquad x \in \Omega(t_0). \tag{6}$$

Here, the div operator is defined as $\operatorname{div} \mathcal{F} = \frac{\partial \mathcal{F}_i}{\partial x_i}$ and the summation convention is used on repeated indices.

The space-time formulation requires the introduction of a space-time slab, elements and faces. First, consider the time interval $\mathcal{I} = [t_0, T]$, partitioned by an ordered series of time levels $t_0 < t_1 < \cdots < t_{N_T} = T$. Denoting the nth time interval as $I_n = (t_n, t_{n+1})$, we have $\mathcal{I} = \cup_n \bar{I}_n$. The length of I_n is defined as $\triangle t_n = t_{n+1} - t_n$. Let $\Omega(t_n)$ be the space-time domain at time $t = t_n$. A space-time slab is then defined as the domain $\mathcal{E}^n = \mathcal{E} \cap I_n$, with boundaries $\Omega(t_n)$, $\Omega(t_{n+1})$ and $\mathcal{Q}^n = \partial\mathcal{E}^n \backslash (\Omega(t_n) \cup \Omega(t_{n+1}))$.

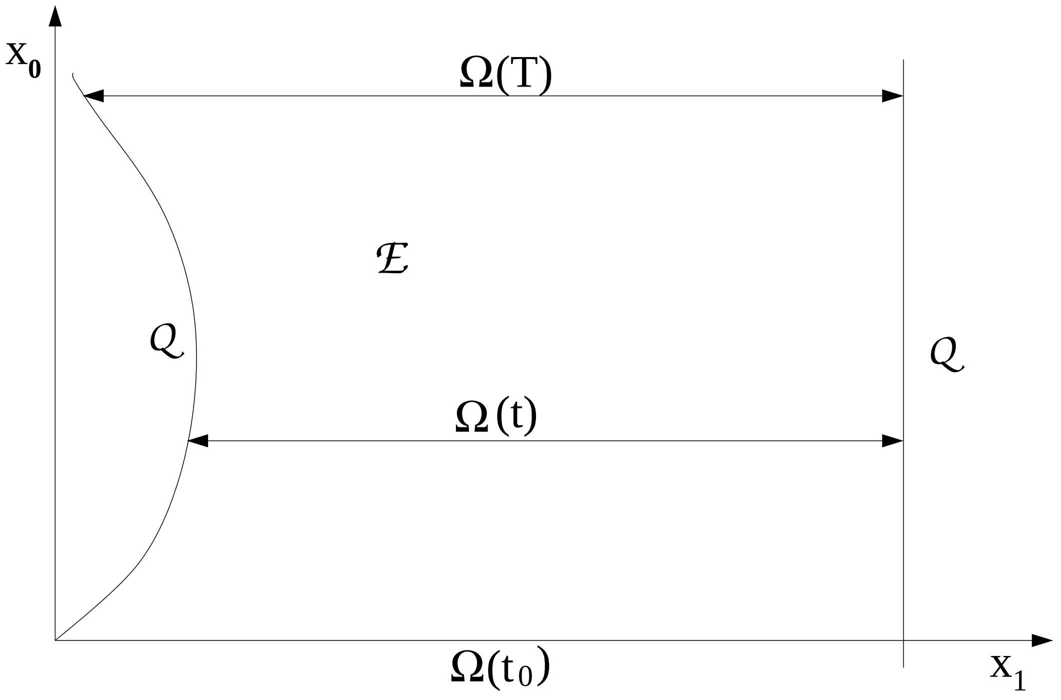


Figure 2: Example of a space-time domain $\mathcal{E}$.

We now describe the construction of the space-time elements $\mathcal{K}$ in the space-time slab $\mathcal{E}^n$. Let the domain $\Omega(t_n)$ be divided into N_n non-overlapping spatial elements K^n. At t_{n+1} the spatial elements K^{n+1} are obtained by mapping the vertices of the elements K^n to their new position at $t = t_{n+1}$. Each space-time element $\mathcal{K}$ is then obtained by connecting the elements K^n and K^{n+1} using linear interpolation in time. A sketch of the space-time slab $\mathcal{E}^n$ is shown in Figure 3. The boundary of the space-time element is denoted as $\partial\mathcal{K}$ and consists of three parts K^n, K^{n+1}, and $\mathcal{Q}^n_{\mathcal{K}} = \partial\mathcal{K}\backslash(K^n \cup K^{n+1})$.

The geometry of the space-time element can be defined by introducing the mapping G^n_K. This mapping connects the space-time element $\mathcal{K}^n$ to the reference element $\hat{\mathcal{K}} = (-1,1)^{d+1}$ and is defined using the following steps. First, we define a smooth, orientation preserving and invertible mapping Φ^n_t in the interval I_n as:

$$\Phi^n_t : \Omega(t_n) \to \Omega(t) : \bar{x} \mapsto \Phi^n_t(\bar{x}), \quad t \in I_n.$$

Next, we split $\Omega(t_n)$ into the tessellation $\bar{\mathcal{T}}^n_h$ with non-overlapping elements K_j. The elements $K \in \bar{\mathcal{T}}^n_h$ are defined using the mapping F^n_K:

$$F^n_K : (-1,1)^d \to K^n : \bar{\xi} \longmapsto \sum_{i=1}^{N_v} x_i(K^n)\chi_i(\bar{\xi}),$$

with $x_i(K^n)$ the spatial coordinates of the N_v nodal points of the space-time element at time $t = t_n$ and χ_i the standard linear finite element shape functions defined on the interval $(-1,1)^d$. Similarly, we use the mapping F^{n+1}_K to define the element K^{n+1}:

$$F^{n+1}_K : (-1,1)^d \to K^{n+1} : \bar{\xi} \longmapsto \sum_{i=1}^{N_v} \Phi^n_{t_{n+1}}(x_i(K^n))\chi_i(\bar{\xi}).$$

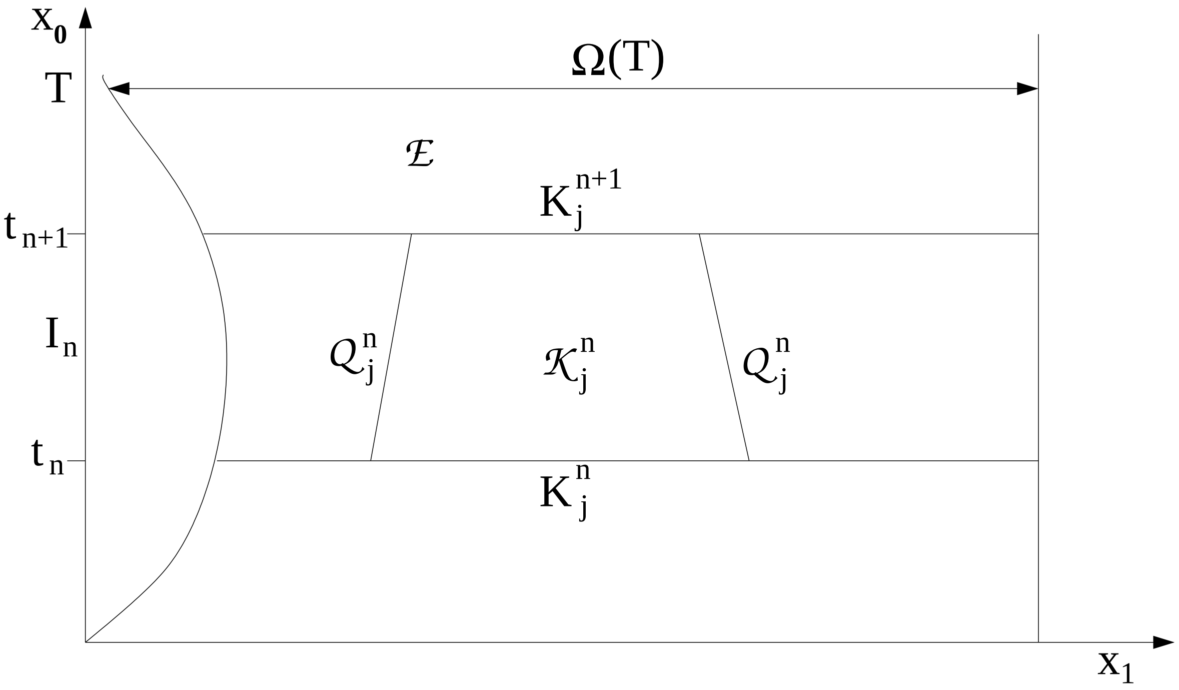


Figure 3: Space-time slab in space-time domain $\mathcal{E}$.

Using linear interpolation in time the space-time element $\mathcal{K}$ can now be defined using the mapping:

$$G_K^n : (-1,1)^{d+1} \to \mathcal{K}^n : (\xi_0, \bar{\xi}) \longmapsto (x_0, \bar{x}), \tag{7}$$

with:

$$(x_0, \bar{x}) = \Big(\tfrac{1}{2}(t_n + t_{n+1}) - \tfrac{1}{2}(t_n - t_{n+1})\xi_0,$$
$$\tfrac{1}{2}(1-\xi_0)F_K^n(\bar{\xi}) + \tfrac{1}{2}(1+\xi_0)F_K^{n+1}(\bar{\xi}) \Big).$$

An overview of the different mappings is given in Figure 4. The space-time tessellation is now defined as:

$$\mathcal{T}_h^n := \{\mathcal{K} = G_K^n(\hat{\mathcal{K}}) \mid K \in \bar{\mathcal{T}}_h^n\}.$$

2.2 Space-time discontinuous Galerkin finite element discretization

The space-time formulation can be used both in continuous and discontinuous finite element methods. The key difference here is if continuous or discontinuous basis functions are used inside a space-time slab. In this section we will discuss the construction of a space-time discontinuous Galerkin discretization for the scalar conservation law (4). The first step is the definition of the basis functions $\hat{\phi}_m, (m = 0, \cdots, M)$ on the master element $\hat{\mathcal{K}} = (-1,1)^{d+1}$, for which we use monomials of total degree p:

$$\hat{\phi}_m(\xi_0, \xi_1) = \xi_0^{i_0} \xi_1^{i_1} \cdots \xi_d^{i_d}.$$

Note, this is the most simple choice for the basis functions. For higher order discretizations hierarchic polynomial basis functions are, however, generally more suitable, see for

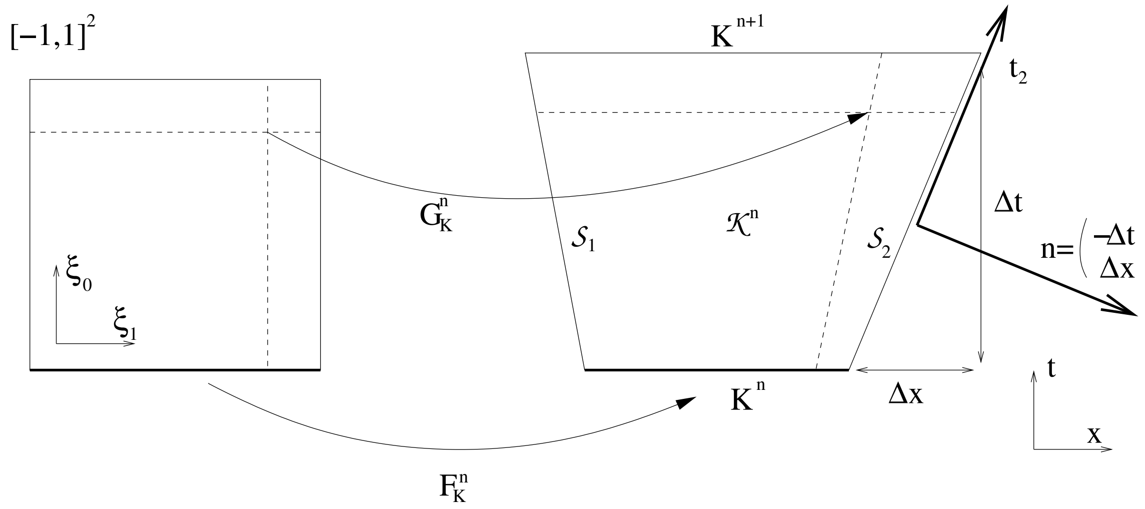


Figure 4: Geometry of 2D space-time element in both computational and physical space.

instance [34; 23]. These basis functions result in better conditioned matrices and changing the polynomial order in an element is relatively straightforward. Using the mapping (7), the basis functions ϕ_m on the space-time element $\mathcal{K}$ can now be defined as:

$$\phi_m(x) = \hat{\phi}_m \circ G_K^{-1}(x).$$

It is important to realize that these polynomial basis functions are defined independently on each element. We do not require any continuity across element faces, both in space and time. This is the main difference with the Runge-Kutta discontinuous Galerkin method where only discontinuous basis functions in space are used and the integration in time is performed with a Runge-Kutta method. For a survey of the RKDG method see [16; 19; 21]. An impression of the basis functions for linear polynomials is given in Figure 5, where we indicated the discontinuities in the polynomial approximation at the element boundaries. For a number of reasons, in particular the definition of a stabilization operator to deal with discontinuous solutions, it is beneficial to split the test and trial functions into an element mean at time t_{n+1} and a fluctuating part. We introduce therefore the basis functions $\psi_m : \mathcal{K} \to \mathbb{R}$:

$$\begin{aligned} \psi_m(x) &= 1, && \text{if } m = 0, \\ &= \phi_m(x) - \frac{1}{|K(t_{n+1})|}\int_{K(t_{n+1})} \phi_m dK, && \text{if } m \geq 1. \end{aligned}$$

For many applications this splitting is, however, not necessary and one can work directly with the ϕ_m basis functions.

The discontinuous Galerkin discretization requires the definition of the following finite element space:

$$V_h^p(\mathcal{T}_h^n) := \Big\{ v_h \,\Big|\, v_h|_{\mathcal{K}} \circ G_{\mathcal{K}}^n \in \mathcal{Q}_p(\mathcal{K}),\, \forall \mathcal{K} \in \mathcal{T}_h^n \Big\},$$

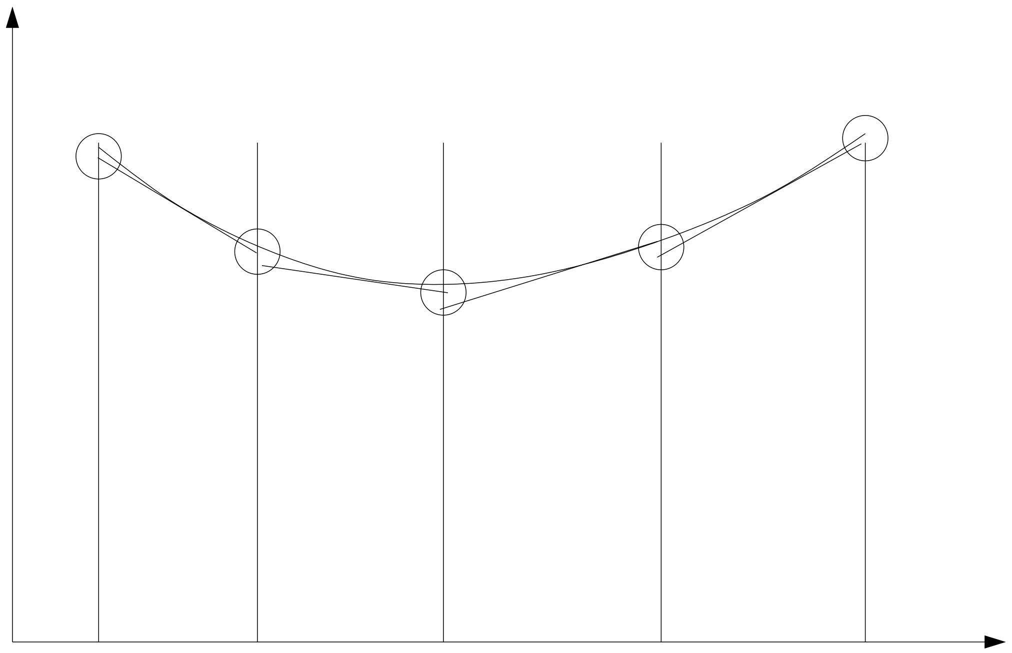


Figure 5: Discontinous Galerkin approximation of a function.

with $\mathcal{Q}_p(\mathcal{K}) = \text{span}\{\psi_m, m = 0, \cdots, M\}$. The trial functions $u_h : \mathcal{K} \to \mathbb{R}^{d+1}$ are defined in each element $\mathcal{K} \in \mathcal{T}_h^n$ as:

$$u_h(x) = \sum_{m=0}^{M} \hat{U}_m(\mathcal{K})\psi_m(x), \quad x \in \mathcal{K}, \tag{8}$$

with $\hat{U}_m$ the expansion coefficients. Due to the definition of the basis functions ψ_m, which ensures that $\int_{K(t_{n+1})} \psi_m(x)dK = 0$ for $m \geq 1$, we have the relation:

$$\bar{u}_h(K(t_{n+1})) := \frac{1}{|K(t_{n+1})|} \int_{K(t_{n+1})} u_h dK = \hat{U}_1,$$

and we can write:

$$u_h(x) = \bar{u}_h(K(t_{n+1})) + \tilde{u}_h(x),$$

with $\bar{u}_h(K(t_{n+1}))$ the mean solution at t_{n+1} and $\tilde{u}_h(x)$ the fluctuations in element $\mathcal{K}^n$ with $\int_{K(t_{n+1})} \tilde{u}(x)dK = 0$. One of the main benefits of this splitting is that the equation for $\hat{U}_0$ is very similar to a first order finite volume discretization and is only weakly coupled to the equations for $\tilde{u}_h$.

