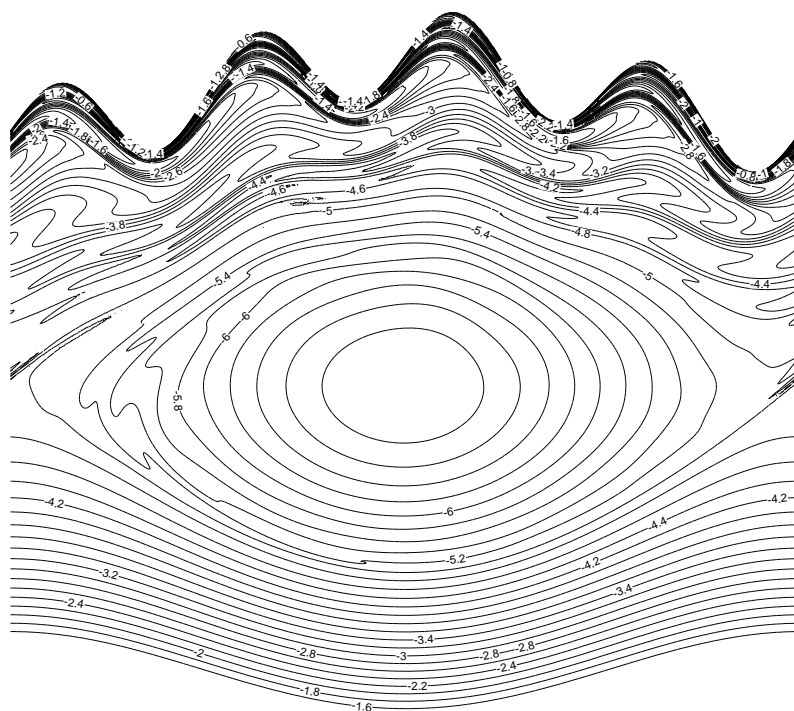# Introduction to (dis)continuous Galerkin finite element methods

Onno Bokhove and Jaap J.W. van der Vegt

Department of Applied Mathematics, University of Twente

March 27, 2008

Contents

[1]

---

[1]Figure title page: the two-dimensional vorticity field developed after the steady-state solution has been perturbed suddenly by a steady oscillatory deformation of the upper channel boundary. See Bernsen et al. (2004).

Chapter 1

Introduction

In many areas of science solutions of partial differential equations (PDE's) are required. Analytical methods to solve PDE's are often complicated and exact solutions are not always available, in which case numerical methods provide an established way to find approximate solutions to the PDE's. Numerical methods are approximate in that solutions of PDE's (generally) depend on coordinates in a continuous way while numerical solutions depend on a finite number of degrees of freedom. A numerical solution is obtained as solution of a finite algebraic system of linear or nonlinear equations, which form a discretization of the PDE's.

Three well-known classes of discretization methods are finite difference, finite volume and finite element methods. Finite volume and finite element methods are especially well suited for computations in complicated domains on (locally) irregular or unstructured meshes. A mesh divides the domain of validity of a PDE into a finite number of elements. On each such element, the discretization acccomplishes that each variable of the PDE is reduced to depend on only a finite number of degrees of freedom per element. While finite element methods appear to be more difficult to understand, they offer more flexibility and superior accuracy in domains with complex boundaries and boundary conditions. Furthermore, the mathematical theory of finite element methods is well-developed.

1.1 Convergence and stability

The quality of a numerical method depends on its rate convergence, its accuracy and stability. A discretization is convergent when for decreasing mesh size and time intervals the numerical solution approaches the exact solution of the PDE.

The accuracy of a discretization concerns the rate of convergence as function of mesh size. The truncation error consist of the discretization applied to the exact solution. The remainder will be expressed as powers of the (spatial and temporal) mesh size. Briefly, consider the diffusion equation $u_t = u_{xx}$ on $x \in [0,1]$ for $t > 0$ with variable $u = u(x,t)$ and

derivatives $u_t = \partial u/\partial t$ and $u_{xx} = \partial^2 u/\partial x^2$. An explicit finite difference discretization is

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} - \frac{U_{j+1}^n - 2\,U_j^n + U_{j-1}^n}{\Delta x^2} = 0 \qquad (1.1.1)$$

with $U_j^n \approx u(x_j = j\,\Delta x, t = n\,\Delta t)$. The truncation error is the same discretization

$$
\begin{aligned}
T(x,t) = &\frac{u\,(j\,\Delta x, (n+1)\,\Delta t) - u(j\,\Delta x, n\,\Delta t)}{\Delta t} \\
&- \frac{u\,((j+1)\,\Delta x, n\,\Delta t) - 2\,u\,(j\,\Delta x, n\,\Delta t) + u((j-1)\,\Delta x, n\,\Delta t)}{\Delta x^2} = 0
\end{aligned}
\qquad (1.1.2)
$$

applied to the exact solution $u(x,t)$. Taylor expansion of the u's in (1.1.2) around (x,t) shows that the discretation error is $T(x,t) = O(\Delta x^2, \Delta t)$, that is, second order in space and first order in time.

For time-dependent solutions, the rate of convergence is discussed for the numerical solution after a finite time. Mesh size is usually characterized by the largest volume, area or length of an element, volume or grid cell in a mesh in three or two dimensions, or one dimension. A discretization is stable when the solution does not become unbounded, provided the PDE has bounded solutions. We refer to Morton (1996) and Morton and Mayers (1994) for further discussion.

1.2 Examples

Finite elements are common in the engineering community but are often perceived to be more difficult in other communities. In this introduction, we aim to make finite element methods accessible by focussing primarily on a few examples of PDE's. These examples include

(i) the two-dimensional Poisson's equation

$$-\nabla^2 \phi(x,y) = f(x,y)$$

with unknown $\phi = \phi(x,y)$, given function $f = f(x,y)$, coordinates $x, y \in \Omega \in \mathbb{R}^2$ with domain Ω and Dirichlet and/or Neumann boundary conditions at the domain boundary $\partial\Omega$;

(ii) the one-dimensional hyperbolic linear advection and Burgers' equations,

$$\partial_t u + \partial_x u = 0 \qquad \text{and} \qquad \partial_t u + u\,\partial_x u = 0$$

with $u = u(x,t)$; and

(iii) the one-dimensional advection-diffusion and viscous Burgers' equations

$$\partial_t u + \partial_x u = \kappa\,\partial_{xx} u \qquad \text{and} \qquad \partial_t u + u\,\partial_x u = \kappa\,\partial_{xx} u$$

with $u = u(x,t)$ and diffusion coefficient $\kappa > 0$. Here, our notation is as follows $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$, $\partial_t = \partial/\partial t$, $\partial_x = \partial/\partial x$, $\partial_{xx} = \partial^2/\partial x^2$.

In the latter two examples (ii) and (iii), the domain is $x \in [0, L]$ with $L > 0$ and the initial condition is $u(x,0) = u_0(x)$. The boundary condition for the linear advection is $u(0,t) = u_b(t)$, while for Burgers' equation there are various choices to be discussed later. For the advection-diffusion and viscous Burger's equations we limit ourselves to periodic boundary conditions.

Poisson's equation is the archetypical elliptic equation and emerges in many problems such as inversion and electrostatics problems. Understanding the finite element discretization for Poisson's equation further provides the tools for discretizing the Helmholtz equation,

$$\nabla^2 \phi - k^2\,\phi = 0$$

for $k^2 > 0$ and real, although the solution of the continuous differential and discretized algebraic system now involves an eigenvalue problem for the eigenvalues k^2. The linear advection equation arises as the advective component of the advection diffusion equation in the limit $\kappa \to 0$, and the quadratic nonlinearity in Burgers' equation is a paradigm for the advective nonlinear terms arising in fluid dynamics.

Note that the three examples coincide with the mathematical classification of PDE's: Poisson's equation and Helmholtz equation are elliptic equations, the linear advection and Burgers' equations are hyperbolic equations, and the advection diffusion and the viscous Burgers' equations are parabolic equations.

1.3 Continuous and discontinuous Galerkin finite element methods

Two finite element methods will be presented: (a) a second-order continuous Galerkin finite element method on triangular, quadrilateral or mixed meshes; and (b) a (space) discontinuous Galerkin finite element method. Consider the triangular mesh in Fig. 1. In the continuous finite element method considered, the function $\phi(x,y)$ will be approximated by a piecewise linear function per element based on the nodal values of ϕ. Hence, the discretization of ϕ denoted as ϕ_h will be continuous.

Similarly, we can consider a piecewise linear discretization of a function $u(x)$ in one dimension, see Fig. 2a, in which the nodal values of $u_h(x)$ determine u_h everywhere else. Hence, this finite element discretization is based on function values at the nodes and is continuous across elements. In contrast, in a discontinuous Galerkin finite element method
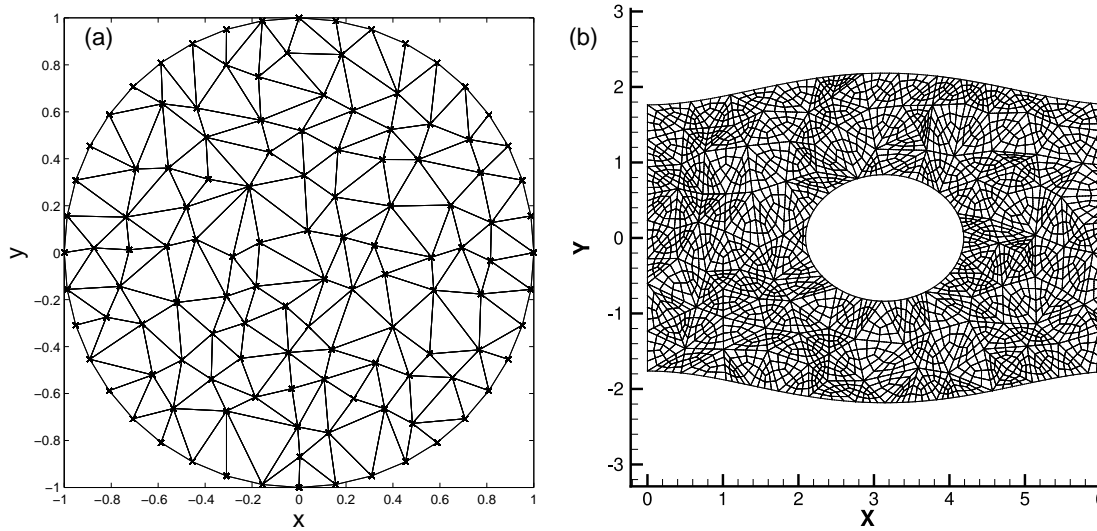
Figure 1.1: (a) This triangular mesh is generated as follows. Boundary points are placed, here regularly, on the circular domain boundary. Subsequently, a specified number of points are randomly placed within the unit circle and accepted when they are larger than a critical distance, proportional to the distance between two boundary points, from all other accepted points. Given these points, the Delaunay triangulation is used to make a balanced triangular mesh including the link between the node and element or assembly information. A counterclockwise orientation is then enforced, after which a "standard" mesh file with the necessary nodal and element information is made. (b) A quadrilateral mesh can be made by placing a point in the middle of each triangle and dividing the triangle into three quadrilaterals. The resulting quadrilateral mesh can then be refined further. The orientation and node to element assembly of this division process is subsequently made. Courtesy Erik Bernsen.

a piecewise linear discretization is found on each element and the limit or trace values approaching the nodes from the element left or right of a node are not assumed to be continuous, see Fig. 2b. Hence, this discretization is continuous in each element but discontinuous across elements. While more degrees of freedom are required in the discontinuous discretization, it offers generally more flexibility and more accuracy.

1.4 Steps in the finite element discretization

Both the continuous and discontinuous Galerkin finite element discretizations usually contain the following steps:
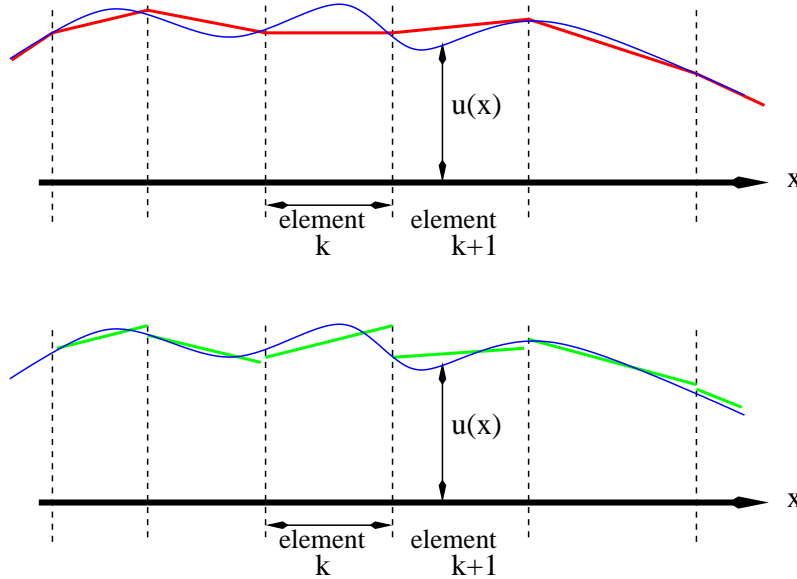
Figure 1.2: In a continuous Galerkin finite element method, the variable $u = u(x)$ is approximated globally in a (piecewise linear) continuous manner (top figure). In contrast, in a discontinuous Galerkin finite element method, the variable $u = u(x)$ is approximated globally in a discontinuous manner and locally in each element in a (piecewise linear) continuous way (bottom figure).

I. *Derive weak formulation:* Each equation is multiplied by its own arbitrary test function, integrated over the domain of validity entirely or as a sum of integrals over all elements, and integrated by parts to obtain the weak formulation.

II. *Form discretized weak formulation/algebraic system:* The variables are expanded in the domain or in each element in a series in terms of a finite number of basis functions. Each basis function has compact support over neighboring elements (for continuous finite elements) or within each element (for discontinuous finite elements). This expansion is then substituted into the weak formulation, and a test function is chosen alternately to coincide with a basis function, to obtain the discretized weak formulation. The resulting system is a linear or nonlinear algebraic system.

III. *Evaluate of integrals in local coordinate system:* A local or reference coordinate system is used to evaluate the integrals. In the continuous finite element discretization global matrices and vectors are assembled in the assembly routine.

IV. *Solve algebraic system:* The resulting algebraic system is solved (iteratively) using forward time stepping methods or linear algebra routines.

Steps I and II are often combined in discontinuous Galerkin methods. If in step III, we have not discretized time, we choose a time discretization of the ordinary differential equations resulting after the spatial finite element discretization. If in step IV the algebraic system is nonlinear, we choose an iterative solution method which essentially will solve a linear system at each iteration step.

Compact support means that the test functions are only nonzero locally over one or a few neighboring elements. In the continuous finite element method (we consider), the basis functions are zero at the edge of their domain of influence, while in the discontinuous case the basis function is (generally) nonzero within an element including the element boundary and zero elsewhere. The careful definition of function spaces for the test and basis functions is common practice in finite element methods. This is often perceived to be complicated, but we will see that all the function spaces make sense, also intuitively.

1.5 Outline

In chapter 2 we discuss the generation of finite element meshes and provide the reference coordinate systems for finite elements in one and two dimensions. A straightforward algorithm is sketched for the generation of unstructured two-dimensional triangular and quadrilateral meshes.

As an introduction, we start in chapter 3 with the well-known piecewise linear continuous Galerkin finite element discretization of the Poisson equation in two dimensions. For more literature we refer to Brenner and Scott (1994), Lucquin and Pirronneau (1998), and Morton (1996).

Space discontinuous Galerkin finite element methods are considered in chapter 4. As an introduction, we consider scalar hyperbolic and parabolic systems in one dimension in section 4.1. We focus therein in particular on the linear advection (diffusion) and (viscous) Burgers' equations as illustrating examples in the definition and explanation of the numerical flux.

One-dimensional systems of hyperbolic equations with additional geometric, source or sink terms are considered in section 4.2. We develop the general discretization but focus on the cross-sectionally averaged shallow water equations with frictional terms to discuss a numerical flux. The choice of numerical flux is different for each hyperbolic system and depends in general on the nature of the eigenvalues in the system. We refer to the work of Cockburn and Shu (1989) and Cockburn et al. (1989) on space discontinuous Galerkin methods for the one-dimensional scalar hyperbolic equation and hyperbolic systems. Bokhove (2005) combined the space discontinuous Galerkin method with Eulerian-Lagrangian free boundary dynamics to simulate flooding and drying for the shallow water equation, including break-up in multiple patches.

The space local discontinuous Galerkin (LDG) method is considered in section 4.1.2 for the one-dimensional advection diffusion and the viscous Burgers' equations. We refer to Cockburn and Shu (1998) and Yan and Shu (2002) for the local discontinuous Galerkin discretizations of (nonlinear) advection diffusion and wave equations. The local discontinuous Galerkin method applied to a one-dimensional nonlinear advection diffusion equation arising in geology is combined with positivity preservation and Lagrangian free boundary dynamics in Bokhove et al. (2005), where also a fractional basis function is used in the free boundary element.

Finally, the space-time discontinuous Galerkin method is considered in chapter 5 for scalar conservation laws.

The presentation of the material is (largely) not new but based on our own work. We have included a list for the interested reader in chapter 6. It is a pleasure to thank Vijaya Ambati, Christiaan Klaij, Willem Ottevanger, Sander Rhebergen, Henk Sollie and Yan Xu for proofreading these notes.

In the JMBC Burger's course 2008, the practical sessions concern exercise 1, especially a–c, in section 4.3. Jaap van der Vegt will present the material on space discontinuous Galerkin methods in section 4.1 and space-time discontinuous Galerkin methods in chapter 5.

Chapter 2

Finite element mesh

2.1 Mesh generation

2.1.1 Mesh generator

There are structured and unstructered meshes. Structured meshes, which do not need to be regular, can be generated relatively straightforward, but three-dimensional unstructered meshes generally require a mesh generation package. There are several mesh generation packages commercially available.

2.1.2 Pseudo two-dimensional meshes using Delaunay triangulation

Three-dimensional meshes, regular in one direction, and unstructured two-dimensional meshes can be made using the well-known Delaunay triangulation. The mesh generation algorithm is outlined in Fig. 1.1 and its caption. It can be used to generate triangular as well as quadrilateral meshes. We note that the quadrilateral mesh displayed is formed from a triangular base mesh and is refined further, see Fig. 1.1. For multiple-connected and non-convex domains, the algorithm requires some extensions to exclude triangles outside the domain that can be generated erroneously in the Delaunay step. In addition, to generate a balanaced quadrilateral mesh near curved boundaries the placement of the point in the parent triangle or quadrilateral requires some care to avoid elements with angles that are too small (see Bernsen et al., 2006). The advantage of the algorithm outlined in Fig. 1.1 lies in its simplicity. It is relatively easily implemented by a user without the need of a complicated mesh generation package. We implemented it in Matlab using Matlab's built-in Delaunay triangulation.

In the end, the mesh file allows us to obtain the global node number $i = Index(\mathsf{k}, \alpha)$ given the element number k and the local node number α. Each element has a local node number, definitely (counterclockwise) oriented. For triangles $\alpha = 0, 1, 2$ and for quadrilaterals $\alpha = 0, 1, \ldots, 3$. Discontinuous Galerkin element methods also require a list of the faces in the

mesh and the element (number) to the left and right of this face.

