



Error Boundedness of Discontinuous Galerkin Methods with Variable Coefficients

Philipp Öffner¹ · Hendrik Ranocha²

Received: 6 June 2018 / Revised: 7 November 2018 / Accepted: 27 December 2018 /
Published online: 7 January 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract

For practical applications, the long time behaviour of the error of numerical solutions to time-dependent partial differential equations is very important. Here, we investigate this topic in the context of hyperbolic conservation laws and flux reconstruction schemes, focusing on the schemes in the discontinuous Galerkin spectral element framework. For linear problems with constant coefficients, it is well-known in the literature that the choice of the numerical flux (e.g. central or upwind) and the selection of the polynomial basis (e.g. Gauß–Legendre or Gauß–Lobatto–Legendre) affects both the growth rate and the asymptotic value of the error. Here, we extend these investigations of the long time error to variable coefficients using both Gauß–Lobatto–Legendre and Gauß–Legendre nodes as well as several numerical fluxes. We derive conditions guaranteeing that the errors are still bounded in time. Furthermore, we analyse the error behaviour under these conditions and demonstrate in several numerical tests similarities to the case of constant coefficients. However, if these conditions are violated, the error shows a completely different behaviour. Indeed, by applying central numerical fluxes, the error increases without upper bound while upwind numerical fluxes can still result in uniformly bounded numerical errors. An explanation for this phenomenon is given, confirming our analytical investigations.

Keywords Flux reconstruction · Discontinuous Galerkin · Summation-by-parts · Error analysis · Error bound · Error growth

Mathematics Subject Classification 65N15 · 65N12 · 65N35 · 65N30 · 65N06

✉ Philipp Öffner
philipp.oeffner@math.uzh.ch
Hendrik Ranocha
h.ranocha@tu-bs.de

¹ Universität Zürich, Winterthurerstrasse 190, 8057 Zürich, Switzerland

² TU Braunschweig, Universitätsplatz 2, 38106 Braunschweig, Germany

1 Introduction

The investigation of the error of numerical solutions to hyperbolic conservation laws has received much interest in the literature [1,7,16,18,20,27,28,31,48]. In some of these papers, a linear error growth (or nearly linear growth) in time is observed, while the numerical error is bounded uniformly in time for others. In [27], the author explains under what conditions the error is or is not bounded in time if a linear problem with constant coefficients is considered. Using finite difference approximations with summation-by-parts (SBP) operators and simultaneous approximation terms (SATs), the error behaviour depends on the choice of boundary procedure of the problem. If one catches the waves in cavities or with periodic boundary conditions, linear growth is observed like in [16], whereas for inflow-outflow problems one obtains uniform boundedness in time. In other words, if the boundary approach has sufficient dissipation, the error is bounded. It does not depend on the internal discretisation.

This investigation is extended to the discontinuous Galerkin spectral element methods (DGSEM) in [20] and to Flux Reconstruction (FR) schemes in [31]. Different from [27], using DG or FR methods, the internal approximation has an influence on the behaviour of the error, since there are additional parameters. The choices of numerical fluxes (upwind and central) and polynomial bases (Gauß–Lobatto–Legendre or Gauß–Legendre) have an impact on the magnitude of the error and the speed at which the asymptotic error is reached.

In all of these works [20,27,31], the model problem under consideration is a linear advection equation with constant coefficients. In this paper, we extend these investigations by considering variable coefficients. The introduction of variable coefficients leads to stability issues and problems in the discretisation of the numerical fluxes as described in [36]. Using split forms in the spatial discretisation [10,26], we are able to construct an error equation in the spirit of [20] for our new model problem.

Furthermore, using this error equation, we formulate conditions on the variable coefficients to guarantee that the error is still bounded uniformly in time. Here, it will be essential that the first derivative of the variable coefficient $a(x)$ is positive. In numerical tests, we demonstrate that if these conditions are fulfilled, the errors behave like in the case of constant coefficients. If these conditions are not satisfied, we have a different behaviour. If central numerical fluxes are applied, the errors tend to infinity, whereas the errors using upwind fluxes in the calculation may still remain bounded uniformly in time. This matches our analysis and the conditions which we derive in the analytical investigations in Sects. 4 and 5.

The paper is organised as follows: in the second section, we introduce the model problem and repeat the stability analysis from the continuous point of view. In Sect. 3, the main idea of SBP-FR methods and the concrete schemes are repeated. Then, we present the different numerical fluxes under consideration and introduce the main focus of our study, the numerical errors. We repeat some approximation results which we need in the following sections. For our analysis, it is essential whether or not boundary points are included in the nodal bases. In Sect. 4, we start by considering Gauß–Lobatto–Legendre nodes. These include the boundary points and we demonstrate that the error is bounded uniformly in time under some conditions on the variable coefficient $a(x)$. Afterwards, in Sect. 5, we adapt the investigation from before to Gauß–Legendre nodes which do not contain the boundary points. We get additional error terms in our error equation and focus finally on the different discretisations of the numerical fluxes. Similar conditions are derived like before on $a(x)$ to guarantee that the error is bounded in time. We confirm our investigation by numerical experiments in Sect. 6, which includes also a physical interpretation of the test cases under consideration. Furthermore, a first analytical study about the error inequalities is given if one of the conditions on a is not

fulfilled. In Sect. 7, we generalize our investigation to systems (linearized Euler equations and magnetic induction equation) and demonstrate problems which arise in these cases. We give an outlook for further research. Finally, we summarise and discuss our results.

2 Model Problem and Continuous Setting

The problem under consideration is the following linear advection equation

$$\begin{aligned} \partial_t u(t, x) + \partial_x(a(x)u(t, x)) &= 0, & t > 0, x \in (x_L, x_R), \\ u(t, x_L) &= g_L(t), & t \geq 0, \\ u(0, x) &= u_0(x), & x \in [x_L, x_R], \end{aligned} \tag{1}$$

with variable speed $a(x) > 0$ and compatible initial and boundary conditions u_0, g_L . Furthermore, the initial and boundary values are chosen in such a way that $u(t, x) \in H^m(x_L, x_R)$ for $m > 1$ and that its norm $\|u(t)\|_{H^m}$ is bounded uniformly in time. This condition is physically meaningful, e.g. for problems with sinusoidal boundary inputs. However, we will also present in Sect. 6 an example where this condition is violated and our whole analysis will break down.

The impact of the boundary condition and the variable coefficient a on the solution is essential and will shortly be repeated from [29,36]. The energy of the solution u of the initial boundary value problem (1) is measured by the classical L^2 -norm $\|u\|^2 = \int_{x_L}^{x_R} u^2 dx$. Focusing on the weak formulation of the advection equation (1), a test function $\varphi \in C^1[x_L, x_R]$ is multiplied and integrated over the domain

$$\int_{x_L}^{x_R} (\partial_t u)\varphi dx + \int_{x_L}^{x_R} (\partial_x(au))\varphi dx = 0. \tag{2}$$

Setting $\varphi = u$, application of the product rule and integration-by-parts yields

$$\begin{aligned} \frac{d}{dt} \|u\|^2 &= 2 \int_{x_L}^{x_R} u \partial_t u dx = -2 \int_{x_L}^{x_R} u \partial_x(au) dx \\ &= - \int_{x_L}^{x_R} (u \partial_x(au) + au \partial_x u + u^2 \partial_x a) dx = -au^2|_{x_L}^{x_R} - \int_{x_L}^{x_R} u^2 \partial_x a dx \\ &= a(x_L)g_L^2 - a(x_R)u(x_R)^2 - \int_{x_L}^{x_R} u^2 \partial_x a dx. \end{aligned}$$

Integration in time over an interval $[0, T]$ leads to

$$\begin{aligned} \|u(T)\|^2 - \|u_0\|^2 &= a(x_L) \int_0^T g_L^2(t) dt - a(x_R) \int_0^T u^2(t, x_R) dt \\ &\quad - \int_{x_L}^{x_R} \left(\int_0^T u^2(t, x) dt \right) \partial_x a(x) dx. \end{aligned} \tag{3}$$

Here, the change of energy at time T can be expressed by the energy added at the left side through the boundary condition minus the energy lost through the right side, and an energy term considering the variation of the coefficient a . If $\partial_x a$ is bounded, the energy is also bounded for a fixed time interval. It can be found in [29, Section 2] that the energy fulfils

$$\begin{aligned} \|u(t)\|^2 &\leq \exp(t\|\partial_x a\|_{\mathbb{L}^\infty}) \cdot \left(\|u_0\|^2 \right. \\ &\quad \left. + \int_0^t \exp(-\tau\|\partial_x a\|_{\mathbb{L}^\infty}) (a(x_L)g_L(\tau)^2 - a(x_R)u(\tau, x_R)^2) d\tau \right). \end{aligned}$$

The numerical scheme has to be constructed such that the approximation imitates this behaviour. Special focus has to be given on an adequate discretisation of the flux function f , which depends on the space coordinate x via the variable coefficients $a(x)$. The numerical fluxes have to be adjusted. We will specify this in Sect. 3.2.

3 Flux Reconstruction with Summation-by-Parts Operators and Numerical Fluxes

In the first part of this section, we shortly repeat the main ideas of Flux Reconstruction (FR), also known as Correction Procedure via Reconstruction, using Summation-by-parts Operators (SBP). A more detailed introduction to this topic can be found in the articles [38,39] and references therein.