The discontinuous Galerkin finite element formulation is now obtained using the following steps. First, equation (4) is multiplied with arbitrary test functions $w_h \in V_h^p(\mathcal{T}_h^n)$ and integrated over the space-time domain $\mathcal{E}$, split into space-time elements. After introducing the trial functions $u_h \in V_h^p(\mathcal{T}_h^n)$, we obtain the weighted residual formulation:

Find a $u_h \in V_h^p(\mathcal{T}_h^n)$, such that for all $w_h \in V_h^p(\mathcal{T}_h^n)$, we have:

$$\sum_{n=0}^{N_T} \sum_{j=1}^{N_n} \Big(\int_{\mathcal{K}_j^n} w_h \, \text{div}\mathcal{F}(u_h) d\mathcal{K} + \int_{\mathcal{K}_j^n} (\text{grad}\, w_h)^T \, \mathfrak{D}(u_h) \, \text{grad}\, u_h d\mathcal{K} \Big) = 0. \tag{9}$$

Here, the second integral is the stabilization operator, with $\mathfrak{D}(u_h) \in \mathbb{R}^{(d+1)\times(d+1)}$ the stabilization matrix, which is necessary to obtain monotone solutions near discontinuities. This contribution will be discussed in Section 2.3. The discontinuous Galerkin weak formulation is now obtained by integrating (9) by parts, resulting in:

Find a $u_h \in V_h^p(\mathcal{T}_h^n)$, such that for all $w_h \in V_h^p(\mathcal{T}_h^n)$, we have:

$$\sum_{n=0}^{N_T}\sum_{j=1}^{N_n}\Big(-\int_{\mathcal{K}_j^n} \operatorname{grad} w_h \cdot \mathcal{F}(u_h) d\mathcal{K} + \int_{\partial\mathcal{K}_j^n} w_h^- n^- \cdot \mathcal{F}(u_h^-) d(\partial\mathcal{K}) \\ + \int_{\mathcal{K}_j^n} (\operatorname{grad} w_h)^T \, \mathfrak{D}(u_h) \operatorname{grad} u_h d\mathcal{K}\Big) = 0. \tag{10}$$

In the evaluation of the integrals in the weak formulation it is important to make a distinction between the traces at the element boundary taken from either the inside or the outside of the element since the basis functions are discontinuous at the element faces. The traces at $\partial\mathcal{K}$ are defined as:

$$w_h^\pm = \lim_{\epsilon\downarrow 0} w_h(x \pm \epsilon n_{\mathcal{K}}),$$

with n the unit outward space-time normal vector at $\partial\mathcal{K}$.

The integrals over the faces $\partial\mathcal{K}$ in the weak formulation (10) can be further evaluated using the fact that the interior faces are counted twice in the summation over the elements. The same applies for boundary faces if we extend the mesh with ghostcells. We can transform the integrals over the element boundaries therefore into:

$$\sum_{\mathcal{K}}\int_{\partial\mathcal{K}} w_h^- n^- \cdot \mathcal{F}^- d(\partial\mathcal{K}) = \sum_{\mathcal{S}}\int_{\mathcal{S}} \tfrac{1}{2}(w_h^- n^- + w_h^+ n^+)\cdot(\mathcal{F}^- + \mathcal{F}^+) + \\ \tfrac{1}{2}(w_h^- + w_h^+)(\mathcal{F}^-\cdot n^- + \mathcal{F}^+\cdot n^+) d\mathcal{S}, \tag{11}$$

with $\mathcal{F}^\pm = \mathcal{F}(u_h^\pm)$, S the faces in the tessellation, and n^-, n^+ the normal vectors at each side of the face $\mathcal{S}$, which satisfy $n^+ = -n^-$. Since the formulation should be conservative, we must impose the condition:

$$\int_{\mathcal{S}} w_h n^- \cdot \mathcal{F}^- d\mathcal{S} = -\int_{\mathcal{S}} w_h n^+ \cdot \mathcal{F}^+ d\mathcal{S}, \qquad \forall w_h \in V_h^p(\mathcal{T}_h^n),$$

hence the second contribution in (11) is zero. The boundary integrals therefore are equal to:

$$\sum_{\mathcal{K}}\int_{\partial\mathcal{K}} w_h^- n^- \cdot \mathcal{F}^- d(\partial\mathcal{K}) = \sum_{\mathcal{S}}\int_{\mathcal{S}} (w_h^- - w_h^+)\tfrac{1}{2} n^- \cdot (\mathcal{F}^- + \mathcal{F}^+) d\mathcal{S},$$

using the relation $n^+ = -n^-$. The next step is to replace the multi-valued trace of the flux $\mathcal{F}$ at $\mathcal{S}$ with a numerical flux function:

$$H(u_h^-, u_h^+, n) \cong \tfrac{1}{2} n \cdot (\mathcal{F}^- + \mathcal{F}^+).$$

Depending on the choice of the numerical flux function this can introduce upwind information into the DG formulation. The use of a numerical flux function shows a close resemblance with upwind finite volume methods. The boundary integrals at the element faces can now be expressed as:

$$\begin{aligned}\sum_{\mathcal{K}}\int_{\partial\mathcal{K}} w_h^- n^- \cdot \mathcal{F}^- d(\partial\mathcal{K}) &= \sum_{\mathcal{S}}\int_{\mathcal{S}} (w_h^- - w_h^+) H(u_h^-, u_h^+, n^-) d\mathcal{S}\\ &= \sum_{\mathcal{K}}\int_{\partial\mathcal{K}} w_h^- H(u_h^-, u_h^+, n^-) d(\partial\mathcal{K}),\end{aligned}$$

where we used the fact that the numerical flux has to be conservative, which imposes the condition $H(u_h^-, u_h^+, n^-) = -H(u_h^+, u_h^-, n^+)$.

In the definition of the numerical flux we have to make a distinction between the faces K^n and K^{n+1} at the time levels $t = t_n$ and t_{n+1}, respectively, and the faces $\mathcal{Q}^n$. The first two faces have a space-time normal vector $n = (\pm 1, 0, \cdots, 0)^T$ and information should only move from the past to the future, whereas through the faces $\mathcal{Q}^n$ information flows both into and out of the element $\mathcal{K}$. The numerical flux at the boundary faces K^n and K^{n+1}, therefore is defined as:

$$\begin{aligned}H(u_h^-, u_h^+, n^-) &= u_h^+ \qquad \text{at } K^n\\ &= u_h^- \qquad \text{at } K^{n+1}.\end{aligned}$$

The numerical flux at the boundary faces $\mathcal{Q}^n$ is chosen as a monotone Lipschitz $H(u_h^-, u_h^+, n)$, which is consistent:

$$H(u, u, n) = n \cdot f(u)$$

and conservative:

$$H(u_h^-, u_h^+, n^-) = -H(u_h^+, u_h^-, n^+).$$

The monotone Lipschitz flux $H(u_h^-, u_h^+, n)$ is obtained by (approximately) solving a Riemann problem with initial states u_h^- and u_h^+ at the element faces $\mathcal{Q}^n$ at time $t = t_n$. Important consistent and monotone Lipschitz fluxes are the:

- Godunov flux

$$H^G(u_h^-, u_h^+, n) = \begin{cases} \min\limits_{u \in [u_h^-, u_h^+]} \hat{f}(u), & \text{if } u_h^- \leq u_h^+ \\ \max\limits_{u \in [u_h^+, u_h^-]} \hat{f}(u), & \text{otherwise,} \end{cases} \tag{12}$$

  with $\hat{f}(u) = f(u) \cdot n$.

- Engquist-Osher flux

$$H^{EO}(u_h^-, u_h^+, n) = \int_0^{u_h^+} \min(\hat{f}'(s), 0) ds + \int_0^{u_h^-} \max(\hat{f}'(s), 0) ds + \hat{f}(0), \tag{13}$$

  where a prime denotes differentiation.

- Local Lax-Friedrichs flux

$$\begin{aligned}H^{LLF}(u_h^-, u_h^+, n) &= \frac{1}{2}(\hat{f}(u_h^-) + \hat{f}(u_h^+) - C(u_h^+ - u_h^-)), \qquad (14)\\ C &= \max_{\min(u_h^-, u_h^+) \leq s \leq \max(u_h^-, u_h^+)} |\hat{f}'(s)|,\end{aligned}$$

- Roe flux with 'entropy fix'

$$H^R(u_h^-,u_h^+,n)=\begin{cases}\hat{f}(u_h^-), & \text{if } \hat{f}'(u)\geq 0 \text{ for } u\in[\min[u_h^-,u_h^+],\max[u_h^-,u_h^+]]\\ \hat{f}(u_h^+), & \text{if } \hat{f}'(u)\leq 0 \text{ for } u\in[\min[u_h^-,u_h^+],\max[u_h^-,u_h^+]]\\ H^{LLF}(u_h^-,u_h^+,n), & \text{otherwise.}\end{cases} \tag{15}$$

For an excellent overview of all these (approximate) Riemann solvers, see [39]. The choice which numerical flux should be used depends on many aspects, e.g. accuracy, robustness, computational complexity, and also personal preference.

The space-time flux can be straightforwardly transformed into an arbitrary Lagrangian Eulerian form. At the boundary faces $\mathcal{Q}^n$ a simple calculation, see Figure 4, shows that the space-time normal vector can be expressed as:

$$n=(-u_g\cdot\bar{n},\bar{n}),$$

with u_g the velocity of the point at $\mathcal{Q}^n$ where the space-time normal vector is computed and $\bar{n}$ the spatial component of the space-time normal vector n. If we introduce this relation into the numerical fluxes defined in (12-15) then

$$\hat{f}(u)=f(u)\cdot\bar{n}-u_g\cdot\bar{n}\,u,$$

which is exactly the flux in an ALE formulation. For more details, see [24; 42].

After introducing the numerical fluxes we can transform the weak formulation (10) into:

Find a $u_h\in V_h^p(\mathcal{T}_h^n)$, such that for all $w_h\in V_h^p(\mathcal{T}_h^n)$, the following variational equation is satisfied:

$$\sum_{j=1}^{N_n}\Big(-\int_{\mathcal{K}_j^n}(\operatorname{grad}w_h)\cdot\mathcal{F}(u_h)d\mathcal{K}+\int_{K_j(t_{n+1})}w_h^-u_h^-dK-\int_{K_j(t_n)}w_h^-u_h^+dK+\int_{\mathcal{Q}_j^n}w_h^-H(u_h^-,u_h^+,n^-)d\mathcal{Q}+\int_{\mathcal{K}_j^n}(\operatorname{grad}\,w_h)^T\mathfrak{D}(u_h)\operatorname{grad}u_h\,d\mathcal{K}\Big)=0. \tag{16}$$

This formulation shows that due to the causality of the time-flux, the solution in a space-time slab now only depends explicitly on the data from the previous space-time slab.

The algebraic equations for the DG discretization are obtained by introducing the polynomial expansions for u_h and w_h, given by (8) into the weak formulation (16) and using the fact that the coefficients $\hat{w}_m$ are arbitrary. The following set of equations for the element mean $\bar{u}_h(K_j(t_{n+1}))$ are then obtained:

$$\big|K_j(t_{n+1})\big|\bar{u}_h(K_j(t_{n+1}))-\big|K_j(t_n)\big|\bar{u}_h(K_j(t_n))+\int_{\mathcal{Q}_j^n}H(u_h^-,u_h^+,n^-)d\mathcal{Q}=0. \tag{17}$$

Note, these equations have a similar structure as a first order accurate finite volume formulation, except that more accurate data are used at the element faces.

The equations for the coefficients $\hat{U}_m(\mathcal{K}_j^n)$, $(m \geq 1)$ related to the fluctuating part of the flow field $\tilde{u}_h$ are equal to:

$$\sum_{m=1}^{M} \hat{U}_m(\mathcal{K}_j^n)\Big(- \int_{\mathcal{K}_j^n} \frac{\partial \psi_l}{\partial t} \psi_m d\mathcal{K} + \int_{K_j^{n+1}} \psi_l(t_{n+1}^-, \bar{x}) \psi_m(t_{n+1}^-, \bar{x}) dK$$
$$+ \int_{\mathcal{K}_j^n} \frac{\partial \psi_l}{\partial x_k} \mathfrak{D}_{kp}(u_h) \frac{\partial \psi_m}{\partial x_p} d\mathcal{K}\Big) - \int_{K_j^n} u_h(t_n^-, \bar{x}) \psi_l(t_n^+, \bar{x}) dK$$
$$+ \int_{\mathcal{Q}_j^n} \psi_l H(u_h^-, u_h^+, n^-) d\mathcal{Q} - \int_{\mathcal{K}_j^n} \frac{\partial \psi_l}{\partial \bar{x}_i} \mathcal{F}_i(u_h) d\mathcal{K} = 0, \qquad l = 1, \cdots, M. \tag{18}$$

The algebraic system given by (17)-(18) is in general non-linear, except for the case of a linear advection equation when $f(u) = au$, with a a constant. The solution of this non-linear system of equations will be discussed in Section 2.4.

2.3 Stabilization operator

The discontinuous Galerkin finite element method without stabilization operator does not guarantee monotone solutions around discontinuities and sharp gradients. In these regions numerical oscillations develop when polynomials of degree one or higher are used. In order to prevent these numerical oscillations frequently a slope limiter is used which reduces the slope of u_h in regions where the solution is oscillatory. The use of a slope limiter in combination with a DG method has become quite popular, but also has serious disadvantages. It may result in an unnecessary reduction in accuracy in smooth parts of the flow field and prevents convergence to steady state, which is particularly important when an implicit time integration method is used. The problems in obtaining a steady state solution originate from an inconsistency in the combination of a discontinuous Galerkin discretization and a slope limiter. Since the limited solution does not satisfy the steady state of the discontinuous Galerkin equations, it is not possible to reduce the residual to machine accuracy. Instead, the scheme tries to converge to the unlimited solution, which suffers, however, from numerical oscillations, and the limiter must remain active to prevent this, resulting in limit cycle behavior.

A better alternative is provided by adding a stabilization operator to the DG discretization, such as proposed by Cockburn and Gremaud [17] and Jaffre, Johnson and Szepessy [25].

The stabilization operator uses the jump in the polynomial representation at the element faces in the discontinuous Galerkin discretization and the element residual. In this way optimal use is made of the information contained in a DG discretization and we maintain the compact stencil of the discontinuous Galerkin method.

The effectiveness of the stabilization operator in (16) strongly depends on the artificial viscosity matrix $\mathfrak{D}(u_h) \in \mathbb{R}^{(d+1)\times(d+1)}$. The definition of the artificial viscosity matrix is more straightforward if the stabilization operator acts independently in all computational coordinate directions. This is achieved by introducing the artificial viscosity matrix $\tilde{\mathfrak{D}} \in \mathbb{R}^{(d+1)\times(d+1)}$ in computational space using the relation:

$$\mathfrak{D}(u_h|_{\mathcal{K}^n}, u_h^*|_{\mathcal{K}^n}) = R^T \, \tilde{\mathfrak{D}}(u_h|_{\mathcal{K}^n}, u_h^*|_{\mathcal{K}^n}) \, R, \tag{19}$$

with $u_h^*|_{\mathcal{K}^n}$ the solution data in the neighboring elements of K^n. The matrix $R \in \mathbb{R}^{(d+1)\times(d+1)}$ is defined as:

$$R = 2\, H^{-1} \, \mathrm{grad}\, G_{\mathcal{K}}. \tag{20}$$

The matrix $H \in \mathbb{R}^{(d+1)\times(d+1)}$ is introduced to ensure that both $\mathfrak{D}$ and $\tilde{\mathfrak{D}}$ have the same mesh dependence as a function of h_i, and is defined as:

$$H = \operatorname{diag}(h_0, h_1, \cdots, h_d),$$

with $h_i \in \mathbb{R}^+$ the leading terms of the expansion of the mapping G_K^n (7) in the computational coordinates ξ_i, $(0 \leq i \leq d)$. The multiplication with the factor two in (20) ensures that for orthogonal cells the matrix R is the rotation matrix from the computational space to the physical space. The stabilization operator in (16) can now be further evaluated, resulting in:

$$\begin{aligned}\int_{\mathcal{K}^n} \frac{\partial \psi_l}{\partial x_k} R_{pk} \tilde{\mathfrak{D}}_{pq}(u_h|_{\mathcal{K}^n}, u_h^*|_{\mathcal{K}^n}) R_{ql} \frac{\partial \psi_m}{\partial x_l} d\mathcal{K} \\ = \int_{\hat{\mathcal{K}}} (H^{-1})_{pi} \frac{\partial \hat{\psi}_l}{\partial \xi_i} \tilde{\mathfrak{D}}_{pq}(u_h|_{\mathcal{K}^n}, u_h^*|_{\mathcal{K}^n}) (H^{-1})_{qj} \frac{\partial \hat{\psi}_m}{\partial \xi_j} |J_{G_\mathcal{K}}| d\hat{\mathcal{K}},\end{aligned}$$

where we used the relation: $(\operatorname{grad} G_{\mathcal{K}}^n)_{ij} = \partial x_j / \partial \xi_i$ and made the assumption that $\tilde{\mathfrak{D}}$ is constant in each element.