2.2 Reference coordinates

2.2.1 One dimension

The one-dimensional domain $\Omega = x \in [a,b]$ $(b > a; a, b \in \mathbb{R})$ is partitioned by points $x_k(t)$, $k = 1, \cdots, N_{\mathrm{el}} + 1$, into N_{el} open elements $K_k = \{x | x \in (x_k, x_{k+1})\}$. The result is a tessellation

$$\mathcal{T}_h = \{K_k | \bigcup_{k=1}^{N_{\mathrm{el}}} \bar{K}_k = \bar{\Omega} \text{ and } K_k \cap K_{k'} = \emptyset \text{ if } k \neq k', 1 \leq k, k' \leq N_{\mathrm{el}}\} \tag{2.2.1}$$

with $\bar{K}_k$ the closure of K_k. For convenience, we will also use the notation $x_{k,L} := x_k$ and $x_{k,R} := x_{k+1}$ below. We define $|K_k| = x_{k,R} - x_{k,L}$. We further introduce a reference element $\hat{K}$ with the local or reference coordinate $\zeta \in (-1, 1)$ such that $x = x(\zeta) = (x_{k+1} + x_k)/2 + |K_k| \zeta/2$. Hence, $dx/d\zeta = |K_k|/2$.

2.2.2 Two dimensions

We consider finite elements in two dimensions spanned by coordinates $\bar{x} = (x, y)^T$ with transpose ^{T}. The domain $\Omega \subset \mathbf{R}^2$ is partitioned into a mixture of N_{el} quadrilateral and triangular elements such that we obtain the tesselation

$$\mathcal{T}_h = \{K_{\mathsf{k}} | \bigcup_{\mathsf{k}=1}^{N_{\mathrm{el}}} \bar{K}_{\mathsf{k}} = \bar{\Omega} \text{ and } K_{\mathsf{k}} \cap K_{\mathsf{k}'} = \emptyset \text{ if } \mathsf{k} \neq \mathsf{k}', 1 \leq \mathsf{k}, \mathsf{k}' \leq N_{\mathrm{el}}\} \tag{2.2.2}$$

with $\bar{K}_{\mathsf{k}}$ the closure of element K_{k}. Each such element K_{k} has $N_{\mathsf{n}}^{\mathsf{k}} = N_{\mathsf{n}} = 3$ or 4 nodes, which are numbered locally in a counterclockwise fashion from 0 to $N_{\mathsf{n}} - 1$. It is convenient to introduce a reference element $\hat{K}$ and define the mapping $F_K : \mathbb{R}^2 \to \mathbb{R}^2$ between the reference element $\hat{K}$ and element K_{k} as follows

$$\bar{x} = F_{K_{\mathsf{k}}}(\bar{\zeta}) = \sum_{\alpha=0}^{N_{\mathsf{n}}^{\mathsf{k}}-1} \bar{x}_{\mathsf{k},\alpha} \, \chi_\alpha(\bar{\zeta}) \tag{2.2.3}$$

with reference coordinate $\bar{\zeta} = (\zeta_1, \zeta_2)^T$ and shape functions

$$\chi_\alpha(\bar{x}) = \chi_\alpha(F_{K_{\mathsf{k}}}(\bar{\zeta}). \tag{2.2.4}$$

Triangles have $N_{\mathsf{n}} = 3$ nodes $\bar{x}_{\mathsf{k},m} = \bar{x}_m$, corresponding to nodes $(0,0)^T, (1,0)^T, (0,1)^T$ in the reference element, an the shape functions are

$$\chi_0(\bar{\zeta}) = \zeta_0 = 1 - \zeta_1 - \zeta_2, \quad \chi_1(\bar{\zeta}) = \zeta_1, \quad \chi_2(\bar{\zeta}) = \zeta_2. \tag{2.2.5}$$

The three normal vectors for triangular elements are

$$\hat{n}_0 = \frac{1}{|\bar{x}_1 - \bar{x}_0|}\,(y_1 - y_0, x_0 - x_1)^T, \qquad \hat{n}_1 = \frac{1}{|\bar{x}_2 - \bar{x}_1|}\,(y_2 - y_1, x_1 - x_2)^T,$$

$$\hat{n}_2 = \frac{1}{|\bar{x}_0 - \bar{x}_2|}\,(y_0 - y_2, x_2 - x_0)^T. \tag{2.2.6}$$

The Jacobian $J_3 = J$ of the transformation between $\bar{x}$ and $\bar{\zeta}$ is

$$J_3 = \begin{pmatrix} \partial_{\zeta_1} x & \partial_{\zeta_1} y \\ \partial_{\zeta_2} x & \partial_{\zeta_2} y \end{pmatrix} = \begin{pmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{pmatrix}. \tag{2.2.7}$$

We computed the inverse of the Jacobian using the following relations

$$\frac{\partial x}{\partial x} = \frac{\partial x}{\partial \zeta_1}\frac{\partial \zeta_1}{\partial x} + \frac{\partial x}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial x} = 1, \qquad \frac{\partial x}{\partial y} = \frac{\partial x}{\partial \zeta_1}\frac{\partial \xi_1}{\partial y} + \frac{\partial x}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial y} = 0,$$

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial \xi_1}\frac{\partial \zeta_1}{\partial x} + \frac{\partial y}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial x} = 0 \qquad \frac{\partial y}{\partial y} = \frac{\partial y}{\partial \xi_1}\frac{\partial \zeta_1}{\partial y} + \frac{\partial y}{\partial \zeta_2}\frac{\partial \zeta_2}{\partial y} = 1,$$

or, in matrix notation

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial \zeta_1} & \frac{\partial x}{\partial \zeta_2} \\ \frac{\partial y}{\partial \zeta_1} & \frac{\partial y}{\partial \zeta_2} \end{pmatrix} \begin{pmatrix} \frac{\partial \zeta_2}{\partial x} & \frac{\partial \zeta_1}{\partial y} \\ \frac{\partial \zeta_2}{\partial x} & \frac{\partial \zeta_2}{\partial y} \end{pmatrix} = J J^{-1}.$$

Hence, gradients in $\bar{x}$ transform as follows

$$\begin{pmatrix} \partial_x V \\ \partial_y V \end{pmatrix} = (J^T)^{-1} \begin{pmatrix} \partial_{\zeta_1} V \\ \partial_{\zeta_2} V \end{pmatrix} = \frac{1}{\det J_3} \begin{pmatrix} y_2 - y_0 & y_0 - y_1 \\ x_0 - x_2 & x_1 - x_0 \end{pmatrix} \begin{pmatrix} \partial_{\zeta_1} V \\ \partial_{\zeta_2} V \end{pmatrix} \tag{2.2.8}$$

with determinant $\det J_3$ and $|J_3| = |\det J_3|$. Triangles have three faces or sides $S_0, \ldots, S_2$ spanned by node pairs $(\bar{x}_0, \bar{x}_1)$, $\ldots$, $(\bar{x}_2, \bar{x}_0)$. The faces $S_0, \ldots, S_2$ correspond to $\zeta_2 = 0, \zeta_1 + \zeta_2 = 1$ and $\zeta_1 = 0$ in the reference element, respectively.

For quadrilaterals there are $N_n = 4$ nodes $\bar{x}_{k,m} = \bar{x}_m$. In the reference element $\bar{\zeta} \in (-1, 1)^2$, and we start counterclockwise with $\alpha = 0$ at node $(-1, -1)^T$ in the reference element. Furthermore, the shape functions are

$$\chi_0(\bar{\zeta}) = (1 - \zeta_1)\,(1 - \zeta_2)/4, \qquad \chi_1(\bar{\zeta}) = (1 + \zeta_1)\,(1 - \zeta_2)/4,$$

$$\chi_2(\bar{\zeta}) = (1 + \zeta_1)\,(1 + \zeta_2)/4, \qquad \chi_3(\bar{\zeta}) = (1 - \zeta_1)\,(1 + \zeta_2)/4. \tag{2.2.9}$$

The four normal vectors for quadrilateral elements are

$$\hat{n}_0 = \frac{1}{|\bar{x}_1 - \bar{x}_0|}\,(y_1 - y_0, x_0 - x_1)^T, \qquad \hat{n}_1 = \frac{1}{|\bar{x}_2 - \bar{x}_1|}\,(y_2 - y_1, x_1 - x_2)^T,$$

$$\hat{n}_2 = \frac{1}{|\bar{x}_3 - \bar{x}_2|}\,(y_3 - y_2, x_2 - x_3)^T, \qquad \hat{n}_3 = \frac{1}{|\bar{x}_0 - \bar{x}_3|}\,(y_0 - y_3, x_3 - x_0)^T. \tag{2.2.10}$$

The Jacobian $J_4 = J$ of the transformation between $\bar{x}$ and $\bar{\zeta}$ is

$$J_4 = J_4(\bar{\zeta}) = \begin{pmatrix} \partial_{\zeta_1} x & \partial_{\zeta_1} y \\ \partial_{\zeta_2} x & \partial_{\zeta_2} y \end{pmatrix}$$

$$= \frac{1}{4} \begin{pmatrix} (1 - \zeta_2)(x_1 - x_0) + (1 + \zeta_2)(x_2 - x_3) & (1 - \zeta_2)(y_1 - y_0) + (1 + \zeta_2)(y_2 - y_3) \\ (1 - \zeta_1)(x_3 - x_0) + (1 + \zeta_1)(x_2 - x_1) & (1 - \zeta_1)(y_3 - y_0) + (1 + \zeta_1)(y_2 - y_1) \end{pmatrix}.$$

$$(2.2.11)$$

Hence, gradients in $\bar{x}$ transform as follows

$$\begin{pmatrix} \partial_x V \\ \partial_y V \end{pmatrix} = (J^T)^{-1} \begin{pmatrix} \partial_{\zeta_1} V \\ \partial_{\zeta_2} V \end{pmatrix} = \frac{1}{\det J_4} \begin{pmatrix} \partial_{\zeta_2} y & -\partial_{\zeta_1} y \\ -\partial_{\zeta_2} x & \partial_{\zeta_1} x \end{pmatrix} \begin{pmatrix} \partial_{\zeta_1} V \\ \partial_{\zeta_2} V \end{pmatrix} \qquad (2.2.12)$$

with the determinant $\det J_4$ and $|J_4| = |\det J_4|$. Quadrilaterals have four faces $S_0, \ldots, S_3$ spanned by node pairs $(\bar{x}_0, \bar{x}_1)$, ..., $(\bar{x}_3, \bar{x}_0)$. The sides $S_0, \ldots, S_3$ correspond to $\zeta_2 = -1, \zeta_1 = 1, \zeta_2 = 1$, and $\zeta_1 = -1$ in the reference element, respectively.

Chapter 3

Continuous finite elements

3.1 Example: Poisson's equation

3.1.1 Equation and boundary conditions

Consider Poisson's equation on the domain $\Omega \subset \mathbb{R}^2$

$$-\triangle\phi = f \qquad \text{in } \Omega \tag{3.1.1a}$$

$$\phi = g \qquad \text{at } \partial\Omega_D \tag{3.1.1b}$$

$$\hat{\mathbf{n}} \cdot \boldsymbol{\nabla}\phi = 0 \qquad \text{at } \partial\Omega_N \tag{3.1.1c}$$

with $\triangle = \nabla^2$, the boundary $\partial\Omega$ given as the union of Dirichlet and Neumann parts such that $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$, and outward normal $\hat{\mathbf{n}}$ at $\partial\Omega$. Given the functions $f = f(\bar{x})$ on Ω and $g(\bar{x})$ for $\bar{x} \in \partial\Omega_D$, we wish to solve $\phi = \phi(\bar{x})$.

Poisson's equation can be integrated over the domain to obtain, after integration by parts and use of the boundary conditions,

$$-\int_{\partial\Omega_D} \hat{\mathbf{n}} \cdot \boldsymbol{\nabla}\phi \, d\Gamma = \int_\Omega f \, d\Omega \tag{3.1.2}$$

with $d\Gamma$ a lign segment along the boundary. Note that in case $\partial\Omega = \partial\Omega_N$ equation (3.1.2) reduces to the constraint $\int_\Omega f \, d\Omega = 0$, in which case function f needs to satisfy this constraint.

3.1.2 Weak formulation

We define the spaces of square integrable functions

$$L^2(\Omega) := \{v \mid \int_\Omega |v|^2 d\Omega < \infty\} \tag{3.1.3}$$

13

and the Hilbert space

$$H_g^1(\Omega) := \{v \in L^2(\Omega) \,|\, \frac{\partial v}{\partial x}, \frac{\partial v}{\partial y} \in L^2(\Omega), \, v|_{\partial \Omega_D} = g\} \tag{3.1.4}$$

with function v satisfying the Dirichlet boundary condition (3.1.1b).

Next, we multiply (3.1.1a) by an arbitrary test function $v \in H_0^1$, integrate over Ω, and use Gauss' law while using the boundary conditions (3.1.1b) and (3.1.1c) to obtain

$$-\int_\Omega v \nabla^2 \phi \, d\bar{x} = \int_\Omega \nabla v \cdot \boldsymbol{\nabla} \phi - \boldsymbol{\nabla} \cdot (v \, \boldsymbol{\nabla} \phi) \, d\bar{x} \tag{3.1.5}$$

$$= \int_\Omega \nabla v \cdot \boldsymbol{\nabla} \phi \, d\bar{x} - \int_{\Omega_D} v \, \hat{\mathbf{n}} \cdot \boldsymbol{\nabla} \phi \, d\Gamma - \int_{\Omega_N} v \, \hat{\mathbf{n}} \cdot \boldsymbol{\nabla} \phi \, d\Gamma \tag{3.1.6}$$

$$= \int_\Omega \nabla v \cdot \boldsymbol{\nabla} \phi \, d\bar{x} = \int_\Omega v \, f \, d\bar{x} \tag{3.1.7}$$

with $\hat{\mathbf{n}}$ the outward normal at the boundary, and $d\Gamma$ an infinitesimal line element along the boundary. Note that we have chosen $v \in H_0^1$ to eliminate the boundary contribution at the Dirichlet boundary, and this is allowed because ϕ is known at $\partial \Omega_D$.

The weak formulation for the Laplace equation thus becomes: Find a $\phi \in H_g^1(\Omega)$, such that for all $v \in H_0^1(\Omega)$ the following relation is satisfied

$$\sum_{K \in \mathcal{T}_h} \int_K \boldsymbol{\nabla} \phi \cdot \boldsymbol{\nabla} v \, d\Omega = \sum_{K \in \mathcal{T}_h} \int_\Omega v \, f \, d\Omega \tag{3.1.8}$$

with $\mathcal{T}_h$ the set of non-overlapping triangular or quadrilateral elements K covering Ω.

3.1.3 Discretized weak formulation

We approximate the unknown function ϕ by ϕ_h defined as an expansion

$$\phi(\bar{x}) \approx \phi_h(\bar{x}) = \sum_{j=1}^{N_{nodes}} \phi_j \, w_j(\bar{x}) \tag{3.1.9}$$

in terms of global basis functions $w_j(\bar{x}$, with $j = 1, \ldots, N_{nodes}$ the node number and N_{nodes} the total number of nodes (not on $\partial \Omega_D$). Often $w_j(\bar{x})$ is chosen such that $w_j = 1$ on node j and zero beyond and on neighboring nodes. Thus w_j has compact support. The test function v is arbitrary as long as it belongs to $H_0^1(\Omega)$. We therefore take $v = w_i$ alternately for all $i = 1, \ldots, N_{nodes}$ to obtain N_{nodes} algebraic equations for the N_{nodes} unknowns. If we take $v = w_i$ in (3.1.8), we obtain the algebraic system

$$\sum_{K \in \mathcal{T}_h} \left(\int_K \boldsymbol{\nabla} w_i \cdot \boldsymbol{\nabla} w_j \, d\Omega \right) \phi_j = \sum_{K \in \mathcal{T}_h} \int_K f \, w_i \, d\Omega, \tag{3.1.10}$$

where we use the summation convention to sum over repeated indices. In matrix form, we write (3.1.10) as follows

$$A_{ij}\,\phi_j = b_i \tag{3.1.11}$$

in which the global matrix components

$$A_{ij} = \sum_{K \in \mathcal{T}_h} \int_K \boldsymbol{\nabla} w_i \cdot \boldsymbol{\nabla} w_j \, d\Omega \qquad i,j \in 0,\dots,N_{nodes} \tag{3.1.12}$$

and the vector slots

$$b_i = \sum_{K \in \mathcal{T}_h} \int_K f\, w_i \, d\Omega \qquad i \in 0,\dots,N_{nodes} \tag{3.1.13}$$

follow from (3.1.10). We desire to obtain the solution vector $\boldsymbol{\phi} = (\phi_1,\dots,\phi_{N_{nodes}})^T$ by inversion of the linear equations (3.1.11). Note that since w_i has compact support, the evaluation of the integrals in A_{ij} and b_i only involves a limited number of neighboring elements around node i.

3.1.4   Evaluation of integrals

The Galerkin basis functions $w_i(x,y)$ with global node number i are now defined on element K_k in a piecewise linear manner as:

$$w_\alpha(x,y) = \hat{w}_\alpha(F_K^{-1}(x,y)) = \chi_\alpha(\bar{\zeta})$$

with α the local element index on element K_k for which $w_\alpha = 1$ on global node i and zero at the other nodes. This choice of basis functions gives formally second-order accuracy. Other (higher-order) basis functions can be found in the literature (e.g., Brenner and Scott, 1994; for an application see Bernsen et al., 2006).

The definition of the global matrix and vector components in (3.1.12) and (3.1.13) suggest the definition of the following two-by-two elemental matrix and two-by-one vector

$$\hat{A}_{\alpha\beta} = \int_K \boldsymbol{\nabla}\chi_\alpha \cdot \boldsymbol{\nabla}\chi_\beta \, d\Omega \tag{3.1.14}$$

$$= \int_{\hat{K}} \left((J^T)^{-1} \begin{pmatrix} \frac{\partial \chi_\alpha}{\partial \zeta_1} \\ \frac{\partial \chi_\alpha}{\partial \zeta_2} \end{pmatrix} \right) \cdot \left((J^T)^{-1} \begin{pmatrix} \frac{\partial \chi_\beta}{\partial \zeta_1} \\ \frac{\partial \chi_\beta}{\partial \zeta_2} \end{pmatrix} \right) |\det(J)| \, d\bar{\zeta}$$

$$= \int_{\hat{K}} \begin{pmatrix} \frac{\partial \chi_\alpha}{\partial \zeta_1} \\ \frac{\partial \chi_\alpha}{\partial \zeta_2} \end{pmatrix}^T J^{-1}(J^T)^{-1} \begin{pmatrix} \frac{\partial \chi_\alpha}{\partial \zeta_1} \\ \frac{\partial \chi_\beta}{\partial \zeta_2} \end{pmatrix} |\det(J)| \, d\bar{\zeta} \tag{3.1.15}$$

$$\hat{b}_\alpha = \int_K f\, w_\alpha \, d\Omega \tag{3.1.16}$$

$$= \int_{\hat{K}} f\big(x(\zeta_1,\zeta_2), y(\zeta_1,\zeta_2)\big)\chi_\alpha(\zeta_1,\zeta_2)|\det(J(\bar{\zeta})|\, d\bar{\zeta} \tag{3.1.17}$$

for $\alpha, \beta = 0, \ldots, N_n^k - 1$ on each reference element $\hat{K}$.