3.1 Flux Reconstruction Using Summation-by-Parts Operators

We consider a one-dimensional scalar conservation law

$$\partial_t u(t, x) + \partial_x f(t, x, u(t, x)) = 0, \quad t > 0, x \in (x_0, x_K), \tag{4}$$

equipped with appropriate initial and boundary conditions. The domain (x_0, x_K) is split into K non-overlapping elements $[x_0, x_K] = [x_0, x_1] \cup \dots \cup [x_{K-1}, x_K]$. The FR method is a semidiscretisation applying a polynomial approximation using a nodal basis on each element. Therefore, each interval $[x_{i-1}, x_i]$ is transferred onto a standard element, which is in our case simply $[-1, 1]$. All calculations are conducted within this reference element. Let \mathbb{P}^N be the space of polynomials of degree $\leq N$, $-1 \leq \zeta_i \leq 1$ ($i = 0, \dots, N$) the interpolation points and $\mathbb{I}^N : \mathbf{L}^2(-1, 1) \cap C(-1, 1) \rightarrow \mathbb{P}^N(-1, 1)$ be the interpolation operator and P_{N-1}^m be the orthogonal projection of u onto \mathbb{P}^{N-1} with respect to the inner product of the Sobolev space $H^m((-1, 1))$. The solution is approximated by a polynomial $U \in \mathbb{P}^N$ and the basic formulation of a nodal Lagrange basis¹ is employed. Instead of working with U one may also express the numerical solution as the vector \underline{u} with coefficients $u_i = U(\zeta_i)$, $i \in \{0, \dots, N\}$. All the relevant information are stored in these coefficients and one may write

$$u(\zeta) \approx U(\zeta) = \sum_{i=0}^N u_i l_i(\zeta), \tag{5}$$

where $l_i(\zeta)$ is the i th Lagrange interpolation polynomial that satisfies $l_i(\zeta_j) = \delta_{ij}$. In finite difference (FD) methods, it is natural to work with the coefficients only and since we are working with SBP operators with origins lying in the FD community [21], we utilise the coefficients. Finally, the flux $f(u)$ is also approximated by a polynomial, where the coefficients are given by $\underline{f}_i = f(\underline{u}_i) = f(U(\zeta_i))$.

Now, with respect to the chosen basis (interpolation points), (an approximation of) the derivative is represented by the matrix \underline{D} . Moreover, a discrete scalar product is represented

¹ Modal bases are also possible [39], but we won't consider these in this paper.

by the symmetric and positive definite mass/norm² matrix $\underline{\underline{M}}$, approximating the usual L^2 scalar product, i.e.

$$\underline{\underline{D}}u \approx \partial_x u \text{ and } (\underline{u}, \underline{v})_N := \underline{u}^T \underline{\underline{M}}v \approx \int_{x_i}^{x_{i+1}} uv \, dx. \tag{6}$$

Using Lagrange polynomials, we get $D_{kj} = l'_j(\zeta_k)$ and $\underline{\underline{M}} = \text{diag}(\omega_0, \dots, \omega_N)$, where ω_j are the quadrature weights associated with the nodes ζ_j . For Gauß–Legendre nodes, $\omega_j = \int_{-1}^1 l_j(x)^2 \, dx$. For other quadrature nodes such as Gauß–Lobatto–Legendre nodes, the mass matrix is in general not exact.

SBP operators are constructed in such way that they mimic integration-by-parts on a discrete level, as described in the review articles [8,44] and references cited therein. Until now, we have expressions/approximations for the derivative as well as for the integration. Hence, only the evaluation on the boundary is missing. Here, we have to introduce two different operators. First, the restriction operator, which is represented by the matrix $\underline{\underline{R}}$, approximates the interpolation of a function to the boundary points $\{x_{i-1}, x_i\}$. Second, the diagonal boundary matrix $\underline{\underline{B}} = \text{diag}(-1, 1)$ gives the difference of boundary values. It is

$$\underline{\underline{R}}u \approx \begin{pmatrix} u(x_{i-1}) \\ u(x_i) \end{pmatrix} \text{ and } (u_L, u_R) \cdot \underline{\underline{B}} \cdot \begin{pmatrix} v_L \\ v_R \end{pmatrix} = u_R v_R - u_L v_L.$$

Finally, all operators are introduced and they have to fulfil the SBP property

$$\underline{\underline{M}}\underline{\underline{D}} + \underline{\underline{D}}^T \underline{\underline{M}} = \underline{\underline{R}}^T \underline{\underline{B}}\underline{\underline{R}}, \tag{7}$$

in order to mimic integration-by-parts on a discrete level

$$\underline{u}^T \underline{\underline{M}}\underline{\underline{D}}v + \underline{u}^T \underline{\underline{D}}^T \underline{\underline{M}}v \approx \int_{x_{i-1}}^{x_i} u (\partial_x v) + \int_{x_{i-1}}^{x_i} (\partial_x u) v = uv|_{x_{i-1}}^{x_i} \approx \underline{u}^T \underline{\underline{R}}^T \underline{\underline{B}}\underline{\underline{R}}v. \tag{8}$$

Here, we investigate the long time error behaviour of linear problems with variable coefficients. To represent these coefficients in our semidiscretisation, multiplication operators are necessary. If the function U is represented by \underline{u} , the discrete operator approximating the linear operator $v \mapsto vU$ is represented by the matrix $\underline{\underline{u}}$, mapping \underline{v} to $\underline{\underline{u}}\underline{v}$. In a nodal basis,³ the standard multiplication operators consider pointwise multiplication. This means that $\underline{\underline{u}}$ is diagonal with $\underline{\underline{u}} = \text{diag}(\underline{u})$ and $(\underline{\underline{u}}v)_i = u_i v_i$.

One central point in our investigation in Sects. 4 and 5 will be whether the boundary points are included in the set of interpolation nodes (Sect. 4) or not (Sect. 5). This is an essential point in this paper and also in others [22,29,34,36,38,39]. If the boundary points $\{x_{i-1}, x_i\}$ are included, the restriction operators are simply

$$\underline{\underline{R}} = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}, \quad \underline{\underline{R}}^T \underline{\underline{B}}\underline{\underline{R}} = \text{diag}(-1, 0, \dots, 0, 1).$$

Thus, restriction to the boundary and multiplication commute, i.e.

$$\begin{pmatrix} u_0 & v_0 \\ \underline{u}_N & \underline{v}_N \end{pmatrix} = (\underline{\underline{R}}u) \cdot (\underline{\underline{R}}v) = \underline{\underline{R}}uv = \underline{\underline{R}}v\underline{u}. \tag{9}$$

² Both names are used. In the DG community [12], the matrix is called mass matrix, whereas the name norm matrix is common for FD methods.

³ For a modal basis see [39].

At the continuous level this property is fulfilled and so we want this property also in our semidiscretisation. However, if the boundary nodes are excluded, restriction and multiplication will not commute in general. Therefore, some corrections have to be applied [33,34,38,39]. It is common to use some linear combination/splitting of the terms $(\underline{Ru}) \cdot (\underline{Rv})$ and \underline{Rvu} to mimic (9) at a discrete level. We have to mention that the construction of these correction terms can be very difficult (e.g. [34] and [35, Section 4.5]) and for some equations like Euler for example, it is still an open problem if such correction terms exist [33].

Now, the general aspects of SBP operators are introduced and we can focus on our FR approach. Contrary to DG methods, we do not apply a variational formulation (i.e. weak form) of (4). Instead, the differential form is used, corresponding to a strong form DG method. To describe the semidiscretisation all operators are introduced. We apply the discrete derivative matrix \underline{D} to f . The divergence is $\underline{D}f$. Since the solutions will probably have discontinuities across elements, we will have this in the discrete flux, too. In order to avoid this problem, a numerical flux f^{num} is introduced which computes a common flux at the boundary using values from both neighbouring elements. The main idea of the FR schemes is that the numerical flux at the boundaries will be corrected by functions in such manner that information of two neighbouring elements interact and basic properties like conservation hold also in the semidiscretisation. Therefore, we add a correction term using a correction matrix \underline{C} at the boundary nodes. This gives *Flux Reconstruction* its name. Hence, a simple FR (or correction procedure via reconstruction, CPR) method for (4) with boundary nodes included reads

$$\partial_t u = -\underline{D}f - \underline{C} \left(f^{num} - \underline{R}f \right). \tag{10}$$

A general choice of the correction matrix \underline{C} recovers the linearly stable flux reconstruction methods of [46,47], as described by [38]. The canonical choice for the correction matrix is

$$\underline{C} := \underline{M}^{-1} \underline{R}^T \underline{B}. \tag{11}$$

It is a generalisation of simultaneous approximation terms (SATs) used in finite difference methods [6] and corresponds to a strong form of the discontinuous Galerkin method [19]. In this paper we concentrate on the correction term using (11). However, a generalisation to the schemes of Vincent et al. [46] is possible and can be done as in [31,38]. However, further problems emerge concerning the interchangeability of coefficients in the broken norms and one has to be careful.

3.2 Numerical Fluxes

Special focus has to be given on an adequate discretisation of the flux function f , which depends on the space coordinate x via the variable coefficients $a(x)$. The numerical fluxes have to be adjusted. The numerical fluxes under consideration will be

$$\text{Edge based central flux } f^{num}(u_-, u_+) = a(x) \frac{u_- + u_+}{2}, \tag{12}$$

$$\text{Split central flux } f^{num}(u_-, u_+) = \frac{a_- u_- + a_+ u_+}{2}, \tag{13}$$

$$\text{Unsplit central flux } f^{num}(u_-, u_+) = \frac{(au)_- + (au)_+}{2}, \tag{14}$$

$$\text{Edge based upwind flux } f^{num}(u_-, u_+) = a(x)u_-, \tag{15}$$

$$\text{Split upwind flux } f^{\text{num}}(u_-, u_+) = a_- u_-, \tag{16}$$

$$\text{Unsplit upwind flux } f^{\text{num}}(u_-, u_+) = (au)_-. \tag{17}$$

If boundary nodes are included and the coefficients a of the discrete version of the function a are obtained by evaluating a at these nodes, (12), (13), (14) and (15), (16), (17) are identical like in the next Sect. 4. From the stability analysis in [36], we know that using the unsplit fluxes (14), (17) may result in stability issues. Furthermore, applying the other fluxes and to guarantee stability, we need that the interpolation speeds have to be exact. In this cases, the edge based (12), (15) and the split numerical fluxes (15), (16) are equivalent. This exactness can be achieved by evaluating the speed $a(x)$ at $N + 1$ Gauss–Lobatto points⁴ and then the unique interpolating polynomial can be evaluated at the nodes used in the basis not including the boundary. We will consider this later in detail in Sect. 5.