The stabilization operator should act only in areas with discontinuities or when the mesh resolution is insufficient. This requirement can be directly coupled to the jump in the solution across element faces and the element residual, respectively, both of which are readily available in the discontinuous Galerkin discretization. In regions with smooth solutions these contributions are of the order of the truncation error and will therefore not reduce the accuracy in these regions.

For problems with discontinuities the artificial viscosity model proposed and analyzed by Jaffre, Johnson and Szepessy [25] is useful. In this model both the jumps at the element faces and the element residual are used to define the artificial viscosity:

$$\begin{aligned}\tilde{\mathfrak{D}}_{qq}(u_h|_{\mathcal{K}^n}, u_h^*|_{\mathcal{K}^n}) &= \max\big(C_2 h_{\mathcal{K}}^{2-\beta} R_q(u_h|_{\mathcal{K}^n}, u_h^*|_{\mathcal{K}^n}), C_1 h_{\mathcal{K}}^{\frac{3}{2}}\big), \qquad && q = 1, \cdots, d, \\ &= 0, && \text{otherwise},\end{aligned}$$

with

$$\begin{aligned}R(u_h|_{\mathcal{K}^n}, u_h^*|_{\mathcal{K}^n}) = &\Big|\sum_{k=0}^{d} \frac{\partial \mathcal{F}(u_h)}{\partial u_h} \frac{\partial u_h(G_{\mathcal{K}}(0))}{\partial x_k}\Big| + C_0 \big| u_h^+(x_{(K^n)}) - u_h^-(x_{(K^{n+1})})\big| / h_{\mathcal{K}} + \\ &\sum_{m=1}^{M_Q} \frac{1}{h_{\mathcal{K}}} \big| \bar{n}_{\mathcal{K}}^T f(u_h^+(x_{(m)})) - \bar{n}_{\mathcal{K}}^T f(u_h^-(x_{(m)}))\big|, \end{aligned} \qquad (21)$$

with $h_{\mathcal{K}} = \sqrt{h_0^2 + h_1^2 + \cdots + h_d^2}$, x_{K^n}, the diameter of the space-time element, $x_{(K^n)}$, $x_{(K^{n+1})}$ the midpoints of the space-time faces at times $t = t_n$ and t_{n+1}, respectively, and M_Q the midpoints of the other space-time faces. The coefficients β, C_0, C_1 and C_2 are positive constants and set equal to $C_0 = 1.2$, $C_1 = 0.1$, $C_2 = 1.0$ and $\beta = 0.1$. For stronger shocks the addition of the quasi-linear form of the conservation law, which is the first contribution on the righthand side of (21), significantly improves the robustness of the numerical scheme since this contribution detects discontinuities very well. Numerical tests showed that the contributions of the element residual of the quasi-linear equations and the contributions in the jump of the flux at the element faces are equally important.

2.4 Solution of algebraic equations for the DG expansion coefficients

The space-time DG formulation results in an implicit time-integration scheme. The equations for the DG expansion coefficients (17)-(18) in the space-time slab $\mathcal{E}^n$ can be represented symbolically as an equation for $\hat{U}^n(\mathcal{K})$:

$$\mathcal{L}(\hat{U}^n(\mathcal{K}); \hat{U}^{n-1}(\mathcal{K})) = 0, \tag{22}$$

with $\hat{U}^{n-1}(\mathcal{K})$ the expansion coefficients in the previous time slab $\mathcal{E}^{n-1}$. In general the equations for the expansion coefficients will be non-linear, only for the flux $f(u) = au$, with a a constant, a linear system will be obtained. There are several ways to solve these non-linear equations. A standard procedure would be to apply a Newton method, but this technique has several disadvantages. In the first place computing the Jacobian matrix, either analytically or numerically, is a non-trivial and computationally expensive task. Also, efficient linear algebra techniques need to be used to solve the linear system. Considering its size this generally have to be iterative, Krylov subspace methods, but finding good preconditioners for the linear system, which strongly influence both the convergence rate and robustness, is difficult. Another disadvantage of a Newton method is that the locality of the DG discretization is lost, because a large global linear system must be solved. This particularly complicates the implementation on a parallel computer.

An alternative technique to solve the non-linear system of algebraic equations (17)-(18) is to use a pseudo-time integration method. In this technique a pseudo-time derivative of the expansion coefficients is added to (22) and this equation is solved by marching the solution with a Runge-Kutta method to a steady state:

$$\frac{\partial \hat{U}^*(\mathcal{K})}{\partial \tau} = \frac{1}{\triangle t}\mathcal{L}(\hat{U}^*(\mathcal{K}); \hat{U}^{n-1}(\mathcal{K})).$$

At steady state, when $\frac{\partial \hat{U}^*(\mathcal{K})}{\partial \tau} = 0$, then $\hat{U}^n(\mathcal{K}) = \hat{U}^*(\mathcal{K})$.

The pseudo-time integration scheme uses a point-implicit five stage Runge-Kutta (RK) scheme, which can be summarized as:

Algorithm 1. Runge-Kutta algorithm for pseudo-time integration.

1. Initialize the first Runge-Kutta stage: $\hat{V}^{(0)} = \hat{U}^{n-1}$.

2. Do for all stages $s = 1$ to 5:

$$(1 + \alpha_s \bar{\lambda})\hat{V}^{(s)} = \hat{V}^{(0)} + \alpha_s \bar{\lambda}\Big(\hat{V}^{(s-1)} - \mathcal{L}^k(\hat{V}^{(s-1)}, \hat{U}^{n-1})\Big) \tag{23}$$

3. End do

4. Update solution: $\hat{U}^n = \hat{V}^{(5)}$.

Here, λ is defined as $\bar{\lambda} = \triangle\tau/\triangle t$ and the Runge-Kutta coefficients are equal to $\alpha_1 = 0.0791451$, $\alpha_2 = 0.163551$, $\alpha_3 = 0.283663$, $\alpha_4 = 0.5$, and $\alpha_5 = 1.0$.

This pseudo-time integration technique is very simple to implement and preserves the locality of the DG discretization. It will, however, only be efficient when fast convergence to steady is achieved. For this purpose the coefficients in the Runge-Kutta scheme have been optimized to damp the transients in the pseudo-time integration as quickly as possible and to allow large pseudo-time steps. In addition, the use of a point implicit RK scheme ensures that the pseudo-time integration method is stable for any positive value of λ. More details about the RK scheme will be given in the next section. Optimizing the Runge-Kutta scheme is, however, not sufficient to obtain an efficient solver. Convergence to steady state is therefore further accelerated using a multigrid technique. In this method the original fine mesh is coarsened a number of times and the solution on the coarse meshes is used to accelerate convergence to steady state on the fine mesh. For the details of the multigrid algorithm, see van der Vegt and van der Ven [42] and Klaij et al. [28] since they are beyond the scope of these notes.

2.5 Stability analysis of the pseudo-time integration method for the linear advection equation

A simple example of a space-time DG discretization is obtained by considering the linear advection equation:

$$u_t + au_x = 0,$$

with $a > 0$. After some lengthy algebra, the space-time discontinuous Galerkin discretization for the linear advection equation using linear basis functions and a mesh with grid velocities $s_j \leq a$, $j = 1, \cdots N$, with N the number of mesh points, can be represented in matrix form as:

$$\mathcal{A}\hat{U}(\mathcal{K}_j^n) - \mathcal{B}\hat{U}(\mathcal{K}_{j-1}^n) = \mathcal{C}\hat{U}(\mathcal{K}_j^{n-1}),$$

with:

$$\mathcal{A} = \begin{pmatrix} \triangle x_j^{n+1} + c^n_{j+\frac{1}{2}} & c^n_{j+\frac{1}{2}} & -c^n_{j+\frac{1}{2}} \\ 2a_1 + c^n_{j+\frac{1}{2}} - 2a\triangle t_n & \frac{1}{3}a_2 + c^n_{j+\frac{1}{2}} + d_{11} & -2a_1 - c^n_{j+\frac{1}{2}} + 2a\triangle t_n \\ -\triangle x_j^n - \triangle x_j^{n+1} - c^n_{j+\frac{1}{2}} & -c^n_{j+\frac{1}{2}} & \frac{2}{3}a_3 + \frac{4}{3}c^n_{j+\frac{1}{2}} + d_{22} \end{pmatrix}$$

$$\mathcal{B} = \begin{pmatrix} c^n_{j-\frac{1}{2}} & c^n_{j-\frac{1}{2}} & -c^n_{j-\frac{1}{2}} \\ -c^n_{j-\frac{1}{2}} & -c^n_{j-\frac{1}{2}} & c^n_{j-\frac{1}{2}} \\ -c^n_{j-\frac{1}{2}} & -c^n_{j-\frac{1}{2}} & \frac{4}{3}c^n_{j-\frac{1}{2}} \end{pmatrix}, \qquad \mathcal{C} = \begin{pmatrix} \triangle x_j^n & 0 & 0 \\ 0 & \frac{1}{3}\triangle x_j^n & 0 \\ -2\triangle x_j^n & 0 & 0 \end{pmatrix},$$

with $\triangle x_j^n = x_{j+1}^n - x_j^n$, $\bar{x}_j^n = \frac{1}{2}(x_j^n + x_{j+1}^n)$, $a_1 = \bar{x}_j^{n+1} - \bar{x}_j^n$, $a_2 = 2\triangle x_j^{n+1} - \triangle x_j^n$, $a_3 = 2\triangle x_j^n + \triangle x_j^{n+1}$, $c^n_{j\pm\frac{1}{2}} = \triangle t_n(a - s_{j\pm\frac{1}{2}})$, and $s^n_{j+\frac{1}{2}} = (x_{j+1}^{n+1} - x_{j+1}^n)/\triangle t_n$. Here x_j^n and x_{j+1}^n denote the begin and end points of the element at time t_n, respectively. The terms d_{11} and d_{22} are determined by the artificial dissipation operator.

The linear advection equation provides a nice model problem to study the stability of the pseudo-time integration method. For this purpose, we assume periodic boundary conditions and use Fourier analysis to investigate the properties of the Runge-Kutta scheme. If we assume that the mesh size is uniform and the time step, element size, and velocity remain constant, i.e. $\triangle t = \triangle t_n$, $\Delta x = \triangle x_j^{n+1} = \triangle x_j^n$, and $s = s^n_{j-\frac{1}{2}} = s^n_{j+\frac{1}{2}}$ for all j and n, and set the artificial viscosity coefficients equal to zero, then the operator $\mathcal{L}$ can be expressed as:

$$\mathcal{L}(\hat{U}^n; \hat{U}^{n-1}) = \mathcal{A}\hat{U}(\mathcal{K}_j^n) - \mathcal{B}\hat{U}(\mathcal{K}_{j-1}^n) - \mathcal{C}\hat{U}(\mathcal{K}_j^{n-1}), \tag{24}$$

with the matrices $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathbb{R}^{3\times 3}$ defined as:

$$\mathcal{A} = \begin{pmatrix} 1+\delta & \delta & -\delta \\ -\delta & \frac{1}{3}+\delta & \delta \\ -2-\delta & -\delta & 2+\frac{4}{3}\delta \end{pmatrix}, \qquad \mathcal{B} = \begin{pmatrix} \delta & \delta & -\delta \\ -\delta & -\delta & \delta \\ -\delta & -\delta & \frac{4}{3}\delta \end{pmatrix}, \qquad \mathcal{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ -2 & 0 & 0 \end{pmatrix},$$

where $\delta = \triangle t(a-s)/\triangle x$ and $s \leq a$.

Consider the spatial Fourier mode $\hat{U}(\mathcal{K}_j^n) = e^{i\theta j}\hat{U}^F$, and introduce this into (24), then the stability of the pseudo-time integration algorithm is determined by the equation:

$$\frac{d\hat{U}^F}{d\tau} = -\frac{1}{\Delta t}\mathcal{P}(\theta)\hat{U}^F,$$

with $\mathcal{P}(\theta) = \mathcal{A} - e^{-i\theta}\mathcal{B}$. Since the linear advection equation is hyperbolic the matrix $\mathcal{P}$ can be written as: $\mathcal{P} = QMQ^{-1}$, with Q the matrix of the right eigenvectors of $\mathcal{P}$ and M a diagonal matrix with the eigenvalues $\mu_m(\theta)$ of $\mathcal{P}(\theta)$. Introducing a new vector $\hat{V}^F = Q^{-1}\hat{U}^F$, we obtain a system of independent ODEs:

$$\frac{d\hat{V}_m^F}{d\tau} = -\frac{\mu_m(\theta)}{\triangle t}\hat{V}_m^F, \qquad \text{for } m = 0, 1, 2.$$

This system of ordinary differential equations is solved with the point-implicit Runge-Kutta scheme given by Algorithm 1. The amplification factor $G(z)$ is defined recursively as:

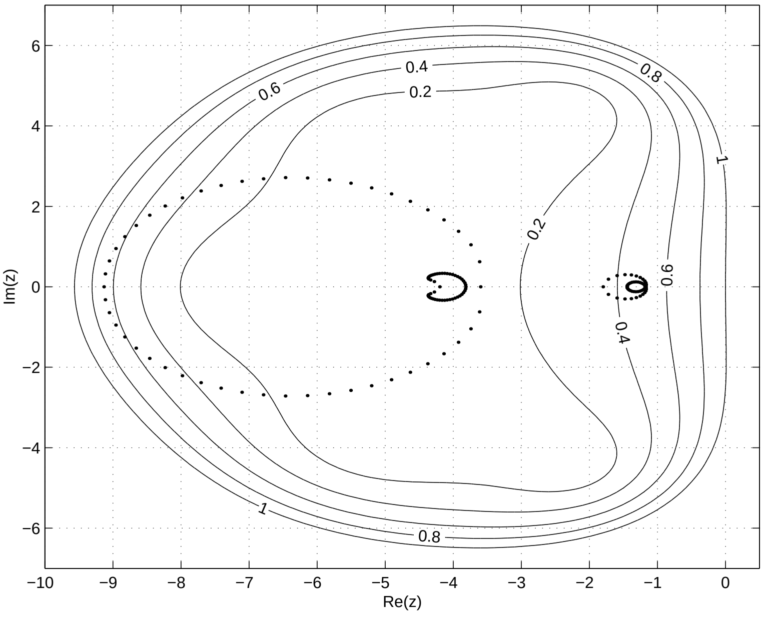
```
$G(z) = 1$
For $s = 1$ to $5$
    $G(z) = \dfrac{1 + \alpha_s(\bar{\lambda} + z)G(z)}{1 + \alpha_s\bar{\lambda}}$
End for
```

The pseudo-time integration method is stable if the amplification factor G satisfies the condition $|G(z_m(\theta))| \leq 1$, for $m = 0, 1, 2$; $\theta \in [0, 2\pi)$ with $z_m(\theta)$ defined as $z_m(\theta) = -\frac{\Delta\tau}{\Delta t}\mu_m(\theta)$. The stability is analyzed for different values of the physical and pseudo-time step CFL-numbers (defined as $CFL_{\triangle t} = a\triangle t/\triangle x$ and $CFL_{\triangle\tau} = a\triangle\tau/\triangle x$, respectively), and the ratio s/a.

In Figure 6 contour values of the stability domain $|G(z)| \leq 1$ for the 5-stage point-implicit Runge-Kutta scheme with optimized coefficients given by Algorithm 1 are shown for the physical CFL numbers $CFL_{\triangle t} = 1$ and 100, respectively. Also shown are the loci of the eigenvalues $z_m(\theta)$, $\theta \in [0, 2\pi)$, which must be inside the stability region to ensure the stability of the pseudo-time integration. For $CFL_{\triangle t} = 1$ the Runge-Kutta scheme is stable for $CFL_{\triangle\tau} \leq 1.94$ and for $CFL_{\triangle t} = 100$ the pseudo-time step CFL number must be less than $CFL_{\triangle\tau} \leq 1.85$, which is unchanged for larger values of $CFL_{\triangle t}$. The large stability domain and excellent smoothing properties of the point-implicit Runge-Kutta method for small values of the physical time step CFL number is important for time-accurate simulations.