Note, the basis functions w_i are only non-zero in the elements connected to the vertex with index i and it is therefore more practical to transform the integrals over the element K into integrals over the reference element $\hat{K}$ The integrals in (3.1.15)-(3.1.17) can be computed with a Gauss quadrature rule as follows, on a square reference element

$$\int_{-1}^{1} \int_{-1}^{1} h(\xi, \eta) d\xi d\eta = \sum_{n=1}^{2} \sum_{m=1}^{2} h(\xi_m, \eta_n)$$

for some function $h(\cdot, \cdot)$ with $\xi_1, \eta_1 = -1/\sqrt{3}$ and $\xi_2, \eta_2 = 1/\sqrt{3}$, and on a triangular reference element

$$\int_{0}^{1} \int_{0}^{1-\xi} h(\xi, \eta) d\xi d\eta = \frac{1}{6} \left(h(1/2, 0) + h(0, 1/2) + h(1/2, 1/2) \right).$$

Given these elemental matrices and vectors we assemble the global matrix, $\mathbf{A}$, and global vector, $\mathbf{b}$, in the following assembly algorithm:

set all components of $\mathbf{A} = 0$ and $\mathbf{b} = 0$ to zero: $A_{ij} = b_i = 0$
for all elements K_k, $k = 1, N_{\text{el}}$, do
.   for $\alpha = 1, N_n^k$ do
. $i = Index(k, \alpha)$
.   for $\beta = 1, N_n^k$ do
. $j = Index(k, \beta)$
. $A_{ij} = A_{ij} + \hat{A}_{\alpha\beta}$
. $b_i = b_i + \hat{b}_\alpha$

The function $Index(k, \alpha)$ maps the local vertex or node numbered α of element K to the global node number. This index is derived or available from the generated mesh file. Note that we only included the interior nodes and the non-Dirichlet boundary nodes i, j.

The matrix A_{ij} is symmetric. Efficient methods to solve (3.1.11) involve Choleski factorizations or conjugate gradients (seee, e.g., Lucquin and Pironneau, 1998; and, Van der Vorst, 2003).

Chapter 4

Space discontinuous Galerkin finite element methods

4.1 Examples

In this section, we consider the space discontinuous Galerkin finite element discretization of the linear advection equation and the nonlinear inviscid Burgers' equation as well as the linear advection diffusion and nonlinear viscous Burgers' equation. The spatial finite element discretization will result in a system of ordinary, coupled algebraic and ordinary differential equations in time. Subsequentlty, we use a standard Runge-Kutta method (Shu and Osher, 1989) to discretize time.

The domain in one dimension is simply a rod of length L divided into N_{el} irregular intervals or elements, see Fig. 4.1. More formally, we define a tessellation $\mathcal{T}_h$ of N_{el} elements K_k in the spatial flow domain $\Omega = [x_1 = 0, x_{N_{\mathrm{el}}+1} = L]$ with $L > 0$ and boundary $\partial\Omega$:

$$\mathcal{T}_h = \{K_k : \bigcup_{k=1}^{N_{\mathrm{el}}} \bar{K}_k = \bar{\Omega} \text{ and } K_k \cap K_{k'} = \emptyset \text{ if } k \neq k', 1 \leq k, k' \leq N_{\mathrm{el}}\}, \qquad (4.1.1)$$

where $\bar{K}_k$ denotes the closure of K_k, et cetera. Element K_k runs from node x_k to node x_{k+1} and has length $|K_k| = x_{k+1} - x_k$.
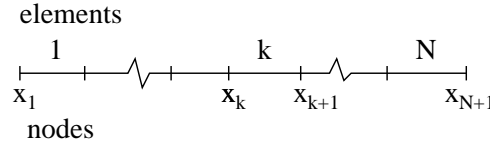

Figure 4.1: A sketch of the one-dimensional finite element mesh with a definition of the node and element numbering. We denote with $N = N_{\mathrm{el}}$ the total number of elements.

4.1.1   Linear advection and inviscid Burgers' equation

Equations

We consider in particular the linear advection equation

$$\partial_t u + a\,\partial_x u = 0 \qquad\qquad (4.1.2)$$

for $a \in \mathbb{R}$ and the inviscid Burgers' equation

$$\partial_t u + u\,\partial_x u = 0 \qquad\qquad (4.1.3)$$

with initial condition $u_0 = u(x,0)$. The boundary conditions for the linear advection equation are either (i) periodic, or (ii) specified at the inflow boundary:

$$u(0,t) = u_{left}(t) \quad \text{if} \quad a > 0 \qquad \text{or} \qquad u(L,t) = u_{right}(t) \quad \text{if} \quad a \le 0. \qquad (4.1.4)$$

The boundary conditions for the Burgers' equation are either (i) periodic, (ii) specified at the inflow boundaries:

$$u(0,t) = u_{left}(t) \quad \text{if} \quad u(0,t) > 0 \qquad \text{or} \qquad u(L,t) = u_{right}(t) \quad \text{if} \quad u(L,t) \le 0, \qquad (4.1.5)$$

or (iii) solid walls with $u(0,t) = u(L,t) = 0$.

It is convenient to set up the discretization for a general hyperbolic scalar equation. We therefore concisely combine (4.1.2) and (4.1.3) as follows

$$\partial_t u + \partial_x f = 0 \qquad\qquad (4.1.6)$$

with the flux $f = f(u)$ in general, and $f(u) = a\,u$ and $f(u) = u^2/2$ for the linear advection and Burgers' equations in particular; and initial condition $u_0(x) = u(x,0)$.

Weak formulation

We multiply (4.1.6) by an arbitrary test function $w = w(x)$ (smooth within each element), integrate by parts over each individual and isolated element, and then add the contribution from all elements to obtain the following weak formulation:

$$\sum_{k=1}^{N_{\text{el}}} \left\{ \int_{K_k} w\,\frac{du}{dt}\,\mathrm{d}x + [f(x_{k+1}^-)\,w(x_{k+1}^-) - f(x_k^+)\,w(x_k^+)] - \int_{K_k} f(u)\,\partial_x w\,\mathrm{d}x \right\} = 0, \quad (4.1.7)$$

where $w(x_{k+1}^-) = \lim_{x\uparrow x_{k+1}} w(x,t)$ and $w(x_k^+) = \lim_{x\downarrow x_k} w(x,t)$. (We only denote these dependencies explicitly when confusion may arise.) Hence, the fluxes at the faces arising in element K_k are evaluated inside the element.

Let $[u] = u_+ - u_-$ and $\bar{u} = (u_+ + u_-)/2$ denote the jump and mean in the quantity u, say, at x_k with $u_- = \lim_{x\uparrow x_k} u(x)$ and $u_+ = \lim_{x\downarrow x_k} u(x)$. Consider the flux at a point

x_{k+1}. Since the elements K_k are isolated from one another at this stage $u_- := u(x_{k+1}^-) \neq u(x_{k+1}^+) =: u_+$.

We rewrite the weak formulation (4.1.7) as a sum over the elements for the interior integrals and a sum over the nodes for the "face" or nodal flux terms

$$\sum_{k=1}^{N_{\mathrm{el}}} \left\{ \int_{K_k} w\,\frac{du}{dt}\,\mathrm{d}x - \int_{K_k} f(u)\,\partial_x w\,\mathrm{d}x \right\} - f(x_1^+)\,w(x_1^+) + f(x_{N_{\mathrm{el}}+1}^-)\,w(x_{N_{\mathrm{el}}+1}^-)$$

$$+ \sum_{k=2}^{N_{\mathrm{el}}} \bigl(f(x_k^-)\,w(x_k^-) - f(x_k^+)\,w(x_k^+) \bigr) = 0. \tag{4.1.8}$$

The flux term at the interior nodes is rewritten as follows

$$f(x_k^-)\,w(x_k^-) - f(x_k^+)\,w(x_k^+) = - \bigl((\gamma_1\,f_+ + \gamma_2\,f_-)\,[w] + [f]\,(\gamma_2\,w_+ + \gamma_1\,w_-) \bigr)$$

$$\overset{[f]=0}{=} - (\gamma_1\,f_+ + \gamma_2\,f_-)\,[w] \tag{4.1.9}$$

if we enforce continuity, i.e. $[f] = 0$, at a node; also $\gamma_1 + \gamma_2 = 1$ and $\gamma_{1,2} \geq 0$.

The heart of the discontinuous Galerkin numerical method hinges now on the choice of the numerical flux. In (4.1.9), we enforced continuity of the flux, which suggests that we should replace $f(x_k^-)$ and $f(x_k^+)$ both by the same numerical flux $\tilde{f} = \tilde{f}(u_-, u_+) = \gamma_1\,f_+ + \gamma_2\,f_-$. This is relevant since $f(x_k^-) \neq f(x_k^+)$ in the numerical discontinuous finite element discretization. It turns out that for arbitrary $\gamma_{1,2}$ with $\gamma_1 + \gamma_2 = 1$ instabilities can occur. In general, the numerical flux is chosen as a function of both u_+ and u_-: $\tilde{f} = \tilde{f}(u_-, u_+)$ or $\tilde{f}(x_k) = \tilde{f}\bigl(u_-(x_k), u_+(x_k)\bigr)$. We discuss choice of the numerical flux shortly.

Discretized weak formulation

We consider finite-element discretizations of the general linear advection and inviscid Burgers' equations (4.1.6) with approximations u_h, w_h to the variable $u = u(x,t)$ and test functions $w = w(x)$. These are such that u_h and w_h belong to the broken space

$$V_h = \bigl\{ v \big| v|_{K_k} \in P^{d_P}(K_k), k = 1, \dots, N_{\mathrm{el}} \bigr\}, \tag{4.1.10}$$

in which $P^{d_P}(K_k)$ denotes the space of polynomials in K_k of degree d_P. Note that u_h is continuous in an element but generally discontinuous at the nodes, i.e. across element boundaries.

After replacement of u, w by u_h, w_h and f at the nodes by the numerical flux $\tilde{f}$ in (4.1.8),

we arrive at the following weak formulation

$$\sum_{k=1}^{N_{\text{el}}} \left\{ \int_{K_k} w_h \frac{du_h}{dt}\, dx - \int_{K_k} f(u_h)\, \partial_x w_h\, dx \right\} - \tilde{f}\left(u_{left}, u_h(x_1^+)\right) w_h(x_1^+)$$

$$+ \tilde{f}\left(u_h(x_{N_{\text{el}}+1}^-), u_{right}\right) w_h(x_{N_{\text{el}}+1}^-) + \sum_{k=1}^{N_{\text{el}}} \tilde{f}\left(u_h(x_k^-), u_h(x_k^+)\right)\left(w_h(x_k^-) - w_h(x_k^+)\right) = 0$$

$$(4.1.11)$$

with u_{left} and u_{right} the boundary values at the left and right boundary chosen to enforce the boundary conditions, if necessary. For the linear advection equation with $a > 0$, with a "wind" a blowing from left to right, no boundary condition is required at $x = L$, and vice versa for $a < 0$.

To reduce the partial differential equations explicitly to ordinary differential equations, we choose a finite number of (orthogonal) polynomials to expand the variables in each element as follows:

$$u_h(x, t) = \sum_{m=0}^{d_P} \hat{U}_m(K_k, t)\, \psi_{m,k}(x), \qquad w_h(x) = \sum_{m=0}^{d_P} \hat{W}_m(K_k)\, \psi_{m,k}(x) \qquad (4.1.12)$$

with polynomial basis functions $\psi_{m,k}(x) \in P^{d_P}(K_k)$. In one dimension the number of coefficients equals equals the degree of the polynomials plus one. These coefficients are chosen such that

$$\hat{U}_0 = \bar{U}(K_k, t) = \int_{K_k} u(x, t)\, dx / |K_k| \quad \text{and}$$

$$\psi_{m,k}(x) = \left\{ \begin{array}{ll} 1 & \text{if } m = 0 \\ \varphi_{m,k}(x) - \int_{K_k} \varphi_{m,k}(x)\, dx / |K_k| & \text{if } m \geq 1 \end{array} \right\}.$$

Here, for example, we approximate u_h, w_h on K_k by a mean and a slope

$$u_h(x, t) = \bar{U}_k + \hat{U}_k\, \psi_{1,k}(x) \quad \text{and} \quad w_h(x) = \bar{W}_k + \hat{W}_k\, \psi_{1,k}(x) \qquad (4.1.13)$$

with $\bar{U}_k = \bar{U}(K_k, t)$ the mean and $\hat{U}_k = \hat{U}_1(K_k, t)$ the slope. We note that $\psi_{1,k} = \zeta$.

Since coefficient $\bar{W}_k$ and $\hat{W}_k$ are arbitrary we can alternately set $\bar{W}_k$ and $\hat{W}_k$ to one in each element and the other coefficients $\bar{W}_k, \hat{W}_k$ to zero. We thus obtain, after substituting (4.1.13) into (4.1.11) and using the arbitrariness of $\bar{W}_k$ and $\hat{W}_k$, the following equations for the mean and fluctuating part per element

$$|K_k| \frac{d\bar{U}_k}{dt} + \tilde{f}(x_{k+1}) - \tilde{f}(x_k) = 0 \qquad (4.1.14a)$$

$$\frac{|K_k|}{3} \frac{d\hat{U}_k}{dt} + [\tilde{f}(x_{k+1}) + \tilde{f}(x_k)] - \int_{-1}^{1} f(U_h)\, d\zeta = 0. \qquad (4.1.14b)$$

Herein, the integrals expressed in the the reference coordinate system are approximated with a third-order two-point Gauss quadrature rule

$$\int_{-1}^{1} f(\zeta)\, d\zeta \approx f(-c_m) + f(c_m) \tag{4.1.15}$$

with $c_m = 1/\sqrt{3}$ for some function $f = f(\zeta)$.

Numerical flux

The numerical flux, $\tilde{f}(u_-, u_+)$, is chosen to (i) be consistent such that $\tilde{f}(u, u) = f(u)$, (ii) be conservative, which we have used in the derivation in (4.1.9), and (iii) reduce to an E-flux, that is,

$$\int_{u_-}^{u_+} f(s) - \tilde{f}(u_-, u_+)\, ds \geq 0.$$

This last property of the E-flux guarantees L^2-stability as we will discuss later. Note that the numerical flux is the only way of communication between elements, and that the flux is determined by the values of u_h immediately left and right of each face.

Given the values u_- and u_+ immediately left and right of each face or node x_k, we wish to obtain a sensible numerical flux. Note that u_h is generally not constant in the neighboring elements $k - 1$ and k adjacent to node x_k, except when only a leading-order expansion is used for which $u(x, t)$ is approximated by its mean per element only. A well-known way is to extend the values $u_\pm$ into the left and right elements, and calculate the exact solution to a local Riemann problem around each node x_k: find the solution $u = u_e(x, t)$ for $t > t_0$ of

$$\partial_t u + \partial_x f(u) = 0 \tag{4.1.16}$$

$$u(x, t_0) = \begin{cases} u_- & x < x_k \\ u_+ & x \geq x_k \end{cases}. \tag{4.1.17}$$

The numerical flux is then defined as $\tilde{f} = f(u_e(x_k, t))$, which is an approximate flux as solution u_e is an exact solution to a local problem with simplified (piecewise constant) initial condition. These local Riemann problems do not interfere with another as long as the time step is small enough. Roughly speaking, the (shock or rarefaction) waves arising from the jumps at the nodes should not cross another. Hence, there is a smoothing step involved in each time step since the numerical solution is a projection onto a finite number of polynomials.

The linear advection equation has characteristics $dx/dt = a$. Hence, $du/dt = 0$ on this characteristic $x = x_0 + a\, t$ with x_0 an integration constant. The solution of the Riemann problem (4.1.16) for the linear advection equation is therefore the upwind solution:

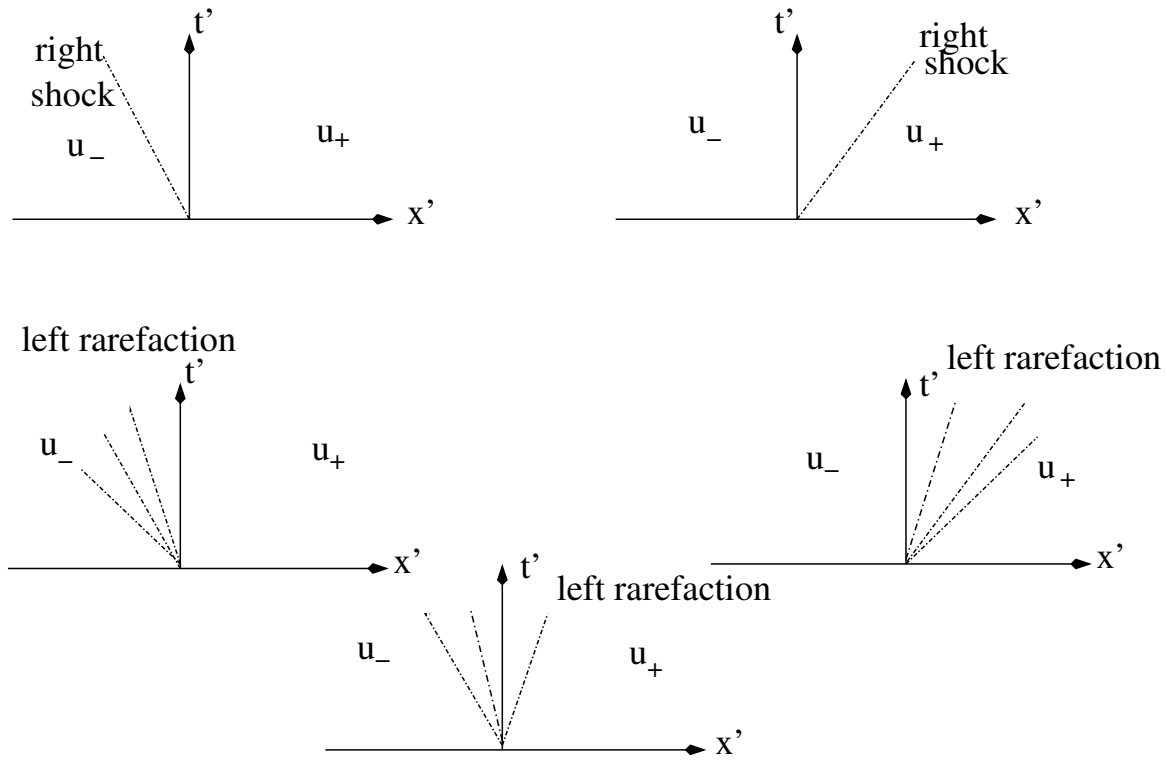$$u_e(x, t') = \begin{cases} u_- & x < x_k + a\, t' \\ u_+ & x \geq x_k + a\, t' \end{cases} \tag{4.1.18}$$

Figure 4.2: The solution to the Riemann problem for Burgers's equation gives either a shock solution for $u_- > u_+$, or a rarefaction wave for $u_- \leq u_+$. The shock wave case divides in a case where the (right) shock moves to the left and one where the shock moves to the right. This depends on the sign of the shock speed s. The (left) rarefaction wave case divides in three cases. The distinction into left and right wave is made for the case where the wave moves into a state of rest with either $u_+ = 0$ (shock case with $u_- > u_+$) or $u_- = 0$ (rarefaction case with $u_- \leq u_+$).

for $t' = t - t_0$. Hence,

$$\tilde{f}_{linear\ advection}(x_k) = \left\{ \begin{array}{ll} a\,u_- & a > 0 \\ a\,u_+ & a \leq 0 \end{array} \right. . \tag{4.1.19}$$

Likewise, the characteristics of Burgers' equation are $dx/dt = u$. Hence, $du/dt = 0$ on these characteristics $x = x_0 + u_0\,t$ with x_0 an integration constant, such that the solution is implicit $u(x,t) = u_0(x - u(x,t))$. For the constant initial data in the Riemann problem, the characteristics are readily solved. We find a shock wave when the characteristics converge for $u_- > u+$, and a rarefaction wave when the characteristics diverge for $u_- \leq u_+$, as a sketch in the x,t–plane reveals. The five different situations around the origin $x' = x - x_k$ are sketched in Fig. 4.2.