As it was described in Sect. 3.1, all calculations are done in a standard element $[-1, 1]$. Therefore, a transformation of every element $e^k = [x_k, x_{k-1}]$ to this standard element is necessary. Equation (2) is transformed to

$$\frac{\Delta x_k}{2} \langle \partial_t u, \varphi^k \rangle + \langle \partial_\xi (a^k u), \varphi^k \rangle = 0, \tag{18}$$

where $\langle \cdot, \cdot \rangle$ is the L^2 -scalar product, φ^k is a test function in the k th element and the factor $\frac{\Delta x_k}{2} = \frac{x_k - x_{k-1}}{2}$ comes from the transformation. Applying the product rule and integration-by-parts to (18) yields

$$\frac{\Delta x_k}{2} \langle \partial_t u, \varphi^k \rangle + \frac{1}{2} \left(\langle \partial_\xi (a^k u), \varphi^k \rangle + \langle a^k \partial_\xi u, \varphi^k \rangle + \langle u \partial_\xi a^k, \varphi^k \rangle \right) = 0, \tag{19}$$

$$\begin{aligned} &\frac{\Delta x_k}{2} \langle \partial_t u, \varphi^k \rangle + \frac{1}{2} \left(a^k u \varphi^k|_{-1} - \langle a^k u, \partial_\xi \varphi^k \rangle \right) \\ &+ \langle a^k \partial_\xi u, \varphi^k \rangle + \langle u \partial_\xi a^k, \varphi^k \rangle = 0. \end{aligned} \tag{20}$$

Formulation (20) will be used to construct the error equations.

3.3 Numerical Errors and Approximation Results

The error in every element is given by $E^k := u^k(t, x(\xi)) - U^k(t, \xi)$, where u represents the solution in the k th element and U is the spatial approximation. Using the interpolation operator and adding zero to the error, E^k can be split in two parts:

$$E^k = \underbrace{(\mathbb{I}^N(u^k) - U^k)}_{=: \varepsilon_1^k \in \mathbb{P}^N} + \underbrace{(u^k - \mathbb{I}^N(u^k))}_{=: \varepsilon_p^k}. \tag{21}$$

With the triangle inequality, one may bound this by

$$\|E^k\|_N \leq \|\varepsilon_1^k\|_N + \|\varepsilon_p^k\|_N, \tag{22}$$

where $\|\cdot\|_N$ is the discrete norm induced by the discrete scalar product (6). ε_p^k is the interpolation error, which is the sum of the series truncation error and the aliasing error. As it was already described in [5,11,15,30,32], its continuous norm converges spectrally fast for the different bases under consideration. It is

⁴ We assume here a nodal basis using $N + 1$ points to represent polynomials of degree $\leq N$.

$$|u|_{H^{m;N}(-1,1)} := \left(\sum_{j=\min(m,N+1)}^m \|u^{(j)}\|_{L^2(-1,1)}^2 \right)^{\frac{1}{2}}$$

the seminorm of the Sobolev space $H^m(-1, 1)$. For Gauß–Lobatto–Legendre/Gauß–Legendre points,

$$\|u - \mathbb{I}(u)\|_{L^2(-1,1)} \leq CN^{-m}|u|_{H^{m;N}(-1,1)}, \tag{23}$$

where C depends on m . In view of our investigation, one needs to consider the interpolation error not only in the standard interval $[-1, 1]$, but in each element e^k . Therefore, the estimation (23) will be transform to every element.⁵ With a combination of [11, Theorem 6.6.1] and [5, Section 5.4.4], for Gauß–Legendre nodes

$$\|\varepsilon_p^k\|_{H^n(e^k)} \leq C (\Delta x_k)^{n-\min\{m,N\}+\frac{1}{2}} N^{n-m+\frac{1}{2}} |u|_{H^{m;N}(e^k)} \tag{24}$$

for $n = 0, 1$. For Gauss–Lobatto–Legendre nodes, delete $\frac{1}{2}$ on the right side of (24). A finite dimensional normed vector space is considered and all norms are equivalent there. This allows to bound the discrete norm in terms of the continuous ones and implies that $\|\varepsilon_p^k\|_N$ in (22) decays spectrally fast in all cases of consideration. In other words, ε_1^k has to be investigated in detail. This error describes the difference of the interpolation of u and the spatial approximation U .

Therefore, we have to consider the numerical schemes under consideration. The semidiscretisation of (1) is given by the following form:

$$\begin{aligned} \partial_t \underline{u} = & -\frac{1}{2} \underline{Dau} - \frac{1}{2} \underline{aDu} - \frac{1}{2} \underline{uDa} \\ & - \underline{M}^{-1} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num}} - \frac{1}{2} \underline{Rau} - \frac{1}{2} (\underline{Ra}) \cdot (\underline{Ru}) \right), \end{aligned} \tag{25}$$

where analogously to the continuous setting a split formulation has been applied. The last term is due to the fact that for Gauss–Legendre nodes the restriction operators \underline{R} do not commute with the multiplication operators. Therefore, corrections have to be used. If boundary nodes are included, multiplication and restriction commute and we can simplify (25) to

$$\partial_t \underline{u} + \frac{1}{2} (\underline{Dau} + \underline{aDu} + \underline{uDa}) + \underline{M}^{-1} \underline{R}^T \underline{B} (\underline{f}^{\text{num}} - \underline{Rau}) = 0. \tag{26}$$

In (25) and (26), the terms 2–4 approximate the split form $\frac{1}{2} (\partial_x(au) + a(\partial_x u) + u(\partial_x a))$ of the flux derivative $\partial_x(au)$ of (1). Since the semidiscretisation is used in every element e^k , one obtains for every element the following form:

$$\begin{aligned} \frac{\Delta x_k}{2} \partial_t \underline{u} + \frac{1}{2} (\underline{Dau} + \underline{aDu} + \underline{uDa}) \\ + \underline{M}^{-1} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num}} - \frac{1}{2} \underline{Rau} - \frac{1}{2} (\underline{Ra}) \cdot (\underline{Ru}) \right) = 0. \end{aligned} \tag{27}$$

Using a Galerkin approach, $\varphi^{k,T} \underline{M}$ is multiplied to (27), resulting due to the SBP property (7) in

⁵ A more detailed analysis can be found in [2,3].

$$\begin{aligned}
 & \frac{\Delta x_k}{2} \underline{\varphi}^{k,T} \underline{M} \partial_t \underline{u} + \frac{1}{2} \underline{\varphi}^{k,T} \underline{M} \left(\underline{D} \underline{a} \underline{u} + \underline{a} \underline{D} \underline{u} + \underline{u} \underline{D} \underline{a} \right) \\
 & + \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num}} - \frac{1}{2} \underline{R} \underline{a} \underline{u} - \frac{1}{2} \left(\underline{R} \underline{a} \right) \cdot \left(\underline{R} \underline{u} \right) \right) = 0, \\
 & \frac{\Delta x_k}{2} \underline{\varphi}^{k,T} \underline{M} \partial_t \underline{u} + \frac{1}{2} \underline{\varphi}^{k,T} \left(\underline{R}^T \underline{B} \underline{R} - \underline{D}^T \underline{M} \right) \underline{a} \underline{u} + \frac{1}{2} \underline{\varphi}^{k,T} \underline{M} \underline{a} \underline{D} \underline{u} \\
 & + \frac{1}{2} \underline{\varphi}^{k,T} \underline{M} \underline{u} \underline{D} \underline{a} + \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num}} - \frac{1}{2} \underline{R} \underline{a} \underline{u} - \frac{1}{2} \left(\underline{R} \underline{a} \right) \cdot \left(\underline{R} \underline{u} \right) \right) = 0. \tag{28}
 \end{aligned}$$

The diagonal multiplication operators are self-adjoint with respect to \underline{M} , i.e. $\underline{M} \underline{a} = \underline{a} \underline{M}$, and $\underline{M} \underline{u} = \underline{u} \underline{M}$. Thus, (28) is

$$\begin{aligned}
 & \frac{\Delta x_k}{2} \underline{\varphi}^{k,T} \underline{M} \partial_t \underline{u} - \frac{1}{2} \underline{\varphi}^{k,T} \underline{D}^T \underline{M} \underline{a} \underline{u} + \frac{1}{2} \underline{\varphi}^{k,T} \underline{a} \underline{M} \underline{D} \underline{u} + \frac{1}{2} \underline{\varphi}^{k,T} \underline{u} \underline{M} \underline{D} \underline{a} \\
 & + \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num}} - \frac{1}{2} \left(\underline{R} \underline{a} \right) \cdot \left(\underline{R} \underline{u} \right) \right) = 0, \tag{29}
 \end{aligned}$$

or with boundary nodes included

$$\begin{aligned}
 & \frac{\Delta x_k}{2} \underline{\varphi}^{k,T} \underline{M} \partial_t \underline{u} - \frac{1}{2} \underline{\varphi}^{k,T} \underline{D}^T \underline{M} \underline{a} \underline{u} + \frac{1}{2} \underline{\varphi}^{k,T} \underline{a} \underline{M} \underline{D} \underline{u} \\
 & + \frac{1}{2} \underline{\varphi}^{k,T} \underline{u} \underline{M} \underline{D} \underline{a} + \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num}} - \frac{1}{2} \underline{R} \underline{a} \underline{u} \right) = 0. \tag{30}
 \end{aligned}$$

The error equations will be derived using both semidiscretisations. Before starting with the Gauß–Lobatto–Legendre case in the next Sect. 4, we shortly repeat for clarification again the notation which will be used in this paper in Table 1.