In Figure 7(top) the effect of not using the point-implicit treatment of $\hat{V}$ in Algorithm 1 is shown for $CFL_{\triangle t} = 1$. In this case (23) is replaced with

$$\hat{V}^{(s)} = \hat{V}^{(0)} - \frac{\alpha_s\bar{\lambda}}{|K^n|}\mathcal{L}^k(\hat{V}^{(s-1)}, \hat{U}^{n-1})\Big) \tag{25}$$



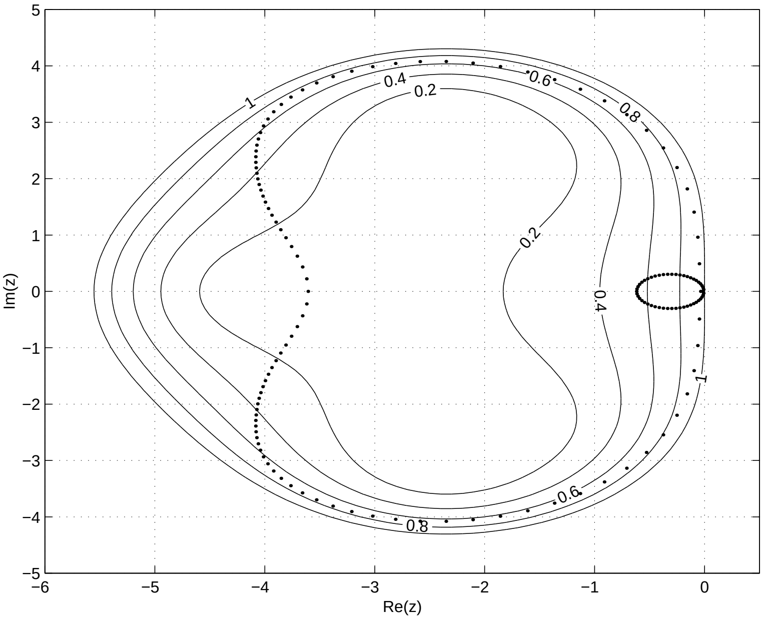


Figure 6: Loci of the eigenvalues $z_m(\theta)$, $\theta \in [0, 2\pi)$, (dots) of the DG discretization of $u_t + au_x = 0$ and the stability domain of the 5-stage point-implicit Runge-Kutta method with optimized coefficients. $CFL_{\triangle t} = 1.0$, $CFL_{\triangle \tau} = 1.8$ (top), $CFL_{\triangle t} = 100.0$, $CFL_{\triangle \tau} = 1.8$ (bottom), no grid velocity.

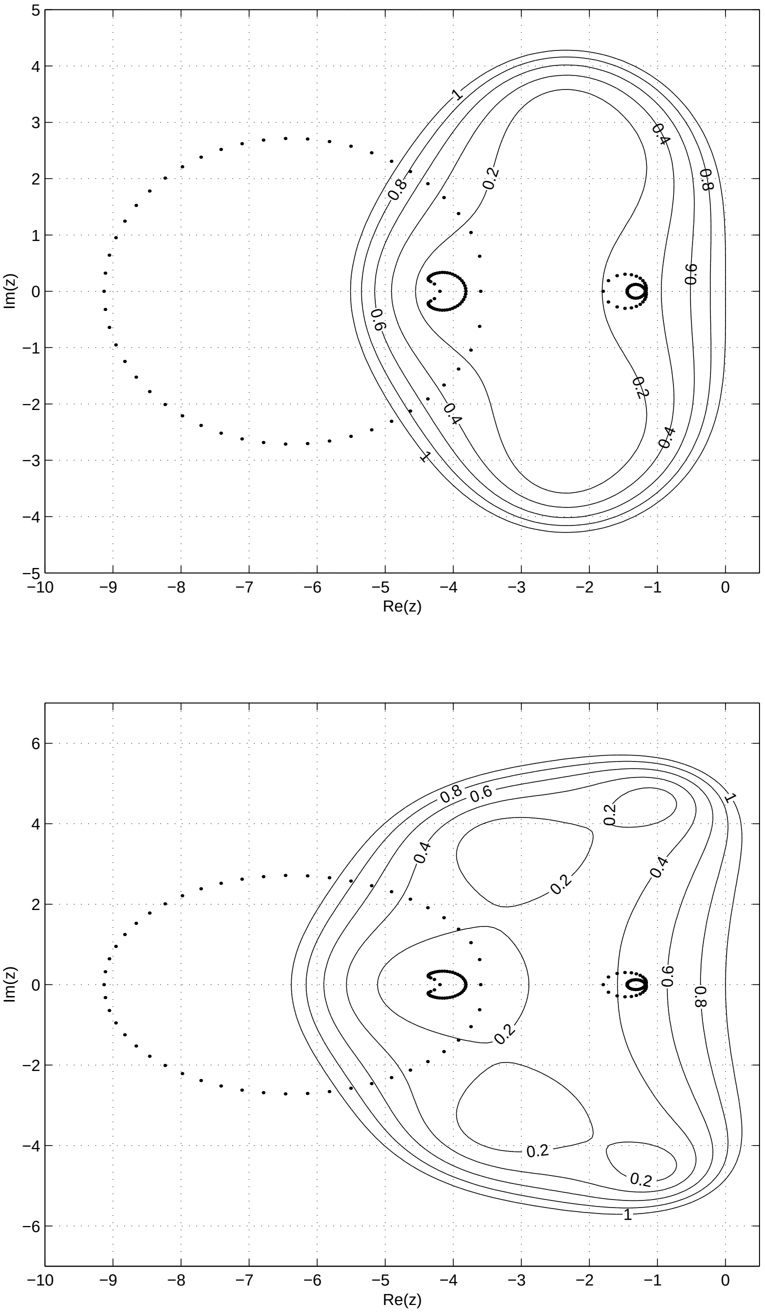


Figure 7: Loci of the eigenvalues $z_m(\theta)$, $\theta \in [0, 2\pi)$, (dots) of the DG discretization of $u_t + au_x = 0$ and the stability domain of the explicit 5-stage Runge-Kutta method (25) with optimized coefficients (top) and the five stage point-implicit Jameson Runge-Kutta scheme (bottom). $CFL_{\triangle t} = 1.0$. $CFL_{\triangle \tau} = 1.8$, no grid velocity.

and the resulting scheme is unstable when the same pseudo-time step CFL numbers are used as for the point-implicit scheme. For small physical time step CFL numbers the stabilizing effect of this technique is very large and the pseudo-time step CFL number must be reduced to 1.08 to ensure stability when the explicit Runge-Kutta scheme (25) is used. For physical CFL numbers larger than 100 the effect of the point-implicit Runge-Kutta scheme is, however, negligible.

The effect of using optimized coefficients in the Runge-Kutta scheme is also large, as can be seen in Figure 7(bottom) where the stability contours for the point-implicit Runge-Kutta scheme with coefficients $\alpha_s = \frac{1}{4}, \frac{1}{6}, \frac{3}{8}, \frac{1}{2}, 1$ for the stages $s = 1, \cdots, 5$ are shown. This are the coefficients for the Jameson Runge-Kutta scheme, which is a popular Runge-Kutta method in computational fluid dynamics and also frequently used as a smoother in multigrid algorithms. For this Runge-Kutta scheme the pseudo-time step CFL number must be reduced to $CFL_{\triangle\tau} \leq 0.88$, when the physical CFL number is equal to $CFL_{\triangle t} = 1$. When the physical CFL number is equal to $CFL_{\triangle t} = 100$ then the pseudo-time step CFL number must be reduced to $CFL_{\triangle\tau} \leq 0.95$ for the Jameson Runge-Kutta scheme. The effect of grid velocity is stabilizing if the grid velocity is in the range $0 \leq s \leq a$. This is a direct consequence of the relation $\delta = CFL_{\triangle t}(1 - s/a)$. When the grid velocity is in this range then it reduces the effective physical time step CFL number and since the pseudo-time integration has a larger stability domain for smaller values of $CFL_{\triangle t}$ this improves stability.

3 Space-time methods for parabolic scalar conservation laws

3.1 Introduction

In this section we will present the extension of the space-time discontinuous Galerkin method for hyperbolic partial differential equations in one dimension, discussed in Section 2, to parabolic scalar conservation laws in multiple dimensions. In addition, the relation between the space-time DG method and the Arbitrary Lagrangian Eulerian (ALE) formulation will be explained.

The development of discontinuous Galerkin finite element methods for elliptic and parabolic partial differential equations initially started in the seventies with the research on interior penalty methods, see for example [3; 22; 47]. The interior penalty method requires, however, a stabilization term with a mesh dependent constant which needs to be properly chosen to ensure stability. Also, the interior penalty method is considerably more complicated than the continuous Galerkin finite element method and did not receive much attention. The suitability of discontinuous Galerkin methods for *hp*-adaptation and parallel computing, which was clearly demonstrated for hyperbolic partial differential equations, recently initiated the development of several new discontinuous Galerkin methods for elliptic and parabolic partial differential equations. The extension of the DG method for hyperbolic partial differential equations to convection-diffusion problems was made by Bassi and Rebay [5], which developed a discontinuous Galerkin finite element method for the compressible Navier-Stokes equations. This method suffered, however, from a weak instability which was removed by Brezzi [13; 14]. During the same period Baumann and Oden [8; 30; 9] developed a DG algorithm without a free parameter, but this algorithm is suboptimal in accuracy and unstable for linear polynomials. A better alternative is provided by the local discontinuous Galerkin method developed by Cockburn and Shu

[20], see also [15; 18]. A detailed survey and analysis of all these method is provided in Arnold et al. [4], which gives a clear framework for the DG methods for elliptic partial differential equations developed so far.

The extension of the space-time DG method to parabolic pde's discussed in this section will be based on the DG algorithms developed by Brezzi [13; 14] and Bassi and Rebay [7; 6]. The main benefit of this approach is that no mesh dependent parameters are required to guarantee the stability of the DG algorithm. In particular, we will consider the so-called primal formulation which does not require the use of auxiliary variables, which are eliminated using lifting operators. The resulting algorithm has optimal order of accuracy and results in a very compact stencil, containing only contributions from nearest neighboring elements. The discussion in this chapter is based on Sudirham, van der Vegt and van Damme [35] to which we refer for more details, including an extensive error and stability analysis of the space-time discontinuous Galerkin method for the linear advection diffusion equation.

3.2 Space-time formulation

In this section we consider a parabolic scalar conservation law defined on a time dependent domain and its formulation in the space-time framework. Let Ω_t be an open, bounded domain in $\mathbb{R}^d$, with d the number of spatial dimensions. The closure of Ω_t is $\overline{\Omega_t}$ and the boundary of Ω_t is denoted by $\partial\Omega_t$. The subscript t denotes the domain at time t as we consider the geometry of the spatial domain to be time-dependent. The outward normal vector to $\partial\Omega_t$ is denoted by $\bar{n} = (n_1, \dots, n_d)$. Denoting $\bar{x} = (x_1, \dots, x_d)$ as the spatial variables, we consider a time-dependent parabolic scalar conservation law:

$$\frac{\partial u}{\partial t} + \sum_{i=1}^{d} \frac{\partial}{\partial x_i} f_i(u(t,\bar{x})) - \sum_{i,j=1}^{d} \frac{\partial}{\partial x_j}\Big(D_{ij}(t,\bar{x})\frac{\partial u}{\partial x_i}\Big) = 0, \text{ in } \Omega_t, \tag{26}$$

where f_i, $i = 1, \cdots, d$ are real-valued flux functions on $\bar{\Omega}_t$ and u is a scalar quantity. Furthermore, $D \in \mathbb{R}^{d\times d}$ is a symmetric matrix of diffusion coefficients on $\bar{\Omega}_t$ whose entries are continuous real-valued functions. This matrix is positive definite in Ω_t and positive semi-definite on $\partial\Omega_t$.

In the space-time discretization we directly consider a domain in $\mathbb{R}^{d+1}$. A point $x \in \mathbb{R}^{d+1}$ has coordinates $(x_0, \bar{x})$, with $x_0 = t$ representing time. We then define the space-time domain $\mathcal{E} \subset \mathbb{R}^{d+1}$. The boundary of the space-time domain $\partial\mathcal{E}$ consists of the hypersurfaces $\Omega_0 := \{x \in \partial\mathcal{E} \mid x_0 = 0\}$, $\Omega_T := \{x \in \partial\mathcal{E} \mid x_0 = T\}$, and $\mathcal{Q} := \{x \in \partial\mathcal{E} \mid 0 < x_0 < T\}$. We reformulate the parabolic scalar conservation law now in the space-time framework. First, we introduce the convective flux $\mathcal{F} \in \mathbb{R}^{d+1}$ and the symmetric matrix $A \in \mathbb{R}^{(d+1)\times(d+1)}$ as:

$$\mathcal{F}(u) = \big(u, f_1(u), \cdots, f_d(u)\big),$$

$$A = \begin{pmatrix} 0 & 0 \\ 0 & D \end{pmatrix}.$$

Then the parabolic scalar conservation law (26) can be transformed into a space-time formulation as:

$$-\nabla\cdot(-\mathcal{F}(u) + A\nabla u) = 0 \quad \text{in} \quad \mathcal{E}, \tag{27}$$

where $\nabla = \big(\frac{\partial}{\partial x_0}, \frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_d}\big)$ denotes the gradient operator in $\mathbb{R}^{d+1}$.

As different boundary conditions are imposed on $\partial\mathcal{E}$, we discuss in more detail the subdivision of $\partial\mathcal{E}$ into different parts. The boundary $\partial\mathcal{E}$ is divided into disjoint boundary subsets Γ_S, Γ_-, and Γ_+, where each subset is defined as follows:

$$\Gamma_S := \{x \in \partial\mathcal{E} : \bar{n}^T D\bar{n} > 0\},$$
$$\Gamma_- := \{x \in \partial\mathcal{E} \setminus \Gamma_S : \lambda(u) < 0\}, \qquad \Gamma_+ := \{x \in \partial\mathcal{E} \setminus \Gamma_S : \lambda(u) \geq 0\},$$

where the unit outward normal vector at $\partial\mathcal{E}$ is denoted with n and $\lambda(u) = \frac{d}{du}(F(u) \cdot n)$. The subscript S denotes the part of $\partial\mathcal{E}$ where matrix D is symmetric positive definite, while the subscripts $-$ and $+$ denote the inflow and outflow boundaries, respectively. We assume that Γ_S has a non-zero surface measure. Note that $\partial\mathcal{E} = \Gamma_S \cup \Gamma_- \cup \Gamma_+$. We subdivide Γ_S further into two sets: $\Gamma_S = \Gamma_{DS} \cup \Gamma_M$, with Γ_{DS} the part of Γ_S with a Dirichlet boundary condition and Γ_M the part of Γ_S with a mixed boundary condition. We also subdivide Γ_- into two parts: $\Gamma_- = \Gamma_{DB} \cup \Omega_0$, with Γ_{DB} the part of Γ_- with a Dirichlet boundary condition and Ω_0 the part of Γ_- with the initial condition. Note that $\Gamma_D = \Gamma_{DS} \cup \Gamma_{DB} \subset \partial\mathcal{E}$ is the part of the space-time domain boundary with a Dirichlet boundary condition. The boundary conditions on different parts of $\partial\mathcal{E}$ are written as

$$\begin{aligned} u = u_0 &\quad \text{on } \Omega_0, \\ u = g_D &\quad \text{on } \Gamma_D, \\ \alpha u + n \cdot (A\nabla u) = g_M &\quad \text{on } \Gamma_M, \end{aligned} \tag{28}$$

with $\alpha \geq 0$ and u_0, g_D, g_M given functions defined on the boundary. There is no boundary condition imposed on Γ_+.

3.3 Space-time description, finite element spaces and trace operators

3.3.1 Definition of space-time slabs, elements and faces

In this section we give a description of the space-time slabs, elements and faces used in the DG discretization. First, consider the time interval $\mathcal{I} = [0, T]$, partitioned by an ordered series of time levels $t_0 = 0 < t_1 < \ldots < t_{N_t} = T$. Denoting the nth time interval as $I_n = (t_n, t_{n+1})$, we have $\mathcal{I} = \cup_n \bar{I}_n$. The length of I_n is defined as $\triangle_n t = t_{n+1} - t_n$. Let Ω_{t_n} be an approximation to the spatial domain Ω at t_n for each $n = 0, \ldots, N_t$. A space-time slab is defined as the domain $\mathcal{E}^n = \mathcal{E} \cap (I_n \times \mathbb{R}^d)$ with boundaries Ω_{t_n}, $\Omega_{t_{n+1}}$ and $\mathcal{Q}^n = \partial\mathcal{E}^n \setminus (\Omega_{t_n} \cup \Omega_{t_{n+1}})$.

We now describe the construction of the space-time elements $\mathcal{K}$ in the space-time slab $\mathcal{E}^n$. Let the domain Ω_{t_n} be divided into N_n non-overlapping spatial elements K^n. At t_{n+1} the spatial elements K^{n+1} are obtained by mapping the elements K^n to their new position. Each space-time element $\mathcal{K}$ is obtained by connecting elements K^n and K^{n+1} using linear interpolation in time. A sketch of the space-time slab $\mathcal{E}^n$ and element $\mathcal{K}$ for two spatial dimensions is shown in Fig. 8. We denote by $h_{\mathcal{K}}$ the radius of the smallest sphere containing each element $\mathcal{K}$. The element boundary $\partial\mathcal{K}$ is the union of open faces of $\mathcal{K}$, which contains three parts K^n, K^{n+1}, and $\mathcal{Q}^n_{\mathcal{K}} = \partial\mathcal{K} \setminus (K^n \cup K^{n+1})$. We denote by $n_{\mathcal{K}}$ the unit outward space-time normal vector on $\partial\mathcal{K}$. The definition of the space-time domain is completed with the tessellation $\mathcal{T}^n_h$, which consists of all space-time elements in $\mathcal{E}^n$, and $\mathcal{T}_h = \cup_n \mathcal{T}^n_h$, which consists of all space-time elements in $\mathcal{E}$.