The discontinuity of the Burgers' shock is located at $x_b(t)$ and has speed $s = dx_b(t)/dt$, which follows from (the Rankine Hugoniot relation)

$$0 = \lim_{\epsilon \to 0} \int_{x_b(t)-\epsilon}^{x_b(t)+\epsilon} \partial_t u + \partial_x(u^2/2)\, dx = -s\,[u] + [u^2/2] \tag{4.1.20}$$

with $\bar{u} = (U_l + U_r)/2$ and $[u] = U_r - U_l$, and U_l and U_r the values immediately left and right of the shock. Hence, $s = (U_l + U_r)/2$. In the Riemann problem the shock wave thus has shock speed $s = (u_- + u_+)/2$ and its position is given by $x' = s\,t'$. Since the flux is evaluated at $x' = x - x_k$, the flux is either $\tilde{f} = u_-^2/2$ when $s > 0$, or $\tilde{f} = u_+^2/2$ when $s \leq 0$ for the shock wave case.

The rarefaction wave in the Riemann problem has characteristics $dx'/dt' = u$ on which u is constant. The tail and the head of the rarefaction wave lie at $x' = u_-\,t'$ and $x' = u_+\,t'$, respectively. Hence the solution is

$$u(x', t') = \begin{cases} u_- & x' < u_-\,t' \\ x'/t' & u_-\,t' < x' < u_+\,t' \\ u_+ & x' > u_+\,t' \end{cases} . \tag{4.1.21}$$

We deduce from this solution that we must indeed have $u_- \leq u_+$. So at $x' = 0$, we find for the rarefaction wave case that $u(0,t') = u_-$ when $u_- > 0$, $u(0,t') = 0$ when $u_- < 0$ and $u_+ > 0$, and $u(0,t') = u_+$ when $u_+ < 0$.

Hence, the numerical flux for Burgers's equation then becomes

$$\tilde{f}_{burgers}(x_k) = \begin{cases} u_-^2/2 & s > 0 \wedge u_- > u_+ \\ u_+^2/2 & s \leq 0 \wedge u_- > u_+ \\ u_-^2/2 & u_- > 0 \wedge u_- \leq u_+ \\ 0 & u_- < 0 \wedge u_+ > 0 \wedge u_- \leq u_+ \\ u_+^2/2 & u_+ < 0 \wedge u_- \leq u_+ \end{cases} . \tag{4.1.22}$$

We note and leave as a check that both these fluxes are consistent and form E-fluxes, see conditions (i) and (iii). For more complex functions and systems of hyperbolic equations, the exact Riemann problem is often too complicated, and approximate Riemann solvers are used to determine the numerical flux in a computationally efficient way.

Initial and boundary conditions

The expansion coefficients $\hat{U}_m(K_k, t)$ defined in (4.1.12) are obtained by projection of the initial condition onto the basis functions $\psi_{m,k}$. For the piecewise linear case, we obtain the initial values of the mean and slope per element as follows

$$\bar{U}_k(0) = \frac{1}{2} \int_{-1}^{1} u_0\left(x(\zeta)\right) d\zeta \tag{4.1.23}$$

$$\hat{U}_k(0) = \frac{3}{2} \int_{-1}^{1} u_0\left(x(\zeta)\right) \zeta\, d\zeta. \tag{4.1.24}$$

For *periodic boundary conditions*, we take $u_{left} = u_h(x^-_{N_{el}+1})$ and $u_{right} = u_h(x^+_1)$. In case we use a piecewise linear approximation, we thus have

$$u_{left} = \bar{U}_{N_{el}+1} + \hat{U}_{N_{el}+1} \qquad \text{and} \qquad u_{right} = \bar{U}_1 - \hat{U}_1. \qquad (4.1.25)$$

For the *linear advection equation with $a > 0$*, we specify $u_{left}(t)$ and, likewise, $u_{right}(t)$ when $a \leq 0$. For *Burgers' equation with solid walls* at $x = 0, L$, we take $u_{left} = -(\bar{U}_1 - \hat{U}_1)$ and $u_{right} = -(\bar{U}_{N_{el}+1} + \hat{U}_{N_{el}+1})$. For *Burgers' equation and open boundaries*, we need to specify $u_{left}(t)$ when $(\bar{U}_1 - \hat{U}_1) > 0$ and $u_{right}(t)$ when $(\bar{U}_{N_{el}+1} + \hat{U}_{N_{el}+1}) < 0$, as the wind u at the relevant boundary then blows into the domain.

Time discretization and time step

We write (4.1.14) as a system of ordinary differential equations

$$\frac{d\mathbf{U}}{dt} = \mathbf{G}(\mathbf{U}, t) \qquad (4.1.26)$$

with $\mathbf{U} = (\bar{U}, \hat{U})^T$ the state vector of unknown coefficients of the basis functions, and the remaining terms are collected in $\mathbf{G}$ and placed on the right-hand-side. We can then use the second-order

$$\begin{aligned}
\mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t\, \mathbf{G}(\mathbf{U}^n, t^n) \\
\mathbf{U}^{n+1} &= \left[\mathbf{U}^n + \mathbf{U}^{(1)} + \Delta t\, \mathbf{G}(\mathbf{U}^{(1)}, t^n + \Delta t)\right]/2 \quad \text{or}
\end{aligned} \qquad (4.1.27)$$

or the third-order Runge-Kutta method

$$\begin{aligned}
\mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t\, \mathbf{G}(\mathbf{U}^n, t^n) \\
\mathbf{U}^{(2)} &= \left[3\, \mathbf{U}^n + \mathbf{U}^{(1)} + \Delta t\, \mathbf{G}(\mathbf{U}^{(1)}, t^n + \Delta t)\right]/4 \\
\mathbf{U}^{n+1} &= \left[\mathbf{U}^n + 2\, \mathbf{U}^{(2)} + 2\, \Delta t\, \mathbf{G}(\mathbf{U}^{(2)}, t^2 + \Delta t/2)\right]/3
\end{aligned} \qquad (4.1.28)$$

of Shu and Osher (1989), for example, to discretize (4.1.26) in time. The numerical flux defined in (4.1.19) and (4.1.22), which was based on the related Riemann problem, for the linear advection and Burgers' equations is time dependent. In the time stepping method, it is evaluated at $t' = 0$, so the latest (intermediate) value of $u_h(x, t)$ is used to define $u_\pm$.

A rough time step estimate is based on the characteristics of both equations:

$$\Delta t = CFL \min_k(|K_k|)/|a| \qquad \text{or} \qquad \Delta t = CFL \min_k(|K_k|)/\max_k(|\bar{U}^n_k + \hat{U}^n_k|, |\bar{U}^n_k - \hat{U}^n_k|) \qquad (4.1.29)$$

with CFL the Courant-Friedrichs-Lewy number; $CFL \leq 1$. The time step can thus vary over time. Via a linear stability analysis of the linear advection equation a more detailed time step can be calculated or, for the nonlinear Burgers' equation, estimated.

Slope limiter

Artificial numerical oscillations can appear around discontuities in the solution, because the projection onto the finite element basis functions are approximate. To diminish these undesirable oscillations, we apply a slope limiter after every time step. We apply the slope limiter as described in [17]. The idea of the slope limiter is to replace the original polynomial P_0 with a new polynomial P that uses the data of the midpoints of the original element u_m and its neighboring elements u_a and u_b. Two linear polynomials, P_1 and P_2 are constructed connecting the data of the neighboring element to that of the original element (see Fig. 4.3). We obtain:

$$P_1 = \frac{u_m(x - x_a) - u_a(x - x_m)}{x_m - x_a}, \quad P_2 = \frac{u_m(x - x_b) - u_b(x - x_m)}{x_m - x_b}. \tag{4.1.30}$$

Now map the polynomials P_j, $j = 0, 1, 2$ onto the DG-space and solve for $(\hat{u}_0)_j$ and $(\hat{u}_1)_j$:

$$\begin{bmatrix} \int_{K_k} \psi_0\psi_0 \, dK & \int_{K_k} \psi_0\psi_1 \, dK \\ \int_{K_k} \psi_1\psi_0 \, dK & \int_{K_k} \psi_1\psi_1 \, dK \end{bmatrix} \begin{bmatrix} (\hat{u}_0)_j \\ (\hat{u}_1)_j \end{bmatrix} = \begin{bmatrix} \int_{K_k} \psi_0 P_j \, dK \\ \int_{K_k} \psi_1 P_j \, dK \end{bmatrix}.$$

After the polynomial reconstruction is performed, an oscillation indicator is sought to assess the smoothness of P_i. The oscillation indicator for the polynomial P_i, $i = 0, 1, 2$, can be defined as $o_i = \partial P_i / \partial x$. The coefficients of the new solution u in element K_k are constructed as the sum of all the polynomials multiplied by a weight, $\hat{u}_q = \sum_{i=0}^{2} w_i(\hat{u}_q)_i$, $q = 0, 1$ in which the weights are computed as:

$$w_i = \frac{(\epsilon + o_i(P_i))^{-\gamma}}{\sum_{j=0}^{2}(\epsilon + o_i(P_j))^{-\gamma}},$$

where γ is a positive number. Take ϵ small, e.g., $\epsilon = 10^{-12}$. Take $\gamma = 1$. If we want more smoothing, take e.g. $\gamma = 10$.

Pseudo-code

The program outline for the discontinuous Galerkin finite element discretization of the one-dimensional scalar hyperbolic equation (4.1.6) is as follows:

```
read input file/make input: get N_el, CFL, T_end, ...
read mesh file/make mesh: make |K_k|, x_k, Ū_k, Û_k, ..., f̃_k, RHS_k
set initial condition (4.1.23)
while (time < T_end) time loop (to solve (4.1.14a),(4.1.14b)),
.        determine time step
.        do intermediate time steps, e.g. RK3 (4.1.28)
.              calculate flux f̃_k at nodes x_k, see, e.g., (4.1.19) & (4.1.22)
```
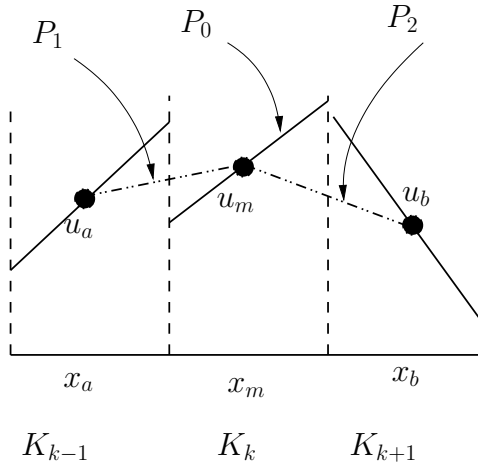
Figure 4.3: Slopelimiter

.   calculate element integrals in (4.1.14b), put in RHS_k
.   solution update
.   do measurements

End.

$$(4.1.31)$$

4.1.2   Advection-diffusion and viscous Burgers' equation

System of first order equations

We consider the linear advection diffusion equation

$$\partial_t u + a\,\partial_x u = \kappa\,\partial_{xx}^2 u \tag{4.1.32}$$

for $a, \kappa \in \mathbb{R}$ and $\kappa > 0$, and the viscous Burgers' equation

$$\partial_t u + u\,\partial_x u = \kappa\,\partial_{xx}^2 u \tag{4.1.33}$$

on a domain $\Omega = [0, L]$ with initial condition $u_0 = u(x, 0)$.

We concisely combine (4.1.32) and (4.1.33) and rewrite it as a system of equations, first order in space,

$$\partial_t u + \partial_x F_u = 0 \qquad \text{and} \qquad \kappa\,q + \partial_x F_q = 0 \tag{4.1.34}$$

with fluxes $F = (F_u, F_q)^T$:

$$F_u = F_u(u,q) = f(u) + \kappa\, q \qquad \text{and} \qquad F_q = F_q(u) = \kappa\, u. \qquad (4.1.35)$$

The flux F_u has a convective part $f(u)$ and a diffusive part $\kappa\, q$. We remember that for the advection diffusion equation we have $f(u) = a\, u$ and for the Burger's equation we have $f(u) = u^2/2$.

We consider periodic boundary conditions. A variety of boundary conditions is considered in the geological application of Bokhove et al. (2005), where a combination of Dirichlet, Neumann, and free boundary conditions are studied for a relevant nonlinear advection-diffusion equation.

Weak formulation

We consider finite-element discretizations of (4.1.34) and (4.1.35) with approximations, $w_h = (u_h, q_h)$, to the state vector, (u,q), and basis functions, $v = (v_u, v_q)$. The discretization is such that u_h, q_h, and (the approximations of) $v_{u,q}$ belong to the broken space (4.1.10).

We multiply (4.1.34) by test functions, $v = \big(v_u(x), v_q(x)\big)$, integrate by parts for each individual and isolated element, and then add the contribution from all elements to obtain the following weak formulation

$$\sum_{k=1}^{N_{\text{el}}} \left\{ \int_{K_k} v_u \frac{du_h}{dt}\, dx + \big(\tilde{F}_u(x_{k+1})\, v_u(x_{k+1}^-) - \tilde{F}_u(x_k)\, v_u(x_k^+)\big) - \int_{K_k} F_u\, \partial_x v_u\, dx \right\} = 0$$

$$\sum_{k=1}^{N_{\text{el}}} \left\{ \int_{K_k} \kappa\, v_q\, q_h\, dx + \big(\tilde{F}_q(x_{k+1})\, v_q(x_{k+1}^-) - \tilde{F}_q(x_k)\, v_q(x_k^+)\big) - \int_{K_k} F_q\, \partial_x v_q\, dx \right\} = 0,$$

$$(4.1.36)$$

where $v_{u,q}(x_{k+1}^-) = \lim_{x\uparrow x_{k+1}} v_{u,q}(x,t)$ and $v_{u,q}(x_k^+) = \lim_{x\downarrow x_k} v_{u,q}(x,t)$. We also replaced the fluxes at the nodes by numerical fluxes $\tilde{F}_u(x_k) = \tilde{F}_u(w_k^-, w_k^+)$ and $\tilde{F}_q(x_k) = \tilde{F}_q(u_k^-, u_k^+)$.

Discretized weak formulation

We approximate u_h and q_h by a mean and slope

$$u_h(x,t) = \bar{U}_k + \hat{U}_k\, \zeta \quad \text{and} \quad q_h(x) = \bar{Q}_k + \hat{Q}_k\, \zeta \qquad (4.1.37)$$

with $\bar{Q}_k = \bar{Q}(K_k, t)$ the mean and $\hat{Q}_k = \hat{Q}_k(K_k, t)$ the slope, and likewise for v_u, v_q. We thus obtain, after substituting (4.1.37) into (4.1.36) and using the arbitrariness of the test

functions, the following system of coupled algebraic and ordinary equations

$$|K_k| \frac{\mathrm{d}\bar{U}_k}{\mathrm{d}t} + \tilde{F}_u(x_{k+1}) - \tilde{F}(x_k) = 0$$

$$\frac{|K_k|}{3} \frac{\mathrm{d}\hat{U}_k}{\mathrm{d}t} + \left(\tilde{F}_u(x_{k+1}) + \tilde{F}_u(x_k)\right) - \int_{-1}^{1} F_u(u_h, q_h)\,\mathrm{d}\zeta = 0$$

$$\kappa\,|K_k|\,\bar{Q}_k + \tilde{F}_q(x_{k+1}) - \tilde{F}_q(x_k) = 0$$

$$\kappa\,\frac{|K_k|}{3}\,\hat{Q}_k + \left(\tilde{F}_q(x_{k+1}) + \tilde{F}_q(x_k)\right) - \int_{-1}^{1} F_q(u_h, q_h)\,\mathrm{d}\zeta = 0.$$

(4.1.38)

The integrals can be approximated with a third-order Gauss quadrature rule, see (4.1.15).

Local discontinuous Galerkin method: numerical flux

The numerical flux, $\tilde{F}_u(w_-, w_+)$, is chosen to (i) be consistent such that $\tilde{F}(w, w) = F(w)$, (ii) be conservative, (iii) ensure a local determination of q_h in terms of u_h, (iv) reduce to an E-flux in the conservative limit when $\kappa = 0$, that is,

$$\int_{u_-}^{u_+} F_u(s, q; \kappa = 0) - \tilde{F}_u(w_-, w_+; \kappa = 0)\,\mathrm{d}s \geq 0$$

with $F_u(u, q; \kappa = 0) = f(u)$, and (v) be L^2-stable, as will be shown. Note that the flux is the only way of communication between elements, and that the flux is determined by the values of u_h and q_h immediately left and right of each face.

The numerical flux chosen is

$$\tilde{F}_u(w_-, w_+) = \tilde{f}(u_-, u_+) + \kappa\,q_+ \qquad \text{and} \qquad \tilde{F}_q = F_q(u_-) = \kappa\,u_-, \qquad (4.1.39)$$

with $\tilde{f}(u_-, u_+)$ the convective flux valid in the inviscid limit $\kappa = 0$, and the diffusive flux is alternating, $\kappa\,q_+$ in $\tilde{F}_u$ versus u_- in $\tilde{F}_q$.

L^2-stability for the discretized equations follows in an analogy of the L^2-stability for the continuous case, and motivates the choice of the numerical flux. To wit, multiply (4.1.34) by u and q, respectively, sum, and integrate over space and time to obtain:

$$\frac{1}{2}\int_0^L u(T)^2 - u_0^2\,dx + \int_0^T\int_0^L \kappa\,q^2\,dx\,dt - \int_0^T\int_0^L (F_u\,\partial_x u + F_q\,\partial_x q)\,dx\,dt +$$

$$\int_0^T (u\,F_u + q\,F_q)_{x=L} - (u\,F_u + q\,F_q)_{x=0}\,dt = 0 \iff$$

$$\frac{1}{2}\int_0^L u(T)^2 - u_0^2\,dx + \int_0^T\int_0^L \kappa\,q^2\,dx\,dt + \int_0^T \left(u\,F_u - \phi(u)\right)|_{x=L} - \left(u\,F_u - \phi(u)\right)|_{x=0}\,dt = 0,$$

(4.1.40)

since

$$F_u \, \partial_x u + F_q \, \partial_x q = f(u) \, \partial_x u + \kappa \, q \, \partial_x u + \kappa \, u \, \partial_x q = \partial_x (\phi(u) + q \, F_q) \qquad (4.1.41)$$

with $\phi(u) = \int_0^u f(s) \, ds$.