4 Error Behaviour Using Gauß–Lobatto Nodes

In this section, Gauß–Lobatto–Legendre nodes will be used in the discretisation, resulting in diagonal norm SBP operators including the boundary nodes. In this case, multiplication and restriction to the boundary commute and the interpolated speed $a(x)$ is automatically continuous. Before starting our investigation in this section, we will briefly summarize our final results for both cases (Gauß–Lobatto–Legendre and Gauß–Legendre).

Result 4.1 $\eta(t)$ is a factor which depends on ε_1 , the values of a and a' . If there exists a positive constant δ , such that the mean value of $\eta(t)$ can be bounded from below, then there exists a constant C such that the errors $\varepsilon_1^k(t)$ of (21) satisfy the inequality

$$\|\varepsilon_1(t)\|_N \leq \frac{1 - \exp(-\delta t)}{\delta} C,$$

in the discrete norm $\|\cdot\|_N$. The total error is bounded in time.

In the following, we will derive the exact conditions when the above inequality is fulfilled and specify in detail what factors play a key role in the definition of η and δ . We outline the steps of our analysis. All steps of the investigation in Sects. 4 and 5 are almost analogous except that in step 5 we have to consider the different flux functions (12)–(17) in our investigation.⁶ The main steps are the following:

⁶ We have an additional error term in Sect. 5, but this does not change the major steps of the study.

Table 1 Summary of the notations used in this article

Notation	Interpretation
u	is the solution of (1)
U	is the spatial approximation of u given by (5)
\underline{u}	are the coefficients of U , evaluated at the interpolation/quadrature nodes
\underline{D}	is the discrete derivative matrix
\underline{R}	is the restriction operator performing interpolation to the boundary
\underline{M}	is the mass/norm matrix
$\langle \cdot, \cdot \rangle$	is the usual L^2 scalar product
$\ \cdot \ $	is the norm induced by the L^2 scalar product
$\langle \cdot, \cdot \rangle_N$	is the discrete scalar product given by (6)
$\ \cdot \ _N$	is the norm induced by the discrete scalar product from above
\mathbb{I}^N	is the interpolation operator
$P_{N-1}^m(u)$	is the orthogonal projection of u onto $\mathbb{P}^{N-1}(-1, 1)$ using the inner product of $H^m(-1, 1)$ (See also “Appendix”)
$E^k = u^k - U^k$	is the total error in the k th element
$\varepsilon_1^k := \mathbb{I}^N(u^k) - U^k$	is the difference between interpolation and spatial approximation in the k th element
$\varepsilon_p^k = u^k - \mathbb{I}^N(u^k)$	is the interpolation error which decays spectrally fast

1. We derive an error equation for ε_1^k of (21) by inserting the error E^k into the continuous equation (20) for every element.
2. By adding zero in a suitable way, we are able to split the equations into a continuous and a discrete part.
3. We add both parts for every element and get the error behaviour for the total domain.
4. We estimate the continuous terms and get an inequality for the error ε_1 in the discrete norms.
5. We split the terms with the numerical fluxes. In the Gauss–Legendre case (Sect. 5), we have to be careful with respect to the used implementation of the numerical fluxes.
6. We estimate the long time error behaviour under some assumptions.

In the following, the error equation for $\varepsilon_1^k = \mathbb{I}^N(u^k) - U^k$ will be derived. Starting by considering Gauß–Lobatto–Legendre nodes in our semidiscretisation and putting $u = \mathbb{I}^N(u^k) + \varepsilon_p^k$ into (20) yields

$$\begin{aligned} & \frac{\Delta x_k}{2} \left\langle \partial_t \mathbb{I}^N(u^k), \varphi^k \right\rangle + \frac{1}{2} \left(a^k \mathbb{I}^N(u^k) \varphi^k \Big|_{-1}^1 - \left\langle a^k \mathbb{I}^N(u^k), \partial_\xi \varphi^k \right\rangle \right. \\ & \quad \left. + \left\langle a^k \partial_\xi \mathbb{I}^N(u^k), \varphi^k \right\rangle + \left\langle \mathbb{I}^N(u^k) \partial_\xi a^k, \varphi^k \right\rangle \right) \\ & = -\frac{\Delta x_k}{2} \left\langle \partial_t \varepsilon_p^k, \varphi^k \right\rangle + \frac{1}{2} \left\langle a^k \varepsilon_p^k, \partial_\xi \varphi^k \right\rangle - \frac{1}{2} \left\langle a^k \partial_\xi \varepsilon_p^k, \varphi^k \right\rangle - \frac{1}{2} \left\langle \varepsilon_p^k \partial_\xi a^k, \varphi^k \right\rangle, \end{aligned}$$

where $\varphi^k \in \mathbb{P}^N$ is a polynomial test function. For Gauß–Lobatto–Legendre nodes, $a^k \varepsilon_p^k = 0$ at the endpoints, since the interpolant is equal to the solution there. Thus, $a^k \varepsilon_p^k \varphi^k \Big|_{-1}^1 = 0$.

Using integration-by-parts for $\left\langle a^k \varepsilon_p^k, \partial_\xi \varphi^k \right\rangle$ yields

$$\begin{aligned} & \frac{\Delta x_k}{2} \left\langle \partial_t \mathbb{I}^N(u^k), \varphi^k \right\rangle + \frac{1}{2} \left(a^k \mathbb{I}^N(u^k) \varphi^k \Big|_{-1}^1 - \left\langle a^k \mathbb{I}^N(u^k), \partial_\xi \varphi^k \right\rangle \right. \\ & \quad \left. + \left\langle a^k \partial_\xi \mathbb{I}^N(u^k), \varphi^k \right\rangle + \left\langle \mathbb{I}^N(u^k) \partial_\xi a^k, \varphi^k \right\rangle \right) \\ & = -\frac{\Delta x_k}{2} \left\langle \partial_t \varepsilon_p^k, \varphi^k \right\rangle - \frac{1}{2} \left\langle \partial_\xi (a^k \varepsilon_p^k), \varphi^k \right\rangle - \frac{1}{2} \left\langle a^k \partial_\xi \varepsilon_p^k, \varphi^k \right\rangle - \frac{1}{2} \left\langle \varepsilon_p^k \partial_\xi a^k, \varphi^k \right\rangle. \end{aligned} \tag{31}$$

We have to transfer the continuous scalar product from (31) to the discrete ones. Therefore, we are following the ideas from [31], add zero to the above equation and rearrange these terms. We will explain this for the first term on the left side of (31) in detail. The third to fifth terms on the left side are handled analogously and details can be found in the ‘‘Appendix’’. Applying the interpolation operator together with discrete norms results in

$$\begin{aligned} \left\langle \partial_t \mathbb{I}^N(u^k), \varphi^k \right\rangle & = \left(\partial_t \mathbb{I}^N(u^k), \underline{\varphi}^k \right)_N \\ & \quad + \left\{ \left\langle \partial_t \mathbb{I}^N(u^k), \varphi^k \right\rangle - \left(\partial_t \mathbb{I}^N(u^k), \underline{\varphi}^k \right)_N \right\}, \end{aligned} \tag{32}$$

Now, we are introducing in the factor Q in the above equation which measures the projection error of a polynomial of degree N to a polynomial of degree $N - 1$. We can rewrite (32) as

$$\left\langle \partial_t \mathbb{I}^N(u^k), \varphi^k \right\rangle = \left(\mathbb{I}^N(u^k), \underline{\varphi}^k \right)_N + \left\{ \left\langle Q(u^k), \varphi^k \right\rangle - \left(\underline{Q}(u^k), \underline{\varphi}^k \right)_N \right\}$$

where $Q(u^k) := \partial_t (\mathbb{I}^N(u^k) - P_{N-1}^m(\mathbb{I}^N(u^k)))$ and P_{N-1}^m is the orthogonal projection⁷ of u onto \mathbb{P}^{N-1} using the inner product of $H^m(e^k)$. We get similar factors ($Q_1 - Q_3$) for the other three terms. Since u and a are bounded, also all of these values have to be bounded. Finally, the values of the interpolation polynomial at the boundaries of the element (-1 and 1) can be approximated by a limitation process from the left side $\mathbb{I}^N(u^k)^-$ and right side $\mathbb{I}^N(u^k)^+$. To simplify the notation, let

$$\begin{aligned} & \underline{f}^{\text{num},k} \left(\mathbb{I}^N(u^k)^-, \mathbb{I}^N(u^k)^+ \right) \\ & := \left(f^{\text{num}} \left(\mathbb{I}_R^N(u^{k-1}), \mathbb{I}_L^N(u^k) \right), f^{\text{num}} \left(\mathbb{I}_R^N(u^k), \mathbb{I}_L^N(u^{k+1}) \right) \right)^T. \end{aligned} \tag{33}$$

For boundary points included, the interpolation is continuous (because the exact solution u is continuous) and all numerical fluxes are exactly the products of the interpolation and the coefficient values. One obtains

$$\frac{1}{2} a^k \mathbb{I}^N(u^k) \varphi^k \Big|_{-1}^1 = \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left(\mathbb{I}^N(u^k)^-, \mathbb{I}^N(u^k)^+ \right) - \frac{1}{2} \underline{R} a^k u \right). \tag{34}$$