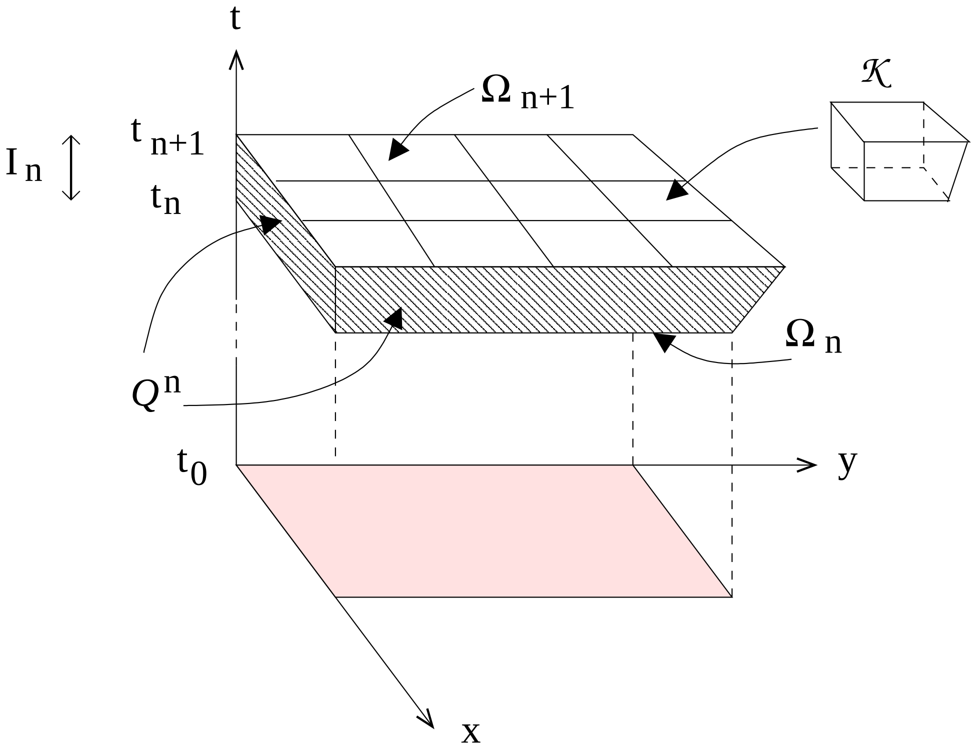


Figure 8: Space-time slab $\mathcal{E}^n$ with space-time element $\mathcal{K}$.

Next, we consider several sets of faces S. The set of all faces in $\bar{\mathcal{E}}$ is denoted with $\mathcal{F}$, the set of all interior faces in $\mathcal{E}$ with $\mathcal{F}_{\text{int}}$, and the set of all boundary faces on $\partial\mathcal{E}$ with $\mathcal{F}_{\text{bnd}}$. In the space-time slab $\mathcal{E}^n$ we denote the set of all faces with $\mathcal{F}^n$ and the set of all interior faces with $\mathcal{S}_I^n$. The faces separating two space-time slabs are denoted as $\mathcal{S}_S^n$. Several sets of boundary faces are defined as follows. The set of faces on Γ_{DS} and Γ_{DB} are denoted with $\mathcal{S}_{DS}^n$ and $\mathcal{S}_{DB}^n$, respectively. These sets are grouped into $\mathcal{S}_D^n$. The set of faces with a mixed boundary condition is denoted with $\mathcal{S}_M^n$. The set of faces with either a Dirichlet or a mixed boundary condition is denoted as $\mathcal{S}_{DM}^n$. The sets $\mathcal{S}_I^n$ and $\mathcal{S}_D^n$ are grouped into $\mathcal{S}_{ID}^n$.

Depending on whether the convective flux on $\mathcal{S}_{DS}^n$ is inflow or outflow, we subdivide $\mathcal{S}_{DS}^n$ further into $\mathcal{S}_{DSm}^n$ and $\mathcal{S}_{DSp}^n$, where $\lambda(u) < 0$ on $\mathcal{S}_{DSm}^n$ and $\lambda(u) \geq 0$ on $\mathcal{S}_{DSp}^n$. The sets $\mathcal{S}_{DB}^n$ and $\mathcal{S}_{DSm}^n$ are grouped into $\mathcal{S}_{DBSm}^n$, while the sets $\mathcal{S}_M^n$ and $\mathcal{S}_{DSp}^n$ are grouped into $\mathcal{S}_{MDSp}^n$. These sets are important when we discuss the convective flux in Section 3.4.2.

3.3.2 Finite element spaces and trace operators

For the definition of the finite element spaces we assume that each element $\mathcal{K} \in \mathcal{T}_h$ is an image of a fixed master element $\hat{\mathcal{K}}$, with $\hat{\mathcal{K}}$ an open unit hypercube in $\mathbb{R}^{d+1}$, constructed via the mapping $G_{\mathcal{K}} : \hat{\mathcal{K}} \to \mathcal{K}$, which is supposed to be a diffeomorphism and is a straightforward extension to d-space dimensions of the mapping defined in (7). We now introduce the finite element spaces associated with the tessellation $\mathcal{T}_h$ that will be used in these notes. To each element $\mathcal{K}$ we assign a pair of nonnegative integers $p_{\mathcal{K}} = (p_{t,\mathcal{K}}, p_{s,\mathcal{K}})$ as local polynomial degrees, where the subscripts t and s denote time and space, and collect them into a vector $\mathbf{p} = \{p_{\mathcal{K}} : \mathcal{K} \in \mathcal{T}_h\}$. Defining $\mathcal{Q}_{p_{t,\mathcal{K}},p_{s,\mathcal{K}}}(\hat{\mathcal{K}})$ as the set of all tensor-product polynomials on $\hat{\mathcal{K}}$ of degree $p_{t,\mathcal{K}}$ in the time direction and degree $p_{s,\mathcal{K}}$ in each spatial coordinate direction, we then introduce the finite element space of discontinuous piecewise polynomial functions as

$$V_h^{(p_t,p_s)} := \{v \in L^2(\mathcal{E}) : v|_{\mathcal{K}} \circ G_{\mathcal{K}} \in \mathcal{Q}_{(p_{t,\mathcal{K}},p_{s,\mathcal{K}})}(\hat{\mathcal{K}}), \forall \mathcal{K} \in \mathcal{T}_h\}.$$

In the derivation and analysis of the numerical discretization we also make use of the auxiliary space $\Sigma_h^{(p_t,p_s)}$:

$$\Sigma_h^{(p_t,p_s)} := \{\tau \in L^2(\mathcal{E})^{d+1} : \tau|_{\mathcal{K}} \circ G_{\mathcal{K}} \in [\mathcal{Q}_{(p_{t,\mathcal{K}},p_{s,\mathcal{K}})}(\hat{\mathcal{K}})]^{d+1}, \forall \mathcal{K} \in \mathcal{T}_h\}.$$

The so called traces of $v \in V_h^{(p_t,p_s)}$ on $\partial\mathcal{K}$ are defined as: $v_{\mathcal{K}}^{\pm} = \lim_{\epsilon \downarrow 0} v(x \pm \epsilon n_{\mathcal{K}})$. The traces of $\tau \in \Sigma_h^{(p_t,p_s)}$ are defined similarly.

Next, we define the *average* $\{\!\{\cdot\}\!\}$ and *jump* $[\![\cdot]\!]$ operators as trace operators for the sets $\mathcal{F}_{\text{int}}$ and $\mathcal{F}_{\text{bnd}}$. Note that functions $v \in V_h^{(p_t,p_s)}$ and $\tau \in \Sigma_h^{(p_t,p_s)}$ are in general multivalued on a face $S \in \mathcal{F}_{\text{int}}$. Introducing the functions $v_i := v|_{\mathcal{K}_i}$, $\tau_i := \tau|_{\mathcal{K}_i}, n_i := n|_{\partial\mathcal{K}_i}$, we define the average operator on $S \in \mathcal{F}_{\text{int}}$ as:

$$\{\!\{v\}\!\} = \frac{1}{2}(v_i^- + v_j^-), \quad \{\!\{\tau\}\!\} = \frac{1}{2}(\tau_i^- + \tau_j^-), \quad \text{on}\, S \in \mathcal{F}_{\text{int}},$$

while the jump operator is defined as:

$$[\![v]\!] = v_i^- n_i + v_j^- n_j, \quad [\![\tau]\!] = \tau_i^- \cdot n_i + \tau_j^- \cdot n_j, \text{ on } S \in \mathcal{F}_{\text{int}},$$

with i and j the indices of the elements $\mathcal{K}_i$ and $\mathcal{K}_j$ which connect to the face $S \in \mathcal{F}_{\text{int}}$. On a face $S \in \mathcal{F}_{\text{bnd}}$, the average and jump operators are defined as:

$$\{\!\{v\}\!\} = v^-, \quad \{\!\{\tau\}\!\} = \tau^-, \quad [\![v]\!] = v^- n, \quad [\![\tau]\!] = \tau^- \cdot n, \quad \text{on}\, S \in \mathcal{F}_{\text{bnd}}.$$

Note that the jump $[\![v]\!]$ is a vector parallel to the normal vector n and the jump $[\![\tau]\!]$ is a scalar quantity. We also need the spatial jump operator $\langle\!\langle\cdot\rangle\!\rangle$ for functions $v \in V_h^{(p_t,p_s)}$, which is defined as:

$$\langle\!\langle v\rangle\!\rangle = v_i^- \bar{n}_i + v_j^- \bar{n}_j, \quad \text{on } S \in \mathcal{F}_{\text{int}}, \qquad \langle\!\langle v\rangle\!\rangle = v^- \bar{n}, \quad \text{on } S \in \mathcal{F}_{\text{bnd}}.$$

3.3.3 Lifting operators

In this section we introduce several lifting operators. The lifting operators discussed in this section are similar to the ones introduced in [4; 14]. These operators are required for the derivation of the space-time DG formulation in Section 3.4.

First, we introduce the local lifting operator $r_S : (L^2(S))^{d+1} \to \Sigma_h^{(p_t,p_s)}$ as:

$$\int_{\mathcal{E}} r_S(\phi) \cdot q \,\mathrm{d}\mathcal{E} = -\int_S \phi \cdot \{\!\{q\}\!\} \,\mathrm{d}S, \quad \forall q \in \Sigma_h^{(p_t,p_s)}, \forall S \in \cup_n \mathcal{S}_{ID}^n. \tag{29}$$

The support of the operator r_S is limited to the element(s) that share the face S. Then we introduce the global lifting operator $R : (L^2(\cup_n \mathcal{S}_{ID}^n))^{d+1} \to \Sigma_h^{(p_t,p_s)}$ as:

$$\int_{\mathcal{E}} R(\phi) \cdot q \,\mathrm{d}\mathcal{E} = \sum_S \int_{\mathcal{E}} r_S(\phi) \cdot q \,\mathrm{d}\mathcal{E}, \quad \forall q \in \Sigma_h^{(p_t,p_s)}, \forall S \in \cup_n \mathcal{S}_{ID}^n. \tag{30}$$

We also specify the above lifting operators for the Dirichlet boundary condition. Let $\mathcal{P}$ be the L^2 projection on $\Sigma_h^{(p_t,p_s)}$, and replace ϕ by $\mathcal{P} g_D n$ in (29). Then on faces $S \in \cup_n \mathcal{S}_D^n$ we have

$$\int_{\mathcal{E}} r_S(\mathcal{P} g_D n) \cdot q \,\mathrm{d}\mathcal{E} = -\int_S g_D n \cdot q \,\mathrm{d}S, \quad \forall q \in \Sigma_h^{(p_t,p_s)}, \forall S \in \cup_n \mathcal{S}_D^n. \tag{31}$$

For the global lifting operators, we proceed in a similar way. Using the projection operator $\mathcal{P}$, we replace ϕ by $\mathcal{P}g_D n$ in (30) and (29) to have:

$$\int_{\mathcal{E}} R(\mathcal{P}g_D n) \cdot q \, \mathrm{d}\mathcal{E} = -\sum_S \int_S g_D n \cdot q \, \mathrm{d}S, \quad \forall q \in \Sigma_h^{(p_t,p_s)}, \forall S \in \cup_n \mathcal{S}_D^n. \tag{32}$$

Using (30) and (32), we then introduce $R_{ID} : (L^2(\cup_n \mathcal{S}_{ID}^n))^{d+1} \rightarrow \Sigma_h^{(p_t,p_s)}$ as:

$$R_{ID}(\phi) = R(\phi) - R(\mathcal{P}g_D n). \tag{33}$$

The spatial part of the lifting operators R and r_S, denoted by $\bar{R}$ and $\bar{r}_S$, are obtained by eliminating the first component of R and r_S, respectively.

3.4 Space-time DG discretization for the parabolic scalar conservation law

In this section, we describe the derivation of the space-time DG weak formulation for the parabolic scalar conservation law. As shown in e.g. [4; 14], it is beneficial for a DG discretization to rewrite the second order partial differential equation (27) into a system of first order equations. Following the same approach, we introduce an auxiliary variable $\sigma = A\nabla u$ to obtain the following system of first order equations:

$$\sigma = A\nabla u, \tag{34}$$

$$-\nabla \cdot (-\mathcal{F}(u) + \sigma) = 0. \tag{35}$$

In the next section we discuss the derivation of the weak formulation for (34)-(35) in the space-time framework.

3.4.1 Weak formulation for the auxiliary variable

First, we consider the auxiliary equation (34). By multiplying this equation with an arbitrary test function $\tau \in \Sigma_h^{(p_t,p_s)}$ and integrating over an element $\mathcal{K} \in \mathcal{T}_h$, we obtain:

$$\int_{\mathcal{K}} \sigma \cdot \tau \, \mathrm{d}\mathcal{K} = \int_{\mathcal{K}} A\nabla u \cdot \tau \, \mathrm{d}\mathcal{K}, \quad \forall \tau \in \Sigma_h^{(p_t,p_s)}.$$

Next, we substitute σ and u with their numerical approximation $\sigma_h \in \Sigma_h^{(p_t,p_s)}$ and $u_h \in V_h^{(p_t,p_s)}$. After integration by parts twice and summation over all elements, we have for all $\tau \in \Sigma_h^{(p_t,p_s)}$ the following formulation:

$$\int_{\mathcal{E}} \sigma_h \cdot \tau \, \mathrm{d}\mathcal{E} = \int_{\mathcal{E}} A\nabla_h u_h \cdot \tau \, \mathrm{d}\mathcal{E} + \sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\partial\mathcal{K}} A(\hat{u}_h - u_h^-) n \cdot \tau^- \, \mathrm{d}\partial\mathcal{K}. \tag{36}$$

The variable $\hat{u}_h$ is the *numerical flux* that must be introduced to account for the multivalued trace on $\partial\mathcal{K}$.

We recall the following relation (see [4] relation (3.3)), which holds for vectors τ and scalars ϕ, piecewise smooth on $\mathcal{T}_h$:

$$\sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\partial\mathcal{K}} (\tau \cdot n)\phi \, \mathrm{d}\partial\mathcal{K} = \sum_{S\in\mathcal{F}} \int_S \{\!\{\tau\}\!\} \cdot [\![\phi]\!] \, \mathrm{d}S + \sum_{S\in\mathcal{F}_{\mathrm{int}}} \int_S [\![\tau]\!] \{\!\{\phi\}\!\} \, \mathrm{d}S. \tag{37}$$

When applied to the last contribution in (36) and using the symmetry of the matrix A, this results in

$$\sum_{\mathcal{K}\in\mathcal{T}_h}\int_{\partial\mathcal{K}} A(\hat{u}_h - u_h^-)n\cdot\tau^- \,\mathrm{d}\partial\mathcal{K}$$
$$= \sum_{S\in\mathcal{F}}\int_S [\![\hat{u}_h - u_h]\!]\cdot\{\!\{A\tau\}\!\}\,\mathrm{d}S + \sum_{S\in\mathcal{F}_{\text{int}}}\int_S \{\!\{\hat{u}_h - u_h\}\!\}[\![A\tau]\!]\,\mathrm{d}S. \tag{38}$$

We consider now the choice for the numerical flux $\hat{u}_h$. There are several options listed in [4]. After a thorough study concerning the consistency, conservation properties, and matrix sparsity of each option, we choose the following numerical flux, which is similar to the choices in [6; 13; 14]:

$$\hat{u}_h = \{\!\{u_h\}\!\} \text{ on } S\in\mathcal{F}_{\text{int}}, \quad \hat{u}_h = g_D \text{ on } S\in\cup_n\mathcal{S}_D^n, \quad \hat{u}_h = u_h^- \text{ elsewhere.} \tag{39}$$

Note that on faces $S\in\mathcal{S}_S^n$, which are the element boundaries K^n and K^{n+1}, the normal vector n has values $n = (\pm 1, \underbrace{0,\dots,0}_{d\,\times})$ and thus $An = (\underbrace{0,\dots,0}_{(d+1)\,\times})$. Hence there is no coupling between the space-time slabs. Substituting the choices for the numerical flux (39) into (38) and using the fact that entries of the matrix A are continuous functions, we obtain for each space-time slab $\mathcal{E}^n$:

$$\sum_{\mathcal{K}\in\mathcal{T}_h^n}\int_{\partial\mathcal{K}} A(\hat{u}_h - u_h^-)n\cdot\tau^- \,\mathrm{d}\partial\mathcal{K}$$
$$= -\sum_{S\in\mathcal{S}_{ID}^n}\int_S [\![u_h]\!]\cdot A\{\!\{\tau\}\!\}\,\mathrm{d}S + \sum_{S\in\mathcal{S}_D^n}\int_S g_D n\cdot A\tau\,\mathrm{d}S. \tag{40}$$

After summation over all space-time slabs, and using the symmetry of matrix A we can introduce the lifting operator (33) into (40) to obtain

$$\sum_{\mathcal{K}\in\mathcal{T}_h}\int_{\partial\mathcal{K}} A(\hat{u}_h - u_h^-)n\cdot\tau^- \,\mathrm{d}\partial\mathcal{K} = \int_{\mathcal{E}} AR_{ID}([\![u_h]\!])\cdot\tau\,\mathrm{d}\mathcal{E}. \tag{41}$$

Introducing (41) into (36), we obtain for all $\tau\in\Sigma_h^{(p_t,p_s)}$:

$$\int_{\mathcal{E}}\sigma_h\cdot\tau\,\mathrm{d}\mathcal{E} = \int_{\mathcal{E}} A\nabla_h u_h\cdot\tau\,\mathrm{d}\mathcal{E} + \int_{\mathcal{E}} AR_{ID}([\![u_h]\!])\cdot\tau\,\mathrm{d}\mathcal{E},$$

which implies that we can express $\sigma_h\in\Sigma_h^{(p_t,p_s)}$ as:

$$\sigma_h = A\nabla_h u_h + AR_{ID}([\![u_h]\!]) \quad \text{a.e. } \forall x\in\mathcal{E}. \tag{42}$$

3.4.2 Weak formulation of the parabolic scalar conservation law

The weak formulation for the parabolic scalar conservation law is obtained if we multiply (35) with arbitrary test functions $v\in V_h^{(p_t,p_s)}$, integrate by parts over element $\mathcal{K}$, and then substitute u,σ with their numerical approximation $u_h\in V_h^{(p_t,p_s)}, \sigma_h\in\Sigma_h^{(p_t,p_s)}$:

$$\int_{\mathcal{E}}(-\mathcal{F}(u_h)+\sigma_h)\cdot\nabla_h v\,\mathrm{d}\mathcal{E} - \sum_{\mathcal{K}\in\mathcal{T}_h}\int_{\partial\mathcal{K}}(-\hat{F}_h+\hat{\sigma}_h)\cdot nv^-\,\mathrm{d}\partial\mathcal{K} = 0. \tag{43}$$

Here we replaced $\mathcal{F}(u_h)$, σ_h on $\partial\mathcal{K}$ with the numerical fluxes $\hat{F}_h$, $\hat{\sigma}_h$, to account for the multivalued traces on $\partial\mathcal{K}$.