The discrete version of L^2-stability for the numerical spatial discretization proceeds as follows (*cf.*, Cockburn and Shu, 1998). We add the weak formulation (4.1.36) of both equations and integrate in time to find

$$\mathcal{B}_h(w_h, v_h) = \int_0^T \int_0^L \frac{du_h}{dt} \, v_u + \kappa \, v_q \, q_h \, dx \, \mathrm{d}t - \int_0^T \sum_{2 \leq k \leq N_{\mathrm{el}}} (\tilde{F} \cdot [v_h])_k \, dt +$$

$$\int_0^T (\tilde{F} \cdot v_h^-)_{k=N_{\mathrm{el}}+1} - (\tilde{F} \cdot v_h^+)_{k=1} \, dt - \int_0^T \sum_{1 \leq k \leq N_{\mathrm{el}}} \int_{K_k} F \cdot \partial_x v_h dx \, dt \qquad (4.1.42)$$

with $w_h = (u_h, q_h)$, $v_h = (v_u, v_q)$, and $[v] = v_+ - v_-$. As in the continuous case, substitute $v_u = u_h$ and $v_q = q_h$ into (4.1.42) to obtain:

$$\mathcal{B}_h(w_h, w_h) = \frac{1}{2} \int_0^L (u_h(T)^2 - u_h(x,0)^2) \, dx + \int_0^T \int_0^L \kappa \, q_h^2 \, \mathrm{d}x \, \mathrm{d}t +$$

$$\int_0^T \left(\phi(u_h^+) + F_q^+ \, q_h^+ - \tilde{F} \cdot w_h^+ \right)_{k=1} \, dt - \qquad (4.1.43)$$

$$\int_0^T \left(\phi(u_h^-) + F_q^- \, q_h^- - \tilde{F} \cdot w_h^- \right)_{k=N_{\mathrm{el}}+1} \, dt + \int_0^T \Theta_{\mathrm{dissipation}}(t) \, \mathrm{d}t,$$

where

$$\Theta_{\mathrm{dissipation}}(t) = - \sum_{2 \leq k \leq N_{\mathrm{el}}} \tilde{F} \cdot [w_h] - \sum_{1 \leq k \leq N_{el}} \int_{K_k} F \cdot \partial_x w_h \, \mathrm{d}x +$$

$$\left(\phi(u_h) + F_q \, q_h \right)_{k=N_{\mathrm{el}}+1}^- - \left(\phi(u_h) + F_q \, q_h \right)_{k=1}^+. \qquad (4.1.44)$$

Rewriting

$$- \sum_{1 \leq k \leq N_{\mathrm{el}}} \int_{K_k} F \cdot \partial_x w_h \, dx = \sum_{2 \leq k \leq N_{\mathrm{]el}}} [\phi(u_h) + F_q \, q_h]_k +$$

$$\left(\phi(u_h) + F_q \, q_h \right)_{k=1}^+ - \left(\phi(u_h) + F_q \, q_h \right)_{k=N_{\mathrm{el}}+1}^- \qquad (4.1.45)$$

is used to evaluate (4.1.44) further. Hence, requiring that

$$\Theta_{\mathrm{dissipation}}(t) = \sum_{2 \leq k \leq N_{el}} \Theta_{\mathrm{dissipation}}^k(t) = \sum_{2 \leq k \leq N_{el}} \left\{ [\phi(u_h) + F_q \, q_h] - \tilde{F} \cdot [w_h] \right\}_k$$

$$= \sum_{2 \leq k \leq N_{el}} \left\{ [\phi(u_h)] + [F_q] \, \bar{q}_h - [u_h] \, \tilde{F}_u + \bar{F}_q \, [q_h] - [q_h] \, \tilde{F}_q \right\}_k > 0 \qquad (4.1.46)$$

motivates the choice (4.1.39), where we used that

$$[F_q \, q_h] = [F_q] \, \bar{q}_h + \bar{F}_q \, [q_h]. \tag{4.1.47}$$

Reordering the chosen convective flux (4.1.39) in the limit $\kappa = 0$ produces

$$\lim_{\kappa \to 0} \Theta^k_{\text{dissipation}}(t) = [\phi(u_h)] - [u_h] \, \tilde{f}(u_-, u_+) = \int_{u_-}^{u_+} F_u(s, q; \kappa = 0) - \tilde{f}(u_-, u_+) \, ds \geq 0 \tag{4.1.48}$$

for $F_u(u, q; \kappa = 0) = f(u)$, which is satisfied provided that the E-flux property (iv) holds. In the diffusive limit, the cancellation is as follows

$$\begin{aligned}
&[F_q] \, \bar{q}_h - [u_h] \, \kappa \, q_+ + \bar{F}_q \, [q_h] - [q_h] \, \tilde{F}_q = \\
&\frac{1}{2} \kappa \left((u_+ - u_-) (q_+ + q_-) - 2 (u_+ - u_-) q_+ \right. \\
&\left. + (u_+ + u_-) (q_+ - q_-) - 2 (q_+ - q_-) u_- \right) = 0.
\end{aligned}$$

Hence, $\Theta^k_{\text{dissipation}} \geq 0$ and thus $\Theta_{\text{dissipation}} \geq 0$. This proves L^2-stability, since the first two quadratic terms in (4.1.42) are now always bounded or even decreasing if the boundary terms cancel. The boundary terms in (4.1.42) cancel for periodic boundary conditions, for example.

Time discretization

We write (4.1.38) as a system of ordinary differential and algebraic equations

$$\frac{d\mathbf{U}}{dt} = \mathbf{G}_U(\mathbf{U}, \mathbf{q}) \qquad \text{and} \qquad \mathbf{q} = \mathbf{G}_q(\mathbf{U}) \tag{4.1.49}$$

with $\mathbf{U} = (\bar{U}, \hat{U})^T$ the state vector of unknown coefficients of the basis functions and $\mathbf{q} = (\bar{Q}, \hat{Q})^T$, and $\mathbf{G}_{U,q}$ the remaining terms placed on the right-hand-side. Using the third-order Runge-Kutta method to discretize (4.1.49) in time, we obtain

$$\begin{aligned}
\mathbf{q}^n &= \mathbf{G}_q(\mathbf{U}^n), & \mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t \, \mathbf{G}_U(\mathbf{U}^n, \mathbf{q}^n) \\
\mathbf{q}^{(1)} &= \mathbf{G}_q(\mathbf{U}^{(1)}), & \mathbf{U}^{(2)} &= \left(3 \, \mathbf{U}^n + \mathbf{U}^{(1)} + \Delta t \, \mathbf{G}_U(\mathbf{U}^{(1)}, \mathbf{q}^{(1)}) \right)/4 \\
\mathbf{q}^{(2)} &= \mathbf{G}_q(\mathbf{U}^{(2)}), & \mathbf{U}^{n+1} &= \left(\mathbf{U}^n + 2 \, \mathbf{U}^{(2)} + 2 \, \Delta t \, \mathbf{G}_U(\mathbf{U}^{(2)}, \mathbf{q}^{(2)}) \right)/3.
\end{aligned} \tag{4.1.50}$$

Note that we can solve for $\mathbf{U}$ and $\mathbf{q}$ in an explicit manner because the new (intermediate) stage of $\mathbf{q}$ can always be found from the new (intermediate) stage of $\mathbf{U}$ before commencing the time update.

4.2 One-dimensional hyperbolic systems

4.2.1 System of equations

We will consider the space discontinuous Galerkin finite element discretization of the following hyperbolic system of conservation laws with extra geometrical, source or sink terms

$$\partial_t U + \partial_x F(U) = S, \tag{4.2.1}$$

where U is the $n_q \times 1$ state vector of variables, $F = F(U)$ the flux vector and S the "source" vector. The flux F is such that $A(u) = \partial F / \partial U$ has real eigenvalues. The system is at least hyperbolic when $S = 0$.

To particularize the numerical flux, we need to specify a particular system. We have chosen as example the equations modelling the cross-sectionally averaged flow of shallow water through a channel with a contraction. The variables involved are the velocity $u = u(x,t)$ and the cross-section $A = A(x,t) = b(x)\,h(x,t)$, and the averaged depth across the channel $h(x,t)$ as function of the coordinate x along the channel and time. The width of the channel $b = b(x)$ is given. This dimensionless system in its pseudo-conservative form reads

$$\partial_t A + \partial_x (A\,u) = 0 \tag{4.2.2a}$$

$$\partial_t(A\,u) + \partial_x\left(A\,u^2 + \frac{A^2}{2\,b\,F_0^2}\right) = \frac{A^2}{2\,b^2\,F_0^2}\,\partial_x b - C_d\,b\,A\,u\,|A\,u|/A^2, \tag{4.2.2b}$$

where $F_0 = u_0/\sqrt{g\,h_0}$ is the Froude number and $C_d = C_d^*\,b_0/h_0$ the friction coefficient. This Froude number F_0 is based on the upstream inflow values of velocity, u_0, and depth, h_0. The dimensional friction $C_d^* \approx 0.004$ stems from the engineering practise (Akers, 2005). We have used the upstream width b_0, velocity and depth u_0 and h_0 to scale the system.

To relate (4.2.2) to (4.2.1) we identify

$$U = (A, m = A\,u)^T, \quad F = \left(A\,u, A\,u^2 + \frac{A^2}{2\,b\,F_0^2}\right)^T \tag{4.2.3}$$

$$S = S_g + S_0 \tag{4.2.4}$$

$$S_g = \left(0, \frac{A^2}{2\,b^2\,F_0^2}\,\partial_x b\right)^T \tag{4.2.5}$$

$$S_0 = \left(0, -C_d\,b\,A\,u\,|A\,u|/A^2\right)^T. \tag{4.2.6}$$

Weak formulation

We choose a test function w_h which is defined and continuous on each element K_k, but generally discontinuous across an element boundary. The weak formulation follows by

multiplication of (4.2.1) with test function w_h, integration (by parts) over each element and summation over all elements

$$\sum_{K_k} \int_{K_k} w_h\, \partial_t U - F(U)\, \partial_x w_h - S\, w_h \mathrm{d}x + F\left(U(x_{k+1}^-)\right) w_h(x_{k+1}^-) - F\left(U(x_k^+)\right) w_h(x_k^+) = 0$$

(4.2.7)

with $x_k^+ = \lim x \downarrow x_k$ and $x_{k+1}^- = \lim x \uparrow x_{k+1}$.

Numerical flux

We rework the summation of the fluxes over elements in (4.2.7) to one over the nodes

$$\sum_{K_k} \int_{K_k} F\left(U(x_{k+1}^-)\right) w_h(x_{k+1}^-) - F\left(U(x_k^+)\right) w_h(x_k^+) =$$

$$- F\left(U(x_1^+)\right) w_h(x_1^+) + F\left(U(x_{N_{el}+1}^-)\right) w_h(x_{N_{el}+1}^-) +$$

$$\sum_{j=2}^{N_{el}} F\left(U(x_j^-)\right) w_h(x_j^-) - F\left(U(x_j^+)\right) w_h(x_j^+) \equiv$$

$$- F\left(U(x_1^+)\right) w_h(x_1^+) + F\left(U(x_{N_{el}+1}^-)\right) w_h(x_{N_{el}+1}^-) + \sum_{j=2}^{N_{el}} F_r\, w_r - F_l\, w_l \quad (4.2.8)$$

with $F_r = F\left(U(x_j^+)\right), F_l = F\left(U(x_j^-)\right), w_{r,l}$ denoting evaluation right or left of node x_j. The last expression we further reorder as follows

$$F_r\, w_r - F_l\, w_l = (w_r - w_l)\, (\alpha\, F_l + \gamma\, F_r) + (\alpha\, w_r + \gamma\, w_l)\, (F_r - F_l)$$
$$= (w_r - w_l)\, (\alpha\, F_l + \gamma\, F_r),$$

(4.2.9)

by imposing flux conservation $F_r = F_l$ of the continuous $F(U)$, and with $\alpha + \gamma = 1$ and $\alpha, \gamma \geq 0$. Flux conservation is thus included explicitly.

We use the HLL or HLLC numerical flux (see Batten, 1997; and Toro, 1999) defined as follows

$$\tilde{F} = \frac{1}{2}\left(F_r + F_l - (|S_l| - |S_m|)\, U_l^* + (|S_r| - |S_m|)\, U_r^* + |S_l|\, U_l - |S_r|\, U_r\right), \quad (4.2.10)$$

$$(S_{l,r} - S_m)\, U_{l,r}^* = (S_{l,r} - u_{l,r})\, U_{l,r} + \begin{pmatrix} 0 \\ P^* - P_{l,r} \end{pmatrix} \quad (4.2.11)$$

$$P^* = P_{l,r} + A_{l,r}\, (S_{l,r} - u_{l,r})\, (S_m - u_{l,r}), \quad (4.2.12)$$

$$S_l = \min(u_l - a_l, u_r - a_r), \quad S_r = \max(u_l + a_l, u_r + a_r), \quad a_{l,r} = \sqrt{h_{l,r}}/F_0, \quad (4.2.13)$$

where $a_{l,r}$ are the gravity wave speeds to the left and rightof a node, and $S_{l,r}$ are the estimate of the fastest left and right going waves.

In the discretization, we now approximate $U \approx U_h$ by expansion of U in terms of basis functions, continuous on each element, and replace $\alpha\, F_l + \gamma\, F_r$ by a numerical flux $\tilde{F}(U_l, U_r)$. From (4.2.7), we then obtain

$$\sum_{K_k} \int_{K_k} w_h\, \partial_t U_h - F(U_h)\, \partial_x w_h - S(U_h)\, w_h \mathrm{d}x + \tilde{F}\left(U_h(x_{k+1}^-), U_h(x_{k+1}^-)\right)\, w_h(x_{k+1}^-)$$

$$-\tilde{F}\left(U_h(x_k^-), U_h(x_k^+)\right)\, w_h(x_k^+) = 0. \tag{4.2.14}$$

Discretized weak formulation

When we expand U_h and w_h, cf. (4.1.12), into first order polynomials we get

$$U_h = \bar{U}_k + \zeta\, \hat{U}_k \qquad \text{and} \qquad w_h = \bar{W}_k + \zeta\, \hat{W}_k \tag{4.2.15}$$

with mean and slope coefficients $\bar{U}_k(t)$ and $\hat{U}_k(t)$. Since w_h is arbitrary, we can only take $\bar{W}_k = 1$ and $\hat{W}_k = 1$ alternately for each element, and obtain the spatial discretization

$$|K_k|\, \frac{\mathrm{d}\bar{U}_k}{\mathrm{d}t} + \tilde{F}\left(U_h(x_{k+1}^-), U_h(x_{k+1}^-)\right) - \tilde{F}\left(U_h(x_k^-), U_h(x_k^+)\right) =$$

$$\int_{-1}^{1} \frac{A^2}{2\, b^2\, F_0^2} \begin{pmatrix} 0 \\ \hat{b}_k \end{pmatrix} \mathrm{d}\zeta + \frac{|K_k|}{2} \int_{-1}^{1} \begin{pmatrix} 0 \\ -C_d\, b\, A\, u\, |A\, u|/A^2 \end{pmatrix} \mathrm{d}\zeta \tag{4.2.16}$$

$$\frac{|K_k|}{3}\, \frac{\mathrm{d}\hat{U}_k}{\mathrm{d}t} + \tilde{F}\left(U_h(x_{k+1}^-), U_h(x_{k+1}^-)\right) + \tilde{F}\left(U_h(x_k^-), U_h(x_k^+)\right) - \int_{-1}^{1} F(U_h)\, \mathrm{d}\zeta =$$

$$\int_{-1}^{1} \frac{A^2\, \zeta}{2\, b^2\, F_0^2} \begin{pmatrix} 0 \\ \hat{b}_k \end{pmatrix} \mathrm{d}\zeta + \frac{|K_k|}{2} \int_{-1}^{1} \zeta \begin{pmatrix} 0 \\ -C_d\, b\, A\, u\, |A\, u|/A^2 \end{pmatrix} \mathrm{d}\zeta. \tag{4.2.17}$$

Time discretization

Abstractly, we write the spatially discrete system (4.2.16)–(4.2.17) as

$$\frac{d\bar{U}}{dt} = \bar{R}_d(U) \qquad \text{and} \qquad \frac{d\hat{U}}{dt} = R_d(U). \tag{4.2.18}$$

Two choices for the time discretization are a second order predictor-corrector Crank-Nicolson scheme, and the third-order Runge-Kutta scheme of Shu and Osher (1989). The

predictor-corrector scheme is

$$\bar{U}^* = \bar{U}^n + \Delta t \, \bar{R}_d(U^n),$$
$$\hat{U}^* = \hat{U}^n + \Delta t \, R_d(U^n)$$
$$\bar{U}^{n+1} = \frac{1}{2} \left(\bar{U}^n + \bar{U}^* + \Delta t \, \bar{R}_d(U^n) + \Delta t \, \bar{R}_d(U^*) \right),$$
$$\hat{U}^{n+1} = \frac{1}{2} \left(\hat{U}^n + \hat{U}^* + \Delta t \, R_d(U^n) + \Delta t \, R_d(U^*) \right). \tag{4.2.19}$$

In the proper Crank-Nicolson scheme the state U^* is replaced by U^{n+1} except the last U^* in the last equation, which is replaced by U^n. The RK3 scheme reads

$$\bar{U}^{(1)} = \bar{U}^n + \Delta t \, \bar{R}_d(U^n),$$
$$\hat{U}^{(1)} = \hat{U}^n + \Delta t \, R_d(U^n)$$
$$\bar{U}^{(2)} = \frac{1}{4} \left(3 \, \bar{U}^n + \bar{U}^{(1)} + \Delta t \, \bar{R}_d(U^{(1)}) \right),$$
$$\hat{U}^{(2)} = \frac{1}{4} \left(3 \, \hat{U}^n + \hat{U}^{(1)} + \Delta t \, R_d(U^{(1)}) \right)$$
$$\bar{U}^{n+1} = \frac{1}{3} \left(\bar{U}^n + 2 \, \bar{U}^{(2)} + 2 \, \Delta t \, \bar{R}_d(U^{(2)}) \right),$$
$$\hat{U}^{n+1} = \frac{1}{3} \left(\hat{U}^n + 2 \, \hat{U}^{(2)} + 2 \, \Delta t \, R_d(U^{(2)}) \right). \tag{4.2.20}$$

4.2.2   Flow at rest

An important test specific to the shallow water equations (4.2.2) is whether rest flow with $u = 0$ and $h = H$ constant remains at rest. One can show that for rest flow with constant h so that $h_l = h_r$ and continuous $b(x)$ so that $b_l = b_r$ the numerical flux becomes $\tilde{F} = \left(0, b \, h^2/(2 \, F_0^2) \right)^T$. If we project the discontinuous Galerkin expansion coefficients of $b(x)$ such that b_h remainscontinuous at the nodes and approximate $\partial_x b$ in terms of this expansion, then rest flow remains at rest in the numerical discetization. This is the reason why the extra S–term has been divided into a gradient, S_g, and non-gradient, S_0, part in (4.2.3).