Using the above investigation and putting (32)–(34) in (31) results in

$$\begin{aligned} & \frac{\Delta x_k}{2} \left(\partial_t \mathbb{I}^N(u^k), \underline{\varphi}^k \right)_N + \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left(\mathbb{I}^N(u^k)^-, \mathbb{I}^N(u^k)^+ \right) - \frac{1}{2} \underline{R} a^k u \right) \\ & \quad - \frac{1}{2} \left(\underline{a}^k \mathbb{I}^N(u^k), \partial_\xi \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\underline{a}^k \partial_\xi \mathbb{I}^N(u^k), \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\underline{\mathbb{I}}^N(u^k) \partial_\xi \underline{a}^k, \underline{\varphi}^k \right)_N \end{aligned}$$

⁷ More details can be found in the ‘‘Appendix’’.

$$\begin{aligned}
 &= + \frac{\Delta x_k}{2} \langle T^k(u), \varphi^k \rangle + \frac{\Delta x_k}{4} \left((\underline{Q}(u^k), \underline{\varphi}^k)_N - (\underline{Q}_1(u^k), \partial_x \underline{\varphi}^k)_N \right. \\
 &\quad \left. + (\underline{a}^k \underline{Q}_2(u^k), \underline{\varphi}^k)_N + (\underline{Q}_3(u^k) \partial_x \underline{a}^k, \underline{\varphi}^k)_N \right) + \frac{\Delta x_k}{4} \langle \underline{Q}_1(u^k), \partial_x \varphi^k \rangle, \tag{35}
 \end{aligned}$$

with

$$\begin{aligned}
 T^k(u) := & - \left\{ \partial_t \varepsilon_p^k + \frac{1}{2} \left(\partial_x (a^k \varepsilon_p^k) + \varepsilon_p^k \partial_x a^k + a^k \partial_x \varepsilon_p^k \right) \right. \\
 & \left. + \frac{1}{2} \left(Q(u^k) + a^k Q_2(u^k) + (Q_3(u^k) \partial_x a^k) \right) \right\}. \tag{36}
 \end{aligned}$$

Here, in definition (36) we have again the derivatives in x since we make the term independent from the transformation. Therefore, we have in (35) a $\frac{\Delta x_k}{2}$ in the T^k terms.

By (24), the interpolation error ε_p^k converges in N to zero, if $m > 1$ and the Sobolev norm of the solution is uniformly bounded in time.⁸ Equation (30) is subtracted from (35) and with $\varepsilon_1^k = \mathbb{I}^N(u^k) - U^k$ one obtains

$$\begin{aligned}
 &\frac{\Delta x_k}{2} \left(\partial_t \varepsilon_1^k, \underline{\varphi}^k \right)_N + \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(f^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \underline{R} a^k \varepsilon_1^k \right) \\
 &\quad - \frac{1}{2} \left(\underline{a}^k \varepsilon_1^k, \partial_\xi \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\underline{a}^k \partial_\xi \varepsilon_1^k, \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\underline{\varepsilon}_1^k \partial_\xi a^k, \underline{\varphi}^k \right)_N \\
 &= + \frac{\Delta x_k}{2} \langle T^k(u), \varphi^k \rangle + \frac{\Delta x_k}{4} \langle \underline{Q}_1(u^k), \partial_x \varphi^k \rangle + \frac{\Delta x_k}{4} \left((\underline{Q}(u^k), \underline{\varphi}^k)_N \right. \\
 &\quad \left. - (\underline{Q}_1(u^k), \partial_x \underline{\varphi}^k)_N + (\underline{a}^k \underline{Q}_2(u^k), \underline{\varphi}^k)_N + (\underline{Q}_3(u^k) \partial_x a^k, \underline{\varphi}^k)_N \right).
 \end{aligned}$$

Putting $\varphi^k = \varepsilon_1^k$ results in the energy equation

$$\begin{aligned}
 &\frac{\Delta x_k}{4} \frac{d}{dt} \|\varepsilon_1^k\|_N^2 + \underline{\varepsilon}_1^{k,T} \underline{R}^T \underline{B} \left(f^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \underline{R} a^k \varepsilon_1^k \right) \\
 &\quad - \frac{1}{2} \left(\underline{a}^k \varepsilon_1^k, \partial_\xi \varepsilon_1^k \right)_N + \frac{1}{2} \left(\underline{a}^k \partial_\xi \varepsilon_1^k, \varepsilon_1^k \right)_N + \frac{1}{2} \left(\underline{\varepsilon}_1^k \partial_x a^k, \varepsilon_1^k \right)_N \\
 &= + \frac{\Delta x_k}{2} \langle T^k(u), \varepsilon_1^k \rangle + \frac{\Delta x_k}{4} \langle \underline{Q}_1(u^k), \partial_x \varepsilon_1^k \rangle \\
 &\quad + \frac{\Delta x_k}{4} \left((\underline{Q}(u^k), \varepsilon_1^k)_N - (\underline{Q}_1(u^k), \partial_x \varepsilon_1^k)_N + (\underline{a}^k \underline{Q}_2(u^k), \varepsilon_1^k)_N \right. \\
 &\quad \left. + (\underline{Q}_3(u^k) \partial_x a^k, \varepsilon_1^k)_N \right). \tag{37}
 \end{aligned}$$

Since $\underline{M}^T = \underline{M}$, we get

$$\begin{aligned}
 \frac{1}{2} \left(\underline{a}^k \varepsilon_1^k, \partial_\xi \varepsilon_1^k \right)_N &= \frac{1}{2} \underline{\varepsilon}_1^{k,T} \underline{a}^{k,T} \underline{M} \underline{D} \varepsilon_1^k, \\
 \frac{1}{2} \left(\partial_\xi \varepsilon_1^k, \underline{a}^k \varepsilon_1^k \right)_N &= \frac{1}{2} \underline{\varepsilon}_1^{k,T} \underline{D}^T \underline{M} \underline{a}^k \varepsilon_1^k = \frac{1}{2} \underline{\varepsilon}_1^{k,T} \underline{a}^{k,T} \underline{M} \underline{D} \varepsilon_1^k, \tag{38}
 \end{aligned}$$

⁸ Therefore, we need the initial and boundary conditions in the model problem (1).

and one obtains in (37)

$$\begin{aligned} & \frac{\Delta x_k}{4} \frac{d}{dt} \|\varepsilon_1^k\|_N^2 + \varepsilon_1^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \underline{Ra}^k \varepsilon_1^k \right) \\ & + \frac{\Delta x_k}{4} \left(\underline{\varepsilon}_1^k \underline{\varepsilon}_1^k, \partial_x \underline{a}^k \right)_N = \frac{\Delta x_k}{2} \left\langle T^k(u), \varepsilon_1^k \right\rangle + \frac{\Delta x_k}{4} \left\langle \underline{Q}_1(u^k), \partial_x \varepsilon_1^k \right\rangle \\ & + \frac{\Delta x_k}{4} \left((\underline{Q}(u^k), \varepsilon_1^k)_N - (\underline{Q}_1(u^k), \partial_x \varepsilon_1^k)_N + (\underline{a}^k \underline{Q}_2(u^k), \varepsilon_1^k)_N \right. \\ & \left. + (\underline{Q}_3(u^k) \partial_x \underline{a}^k, \varepsilon_1^k)_N \right). \end{aligned} \tag{39}$$

Summing this up over all elements and by defining the numerical flux of the error as $\varepsilon_1^{\text{num},k} := \underline{f}^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right)$, the global energy of the error is

$$\begin{aligned} & \frac{1}{2} \frac{d}{dt} \sum_{k=1}^K \frac{\Delta x_k}{2} \|\varepsilon_1^k\|_N^2 + \sum_{k=1}^K \varepsilon_1^{k,T} \underline{R}^T \underline{B} \left(\varepsilon_1^{\text{num},k} - \frac{1}{2} \underline{Ra}^k \varepsilon_1^k \right) \\ & + \frac{1}{2} \sum_{k=1}^K \frac{\Delta x_k}{2} \left(\underline{\varepsilon}_1^k \underline{\varepsilon}_1^k, \partial_x \underline{a}^k \right)_N \\ & = \sum_{k=1}^K \frac{\Delta x_k}{2} \left\langle T^k(u), \varepsilon_1^k \right\rangle + \sum_{k=1}^K \frac{\Delta x_k}{4} \left\langle \underline{Q}_1(u^k), \partial_x \varepsilon_1^k \right\rangle - \sum_{k=1}^K \frac{\Delta x_k}{4} \left\langle \underline{Q}_1(u^k), \partial_x \varepsilon_1^k \right\rangle_N \\ & + \sum_{k=1}^K \frac{\Delta x_k}{4} \left((\underline{Q}(u^k), \varepsilon_1^k)_N + (\underline{a}^k \underline{Q}_2(u^k), \varepsilon_1^k)_N + (\underline{Q}_3(u^k) \partial_x \underline{a}^k, \varepsilon_1^k)_N \right). \end{aligned} \tag{40}$$

The right-hand side of (40) will be estimated using the Cauchy–Schwarz inequality. For example (the others terms are handled similarly),

$$\sum_{k=1}^K \frac{\Delta x_k}{2} \left\langle T^k(u), \varepsilon_1^k \right\rangle \leq \sqrt{\sum_{k=1}^K \frac{\Delta x_k}{2} \|T^k(u)\|^2} \sqrt{\sum_{k=1}^K \frac{\Delta x_k}{2} \|\varepsilon_1^k\|^2}, \tag{41}$$

$$\sum_{k=1}^K \frac{\Delta x_k}{4} \left(\underline{Q}_1(u^k), \partial_x \varepsilon_1^k \right)_N \leq \frac{1}{2} \sqrt{\sum_{k=1}^K \frac{\Delta x_k}{2} \|\underline{Q}_1(u^k)\|_N^2} \sqrt{\sum_{k=1}^K \frac{\Delta x_k}{2} \|\partial_x \varepsilon_1^k\|_N^2}, \tag{42}$$

Using an estimation for the differential operator ∂_x and the fact that $\varepsilon_1 \in \mathbb{P}^N$, it is $\|\partial_x \varepsilon_1^k\|_N^2 \leq c_1 N^2 \|\varepsilon_1^k\|_N^2$ with a positive constant c_1 . This is due to the fact that all norms are equivalent and we can estimate with a Markov–Bernstein type inequality, see [13]. The estimation is used for example in (42). An alternative approach would have been to use the summation-by-parts property (7) and estimate analogously.