The next step is to find appropriate choices for the numerical fluxes. We separate the numerical fluxes into an *convective flux* $\hat{F}_h$ and a *diffusive flux* $\hat{\sigma}_h$. For the convective flux, the obvious choice is an upwind flux, as described in [42] and we use here the Lax-Friedrichs flux for convenience. For more details, see [39]. Thus, we write the numerical flux $\hat{F}_h$ as:

$$\hat{F}_h(u_h^-, u_h^+) = \{\!\{\mathcal{F}(u_h)\}\!\} + C_S[\![u_h]\!]. \tag{44}$$

The parameter C_S is chosen as:

$$C_S = \max_{u\in[u_h^-, u_h^+]} |\lambda(u)| \quad \text{on } S \in \mathcal{F}_{\text{int}}. \tag{45}$$

If we substitute τ and ϕ in relation (37) with $\{\!\{\mathcal{F}(u_h)\}\!\} + C_S[\![u_h]\!]$ and v, respectively, the summation over the boundaries $\partial\mathcal{K}$ can be written as a sum over all faces as follows:

$$\begin{aligned}\sum_{\mathcal{K}\in\mathcal{T}_h} &\int_{\partial\mathcal{K}} (\{\!\{\mathcal{F}(u_h)\}\!\} + C_S[\![u_h]\!]) \cdot nv^- \,\mathrm{d}\partial\mathcal{K} \\ &= \sum_{S\in\mathcal{F}_{\text{int}}} \int_S (\{\!\{\mathcal{F}(u_h)\}\!\} + C_S[\![u_h]\!]) \cdot [\![v]\!] \,\mathrm{d}S + \sum_{S\in\mathcal{F}_{\text{bnd}}} \int_S \mathcal{F}(u_h) \cdot nv \,\mathrm{d}S.\end{aligned} \tag{46}$$

Now we consider the numerical flux $\hat{\sigma}_h$. From [4], we have several options for this numerical flux. For similar reasons as in Section 3.4.1, we choose $\hat{\sigma}_h = \{\!\{\sigma_h\}\!\}$, which is the same as in [13; 14]. By replacing $\hat{\sigma}_h$ with $\{\!\{\sigma_h\}\!\}$, then using (37) the contribution with $\hat{\sigma}_h$ in (43) can also be written as a sum over all faces $S \in \mathcal{F}$ as:

$$\sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\partial\mathcal{K}} \{\!\{\hat{\sigma}_h\}\!\} \cdot nv^- \,\mathrm{d}\partial\mathcal{K} = \sum_{S\in\mathcal{F}} \int_S \{\!\{\sigma_h\}\!\} \cdot [\![v]\!] \,\mathrm{d}S. \tag{47}$$

Using (46)-(47) and (42) (to eliminate σ_h), the primal formulation for u_h is obtained:

$$\begin{aligned}&\int_{\mathcal{E}} \big(-\mathcal{F}(u_h) + A\nabla_h u_h + AR_{ID}([\![u_h]\!])\big) \cdot \nabla_h v \,\mathrm{d}\mathcal{E} \\ &+ \sum_{S\in\mathcal{F}_{\text{int}}} \int_S (\{\!\{\mathcal{F}(u_h)\}\!\} + C_S[\![u_h]\!]) \cdot [\![v]\!] \,\mathrm{d}S + \sum_{S\in\mathcal{F}_{\text{bnd}}} \int_S F_h(u_h) \cdot nv \,\mathrm{d}S \\ &- \sum_{S\in\mathcal{F}} \int_S \big(A\{\!\{\nabla_h u_h\}\!\} + A\{\!\{R_{ID}([\![u_h]\!])\}\!\}\big) \cdot [\![v]\!] \,\mathrm{d}S = 0.\end{aligned} \tag{48}$$

This relation can be simplified using the following steps. Due to the symmetry of the matrix A and using the lifting operator R_{ID} (33) we have the relation

$$\begin{aligned}&\int_{\mathcal{E}} AR_{ID}([\![u_h]\!]) \cdot \nabla_h v \,\mathrm{d}\mathcal{E} \\ &\quad = -\sum_{S\in\cup_n\mathcal{S}_{ID}^n} \int_S A[\![u_h]\!] \cdot \{\!\{\nabla_h v\}\!\} \,\mathrm{d}S + \sum_{S\in\cup_n\mathcal{S}_D^n} \int_S Ag_D n \cdot \nabla_h v \,\mathrm{d}S.\end{aligned} \tag{49}$$

Further, the lifting operator R_{ID} has nonzero values only on faces $S \in \mathcal{S}_{ID}^n$. Using R, R_{ID} (see (30) and (33)) we obtain the following relation

$$\begin{aligned}&-\sum_{S\in\mathcal{F}} \int_S A\{\!\{R_{ID}([\![u_h]\!])\}\!\} \cdot [\![v]\!] \,\mathrm{d}S \\ &\quad = \int_{\mathcal{E}} AR([\![u_h]\!]) \cdot R([\![v]\!]) \,\mathrm{d}\mathcal{E} - \int_{\mathcal{E}} AR(\mathcal{P}g_D n) \cdot R([\![v]\!]) \,\mathrm{d}\mathcal{E}.\end{aligned} \tag{50}$$

Following a similar approach as in [14], we replace each term in (50) with the local lifting operator r_S, defined in Section 3.3.3, and make the following simplifications:

$$\int_{\mathcal{E}} AR(\llbracket u_h \rrbracket) \cdot R(\llbracket v \rrbracket)\ \mathrm{d}\mathcal{E} \cong \sum_{S \in \cup_n \mathcal{S}_{ID}^n} \sum_{\mathcal{K} \in \mathcal{T}_h} \eta_{\mathcal{K}} \int_{\mathcal{K}} Ar_S(\llbracket u_h \rrbracket) \cdot r_S(\llbracket v \rrbracket)\ \mathrm{d}\mathcal{K}, \tag{51}$$

$$\int_{\mathcal{E}} AR(\mathcal{P} g_D n) \cdot R(\llbracket v \rrbracket)\ \mathrm{d}\mathcal{E} \cong \sum_{S \in \cup_n \mathcal{S}_{D}^n} \sum_{\mathcal{K} \in \mathcal{T}_h} \eta_{\mathcal{K}} \int_{\mathcal{K}} Ar_S(\mathcal{P} g_D n) \cdot r_S(\llbracket v \rrbracket)\ \mathrm{d}\mathcal{K}. \tag{52}$$

A sufficient condition for the constant $\eta_{\mathcal{K}}$ to guarantee a stable and unique solution is $\eta_{\mathcal{K}} > n_f$, with n_f the number of faces of an element. For a proof, see [35]. The advantage of this replacement is that the stiffness matrix in the weak formulation using the local lifting operators is considerably sparser than the stiffness matrix resulting from the weak formulation with global lifting operators. We refer to [4; 14] for a further explanation.

Substituting relations (49)-(50) into (48), using relations (51)-(52), and considering the structure of matrix A, we then obtain:

$$\begin{aligned}
&- \int_{\mathcal{E}} \mathcal{F}(u_h) \cdot \nabla_h v\ \mathrm{d}\mathcal{E} + \int_{\mathcal{E}} D \overline{\nabla}_h u_h \cdot \overline{\nabla}_h v\ \mathrm{d}\mathcal{E} - \sum_{S \in \cup_n \mathcal{S}_{ID}^n} \int_S D \langle\!\langle u_h \rangle\!\rangle \cdot \{\!\{ \overline{\nabla}_h v \}\!\}\ \mathrm{d}S \\
&+ \sum_{S \in \cup_n \mathcal{S}_{D}^n} \int_S g_D D \bar{n} \cdot \overline{\nabla}_h v\ \mathrm{d}S + \sum_{S \in \mathcal{F}_{\mathrm{int}}} \int_S (\{\!\{ \mathcal{F}(u_h) \}\!\} + C_S \llbracket u_h \rrbracket) \cdot \llbracket v \rrbracket\ \mathrm{d}S \\
&+ \sum_{S \in \mathcal{F}_{\mathrm{bnd}}} \int_S \mathcal{F}(u_h) \cdot nv\ \mathrm{d}S - \sum_{S \in \cup_n \mathcal{S}_{ID}^n} \int_S D \{\!\{ \overline{\nabla}_h u_h \}\!\} \cdot \langle\!\langle v \rangle\!\rangle\ \mathrm{d}S \\
&- \sum_{S \in \mathcal{F}_{\mathrm{bnd}} \setminus \cup_n \mathcal{S}_{D}^n} \int_S D \overline{\nabla}_h u_h \cdot \bar{n} v\ \mathrm{d}S + \sum_{S \in \cup_n \mathcal{S}_{ID}^n} \sum_{\mathcal{K} \in \mathcal{T}_h} \eta_{\mathcal{K}} \int_{\mathcal{K}} D \bar{r}_S(\llbracket u_h \rrbracket) \cdot \bar{r}_S(\llbracket v \rrbracket)\ \mathrm{d}\mathcal{K} \\
&- \sum_{S \in \cup_n \mathcal{S}_{D}^n} \sum_{\mathcal{K} \in \mathcal{T}_h} \eta_{\mathcal{K}} \int_{\mathcal{K}} D \bar{r}_S(\mathcal{P} g_D n) \cdot \bar{r}_S(\llbracket v \rrbracket)\ \mathrm{d}\mathcal{K} = 0.
\end{aligned} \tag{53}$$

Here we used the spatial gradient operator $\overline{\nabla}$, the spatial jump operator $\langle\!\langle \cdot \rangle\!\rangle$ (see Section 3.3.2) and the spatial lifting operator $\bar{r}_S$ (see Section 3.3.3). Next, we introduce the following boundary and initial conditions:

$$\begin{aligned}
D \overline{\nabla}_h u_h \cdot \bar{n} &= g_M - \alpha u_h && \text{on } S \in \cup_n \mathcal{S}_M^n, \\
u_h &= g_D && \text{on } S \in \cup_n \mathcal{S}_{DBSm}^n, \\
u_h &= u_0 && \text{on } \Omega_0,
\end{aligned}$$

into (53). We introduce $a : V_h^{(p_t,p_s)} \times V_h^{(p_t,p_s)} \to \mathbb{R}$:

$$a(u_h, v) = a_a(u_h, v) + a_d(u_h, v), \tag{54}$$

with $a_a : V_h^{(p_t,p_s)} \times V_h^{(p_t,p_s)} \to \mathbb{R}$, $a_d : V_h^{(p_t,p_s)} \times V_h^{(p_t,p_s)} \to \mathbb{R}$ defined as:

$$\begin{aligned}
a_a(u_h, v) = &- \int_{\mathcal{E}} \mathcal{F}(u_h) \cdot \nabla_h v\ \mathrm{d}\mathcal{E} + \sum_{S \in \mathcal{F}_{\mathrm{int}}} \int_S (\{\!\{ \mathcal{F}(u_h) \}\!\} + C_S \llbracket u_h \rrbracket) \cdot \llbracket v \rrbracket\ \mathrm{d}S \\
&+ \sum_{S \in (\cup_n \mathcal{S}_{MDSp}^n \cup \Gamma_+)} \int_S \mathcal{F}(u_h) \cdot nv\ \mathrm{d}S,
\end{aligned} \tag{55}$$

$$
\begin{aligned}
a_d(u_h,v) = & \int_{\mathcal{E}} D\overline{\nabla}_h u_h \cdot \overline{\nabla}_h v \,\mathrm{d}\mathcal{E} \\
& - \sum_{S\in\cup_n\mathcal{S}^n_{ID}} \int_S \big(D\langle\!\langle u_h\rangle\!\rangle \cdot \{\!\{\overline{\nabla}_h v\}\!\} + D\{\!\{\overline{\nabla}_h u_h\}\!\} \cdot \langle\!\langle v\rangle\!\rangle\big) \,\mathrm{d}S \\
& + \sum_{S\in\cup_n\mathcal{S}^n_{ID}} \sum_{\mathcal{K}\in\mathcal{T}_h} \eta_{\mathcal{K}} \int_{\mathcal{K}} D\bar{r}_S([\![u_h]\!]) \cdot \bar{r}_S([\![v]\!]) \,\mathrm{d}\mathcal{K} \\
& + \sum_{S\in\cup_n\mathcal{S}^n_{M}} \int_S \alpha u_h v \,\mathrm{d}S,
\end{aligned} \tag{56}
$$

and the linear form $\ell : V_h^{(p_t,p_s)} \to \mathbb{R}$ defined as:

$$
\begin{aligned}
\ell(v) = & - \sum_{S\in\cup_n\mathcal{S}^n_{D}} \int_S g_D D\bar{n} \cdot \overline{\nabla}_h v \,\mathrm{d}S \\
& + \sum_{S\in\cup_n\mathcal{S}^n_{D}} \sum_{\mathcal{K}\in\mathcal{T}_h} \eta_{\mathcal{K}} \int_{\mathcal{K}} D\bar{r}_S(\mathcal{P}g_D n) \cdot \bar{r}_S([\![v]\!]) \,\mathrm{d}\mathcal{K} + \sum_{S\in\cup_n\mathcal{S}^n_{M}} \int_S g_M v \,\mathrm{d}S \\
& - \sum_{S\in\cup_n\mathcal{S}^n_{DBSm}} \int_S F(g_D)\cdot n v \,\mathrm{d}S + \int_{\Omega_0} c_0 v \,\mathrm{d}\Omega.
\end{aligned} \tag{57}
$$

Note that the term $\sum_{S\in\mathcal{F}_{\text{bnd}}\setminus\cup_n\mathcal{S}^n_{DM}} \int_S D\overline{\nabla}_h u_h \cdot \bar{n} v \,\mathrm{d}S$ is dropped from the bilinear form $a_d(\cdot,\cdot)$ since on $S \in \mathcal{F}_{\text{bnd}} \setminus \cup_n\mathcal{S}^n_{DM}$ the matrix D is zero.

The space-time DG discretization for (26) can now be formulated as follows. Find a $u_h \in V_h^{(p_t,p_s)}$ such that:

$$
a(u_h,v) = \ell(v), \qquad \forall v \in V_h^{(p_t,p_s)}. \tag{58}
$$

This formulation is the most straightforward for the theoretical analysis discussed in [35], but for practical implementations, an arbitrary Lagrangian Eulerian (ALE) formulation is preferable. Therefore, in these notes, we also present the ALE form of the space-time weak formulation (58). The relation between the space-time and ALE formulation discussed here follows the derivation in [42].

Using a result from [42], the space-time normal vector n can be split into two parts: $n = (n_t, \bar{n})$, with n_t the temporal part and $\bar{n}$ the spatial part of the space-time normal vector n. Next, we consider the normal vector n on the faces $S \in \mathcal{F}_{\text{int}}$, which consist of two sets: $\mathcal{F}_{\text{int}} = \cup_n(\mathcal{S}^n_I \cup \mathcal{S}^n_S)$. On $S \in \mathcal{S}^n_S$, the space-time normal vector is $n = (\pm 1, \underbrace{0,\dots,0}_{d\,\times})$ and is not affected by the mesh velocity. On the faces $S \in \mathcal{S}^n_I$ the space-time normal vector depends on the mesh velocity u_g:

$$
n = (-u_g \cdot \bar{n}, \bar{n}), \tag{59}
$$

which also holds on the boundary faces $S \in \mathcal{F}_{\text{bnd}} \setminus (\Omega_0 \cup \Omega_T)$.