4.3   Exercises

1. The goal of this exercise is to make a numerical implementation and verification of the space discontinuous finite element method for the scalar hyperbolic system, and for the linear advection and inviscid Burgers' equations considered in section 4.1.1 in particular.

   A well-known strategy is to test the code first for mean values only, that is, for a leading order finite volume discretization (4.1.14a) before including the higher order expansions in (4.1.12).

(a) Write down two exact solutions for the linear advection equation: for the Riemann problem and for a general initial condition in a periodic domain.

(b) Write down exact solutions for Burgers' equation, one for the Riemann problem and one for general initial conditions using the method of characteristics. Predict when wave breaking occurs for an initially smooth profile $u_0(x)$.

(c) Implement the algorithm, e.g. using Matlab. Use the pseudo code (4.1.31), if necessary. For example, for Burger's equation $\partial_t u + u\,\partial_x u = 0$ we can use initial data $u_0(x) = \sin(\frac{\pi}{6}x)$ for a periodic domain $0 < x \leq 12$. For Burger's equation $\partial_t u + u\,\partial_x u = 0$, we can alternatively use Riemann initial data $u_0(x) = 1$ for $x < 1$ and $u_0(x) = 0$ for $x \geq 1$ (and vice versa) for a domain $0 < x \leq 12$ with inflow and outflow boundaries.

(d) Take $f(u) = a\,u$ and verify the implementation against exact solutions of the linear advection equation. Show that the method is second order acccurate in space for smooth and nonsmooth solutions by making convergence tables for the L^1, L^2 and L^∞ errors. These errors are defined as follows

$$L^p(t) = \left(\int_0^L |u_h(x,t) - u_{exact}(x,t)|^p dx \right)^{1/p} \tag{4.3.1}$$
$$\|u_h(x,t) - u_{exact}(x,t)\|_{L^\infty} = \operatorname{ess\,sup}\{|u_h(x,t) - u_{exact}(x,t)| : x \in [0,L]\}.$$

(e) Take $f(u) = u^2/2$ and verify the implementation against exact solutions of the Burgers' equation. Display the accuracy of the method for smooth solutions and for solutions with discontinuities. Confirm by simulation that the time of shock formation in Burgers' equation coincides with the analytical prediction.

(f) Implement the slope limiter discussed in (4.1.30). Check the accuracy of the method.

(Bonus) Take $f(u) = au + bu^2$. Derive a numerical flux, implement this numerical flux and compare your numerical results with the exact solution of a Riemann problem.

2. Show that instead of the flux (4.1.39), the following numerical flux

$$\tilde{F}_u(w_-, w_+) = \tilde{f}(u_-, u_+) + \kappa\,q_- \qquad \text{and} \qquad \tilde{F}_q = F_q(u_+) = \kappa\,u_+ \tag{4.3.2}$$

also satisfies L^2-stability. Does the flux

$$\tilde{F}_u(w_-, w_+) = \tilde{f}(u_-, u_+) + \kappa\,\bar{q} \qquad \text{and} \qquad \tilde{F}_q = F_q(\bar{u}) = \kappa\,\bar{u} \tag{4.3.3}$$

satisfy L^2-stability?

Chapter 5

Space-time discontinuous Galerkin finite element methods

5.1 Space-time methods for scalar conservation laws

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 and the interior mesh points have to move to maintain a consistent mesh without grid folding. This mesh movement imposes additional complications for the numerical discretization relative to the case without mesh movement. In particular, ensuring that the numerical discretization is conservative on time-dependent meshes is non-trivial. This is important for many reasons. In the first place the equations of fluid dynamics express conservation of mass, momentum and energy, which should preferrably also be satisfied at the discrete level, but also problems with discontinuities require a conservative scheme since otherwise the shock speed will be inaccurate.

A natural way to derive numerical discretizations for problems which require deforming and moving meshes is to use the space-time approach. In this technique time is considered as an extra dimension and treated in the same way as space. In this chapter we will extend the discontinuous Galerkin finite element discretization discussed chapter 4 to a space-time discretization. We will only discuss the main aspects of this space-time DG method using a scalar hyperbolic partial differential equation in one space dimension as model problem. The extension to parabolic and incompletely parabolic partial differential equations and systems are beyond the scope of these notes. More details on space-time DG methods, including applications to inviscid compressible flow described by the Euler equations of gas dynamics, can be found in Van der Vegt and Van der Ven [23], [24]. Extensions of the space-time DG method to the advection-diffusion equation are given in Sudirham, Van der

Vegt and Van Damme [21] and for the compressible Navier-Stokes equations in Klaij, Van der Vegt and Van der Ven [13].

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

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x_1} = 0, \qquad x_1 \in \Omega(t),\ t \in (t_0, T), \tag{5.1.1}$$

with boundary conditions:

$$u(t, x_1) = \mathcal{B}(u, u_w), \qquad x_1 \in \partial\Omega(t),\ t \in (t_0, T), \tag{5.1.2}$$

and initial condition:

$$u(0, x_1) = u_0(x_1), \qquad x_1 \in \Omega(t_0). \tag{5.1.3}$$

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 5.1.

If we would directly discretize (5.1.1)-(5.1.3) 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 [15].

An alternative approach is provided by the space-time discretization method. In a space-time discretization we directly consider the domain in $\mathbb{R}^2$. A point $x \in \mathbb{R}^2$ has coordinates (x_0, x_1), with $x_0 = t$ representing time and $x \in \mathbb{R}$ the spatial coordinate. We define the space-time domain as the open domain $\mathcal{E} \subset \mathbb{R}^2$. The boundary $\partial\mathcal{E}$ of the space-time domain $\mathcal{E}$ consists of the hypersurfaces $\Omega(t_0) := \{x \in \partial\mathcal{E} \,|\, x_0 = t_0\}$, $\Omega(T) := \{x \in \partial\mathcal{E} \,|\, x_0 = T\}$, 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 (5.1.1) can now be reformulated in the space-time framework. For this purpose we consider the space-time domain $\mathcal{E}$ in $\mathbb{R}^2$, see Figure 5.2. The space-time formulation of the scalar conservation law (5.1.1) is obtained by introducing the space-time flux vector:

$$\mathcal{F}(u) := (u, f(u))^T.$$
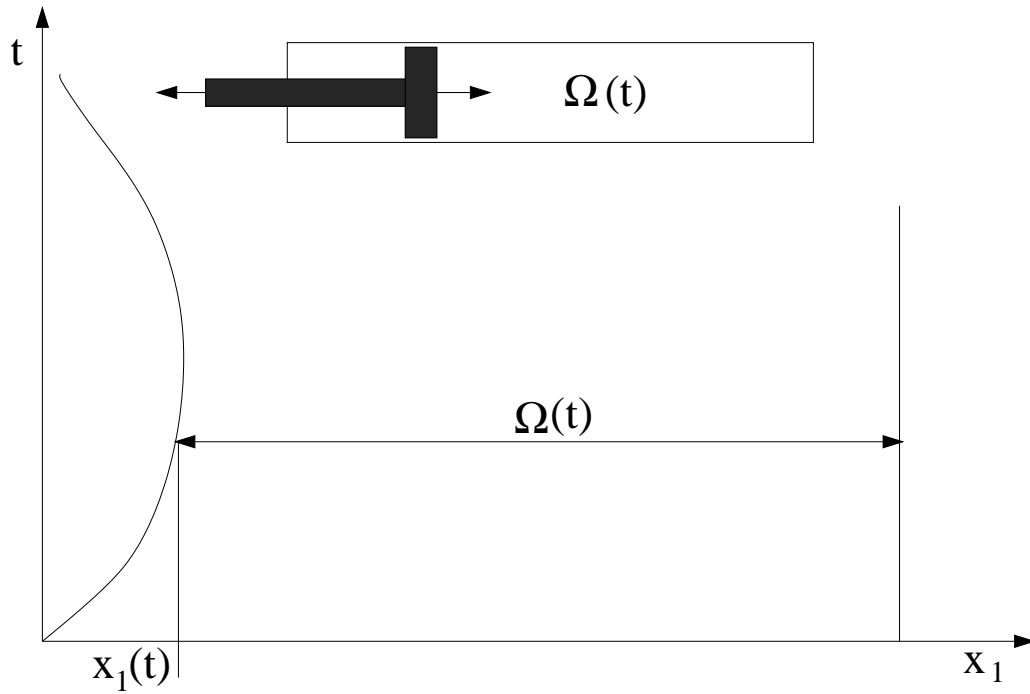
Figure 5.1: Example of a time dependent flow domain $\Omega(t)$.

The scalar conservation law then can be written as:

$$\text{div } \mathcal{F}(u(x)) = 0, \qquad x \in \mathcal{E}, \tag{5.1.4}$$

with boundary conditions:

$$u(x) = \mathcal{B}(u, u_w), \qquad x \in \mathcal{Q}, \tag{5.1.5}$$

and initial condition:

$$u(x) = u_0(x), \qquad x \in \Omega(t_0). \tag{5.1.6}$$

Here, the div operator is defined as $\text{div } \mathcal{F} = \frac{\partial \mathcal{F}_i}{\partial x_i}$.

The space-time formulation requires the introduction of a space-time slab, element 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
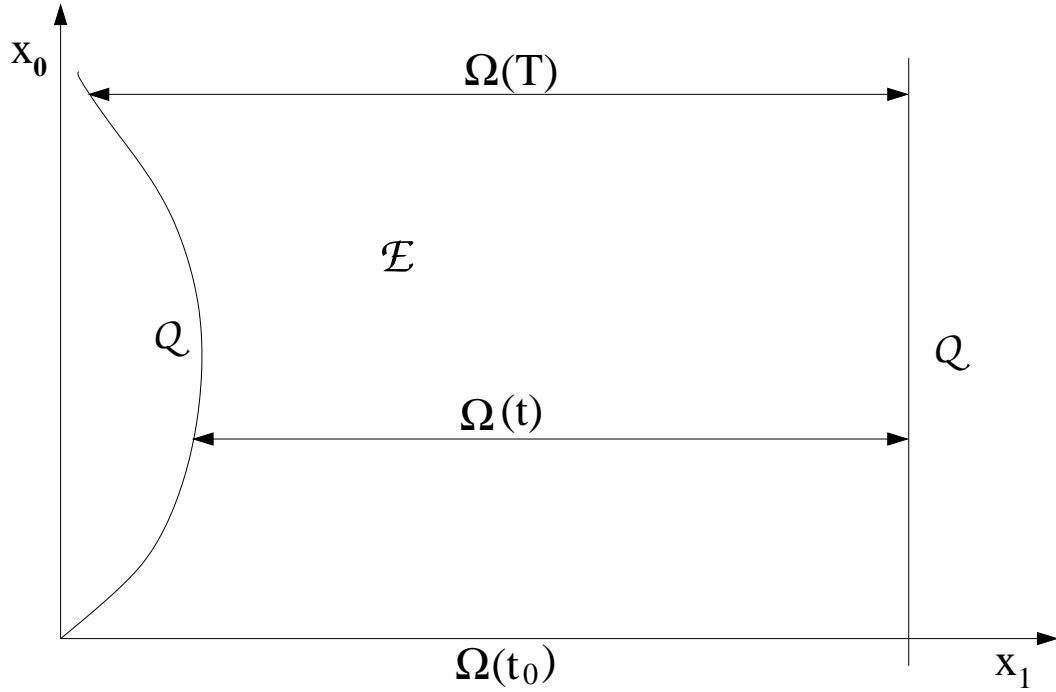
Figure 5.2: Example of a space-time domain $\mathcal{E}$.

domain at time $t = t_n$. A space-time slab is then defined as the domain $\mathcal{E}^n = \mathcal{E} \cap (I_n \times \mathbb{R})$, with boundaries $\Omega(t_n)$, $\Omega(t_{n+1})$ and $\mathcal{Q}^n = \partial\mathcal{E}^n \backslash (\Omega(t_n) \cup \Omega(t_{n+1}))$.

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 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 5.3. The element boundary of the space-time slab is denoted as $\partial\mathcal{K}$ and consists of three parts K^n, K^{n+1}, and $\mathcal{Q}_{\mathcal{K}}^n = \partial\mathcal{K} \backslash (K^n \cup K^{n+1})$.

The geometry of the space-time element can be defined by introducing the mapping G_K^n. This mapping connects the space-time element $\mathcal{K}^n$ to the reference element $\hat{\mathcal{K}} = [-1, 1]^2$ and is defined using the following steps. First, we define a smooth, orientation preserving and invertible mapping Φ_t^n in the interval I_n as:

$$\Phi_t^n : \Omega(t_n) \rightarrow \Omega(t) : x_1 \mapsto \Phi_t^n(x_1), \quad t \in I_n.$$

Next, we split $\Omega(t_n)$ into the tessellation $\bar{\mathcal{T}}_h^n$ with non-overlapping elements K_j. The ele-
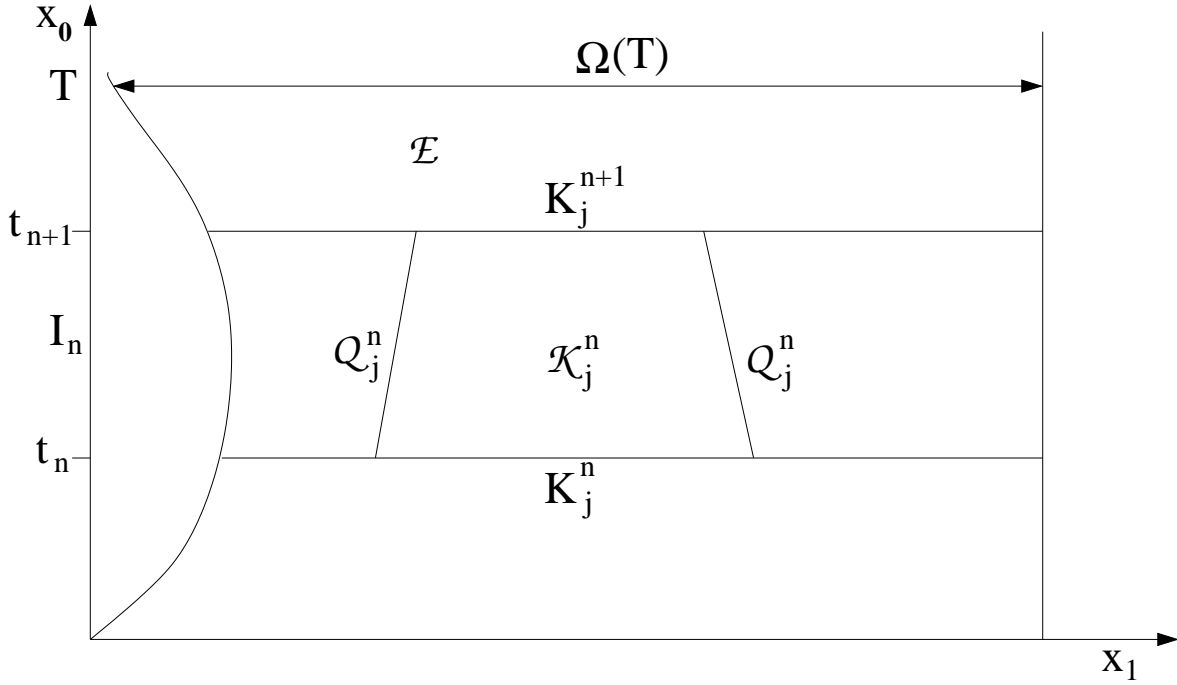
Figure 5.3: Space-time slab in space-time domain $\mathcal{E}$.

ments K_j are defined using the mapping F_K^n:

$$F_K^n : [-1, 1] \to K^n : \xi_1 \longmapsto \sum_{i=1}^{2} x_i(K^n)\chi_i(\xi_1),$$

with $x_i(K^n)$ the spatial coordinates of the space-time element at time $t = t_n$ and χ_i the standard linear finite element shape functions defined on the interval $[-1, 1]$ as:

$$\chi_1(\xi_1) = \tfrac{1}{2}(1 - \xi_1),$$
$$\chi_2(\xi_1) = \tfrac{1}{2}(1 + \xi_1).$$

Similarly, we use the mapping F_K^{n+1} to define the element K^{n+1}:

$$F_K^{n+1} : [-1, 1] \to K^{n+1} : \xi_1 \longmapsto \sum_{i=1}^{2} \Phi_{t_{n+1}}^n(x_i(K^n))\chi_i(\xi_1).$$

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

$$G_K^n : [-1, 1]^2 \to \mathcal{K}^n : (\xi_0, \xi_1) \longmapsto (x_0, x_1), \tag{5.1.7}$$
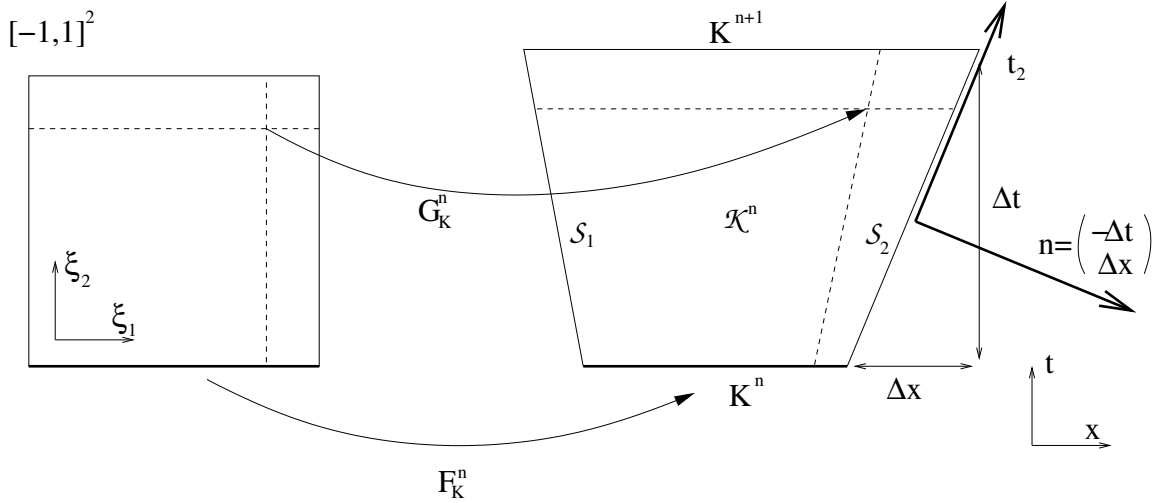
Figure 5.4: Geometry of 2D space-time element in both computational and physical space.

with:

$$(x_0, x_1) = \left(\tfrac{1}{2}(t_n + t_{n+1}) - \tfrac{1}{2}(t_n - t_{n+1})\xi_0, \right.$$

$$\left. \tfrac{1}{2}(1 - \xi_0)F_K^n(\xi_1) + \tfrac{1}{2}(1 + \xi_0)F_K^{n+1}(\xi_1) \right).$$

An overview of the different mappings is given in Figure 5.4. The space-time tessellation is now defined as:

$$\mathcal{T}_h^n := \{\mathcal{K} = G_k^n(\hat{\mathcal{K}}) \,|\, K \in \bar{\mathcal{T}}_h^n\}.$$