With the global norm over all elements and the equivalence between the continuous and discrete norms, we obtain

$$\begin{aligned} & \frac{1}{2} \frac{d}{dt} \sum_{k=1}^K \frac{\Delta x_k}{2} \|\varepsilon_1^k\|_N^2 + \sum_{k=1}^K \varepsilon_1^{k,T} \underline{R}^T \underline{B} \left(\varepsilon_1^{\text{num},k} - \frac{1}{2} \underline{Ra}^k \varepsilon_1^k \right) \\ & + \frac{1}{2} \sum_{k=1}^K \frac{\Delta x_k}{2} \left(\underline{\varepsilon}_1^k \underline{\varepsilon}_1^k, \partial_x \underline{a}^k \right)_N \leq \left\{ c \|T\| + \frac{cN}{2} \|\underline{Q}_1\| + \frac{1}{2} \left(\|\underline{Q}\|_N + N \tilde{c}_1 \|\underline{Q}_1\|_N \right. \right. \\ & \left. \left. + \|\underline{a} \underline{Q}_2\|_N + \|\underline{Q}_3 \partial_x \underline{a}\|_N \right) \right\} \|\varepsilon_1\|_N \equiv \hat{\mathbb{E}}(t, N) \|\varepsilon_1\|_N \end{aligned} \tag{43}$$

Applying the same approach like in [20] and splitting the sum into three parts (one for the left physical boundary, one for the right physical boundary and a sum over the internal element endpoints), it is

$$\begin{aligned} & \sum_{k=1}^K \varepsilon_1^{k,T} \underline{\underline{R}}^T \underline{\underline{B}} \left(\varepsilon_1^{\text{num},k} - \frac{1}{2} \underline{\underline{R}} a^k \varepsilon_1^k \right) \\ &= \sum_{k=1}^K \varepsilon_1^{k,T} \underline{\underline{R}}^T \underline{\underline{B}} \left(f_L^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \underline{\underline{R}} a^k \varepsilon_1^k \right) \\ &= -\mathbf{E}_L^1 \left(f_L^{\text{num},1} - \frac{1}{2} a_L^1 \mathbf{E}_L^1 \right) + \sum_{k=2}^K \left(f_L^{\text{num},k} - \frac{1}{2} a_R^{k-1} \left(\mathbf{E}_R^{k-1} + \mathbf{E}_L^k \right) \right) \\ & \quad \left(\mathbf{E}_R^{k-1} - \mathbf{E}_L^k \right) + \mathbf{E}_R^K \left(f_R^{\text{num},K} - \frac{1}{2} a_R^K \mathbf{E}_R^K \right). \end{aligned}$$

Here, \mathbf{E}_i^k ($i = L, R; k = 1, \dots, K$) represents the error ε_1^k at the the position in the elements, and (to shorten the notation) $f_L^{\text{num},k} := f^{\text{num},k}(\mathbf{E}_R^{k-1}, \mathbf{E}_L^k)$, $f_L^{\text{num},1} := f^{\text{num},1}(0, \mathbf{E}_L^1)$ and $f_R^{\text{num},K} := f^{\text{num},1}(\mathbf{E}_R^K, 0)$. The external states for the physical boundary contributions are zero, because $\mathbb{I}^N(u^1) = g$ at the left boundary and the external state for U^1 is set to g . At the right boundary, where the upwind numerical flux is used, it doesn't matter what the external state is, since its coefficient in the numerical flux is zero. One gets for the inner element with

$$\begin{aligned} \llbracket \mathbf{E}^k \rrbracket &= \mathbf{E}_R^{k-1} - \mathbf{E}_L^k \\ \sum_{k=2}^K \left(f_L^{\text{num},k} - \frac{1}{2} a_R^{k-1} \left(\mathbf{E}_R^{k-1} + \mathbf{E}_L^k \right) \right) \left(\mathbf{E}_R^{k-1} - \mathbf{E}_L^k \right) &= \sum_{k=2}^K \frac{\sigma a_R^{k-1}}{2} \left(\llbracket \mathbf{E}^k \rrbracket \right)^2 \geq 0, \\ & \text{with } \begin{cases} \sigma = 0 & \text{central flux,} \\ \sigma = 1 & \text{upwind flux.} \end{cases} \end{aligned}$$

For the left and right boundaries, it is finally

$$\begin{aligned} \text{left: } & -\mathbf{E}_L^1 \left(f_L^{\text{num},1} - \frac{1}{2} a_L^1 \mathbf{E}_L^1 \right) = \frac{\sigma a_L^1}{2} \left(\mathbf{E}_L^1 \right)^2, \\ \text{right: } & \mathbf{E}_R^K \left(f_R^{\text{num},K} - \frac{1}{2} a_R^K \mathbf{E}_R^K \right) = \frac{\sigma a_R^K}{2} \left(\mathbf{E}_R^K \right)^2. \end{aligned}$$

Therefore, the energy growth rate is bounded by

$$\begin{aligned} & \frac{1}{2} \frac{d}{dt} \|\varepsilon_1\|_N^2 + \underbrace{\frac{\sigma}{2} \left(a_R^K \left(\mathbf{E}_R^K \right)^2 + a_L^1 \left(\mathbf{E}_L^1 \right)^2 \right) + \frac{\sigma}{2} \sum_{k=2}^K a_R^{k-1} \left(\llbracket \mathbf{E}^k \rrbracket \right)^2}_{BTs} \\ & + \underbrace{\frac{1}{2} \sum_{k=1}^K \frac{\Delta x_k}{2} \left(\underline{\underline{\varepsilon}}_1^k \underline{\underline{\varepsilon}}_1^k, \partial_x \underline{\underline{a}}^k \right)_N}_{Int_d} \leq \mathbb{E}(t, N) \|\varepsilon_1\|_N. \end{aligned} \tag{44}$$

It is $BTs \geq 0$. If $Int_d \geq 0$, then (44) has the same form as in in [27] and one may estimate/bound analogously to [20,27] the error in time. The \mathbb{E} term depends also on N , but this has no influence in the estimation here. We rewrite (44) as

$$\frac{d}{dt} \|\varepsilon_1\|_N + \underbrace{\frac{BTs + Int_d}{\|\varepsilon_1\|_N^2}}_{\eta(t)} \|\varepsilon_1\|_N \leq \mathbb{E}(t). \tag{45}$$

Like it was described in [27], it is assumed that the mean value of $\eta(t)$ over any finite time interval is bounded by a positive constant δ_0 from below. This means that $\bar{\eta} \geq \delta_0 > 0$. Under the assumption for u , the right hand side $\mathbb{E}(t, N)$ is also bounded in time and one can put $\max_{s \in [0, \infty)} \mathbb{E}(s, N) \leq C_1 < \infty$. Applying these facts in (45) and integrating over time, the following inequality for the error is obtained

$$\|\varepsilon_1(t)\|_N \leq \frac{1 - \exp(-\delta_0 t)}{\delta_0} C_1, \tag{46}$$

see [27, Lemma 2.3] for details.

Remark 4.1 The term Int_d is a crucial factor. If $\partial_x a^k > 0$, one may estimate the left side of (45) using the minimum of the discrete values of a . Then, $Int_d \geq \frac{1}{2} \min\{\partial_x a^k\} \|\varepsilon_1\|_N^2 > 0$ and the above assumption on η is inevitably fulfilled.

Simultaneously, the term Int_d can also destroy the error boundedness if the derivatives of a are negative. It depends then on the sum of BTs and Int_d . The upwind fluxes can therefore rescue the error boundedness (46) whereas applying the central flux ($\sigma = 0$) will contribute to an unlimited growth of the error. We demonstrate this in some examples in Sect. 6 and make a first analytical estimation in Sect. 6.5.

5 Error Behaviour Using Gauß–Legendre Nodes

Here, Gauß–Legendre nodes are used, yielding diagonal norm SBP operators not including the boundary nodes, contrary to Gauß–Lobatto–Legendre nodes discussed in the previous Sect. 4. Thus, care has to be taken of several potential problems. Firstly, the restriction to the boundary and multiplication do not commute. Secondly, the numerical flux functions (12)–(16) are now different from each other and have to be considered separately.