If we recall the bilinear and linear forms in (55)-(57), then only $a_a(\cdot,\cdot)$ and $\ell(\cdot)$ need to be rewritten into the ALE formulation by splitting the normal vector n into a temporal and spatial part. The bilinear form a_d in (56) remains valid for the ALE formulation since it does not depend on n_t. We now consider the contribution $\{\!\{\mathcal{F}(u_h)\}\!\} \cdot [\![v]\!]$ in (55). On $S \in \cup_n\mathcal{S}^n_I$, this contribution can be written in the ALE formulation using (59) as:

$$
\{\!\{\mathcal{F}(u_h)\}\!\} \cdot [\![v]\!] = \{\!\{f(u_h) - u_g u_h\}\!\} \cdot \langle\!\langle v\rangle\!\rangle,
$$

while on $S \in \mathcal{S}_S^n$ this term does not change. Here, $f(u) = (f_1(u), \cdots, f_d(u))^T$. Next, consider the term $[\![u_h]\!] \cdot [\![v]\!]$. Since the normal vector n has length one, we immediately obtain

$$[\![u_h]\!] \cdot [\![v]\!] = (u_h^+ - u_h^-)(v^+ - v^-),$$

and thus this contribution also does not depend on the mesh velocity u_g.

The form $a_a(\cdot,\cdot)$ and linear functional $\ell(\cdot)$ in the ALE formulation are now equal to:

$$\begin{aligned} a_a(u_h, v) = &-\int_{\mathcal{E}} \mathcal{F}(u_h)\cdot\nabla_h v \,\mathrm{d}\mathcal{E} + \sum_{S\in\cup_n \mathcal{S}_I^n} \int_S (\{\!\{f(u_h) - u_g u_h\}\!\}\cdot\langle\!\langle v\rangle\!\rangle + C_S[\![u_h]\!]\cdot[\![v]\!]) \,\mathrm{d}S \\ &+ \sum_{S\in\cup_n \mathcal{S}_S^n} \int_S (\{\!\{\mathcal{F}(u_h)\}\!\} + C_S[\![u_h]\!])\cdot[\![v]\!] \,\mathrm{d}S \\ &+ \sum_{S\in(\cup_n \mathcal{S}_{MDSp}^n \cup \Gamma_+)} \int_S (f(u_h) - u_g u_h)\cdot\bar{n} v \,\mathrm{d}S, \end{aligned} \tag{60}$$

$$\begin{aligned} \ell(v) = &-\sum_{S\in\cup_n \mathcal{S}_D^n} \int_S g_D D\bar{n}\cdot\overline{\nabla}_h v \,\mathrm{d}S \\ &+ \sum_{S\in\cup_n \mathcal{S}_D^n} \sum_{\mathcal{K}\in\mathcal{T}_h} \eta_{\mathcal{K}} \int_{\mathcal{K}} D\bar{r}_S(\mathcal{P}g_D n)\cdot\bar{r}_S([\![v]\!]) \,\mathrm{d}\mathcal{K} + \sum_{S\in\cup_n \mathcal{S}_M^n} \int_S g_M v \,\mathrm{d}S \\ &- \sum_{S\in\cup_n \mathcal{S}_{DBSm}^n} \int_S (f(g_D) - g_D u_g)\cdot\bar{n} v \,\mathrm{d}S + \int_{\Omega_0} c_0 v \,\mathrm{d}\Omega, \end{aligned} \tag{61}$$

with $a_d(\cdot,\cdot)$ is given by (56).

4 Extension to 3D compressible flow

4.1 Space-time formulation for compressible flows

The space-time discontinuous Galerkin method for parabolic partial differential equations discussed in Section 3 provides the starting point for the extension to compressible flows. In this section we will only give a brief overview of the space-time DG discretization for the Euler equations of inviscid flow and the compressible Navier-Stokes equations. For more detailed information we refer to [42; 44; 26; 27; 28].

We consider the compressible Navier-Stokes equations in a time-dependent flow domain $\Omega(t)$. Since the flow domain boundary $\partial\Omega(t)$ is moving and deforming in time we do not make an explicit separation between the space and time variables and consider the Navier-Stokes equations directly in $\mathbb{R}^4$. Let $\mathcal{E} \subset \mathbb{R}^4$ be an open domain. Let $F^e : \mathbb{R}^5 \to \mathbb{R}^{5\times 4}$ denote the inviscid flux tensor, which is defined as:

$$F^e = \begin{pmatrix} \rho & \rho u_1 & \rho u_2 & \rho u_3 \\ \rho u_1 & \rho u_1^2 + p & \rho u_1 u_2 & \rho u_1 u_3 \\ \rho u_2 & \rho u_1 u_2 & \rho u_2^2 + p & \rho u_2 u_3 \\ \rho u_3 & \rho u_1 u_3 & \rho u_2 u_3 & \rho u_3^2 + p \\ \rho E & (\rho E + p)u_1 & (\rho E + p)u_2 & (\rho E + p)u_3 \end{pmatrix},$$

with ρ, p, and E the density, pressure, and specific total energy, respectively, and u_i the velocity components in the Cartesian coordinate directions x_i, $i \in \{1, 2, 3\}$ of the velocity vector $u : \mathcal{E} \to \mathbb{R}^3$. The vector $U : \mathcal{E} \to \mathbb{R}^5$ denotes the conservative flow variables with components U_i and is the first column in F^e:

$$U_i = F_{i0}.$$

We also introduce the viscous flux tensor $F^v : \mathbb{R}^5 \to \mathbb{R}^{5\times 4}$, which is defined as:

$$F^v = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \tau_{11} & \tau_{12} & \tau_{13} \\ 0 & \tau_{21} & \tau_{22} & \tau_{23} \\ 0 & \tau_{31} & \tau_{32} & \tau_{33} \\ 0 & \tau_{1j}u_j - q_1 & \tau_{2j}u_j - q_2 & \tau_{3j}u_j - q_3 \end{pmatrix},$$

where the summation convention is used on repeated indices in this section. The total stress tensor $\tau \in \mathbb{R}^{3\times 3}$ is defined as

$$\tau = \lambda \bar{\nabla} \cdot u + \mu(\bar{\nabla} u + (\bar{\nabla} u)^T),$$

with the dynamic viscosity coefficient μ given by Sutherlands's law:

$$\frac{\mu}{\mu_\infty} = \frac{T_\infty + T_S}{T + T_S} \left(\frac{T}{T_\infty}\right)^{\frac{3}{2}},$$

where T is the temperature, T_S a constant and $(\cdot)_\infty$ refers to free stream values. For the thermal diffusivity coefficient λ, we assume $\lambda = -\frac{2}{3}\mu$. The heat flux $q \in \mathbb{R}^3$ is defined as

$$q = -\kappa \bar{\nabla} T,$$

with κ the thermal conductivity coefficient.

The compressible Navier-Stokes equations can now be defined as

$$\nabla \cdot \big(F^e\big(U(x)\big) - F^v\big(U(x), \nabla U\big)\big) = 0, \qquad x \in \mathcal{E}, \tag{62}$$

together with the initial and boundary conditions:

$$\begin{aligned} U(x) &= U_0(x), & x &\in \Omega(t_0), \\ U(x) &= \mathcal{B}(U, U_w), & x &\in \mathcal{Q}. \end{aligned}$$

Here $U_0 : \Omega(t_0) \to \mathbb{R}^5$ denotes the initial flow field, $\mathcal{B} : \mathbb{R}^5 \times \mathbb{R}^5 \to \mathbb{R}^5$ the boundary operator and $U_w : \mathcal{Q} \to \mathbb{R}^5$ the prescribed boundary flow field data. The divergence of a second order tensor is defined as: $\nabla \cdot \mathcal{F} = \frac{\partial \mathcal{F}_{ij}}{\partial x_j}$. The Navier-Stokes equations are completed with equations of state for the pressure and temperature in a calorically perfect gas: $p = (\gamma - 1)\rho(E - \frac{1}{2}u_i u_i)$ and $T = \frac{1}{c_v}(E - \frac{1}{2}u_i u_i)$, with c_v the specific heat at constant volume and γ the ratio of specific heats at constant pressure and volume. The Euler equations for inviscid compressible flow are obtained by neglecting the viscous flux tensor F^v.

The space-time discontinuous Galerkin discretization for the compressible Navier-Stokes equations is now obtained by transforming (62) into a first order system in the same way as done in Section 3.4 for the parabolic scalar conservation law. For this purpose we introduce a homogeneity tensor $A \in \mathbb{R}^{5\times 4\times 5\times 4}$ defined as

$$A_{ikrs}(U) := \frac{\partial F^v_{ik}(U, \nabla U)}{\partial(U_{r,s})},$$

with $i, r = 1, \cdots 5$ and $k, s = 1, \cdots, 4$, and where a comma denotes partial differentiation with respect to the indicated coordinate direction. Note, the homogeneity tensor A is not symmetric nor positive definite for given indices r, s. This is a significant complication compared to the parabolic scalar conservation law discussed in Section 3. The viscous flux then satisfies the relation

$$F^v_{ik}(U, \nabla U) = A_{ikrs}(U)U_{r,s}.$$

Introducing the auxiliary variable $\Theta \in \mathbb{R}^{5\times 4}$, the compressible Navier-Stokes equations can now be written as

$$\nabla \cdot (F^e(U) - \Theta(U)) = 0, \tag{63}$$

$$\Theta(U) - A(U)\nabla U = 0. \tag{64}$$

For the space-time discontinuous Galerkin discretization we introduce the following finite element spaces on the finite element tessellation $\mathcal{T}_h$:

$$W_h = \{W \in (L^2(\mathcal{E}))^5 \,:\, W|_{\mathcal{K}} \circ G_{\mathcal{K}} \in P^k(\hat{\mathcal{K}})^5, \quad \forall \mathcal{K} \in \mathcal{T}_h\},$$

with $L^2(\mathcal{E})$ the space of square integrable functions on $\mathcal{E}$, P^k the space of polynomials of total degree k on the reference element $\hat{\mathcal{K}}$, and $G_{\mathcal{K}} : \hat{\mathcal{K}} \to \mathcal{K}$ the mapping connecting the reference element to the space-time element. Similarly, we define the space

$$V_h = \{V \in (L^2(\mathcal{E}))^{5\times 4} \,:\, V|_{\mathcal{K}} \circ G_{\mathcal{K}} \in P^k(\hat{\mathcal{K}})^{5\times 4}, \quad \forall \mathcal{K} \in \mathcal{T}_h\}.$$

The space-time weak formulation can now be obtained by multiplying (63-64) with arbitrary test functions $W \in W_h$ and $V \in V_h$, respectively, and integration by parts over each space-time element (twice for (64)) and summation over all elements in the tessellation $\mathcal{T}_h$. The weak formulation then is equal to:

Find a $U \in W_h$ and a $\Theta \in V_h$, such that for all $W \in W_h$ and $V \in V_h$ the following relation is satisfied

$$-\sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\mathcal{K}} \Big(W_{i,k}(F^e_{ik}(U) - \Theta_{ik}(U)\Big) d\mathcal{K} + \sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\partial\mathcal{K}} W_i^- n_k\big(\hat{F}^e_{ik}(U) - \hat{\Theta}_{ik}(U)\big) d(\partial\mathcal{K}) = 0, \tag{65}$$

$$\sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\mathcal{K}} V_{ik}\Theta_{ik}(U) d\mathcal{K} - \sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\mathcal{K}} V_{ik} A_{ikrs} U_{r,s} d\mathcal{K} - \sum_{\mathcal{K}\in\mathcal{T}_h} \int_{\mathcal{Q}} V^-_{ik} A^-_{ikrs}(\hat{U}_r - U^-_r) n_s d\mathcal{Q} = 0. \tag{66}$$

Just as in the case of the scalar conservation laws we can obtain the discontinuous Galerkin finite element discretization by representing the flow variables U, auxiliary variables Θ and test functions V and W as polynomials in each element, which are discontinuous across element faces. This discontinuity at the element faces requires the introduction of numerical fluxes which are indicated with a hat in (65-66) and form an essential ingredient in the DG discretization. After the definition of the numerical fluxes, it is possible to eliminate the auxiliary variables Θ using similar lifting operators as defined in Section 3.3.3 and we obtain the so-called primal formulation. The details of this derivation are beyond the scope of these notes and we refer to Klaij, van der Vegt and van der Ven [26; 27; 28].

The space-time DG finite element discretization for the Euler equations is less involved since it does not require the use of the auxiliary variables Θ. A detailed description of this algorithm, which closely follows the lines discussed in Section 2, can be found in van der Vegt and van der Ven [42; 44].

4.2 Some applications

In this section we present a few examples of results obtained with space-time discontinuous Galerkin methods. This section serves to highlight some of the possibilities of computations which can be performed with these algorithms. For more extensive applications we refer to the references cited in Section 1.

The first example is the inviscid flow about an oscillating NACA 0012 airfoil. The freestream Mach number is 0.8, the pitching angle ranges between -0.5 and 4.5 degrees and the oscillation period is $T = 20$ (normalized with L/a_∞, where L is the chord length and a_∞ is the freestream speed of sound), which results in a circular frequency $\omega = \pi/10$. The flow field is computed both on a fine mesh with 32,768 elements and an adapted mesh, which has approximately 9,400 elements during the simulation. During each time step the coarse mesh is adapted, with first coarsening followed by refinement. Both simulations used a time step of 1.0 for the interval $[3.0, 13.0]$ of a period, and a time step of 0.5 in the remaining part of the period. The smaller time steps during this part of the oscillation period are necessary since the shock at the lower side of the airfoil has a greater velocity than the shock at the upper side. If the shock moves through several cells during a time step this will result in numerical oscillations, since no artificial dissipation or limiting is applied in the time direction. The results on the fine and adapted mesh are nearly identical, where the difference in the lift coefficient can be attributed to the improved accuracy in the shock due to the mesh adaptation. This can be inferred from the pressure coefficients C_p at the wing shown in Figure 9. The pressure coefficients for the fine and adapted mesh are nearly identical, except in the shock, where the adapted mesh captures the discontinuity better. Figure 9 also shows that the mesh adaptation does not negatively influence the time accuracy and is very efficient in capturing the flow discontinuities, also for the weak shock at the lower side of the wing which periodically disappears.

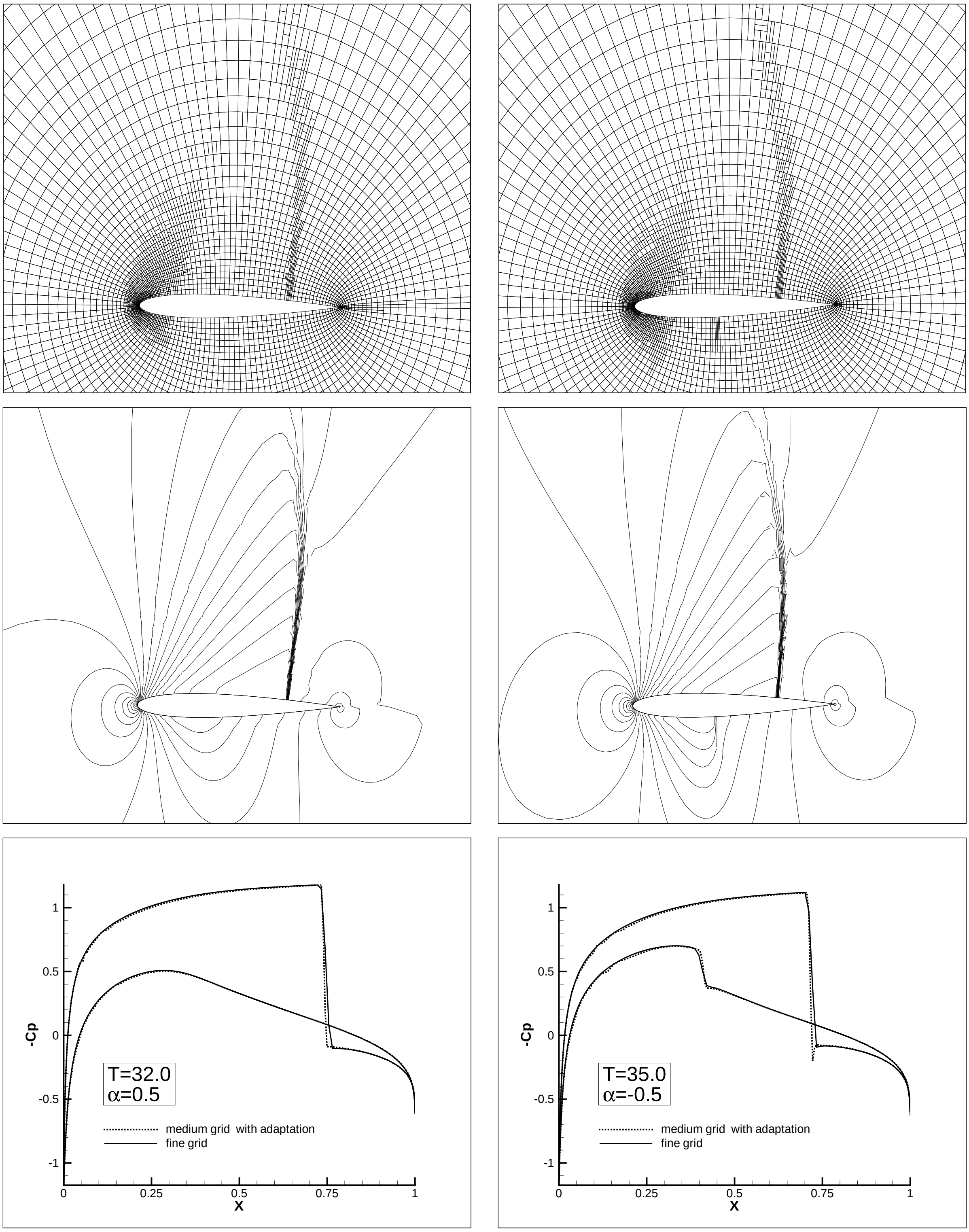


Figure 9: Adapted mesh around oscillating NACA 0012 airfoil, contours of density, and pressure coefficient C_p on the airfoil surface for $\alpha = 0.5°$ (pitching downward) and $\alpha = -0.5°$ ($M_\infty = 0.8$, $\omega = \pi/10$)., (Taken from [42])