5.2   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 (5.1.1). The first step is the definition of the basis functions $\hat{\phi}_m, (m = 0, 1, 2)$ on the master element $\hat{\mathcal{K}} = [-1, 1]^2$, for which we use linear basis functions, both in space and time:

$$\hat{\phi}_m(\xi_0, \xi_1) = 1, \qquad\qquad m = 0,$$
$$= \xi_{m-1}, \qquad m = 1, 2$$

The basis functions ϕ_m on the space-time element $\mathcal{K}$ then can 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 and we do not require any continuity across element faces, both in space and time. This is the main difference with the discontinuous Galerkin method discussed in chapter 4, where only discontinuous basis functions in space are used. An impression of the basis functions is given in Figure 5.5. 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 $\psi_m : \mathcal{K} \to \mathbb{R}$:

$$\psi(x_0, x_1) = 1, \qquad\qquad\qquad\qquad m = 0,$$
$$= \phi_m(x_0, x_1) - \frac{1}{|K_j(t_{n+1})|} \int_{K_j(t_{n+1})} dK, \quad m = 1, 2.$$

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

$$V_h^1(\mathcal{T}_h^n) := \left\{ v_h \,\middle|\, v_h|_{\mathcal{K}_j^n} \in P^1(\mathcal{K}_j^n), \, j = 1, \cdots, N_n \right\},$$

with $P^1(\mathcal{K}) = \text{span}\{\psi_m, m = 0, 1, 2\}$. The trial functions $u_h : \mathcal{T}_h^n \to \mathbb{R}^2$ are defined in each element $\mathcal{K}_j \in \mathcal{T}_h^n$ as:

$$u_h(x) = \mathcal{P}(u(x)|_\mathcal{K}) = \sum_{m=0}^{2} \hat{u}_m(\mathcal{K}_j)\psi_m(x), \quad x \in \mathcal{K}_j, \tag{5.2.1}$$

with $\mathcal{P}$ the projection operator to the finite element space $V_h^1(\mathcal{T}_h^n)$ and $\hat{u}_m$ 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 $m = 1, 2$, we have the relation:

$$\bar{u}_h(K_j(t_{n+1})) := \frac{1}{|K(t_{n+1})|} \int_{K(t_{n+1})} dK = \hat{u}_0,$$

and we can write:

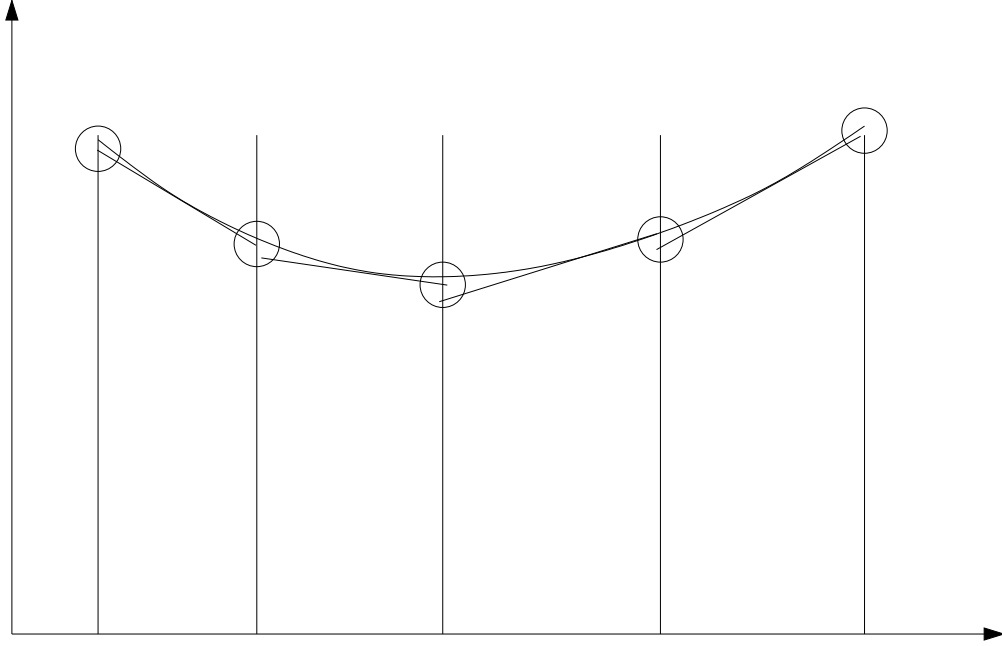$$u_h(x) = \bar{u}_h(K_j(t_{n+1})) + \tilde{u}_h(x),$$

Figure 5.5: Discontinous Galerkin approximation of a function.

with $\bar{u}_h(K_j(t_{n+1}))$ 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 (5.1.4) is multiplied with arbitrary test functions $w_h \in V_h^1$ and integrated over the space-time domain $\mathcal{E}$, split into space-time elements. After introducing the trial functions $u_h \in V_h^1$, we obtain the weighted residual formulation:

Find a $u_h \in V_h^1$, such that for all $w_h \in V_h^1$, we have:

$$\sum_{n=0}^{N_T} \sum_{j=1}^{N_n} \Big(\int_{\mathcal{K}_j^n} w_h \, \mathrm{div}\mathcal{F}(u_h)d\mathcal{K} + \int_{\mathcal{K}_j^n} (\mathrm{grad}\, w_h)^T \, \mathfrak{D}(u_h) \, \mathrm{grad}\, u_h d\mathcal{K} \Big) = 0. \qquad (5.2.2)$$

Here, the second integral is the stabilization operator, with $\mathfrak{D}(u_h) \in \mathbb{R}^{2 \times 2}$ the stabilization matrix, which is necessary to obtain monotone solutions near discontinuities. This contribution will be discussed in Section 5.3. The discontinuous Galerkin weak formulation is now obtained by integrating (5.2.2) by parts, resulting in:
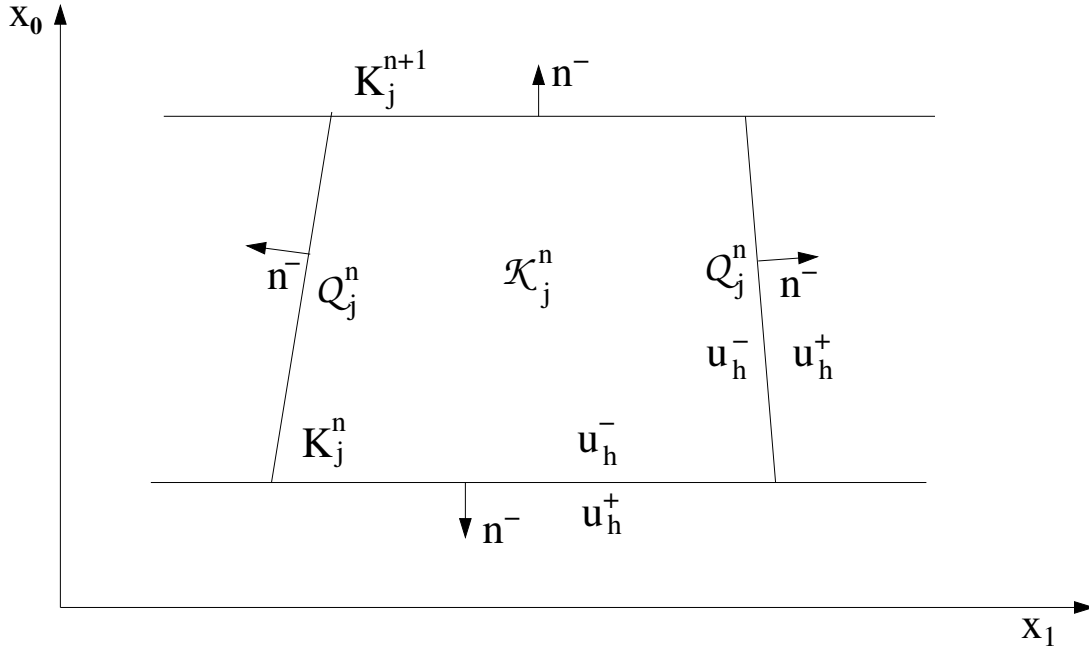
Figure 5.6: Definition of the traces in a space-time slab.

Find a $u_h \in V_h^1$, such that for all $w_h \in V_h^1$, we have:

$$\sum_{n=0}^{N_T} \sum_{j=1}^{N_n} \left(- \int_{\mathcal{K}_j^n} \operatorname{grad} w_h \cdot \mathcal{F}(u_h) d\mathcal{K} + \int_{\partial \mathcal{K}_j^n} w_h^- n^- \cdot \mathcal{F}(u_h^-) d(\partial \mathcal{K}) \right.$$

$$\left. + \int_{\mathcal{K}_j^n} (\operatorname{grad} w_h)^T \mathfrak{D}(u_h) \operatorname{grad} u_h d\mathcal{K} \right) = 0. \tag{5.2.3}$$

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. 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 traces are also shown in Figure 5.6.

The integrals over the faces $\partial \mathcal{K}$ in the weak formulation (5.2.3) 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:

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

$$\tfrac{1}{2}(w_h^- + w_h^+)(\mathcal{F}^- n^- + \mathcal{F}^+ n^+) d\mathcal{S}, \qquad (5.2.4)$$

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, which must impose the condition:

$$\int_{\mathcal{S}} w_h n^- \mathcal{F}^- d\mathcal{S} = - \int_{\mathcal{S}} w_h n^+ \mathcal{F}^+ d\mathcal{S}, \qquad \forall w_h \in V_h^1(\mathcal{T}_h^n),$$

hence the second contribution in (5.2.4) is zero. The boundary integrals therefore are equal to:

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

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:

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

this introduces upwind information into the DG formulation and this technique shows a close resemblance with upwind finite volume methods. The boundary integrals at the element faces can therefore be expressed as:

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

$$= \sum_{\mathcal{K}} \int_{\partial \mathcal{K}} w_h^- H(u_h^-, u_h^+, n^-) d(\partial \mathcal{K}),$$

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

$$H(u_h^-, u_h^+, n^-) = u_h^+ \qquad \text{at } K_j^n$$

$$= u_h^- \qquad \text{at } K_j^{n+1}.$$

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

$$H(u, u, n) = n \cdot \mathcal{F}(u)$$

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 $\mathcal{Q}_j^n$ at time $t = t_n$ and is discussed in detail in chapter 4. Important consistent and monotone Lipschitz fluxes are the:

- Godunov flux

- Engquist-Osher flux

- Lax-Friedrichs flux

- Roe flux with 'entropy fix'

- HLLC flux.

The choice which numerical flux should be used depends on many aspects, e.g. accuracy, robustness, computational complexity, and also personal preference. After introducing the numerical fluxes we can transform the weak formulation (5.2.3) into:

Find a $u_h \in V_h^1(\mathcal{T}_h^n)$, such that for all $w_h \in V_h^1(\mathcal{T}_h^n)$, the following variational equation is satisfied:

$$\sum_{j=1}^{N_n} \left(-\int_{\mathcal{K}_j^n} (\operatorname{grad} w_h) \cdot \mathcal{F}(u_h) d\mathcal{K} + \int_{K_j(t_{n+1})} w_h^- u_h^- dK - \right.$$

$$\int_{K_j(t_n)} w_h^- u_h^+ dK + \int_{\mathcal{Q}_j^n} w_h^- H(u_h^-, u_h^+, n^-) d\mathcal{Q} +$$

$$\left. \int_{\mathcal{K}_j^n} (\operatorname{grad} w_h)^T \mathfrak{D}(u_h) \operatorname{grad} u_h \, d\mathcal{K} \right) = 0. \tag{5.2.5}$$

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 equation for the DG discretization are obtained by introducing the polynomial expansions for u_h and w_h, given by (5.2.1) into the weak formulation (5.2.5) 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:

$$\left|K_j(t_{n+1})\right| \bar{u}_h(K_j(t_{n+1})) - \left|K_j(t_n)\right| \bar{u}_h(K_j(t_n)) + \int_{\mathcal{Q}_j^n} H(u_h^-, u_h^+, n^-) d\mathcal{Q} = 0. \tag{5.2.6}$$

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 $\hat{u}_m(\mathcal{K}_j^n)$, $(m = 1, 2)$ related to the fluctuating part of the flow field $\tilde{u}_h$ are equal to:

$$
\sum_{m=1}^{2} \hat{u}_m(\mathcal{K}_j^n) \Big(-\int_{\mathcal{K}_j^n} \frac{\partial \psi_l}{\partial t} \psi_m d\mathcal{K} + \int_{K_j^{n+1}} \psi_l(x_1, t_{n+1}^-) \psi_m(x_1, t_{n+1}^-) dK
$$

$$
+ \int_{\mathcal{K}_j^n} \frac{\partial \psi_l}{\partial x_k} \mathfrak{D}_{kp}(u_h) \frac{\partial \psi_m}{\partial x_p} d\mathcal{K} \Big) - \int_{K_j^n} u_h(x_1, t_n^-) \psi_l(x_1, t_n^+) dK
$$

$$
+ \int_{\mathcal{Q}_j^n} \psi_l H(u_h^-, u_h^+, n^-) d\mathcal{Q} - \int_{\mathcal{K}_j^n} \frac{\partial \psi_l}{\partial x_1} \mathcal{F}_1(u_h) d\mathcal{K} = 0, \qquad l = 1, 2. \quad (5.2.7)
$$

The algebraic system given by (5.2.6)-(5.2.7) 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 5.4.

5.3   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 results 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 [8] and Jaffre, Johnson and Szepessy [12].

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 (5.2.5) strongly depends on the artificial viscosity matrix $\mathfrak{D}(u_h) \in \mathbb{R}^{2\times 2}$. 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 $\tilde{\mathfrak{D}} \in \mathbb{R}^{2\times 2}$ in computational space using the relation:

$$\mathfrak{D}(u_h|_{\mathcal{K}_j^n}, u_h^*|_{\mathcal{K}_j^n}) = R^T \, \tilde{\mathfrak{D}}(u_h|_{\mathcal{K}_j^n}, u_h^*|_{\mathcal{K}_j^n}) \, R, \tag{5.3.1}$$

with $u_h^*|_{\mathcal{K}_j^n}$ the solution data in the neighboring elements of K_j^n. The matrix $R \in \mathbb{R}^{2\times 2}$ is defined as:

$$R = 2 \, H^{-1} \operatorname{grad} G_K. \tag{5.3.2}$$

The matrix $H \in \mathbb{R}^{2\times 2}$ 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:

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

with $h_i \in \mathbb{R}^+$ the leading terms of the expansion of the mapping G_K (5.1.7) in the computational coordinates ξ_i, $(0 \le i \le 1)$. The multiplication with the factor two in (5.3.2) ensures that for orthogonal cells the matrix R is the rotation matrix from the computational space to the physical space. The stabilization operator in (5.2.5) can now be further evaluated, resulting in:

$$\int_{\mathcal{K}_j^n} \frac{\partial \psi_n}{\partial x_k} R_{pk} \tilde{\mathfrak{D}}_{pq}(u_h|_{\mathcal{K}_j^n}, u_h^*|_{\mathcal{K}_j^n}) R_{ql} \frac{\partial \psi_m}{\partial x_l} d\mathcal{K} = 4 \int_{\hat{\mathcal{K}}} (H^{-1})_{pn} \tilde{\mathfrak{D}}_{pq}(u_h|_{\mathcal{K}_j^n}, u_h^*|_{\mathcal{K}_j^n})(H^{-1})_{qm} |J_{G_K}| d\hat{\mathcal{K}},$$

$$= \frac{4|\mathcal{K}_j^n|}{h_n^2} \delta_{nm} \tilde{\mathfrak{D}}_{nn}(u_h|_{\mathcal{K}_j^n}, u_h^*|_{\mathcal{K}_j^n})$$

(no summation on n), where we used the relations: $(\operatorname{grad} G_K)_{ij} = \partial x_j / \partial \xi_i$ and $\partial \psi_n / \partial \xi_p = \delta_{np}$ 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 [12] is useful. In this model both the jumps at the element

faces and the element residual are used to define the artificial viscosity:

$$\tilde{\mathfrak{D}}_{qq}(u_h|_{\mathcal{K}_j^n}, u_h^*|_{\mathcal{K}_j^n}) = \max\left(C_2 h_{\mathcal{K}}^{2-\beta} R_q(u_h|_{\mathcal{K}_j^n}, u_h^*|_{\mathcal{K}_j^n}), C_1 h_{\mathcal{K}}^{\frac{3}{2}}\right), \qquad q = 1,$$

$$= 0, \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{otherwise,}$$

with

$$R(u_h|_{\mathcal{K}_j^n}, u_h^*|_{\mathcal{K}_j^n}) = \Big|\sum_{k=0}^{1} \frac{\partial \mathcal{F}(u_h)}{\partial u_h} \frac{\partial u_h(G_K(0))}{\partial x_k}\Big| + C_0 |u_h^+(x_{(3)}) - u_h^-(x_{(3)})|/h_{\mathcal{K}} +$$

$$\sum_{m=1}^{2} \frac{1}{h_{\mathcal{K}}} |\bar{n}_{\mathcal{K}}^T f(u_h^+(x_{(m)})) - \bar{n}_{\mathcal{K}}^T f(u_h^-(x_{(m)}))|, \qquad (5.3.3)$$

with $h_{\mathcal{K}} = \sqrt{h_0^2 + h_1^2}$. 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 $\beta = 0.1$. The points $x_{(1)}$ and $x_{(2)}$ are the midpoints of the faces $\mathcal{Q}_j^n$ and $x_{(3)}$ the midpoint of K_j^n. For stronger shocks the addition of the quasi-linear form of the conservation law, which is the first contribution on the righthand side of (5.3.3), 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.

5.4 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 (5.2.6)-(5.2.7) in the space-time slab $\mathcal{E}^n$ can be represented symbolically as an equation for $\hat{u}^n$:

$$\mathcal{L}(\hat{u}^n; \hat{u}^{n-1}) = 0, \qquad (5.4.1)$$

with $\hat{u}^{n-1}$ 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 (5.2.6)-(5.2.7) is to use a pseudo-time integration method. In this technique a pseudo-time derivative of the expansion coefficients is added to (5.4.1) and this equation is solved by marching the solution with a Runge-Kutta method to a steady state:

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

At steady state, when $\frac{\partial \hat{u}^*}{\partial \tau} = 0$, then $\hat{u}^n = \hat{u}^*$.

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:

$$(1 + \alpha_s \bar{\lambda})\hat{v}^{(s)} = \hat{v}^{(0)} + \alpha_s \bar{\lambda}\Big(\hat{v}^{(s-1)} - \mathcal{L}^k(\hat{v}^{(s-1)}, \hat{u}^{n-1})\Big)$$

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 we refer to van der Vegt and van der Ven [23] since they are beyond the scope of these notes.