However, even if there are more problems, there are also some reasons to consider Gauß–Legendre nodes. Indeed, Gauß–Legendre nodes have a higher order of accuracy in the quadrature and as investigated in [31], for the linear advection equation with constant coefficients using Gauß–Legendre nodes, the error reaches always faster its asymptotic value. Moreover, this asymptotic value is lower than the corresponding one using Gauß–Lobatto–Legendre nodes. Furthermore, the influence of the numerical fluxes is not that essential. Using $u = \mathbb{I}^N(u^k) + \varepsilon_p^k$ in (20), where the terms are rearranged similar to Sect. 4, we get analogously an equation similar to (35) except an additional error term due to the fact that boundary terms are not included. We obtain

$$\begin{aligned} & \frac{\Delta x_k}{2} \left(\partial_t \mathbb{I}^N(u^k), \underline{\varphi}^k \right)_N + \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{num,k} \left(\mathbb{I}^N(u^k)^-, \mathbb{I}^N(u^k)^+ \right) - \frac{1}{2} \left(\underline{Ra}^k \right) \cdot \left(\underline{Ru} \right) \right) \\ & + \underbrace{\left(\frac{1}{2} a^k \mathbb{I}^N(u^k) \varphi^k \Big|_{-1}^1 - \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{num,k} \left(\mathbb{I}^N(u^k)^-, \mathbb{I}^N(u^k)^+ \right) - \frac{1}{2} \left(\underline{Ra}^k \right) \cdot \left(\underline{Ru} \right) \right) \right)}_{=: \varepsilon_2^k(a^k)} \\ & - \frac{1}{2} \left(\underline{a}^k \mathbb{I}^N(u^k), \partial_\xi \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\partial_\xi \mathbb{I}^N(u^k), \underline{a}^k \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\partial_\xi a^k, \mathbb{I}^N(u^k) \varphi^k \right)_N \end{aligned}$$

$$\begin{aligned}
 &= \frac{\Delta x_k}{2} \left\langle \hat{T}^k(u^k), \varphi^k \right\rangle + \frac{\Delta x_k}{4} \left\langle Q_1(u^k), \partial_x \varphi^k \right\rangle \\
 &+ \frac{\Delta x_k}{4} \left\{ \left(\underline{Q}(u^k), \underline{\varphi}^k \right)_N - \left(\underline{Q}_1(u^k), \partial_x \underline{\varphi}^k \right)_N + \left(\underline{Q}_2(u^k), \underline{a}^k \underline{\varphi}^k \right)_N \right. \\
 &\left. + \left(\partial_x \underline{a}^k, \underline{Q}_3(u^k) \varphi^k \right)_N \right\} \tag{47}
 \end{aligned}$$

with

$$\begin{aligned}
 \hat{T}^k(u^k) := & - \left\{ \partial_t \varepsilon_p^k + \frac{1}{2} \left(\partial_x \left(a^k \varepsilon_p^k \right) + \varepsilon_p^k \partial_x a^k \partial_x \varepsilon_p^k \right) \right. \\
 & \left. + \frac{1}{2} \left(Q(u^k) + a^k Q_2(u^k) + Q_3(u^k) \partial_x a^k \right) \right\}.
 \end{aligned}$$

Following the approach from Sect. 4 we get the estimate⁹

$$\begin{aligned}
 &\frac{1}{2} \frac{d}{dt} \sum_{k=1}^K \frac{\Delta x_k}{2} \|\varepsilon_1^k\|_N^2 + \sum_{k=1}^K \varepsilon_1^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \left(\underline{R}a^k \right) \cdot \left(\underline{R}\varepsilon_1^k \right) \right) \\
 &+ \underbrace{\frac{1}{2} \sum_{k=1}^K \frac{\Delta x_k}{2} \left(\partial_x \underline{a}^k, \underline{\varepsilon}_1^k \varepsilon_1^k \right)_N}_{Int_d} \leq - \underbrace{\frac{1}{2} \sum_{k=1}^K \frac{\Delta x_k}{2} \varepsilon_2^k(a^k)}_{:=\Theta_2} \\
 &+ \underbrace{\left\{ c_1 \|T\| + \frac{cN}{2} \|Q_1\| + \frac{1}{2} \left(\|Q\|_N + N \|Q_1\|_N + \|a Q_2\|_N + \|Q_3 \partial_x a\|_N \right) \right\}}_{:=\hat{\mu}_G(t,N)} \|\varepsilon_1\|_N. \tag{48}
 \end{aligned}$$

Remark 5.1 The sum of the terms ε_2^k depends on a and the interpolation of the flux functions. It is given by the formula

$$\begin{aligned}
 \varepsilon_2^k(a^k) := & \left(\frac{1}{2} a^k \varepsilon_1^k \mathbb{I}^N(u^k) \right) \Big|_{-1}^1 - \varepsilon_1^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left(\mathbb{I}^N(u^k)^-, \mathbb{I}^N(u^k)^+ \right) \right. \\
 & \left. - \frac{1}{2} \left(\underline{R}a^k \right) \left(\underline{R}u \right) \right).
 \end{aligned}$$

Using Gauß-Lobatto nodes and an upwind flux, these terms are zero, see Sect. 4. If the sum over all elements is positive, i.e. $\Theta_2 \geq 0$, then this term decreases the upper bound of the error ε_1 .

If $\Theta_2 < 0$, then it increases the total error. The error depends on u , a and the jumps between interfaces. Under the assumption that u is continuous, Θ_2 will be bounded from below, resulting in an upper bound on the right side. Nevertheless, this makes it hard to study the behaviour of the total error analytically.

We consider the first line of (48), especially the term

$$\sum_{k=1}^K \varepsilon_1^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \left(\underline{R}a^k \right) \cdot \left(\underline{R}\varepsilon_1^k \right) \right)$$

⁹ Details of main steps can also be found in the ‘‘Appendix’’.

with different flux functions (12)–(16). In [36], different assumptions on a have already been formulated for stability and conservation of the numerical schemes. First, we consider the general case. One may recognise the problems which arise by considering variable coefficients in the model problem (1). Following this, we will formulate analogous assumptions to [36, Theorem 3.4] and proceed with our analysis.

We split the sum in three terms (one for the left physical boundary, one for the right physical boundary and a sum over the internal element endpoints), and we get

$$\begin{aligned} & \sum_{k=1}^K \varepsilon_1^{k,T} \underline{\mathbf{R}}^T \underline{\mathbf{B}} \left(\underline{f}^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \left(\underline{\mathbf{R}} a^k \right) \cdot \left(\underline{\mathbf{R}} \varepsilon_1^k \right) \right) \\ &= -\mathbf{E}_L^1 \left(f_L^{\text{num},1} - \frac{1}{2} a_L^1 \mathbf{E}_L^1 \right) + \sum_{k=2}^K \left(f_L^{\text{num},k} \left(\mathbf{E}_R^{k-1} - \mathbf{E}_L^k \right) \right. \\ & \quad \left. - \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) \right) + \mathbf{E}_R^K \left(f_R^{\text{num},K} - \frac{1}{2} a_R^K \mathbf{E}_R^K \right). \end{aligned}$$

We describe with \mathbf{E}_i ($i = L, R$) the approximation error ε_1 , the indices give the position in the elements, $f_L^{\text{num},k} := f^{\text{num},k} \left(\mathbf{E}_R^{k-1}, \mathbf{E}_L^k \right)$, $f_L^{\text{num},1} := f^{\text{num},1} \left(0, \mathbf{E}_L^1 \right)$ and $f_R^{\text{num},K} := f^{\text{num},1} \left(\mathbf{E}_R^K, 0 \right)$. The external states for the physical boundary contributions are zero, because $\mathbb{I}^N(u)^1 = g$ at the left boundary and the external state for U^1 is set to g . The selection of the numerical flux functions (12)–(16) has an influence on the behaviours of the errors and we have to be careful in our study. If the interpolation of a is exact and a is continuous over the inter-element boundaries, then the influence of the numerical fluxes can be simplified essentially and we are able to analyse the long time error behaviours. We will formulate this in detail for the first flux under consideration, the edge based central flux (12).

– Edge based central flux $f^{\text{num}}(u_-, u_+) = a(x) \frac{u_- + u_+}{2}$: We get for the terms in the sum

$$\begin{aligned} & \frac{1}{2} a^k(x_L) \left(\mathbf{E}_R^{k-1} + \mathbf{E}_L^k \right) \left(\mathbf{E}_R^{k-1} - \mathbf{E}_L^k \right) - \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) \\ &= \frac{1}{2} a^k(x_L) \left(\left(\mathbf{E}_R^{k-1} \right)^2 - \left(\mathbf{E}_L^k \right)^2 \right) - \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) \\ &= \frac{1}{2} \left(\mathbf{E}_R^{k-1} \right)^2 \left(a^k(x_L) - a_R^{k-1} \right) + \frac{1}{2} \left(\mathbf{E}_L^k \right)^2 \left(a_L^k - a(x_L) \right) = 0. \end{aligned}$$

If the interpolation of a is exact and a is continuous, the brackets of a will be zero, because $a^k(x_L) = a_L^k = a^{k-1}(x_R) = a_R^{k-1}$. If this is not the case, we get additional terms that can be positive or negative depending on brackets. On the boundaries, one obtains

$$\begin{aligned} \text{left: } & -\mathbf{E}_L^1 \left(f_L^{\text{num},1} - \frac{1}{2} a_L^1 \mathbf{E}_L^1 \right) = -\mathbf{E}_L^1 \left(\frac{a^1(x_L)}{2} \mathbf{E}_L^1 - \frac{a_L^1}{2} \mathbf{E}_L^1 \right) \\ &= \frac{1}{2} \left(\mathbf{E}_L^1 \right)^2 \left(a_L^1 - a^1(x_L) \right) = 0, \\ \text{right: } & \mathbf{E}_R^K \left(f_R^{\text{num},K} - \frac{1}{2} a_R^K \mathbf{E}_R^K \right) = \frac{1}{2} \left(\mathbf{E}_R^K \right)^2 \left(a^K(x_R) - a_R^K \right) = 0. \end{aligned}$$

– Using this approach, we get the following results where the details of the calculation can be found in the ‘‘Appendix’’:

Table 2 Error terms of the numerical fluxes

Fluxes	Interiour	Left	Right
Split central	0	0	0
Edge bases upwind	$\frac{1}{2} a_R^{k-1} \left[\left[E_R^{k-1} \right] \right]^2$	$\frac{1}{2} \left(\mathbf{E}_L^1 \right) a_L^1$	$\left(\mathbf{E}_R^k \right)^2 \left(\frac{a_R^K}{2} \right)$
Split upwind	$\frac{1}{2} a_R^{k-1} \left[\left[E_R^{k-1} \right] \right]^2$	$\frac{1}{2} \left(\mathbf{E}_L^1 \right) a_L^1$	$\left(\mathbf{E}_R^k \right)^2 \left(\frac{a_R^K}{2} \right)$

For the calculation of the split upwind flux, we apply the assumptions of the exactness of the interpolation and the continuity of a .