The second example is the steady viscous flow about a delta wing computed with the space-time discontinuous Galerkin discretization of the compressible Navier-Stokes equations. The Reynolds number is $Re = 4 \cdot 10^4$, the Mach number $M_\infty = 0.3$ and the angle of attack $\alpha = 12.5°$. The finite element mesh contains 1.671.168 elements. The

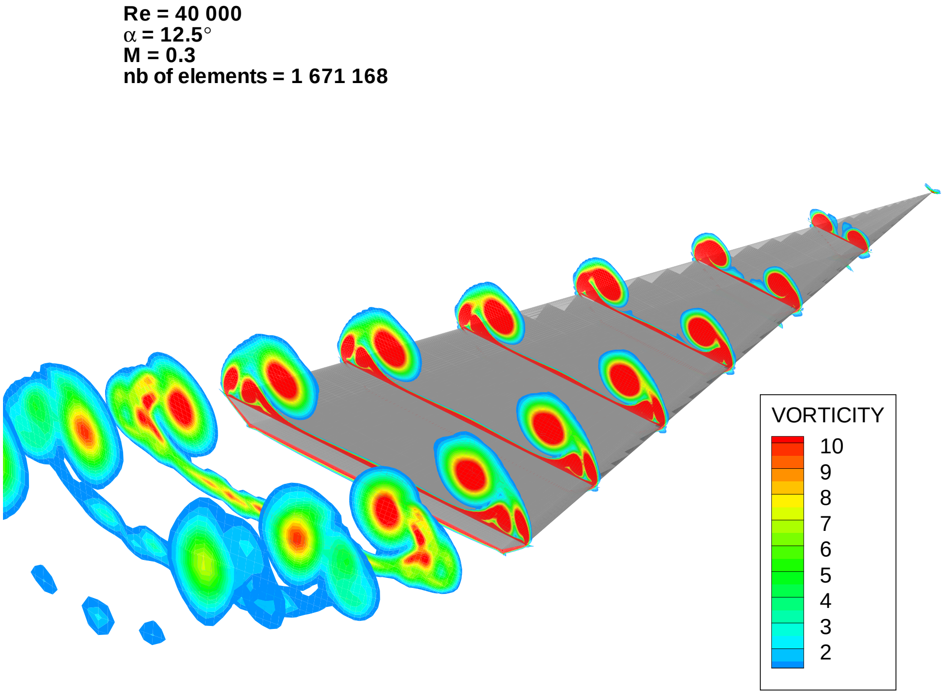


Figure 10: Viscous flow field about a delta wing with a sharp leading edge ($Re = 4 \cdot 10^4$, $M_\infty = 0.3$, $\alpha = 12.5°$).

delta wing has a sharp leading edge, which creates a vortex sheet which rolls up into the primary vortex above the wing. This vortex has a nearly conical shape and induces a large velocity near the upper wing surface, which causes flow separation at the wing surface and creates a secondary vortex. Further downstream, the induced velocity of the secondary vortex is large enough to generate even a tertiary vortex near the leading edge. The vortex structures roll up into a concentrated vortex further downstream in the wake. The space-time DG algorithm is well capable of capturing these detailed flow structures which can be further improved using local mesh refinement [26; 27].

The third example is the unsteady viscous flow about a NACA 0012 airfoil which undergoes a rapid pitch up maneuver to large angles of attack. This will result in the generation of large unsteady vortical structures on the lee-side of the profile. The Reynolds number in these simulations is $Re = 10^4$, the Mach number $M_\infty = 0.2$ and the mesh contains 4256 elements. During the pitch-up maneuver the mesh is deforming to accommodate for the boundary movement.

5 Acknowledgement

It is a pleasure to thank J.J. Sudirham and C.M. Klaij (University of Twente) and H. van der Ven (National Aerospace Laboratory, NLR) for the collaboration in the development of the space-time discontinuous Galerkin finite element methods discussed in these notes.

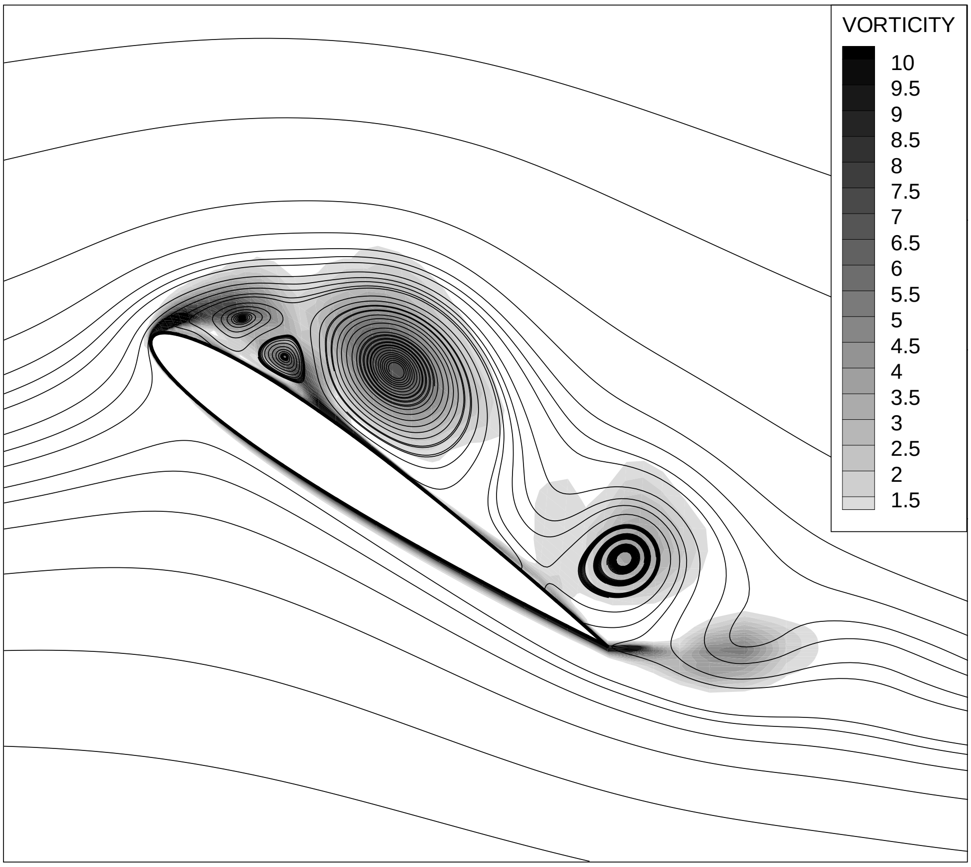


Figure 11: Streamlines in flow about a NACA 0012 airfoil in pitch-up maneuver. ($Re = 10000$, $M_\infty = 0.2$, $\alpha = 35°$.

The financial support of the Technology Foundation STW and NLR for the research on space-time discontinuous Galerkin methods is gratefully acknowledged.

References

[1] V.R. Ambati and O. Bokhove, Space-time finite element shallow water flows. *J. Comp. Appl. Math.*, **204**(2), 452-462 (2007).

[2] V.R. Ambati and O. Bokhove, Space-time discontinuous Galerkin discretization of rotating shallow water equations. *J. Comp. Phys.*, **225**, 1233-1261 (2007).

[3] D.N. Arnold, An interior penalty finite element method with discontinuous elements, *SIAM J. Num. Anal.*, **19**(4), 742-760 (1982).

[4] D.N. Arnold, F. Brezzi, B. Cockburn and L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, *SIAM J. Numer. Anal.*, **39**(5), 1749-1779 (2002).

[5] F. Bassi and S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, *J. Comput. Phys.*, **131**, 267-279 (1997).

[6] F. Bassi and S. Rebay, Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations, *Int. J. Numer. Meth. Fluids.*, **40**,

197-207 (2002).

[7] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti and M. Savini, A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows, in *Proceedings of 2nd European Conference on Turbomachinery, Fluid Dynamics and Thermodynamics*, Technologisch Instituut, Antwerpen, Belgium, 99-108, 1997.

[8] C.E. Baumann, An hp-adaptive discontinuous finite element method for computational fluid dynamics, Ph.D. dissertation, The University of Texas at Austin, Aug. 1997.

[9] C.E. Baumann and J.T. Oden, A discontinuous *hp* finite element method for the Euler and Navier-Stokes equations. Tenth international conference on finite elements in fluids (Tucson, AZ, 1998). *Intern. J. Numer. Methods Fluids* **31**(1), 79-95 (1999).

[10] E. Bernsen, O. Bokhove and J.J.W. an der Vegt, A (Dis)continuous finite element model for generalized 2D vorticity dynamics, *J. Comput. Phys.*, **211**, 719-747 (2006).

[11] O.J. Boelens, H. van der Ven, B. Oskam and A.A. Hassan, The boundary conforming discontinuous Galerkin finite element approach for rotorcraft simulations, *J. of Aircraft*, **39**(5), 776-785 (2002).

[12] O. Bokhove, Flooding and drying in finite-element Galerkin discretizations of shallow-water equations. Part I: One dimension, *J. Sci. Comput.* **22**, 47-82 (2005).

[13] F. Brezzi, G. Manzini, D. Marini, P. Pietra, A. Russo, Discontinuous finite elements for diffusion problems, Atti del Convegno in Memoria di F. Brioschi, Milano, Instituto Lombardo di Scienze e Lettere, 1997.

[14] F. Brezzi, G. Manzini, D. Marini, P. Pietra, A. Russo, Discontinuous Galerkin approximations for elliptic problems, *Numer. Meth. Part. Diff. Eq.*, **16**, 365-378 (2000).

[15] P. Castillo, B. Cockburn, I. Perugia and D. Schotzau, Local discontinuous Galerkin methods for elliptic problems, *Comm. Numer. Meth. Engng*, **18**, 69-75 (2002).

[16] B. Cockburn, Discontinuous Galerkin methods for convection-dominated problems, in T.J. Barth and H. Deconinck (Eds.), *Lect. Notes in Comp. Sci. and Eng.*, **9** (Springer Verlag, 1999).

[17] B. Cockburn and P.A. Gremaud, Error estimates for finite element methods for nonlinear conservation laws, *SIAM J. Numer. Anal.*, **33**, 522-554 (1996).

[18] B. Cockburn, G. Kanschat, I. Perugia and D. Schotzau, Superconvergence of the local discontinuous Galerkin method for elliptic problems on Cartesian grids, *SIAM J. Numer. Anal.*, **39**(1), 264-285 (2002).

[19] B. Cockburn, G.E. Karniadakis, C.-W. Shu, (Eds.), Discontinuous Galerkin methods. Theory, computation and applications, *Lect. Notes in Comp. Sci. and Eng.*, **11**, Springer Verlag, 2000.

[20] B. Cockburn and C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM J. Numer. Anal.*, **35**(6), 2440-2463 (1998).

[21] B. Cockburn and C.W. Shu, Runge-Kutta discontinuous Galerkin method for convection dominated problems, *J. Sci. Comput.*, **6**(3), 173-261 (2001).

[22] J.Jr. Douglas and T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods, in *Lecture Notes in Physics*, **58**, Springer-Verlag, 1976.

[23] A. Ern and J.-L. Guermond, Theory and practice of finite elements, Springer Verlag, 2004.

[24] A. Harten and J.M. Hyman, Self adjusting grid methods for one-dimensional hyperbolic conservation laws, *J. of Comput. Phys.*, **50**, 235-269 (1983).

[25] J. Jaffre, C. Johnson and A. Szepessy, Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws, *Math. Models and Meth. in Appl. Sci.* **5**, 367-386 (1995).

[26] C.M. Klaij, J.J.W. van der Vegt and H. van der Ven, Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations, *J. Comput. Phys.*, **217**(2), 589-611 (2006).

[27] C.M. Klaij, J.J.W. van der Vegt and H. van der Ven, Pseudo-time stepping methods for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, *J. Comput. Phys.*, **219**, 622-643 (2006).

[28] C.M. Klaij, M.H. van Raalte, H. van der Ven, and J.J.W. van der Vegt, h-Multigrid for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, *J. Comput. Phys.*, **227**(2), 1024-1045 (2007).

[29] M. Lesoinne and C. Farhat, Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations, *Comput. Meth. Appl. Mech. Engrg.*, **134**, 71-90 (1996).

[30] J.T. Oden, I. Babuska and C.E. Baumann, A discontinuous *hp* finite element method for diffusion problems, *J. Comput. Phys.*, **146**, 491-519 (1998).

[31] L. Pesch, A. Bell, W.E.H. Sollie, V.R. Ambati, O. Bokhove and J.J.W. van der Vegt, hpGEM- A software framework for Discontinuous Galerkin finite element methods, *ACM Trans. Math. Softw.*, **33**(4) (2007).

[32] L. Pesch and J.J.W. van der Vegt, A discontinuous Galerkin finite element discretization of the Euler equations for compressible and incompressible fluids, *J. Comput. Phys.*, **227**(11), 5426-5446 (2008).

[33] S. Rhebergen, O. Bokhove and J.J.W. van der Vegt, Discontinuous Galerkin finite element methods for hyperbolic nonconservative partial differential equations, *J. Comput. Phys.*, **227**(3), 1887-1922, (2008).

[34] P. Solin, K. Segeth and I. Dolezel, Higher-order finite element methods, Chapman & Hall/CRC, 2004.

[35] J.J. Sudirham, J.J.W. van der Vegt and R.M.J. van Damme, Space-time discontinuous Galerkin method for advection-diffusion problems on time-dependent domains, *Appl. Numer. Math.*, **56**, 1491-1518 (2006).

[36] P. Tassi, O. Bokhove, and C. Vionnet, Space discontinuous Galerkin method for shallow water flows -kinetic and HLLC flux, and potential vorticity-generation. *Advanc. Water Res.*, **30**, 998-1015 (2007).

[37] P.A. Tassi, S. Rhebergen, C.A. Vionnet and O. Bokhove, A discontinuous Galerkin finite element model for morphological evolution under shallow flows, *Comput. Meth. Appl. Mech. Engrg.*, in press (2008).

[38] S.K. Tomar and J.J.W. van der Vegt, A Runge-Kutta discontinuous Galerkin method for linear free-surface gravity waves using high order velocity recovery, *Comput. Meth. Appl. Mech. Engrg.*, **196**(13-16), 1984-1996 (2007).

[39] E.F. Toro, *Riemann solvers and numerical methods for fluid dynamics. A practical introduction* 2nd edition, Springer Verlag, 1999.

[40] J.J.W. van der Vegt and S.K. Tomar, Discontinuous Galerkin method for linear free-surface gravity waves, *J. Sci. Comput.*, **22-23**, 531-567 (2005).

[41] J.J.W. van der Vegt and J.J. Sudirham, A space-time discontinuous Galerkin method for the Time-Dependent Oseen Equations, article in print *Appl. Numer. Math.*, 2008.

[42] J.J.W. van der Vegt and H. van der Ven, Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. Part I. General formulation., *J. Comput. Phys.*, **182**, 546-585 (2002).

[43] J.J.W. van der Vegt and Y. Xu, Space-time discontinuous Galerkin method for non-linear water waves, *J. Comput. Phys.*, **224**(1), 17-39 (2007).

[44] H. van der Ven and J.J.W. van der Vegt, Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. II. Efficient flux quadrature, *Comput. Meth. Appl. Mech. Engrg.*, **191**, pp. 4747-4780 (2002).

[45] H. van der Ven and O.J. Boelens, A framework for aeroelastic simulations of trimmed rotor systems in forward flight. In *Proc. 30th European Rotorcraft Forum, Marseiile, France, Sept. 14-16*, 2004.

[46] H. van der Ven, J.J.W. van der Vegt and E.G. Bouwman, Space-time discontinuous Galerkin finite element method for inviscid compressible flows. In *Computational fluid and solid mechanics 2003 (MIT) Boston)*, Vol. 1, 1181-1143, Elsevier Sci., Oxford, U.K., 2003.

[47] M.F. Wheeler, An elliptic collocation-finite element method with interior penalties, *SIAM J. Numer. Anal.*, **15**, 152-161 (1978).

[48] Y. Xu, J.J.W. van der Vegt, and O. Bokhove, 2008: Discontinuous Hamiltonian Finite Element Method for a Bilinear Poisson Bracket. article in print, *J. Sci. Comput.*, 2008.