5.5 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:

$$u_t + a u_x = 0,$$

with $a > 0$. After some lengthy algebra, the space-time discontinuous Galerkin discretization for the linear advection equation using 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:

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

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

with $\triangle x_j^n = x_{j+1}^n - x_j^n$, $\bar{x}_j^n = \frac{1}{2}(x_j^n + x_{j+1}^n)$, $a_1 = \bar{x}_j^{n+1} - \bar{x}_j^n$, $a_2 = 2\triangle x_j^{n+1} - \triangle x_j^n$, $a_3 = 2\triangle x_j^n + \triangle x_j^{n+1}$, $c_{j\pm\frac{1}{2}}^n = \triangle t_n(a - s_{j\pm\frac{1}{2}})$, and $s_{j+\frac{1}{2}}^n = (x_{j+1}^{n+1} - x_{j+1}^n)/\triangle t_n$. 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 mesh 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 $s = s_{j-\frac{1}{2}}^n = s_{j+\frac{1}{2}}^n$ for all j and n, and set the artificial viscosity coefficients equal to zero, then the operator $\mathcal{L}$ can be expressed as:

$$\mathcal{L}(\hat{U}^n, \hat{U}^{n-1}) = \mathcal{A}\hat{U}(\mathcal{K}_j^n) - \mathcal{B}\hat{U}(\mathcal{K}_{j-1}^n) - \mathcal{C}\hat{U}(\mathcal{K}_j^{n-1}), \tag{5.5.1}$$

with the matrices $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathbb{R}^{3\times3}$ defined as:

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

with $\delta = \triangle t(a-s)/\triangle x$ and $s \le a$.

Consider the spatial Fourier mode $\hat{U}(\mathcal{K}_j^n) = e^{i\theta j}\hat{U}^F$, and introduce this into (5.5.1), 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:

$$G(z) = 1$$
$$\text{For } s = 1 \text{ to } 5$$
$$G(z) = \frac{1.0 + \alpha_s(\bar{\lambda} + z)G(z)}{1.0 + \alpha_s\bar{\lambda}}$$
$$\text{End for}$$

The pseudo-time integration method is stable if the amplification factor G satisfies the condition $|G(z_m(\theta))| \le 1$, 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 5.7 contour values of the stability domain $|G(z)| \le 1$ for the 5-stage semi-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 $CFL_{\triangle\tau} \le 1.94$ and for $CFL_{\triangle t} = 100$ the pseudo-time step CFL number must be less than $CFL_{\triangle\tau} \le 1.85$, which is unchanged for larger values of $CFL_{\triangle t}$. The large stability domain and excellent smoothing properties of the semi-implicit Runge-Kutta method for small values of the physical time step CFL number is important for time-accurate simulations.

In Figure 5.8 the effect of the semi-implicit treatment of $\hat{v}$ in Algorithm 1 is shown for $CFL_{\triangle t} = 1$. 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
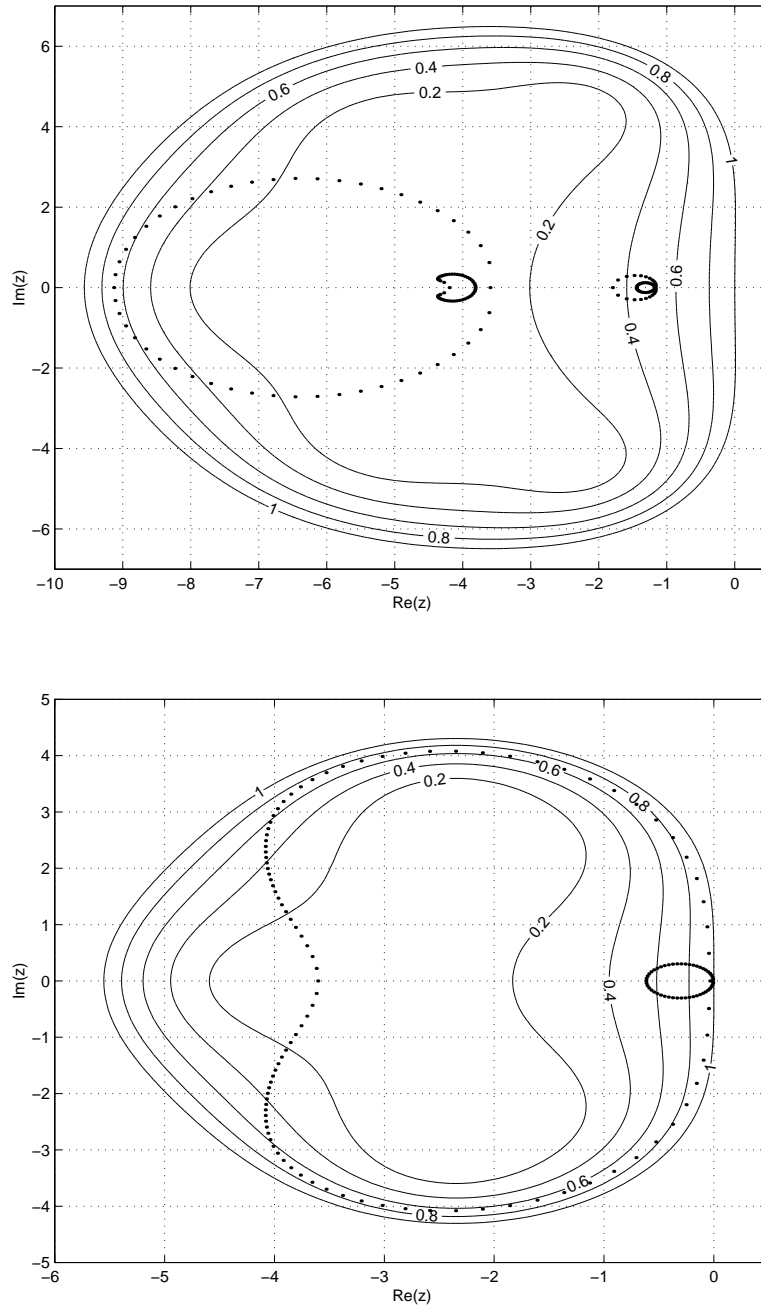
Figure 5.7: Loci of the eigenvalues $z_m(\theta)$, $\theta \in [0, 2\pi)$, (dots) of the DG discretization of $u_t + au_x = 0$ and the stability domain of the 5-stage semi-implicit Runge-Kutta method with optimized coefficients. $CFL_{\triangle t} = 1.0$, $CFL_{\triangle \tau} = 1.8$ (top), $CFL_{\triangle t} = 100.0$, $CFL_{\triangle \tau} = 1.8$ (bottom), no grid velocity.

ensure stability when the semi-implicit technique is not 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 5.8 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.

5.6   Conclusions

The space-time discontinuous Galerkin finite element method provides a versatile conservative numerical technique to solve partial differential equations on time dependent domains which require moving and deforming meshes. The space-time DG method results in a very local, element wise discretization, which is beneficial for mesh adaption using local grid refinement or adjustment of the polynomial order and parallel computing. The use of a pseudo-time integration method using a Runge-Kutta time integration method in combination with a multigrid technique provides an efficient algorithm to solve the non-linear algebraic equations for the expansion coefficients in a space-time DG method.
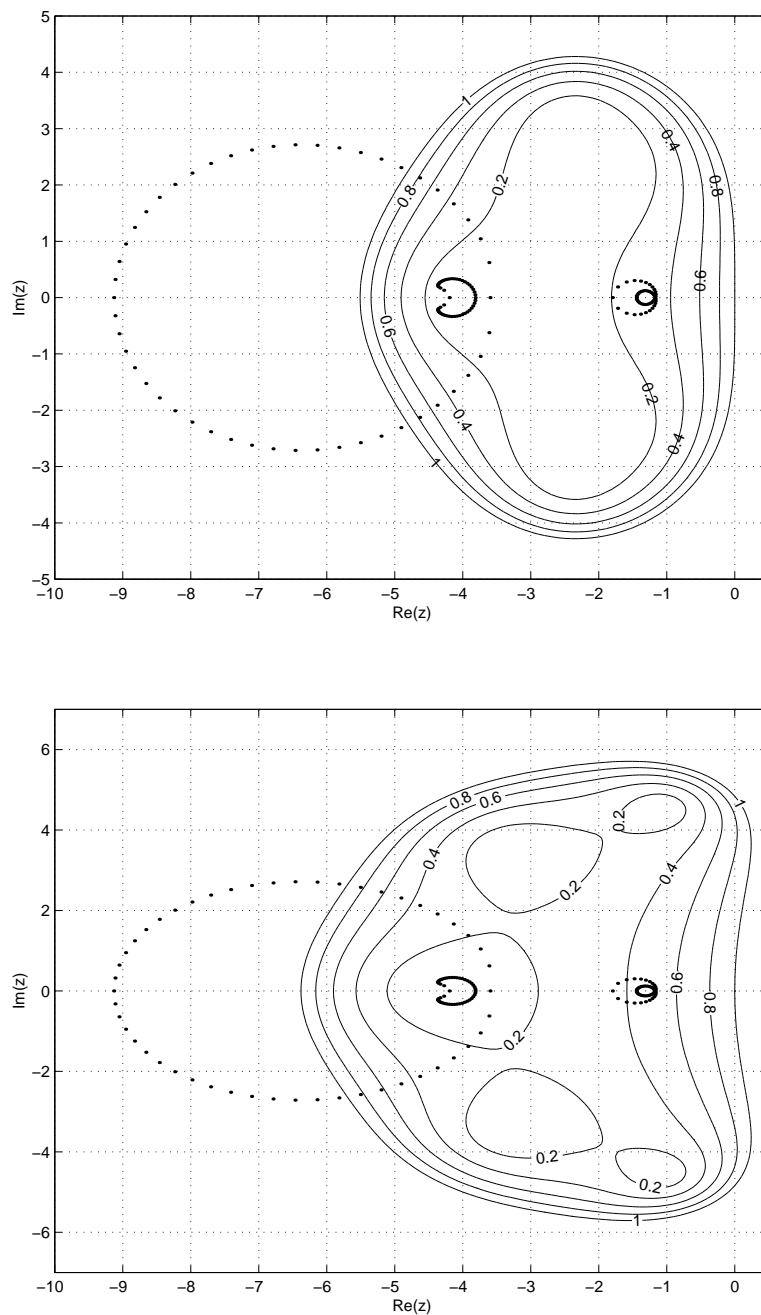
Figure 5.8: Loci of the eigenvalues $z_m(\theta)$, $\theta \in [0, 2\pi)$, (dots) of the DG discretization of $u_t + au_x = 0$ and the stability domain of the explicit 5-stage Runge-Kutta method with optimized coefficients (top) and the five stage semi-implicit Jameson Runge-Kutta scheme (bottom). $CFL_{\triangle t} = 1.0$. $CFL_{\triangle \tau} = 1.8$, no grid velocity.

Chapter 6

Further reading

For the interested reader we have included here a partial list of our own work since 2000.

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

- V.R. Ambati and O. Bokhove, 2007: Space-time discontinuous Galerkin discretization of rotating shallow water equations. J. Comp. Phys. 225, 1233-1261.

- Bernsen, E., Bokhove, O.and Van der Vegt, J.J.W. 2006: A (Dis)Continuous Finite Element Model for Generalized 2D Vorticity Dynamics. J. Comp. Phys. 212, 719-747. technical appendices. Memo. 1787 Dept. of Applied Math., Univ. of Twente. ISSN 0169-2690.

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

- M.A. Botchev, D. Harutyunyan and J.J.W. van der Vegt, 2006: The Gautschi time stepping scheme for edge finite element discretizations of the Maxwell equations, Journal of Computational Physics, Vol. 216, Issue 2, pp. 654-686, doi:10.1016/j.jcp.2006.01.014.

- D. Harutyunyan, F. Izsk, J.J.W. van der Vegt and M.A. Botchev, Adaptive finite element techniques for the Maxwell equations using implicit a posteriori error estimates, article in print, Computer Methods in Applied Mechanics and Engineering, December 2007, doi:10.1016/j.cma.2007.12.006.

- C.M. Klaij, J.J.W. van der Vegt and H. van der Ven, 2006: Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations, Journal of Computational Physics, Vol. 217, Issue 2, pp. 589-611, doi:10.1016/j.jcp.2006.01.018.

56

- 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, Journal of Computational Physics, Vol. 219, pp. 622-643, 2006, doi:10.1016/j.jcp.2006.04.003.

- 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, , Journal of Computational Physics, Vol. 227, No. 2, pp. 1024-1045, 2007, doi:10.1016/j.jcp.2007.08.034.

- L. Pesch, A. Bell, W.E.H. Sollie, V.R. Ambati, O. Bokhove and J.J.W. van der Vegt, 2007: hpGEM- A software framework for Discontinuous Galerkin finite element methods, ACM Transactions on Software 33 (4).

- P. Tassi, O. Bokhove, and C. Vionnet, 2007: Space discontinuous Galerkin method for shallow water flows -kinetic and HLLC flux, and potential vorticity-generation. Advances in water resources 30, 998-1015.

- P.A. Tassi, S. Rhebergen, C.A. Vionnet and O. Bokhove, 2008: A discontinuous Galerkin finite element model for morphological evolution under shallow flows, Comput. Methods Appl. Mech. Engrg., in press.

- 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., Vol 227/3 pp 1887-1922, (2008), doi: 10.1016/j.jcp.2007.10.007.

- D. Sarmany, M.A. Botchev and J.J.W. van der Vegt, 2007: Dispersion and Dissipation Error in High-Order Runge-Kutta Discontinuous Galerkin Discretisations of the Maxwell Equations, Journal of Scientific Computing, Vol. 33, no. 1, pp. 47-74, doi:10.1007/s10915-007-9143-y.

- 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, Applied Numerical Mathematics, Vol. 56, Issue 12, pp. 1491-1518, 2006, doi:10.1016/j.apnum.2005.11.003.

- 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, Computer Methods in Applied Mechanics and Engineering, Vol. 196, Issues 13-16, pp. 1984-1996, 2007, doi:10.1016/j.cma.2006.11.007.

- Y. Xu, J.J.W. van der Vegt, and O. Bokhove, 2008: Discontinuous Hamiltonian Finite Element Method for a Bilinear Poisson Bracket. Submitted Oct. 19th J. Sci. Comput. In press.

- Van der Vegt, J.J.W., Iszak, F, and Bokhove, O. 2007: Error analysis of a continuous-discontinuous Galerkin finite element model for generalized 2D vorticity dynamics. Siam J. Num. Anal. 45, 1349.

- J.J.W. van der Vegt and Y. Xu, Space-time discontinuous Galerkin method for non-linear water waves, Journal of Computational Physics, Vol. 224, Issue 1, pp. 17-39, 2007, doi:10.1016/j.jcp.2006.11.031.

- 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 Applied Numerical Mathematics, 2007, doi:10.1016/j.apnum.2007.11.010.

- 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 I. General Formulation, Journal of Computational Physics, Vol. 182(2), pp. 546-585, 2002, doi:10.1006/jcph.2002.7185.

- 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, Computer Methods in Applied Mechanics and Engineering, Vol. 191, pp. 4747-4780, 2002, doi:10.1016/S0045-7825(02)00403-6.

Bibliography

[1] Batten, P., Leschziner, M.A., & Goldberg, U.C. (1997) 'Average-state Jacobians and implicit methods for compressible viscous and turbulent flows'. *J. Comp. Phys.* 137, 38–78.

[2] Bernsen, E., Bokhove, O., & Vanneste, J. (2004) 'On two-dimensional vortical flows with separatrices'. Poster at the Symposium on nonlinear systems, June 10-11 2004, Enschede, The Netherlands. wwwhome.math.utwente.nl/∼bokhoveo/obpubs.html

[3] Bernsen, E,, Bokhove, O., & Vegt van der, J.J.W. (2006) 'Discontinuous Galerkin finite element approximations of 2D vorticity equations with linear elliptic inversions and circulation'. Online *J. Comp. Phys.* **212**, 719–747.

[4] Bokhove, O. (2003) 'Flooding and drying in discontinuous Galerkin finite-element discretizations of shallow-water equations. Part 2: two dimensions'. Memorandum 1684, *Mathematical Communications 2003* University of Twente, online available.

[5] Bokhove, O. (2005) 'Flooding and drying in discontinuous Galerkin finite-element discretizations of shallow-water equations. Part 1: one dimension'. *J. of Sci. Comp.* **22**, 47–82.

[6] Bokhove, O., Woods, A.W. & Boer de A. (2005) 'Magma flow through elastic-walled dikes'. Reviewed at Southwest Research Institute, San Antonio Texas. Reviewed by CNWRA for the U.S. Nuclear Regulatory Committee (NRC). *Theor. Comput. Fluid Mech.* **19**, 261–286.

[7] Brenner, C.B., & Scott, L.R. (1994) *The mathematical theory of finite elements.* Springer Verlag, 297 pp.

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

[9] Cockburn, B., & Shu, C.-W. (1989) 'TVB Runge-Kutta local projection discontinuous Galerkin method for scalar conervation laws II: General framework'. *Math. Comp.* **52**, 411–435.

[10] Cockburn, B., Lin, S., & Shu, C.-W. (1989) 'TVB Runge-Kutta local projection discontinuous Galerkin method for scalar conervation laws III: One dimensional systems'. *J. Comp. Phys.* **84**, 90-113.

[11] Cockburn, B., & Shu, C.-W.: The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *Siam. J. Numer. Anal.* **135**, 2240–2463 (1998)

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

[13] C.M. Klaij, J.J.W. van der Vegt and H. van der Ven, (2005) 'Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations'. Submitted to *J. Comp. Phys.*, see also Technical Memorandum 1774, Department of Applied Mathematics, University of Twente, http://www.math.utwente.nl/publications/

[14] Krivodonova, L., Xin, J., Remacle, J.-F., Chevaugeon, N., & Flaherty, J.E. (2003) 'Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws'. *Applied Num. Math.* **48**, 323–338.

[15] Lesoinne, M. and Farhat, C., 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).

[16] Lucquin, B, & Pironneau, O. (1998) *Introduction to scientific computing.* Wiley, 361 pp.

[17] Luo, L., Baum, J.D. and Löhner, R. (2007) 'A Hermite WENO-based limiter for discontinuous Galerkin method on unstructered grids'. *J. Comp. Phys.* **225**, 686–713.

[18] Morton, K.W., & Mayers, D.F. (1994) *Numerical solution of partial differential equations.* Cambridge University Press.

[19] Morton, K.W. (1996) *Numerical Solution of Convection-Diffusion Problems.* Chapman & Hall, 372 pp.

[20] Shu, C-W. & Osher, S. (1989) Efficient implementation of essentially non-oscillatory shock-capturing schemes II. *J. Comp. Phys.* **83**, 32–78.

[21] Sudirham, J.J., Van der Vegt, J.J.W. and Van Damme, R.M.J., (2004) 'Space-time discontinuous Galerkin method for advection-diffusion problems on time-dependent domains'. Submitted to *Appl. Numer. Math.*

[22] Toro, E.F. (1999) *Shock capturing methods for free-surface flows.* Wiley, Toronto, 309 pp.

[23] Van der Vegt, J.J.W., & Van der Ven, H. (2002) 'Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. Part I. General formulation'. *J. Comp. Phys.* **182**, 546–585.

[24] Van der Ven, H., & Van der Vegt, J.J.W. (2002) '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**, 4747–4780.

[25] Vorst van der, H.A. (2003) *Iterative Krylov methods for large linear systems.* Cambridge University Press, 221 pp.

[26] Yan, J, & Shu, C.-W. (2002) Local discontinuous Galerkin methods for partial differential equations with higher order derivatives. *J. Sci. Comp.* **17**, 27–47.