- Unsplit upwind flux $f^{\text{num}}(u_-, u_-) = (au)_-$.

Unfortunately, for the unsplit numerical fluxes (14), (17) we are not able to find such a simplification as above, since the restriction of the product can not be compared to the product of the restriction. This issue triggers also stability problems, see [36] for details. We formulate this now for the unsplit upwind flux as an example. It is:

$$\begin{aligned} & (a\mathbf{E})_R^{k-1} \left(\mathbf{E}_R^{k-1} - \mathbf{E}_L^k \right) - \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) \\ &= \frac{1}{2} \left(\left(2(a\mathbf{E})_R^{k-1} \mathbf{E}_R^{k-1} - a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 \right) - 2a_R^{k-1} \mathbf{E}_L^k \mathbf{E}_R^{k-1} + a_L^k \left(\mathbf{E}_L^k \right)^2 \right). \end{aligned}$$

Because of $(a\mathbf{E})_R^{k-1} \neq a_R^{k-1} \mathbf{E}_R^{k-1}$ in general, a further simplification is in this case not possible anymore. The following error bounds are **only** valid for the split numerical fluxes. Nevertheless, we test also the unsplit fluxes in the next section.

By comparison, one may recognise that the split upwind flux is equal the edge upwind flux and analogously for the central fluxes under assumptions. Using central fluxes leads to no additional terms in the inequality (48), whereas using upwind fluxes does. If the restrictions $a_{L/R}$ to the boundary are positive,¹⁰ all of these terms are positive. We reformulate the energy inequality (48) as

$$\begin{aligned} & \frac{1}{2} \frac{d}{dt} \|\varepsilon_1\|_N^2 + \underbrace{\frac{\sigma}{2} \left(a_R^K \left(\mathbf{E}_R^K \right)^2 + a_L^1 \left(\mathbf{E}_L^1 \right)^2 \right) + \frac{\sigma}{2} \sum_{k=2}^K a_R^{k-1} \left(\left[\mathbf{E}^k \right] \right)^2}_{BTs} + Int_d \\ & \leq \hat{\mathbb{E}}_G(t, N) \|\varepsilon_1\|_N - \Theta_2, \end{aligned} \tag{49}$$

where σ is zero (central flux) or one (upwind flux). The energy growth energy inequality (49) is similar to (48). The only difference is the term Θ_2 , which will yield a smaller upper bound under the condition $\Theta_2 \geq 0$. We follow the steps of Sect. 4 and get

$$\frac{d}{dt} \|\varepsilon_1\|_N + \underbrace{\frac{BTs + Int_d + \Theta_2}{\|\varepsilon_1\|_N^2}}_{\eta_G(t)} \|\varepsilon_1\|_N \leq \hat{\mathbb{E}}_G(t, N). \tag{50}$$

¹⁰ This assumption is already formulated in [36, Theorem 3.4] to guarantee stability and conservation of the numerical schemes.

We have to assume that we can bound the mean value of $\eta_G(t)$ by a positive constant δ_G from below. If already $Int_d + \Theta_2 > 0$, this is actually met without restrictions. So, using the central fluxes ($\sigma = 0$) does not yield to problems. Simultaneously, if $BTs + Int_d + \Theta_2$ overall is positive, the requirement on every $a_{L,R}^k$ to be non-negative can be weakened to make the estimations, but one should have in mind that the positivity of $a_{L,R}^k$ is a condition to prove stability. This means¹¹ $\overline{\eta_G}(t) \geq \delta_G > \delta_0 > 0$. Under the assumption of u , the right hand side $\hat{\mathbb{E}}_G(t, N)$ is also bounded in time and one can put $\max_{s \in [0, \infty)} \hat{\mathbb{E}}_G(t, N) \leq C_2 < \infty$. Applying this in (45) and integrating over time, the inequality for the error follows as

$$\|\varepsilon_1(t)\|_N \leq \frac{1 - \exp(-\delta_G t)}{\delta_G} C_2. \tag{51}$$

Since $\delta_G > \delta_0$, the error using Gauß–Legendre nodes will reach its asymptotic value faster than the error using a Gauß–Lobatto–Legendre basis. We see this behaviour in our numerical simulations in the next section.

6 Numerical Examples

In this section, we present some numerical experiments using the constructed schemes. We focus on the influence of the different numerical fluxes on the long time behaviour of the error. From [20,31], we know that in case of constant coefficients the choice of the numerical flux has an essential influence on the error behaviour, especially in the Gauß–Lobatto–Legendre case.

We consider our model problem, the linear advection equation

$$\begin{aligned} \partial_t u(t, x) + \partial_x(a(x)u(t, x)) &= 0, & t > 0, x \in (x_L, x_R), \\ u(t, x_L) &= g_L(t), & t \geq 0, \\ u(0, x) &= u_0(x), & x \in (x_L, x_R), \end{aligned} \tag{1}$$

with smooth speed $a(x) > 0$, initial condition u_0 and boundary condition g_L . The solution u of the corresponding Cauchy problem can be calculated by the method of characteristics, see e.g. [4, Chapter 3]. As time integrator, we use the fourth order, ten stage, strong stability preserving Runge–Kutta method of [17] and the time step is chosen such that the time integration error is negligible. Although the term “strong-stability preserving” means the preservation of stability properties of the explicit Euler method and the explicit Euler method is not stable for our numerical experiments, this fourth order Runge–Kutta method is strongly stable for linear equations [37]. All elements are of uniform size.

6.1 Coefficient $a(x) = x$

In our first experiment, we choose $a(x) = x$ with initial condition $u_0(x) = \sin(12(x - 0.1))$. The interval is $[x_L, x_R] = [0, 2\pi]$ and we choose the inflow boundary condition such that we get the solution

$$u(t, x) = \exp(-t)u_0(x \exp(-t)).$$

For the coefficient $a(x) = x$, the first derivative of a is strictly positive, implying $Int_d > 0$.

¹¹ δ_0 from Sect. 4, inequality (46).

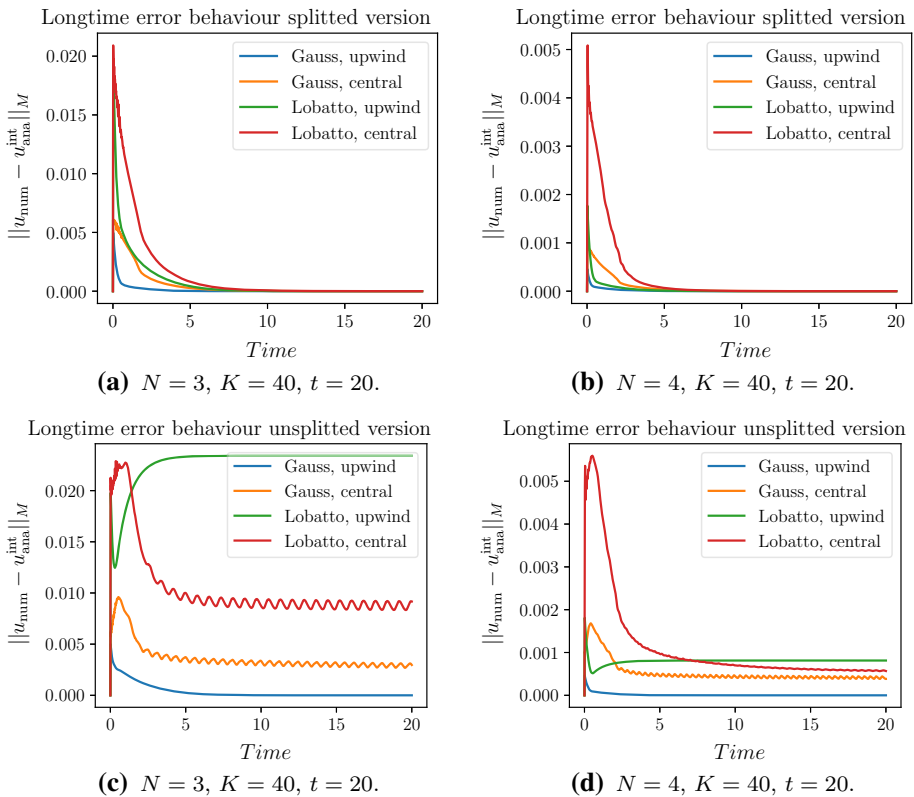


Fig. 1 Errors of numerical solutions using split and unsplit discretisations, $a(x) = x$

In our first simulation, we use $K = 40$ elements and calculate the solutions up to $t = 20$ with 200,000 time steps. In Fig. 1, we plot the long time error behaviour using polynomial degrees three and four. One recognizes that in all cases the error remains bounded in time.

In the first row of Fig. 1, all terms (surface, flux and volume) are split whereas in the second row they are not. We see that the error for the split version behaves like in the case of constant coefficients [20,31]. We mean that the errors using the upwind fluxes are always lower than the ones using central fluxes and one may recognize that we have some noisy behaviour using the central fluxes. Using upwind fluxes, the error reaches its asymptotic value faster than for the central fluxes.

In the second row, the unsplit discretisation is used. We recognize that we lose the predictions from [20,31] that applying the upwind flux yields a more accurate solution. The absolute value is also bigger applying the unsplit versions and we have again the noisy behaviour by applying the central fluxes.

Comparing all four plots, we recognize that the best results are obtained by using Gauß–Legendre nodes and the split discretisation. Therefore, we have a closer look on this. In Fig. 2, we consider only Gauß nodes and compare the split numerical fluxes and the unsplit numerical fluxes (with split surface and volume terms). *True* in the legend indicates the split numerical fluxes and *false* the unsplit ones. The experiment on the left-hand side demonstrates clearly that the noisy behavior for the central flux transfers also to the application of Gauß–Legendre nodes if all terms are split. Furthermore, we can hardly indicate some difference between the usage of split and unsplit upwind fluxes here, whereas we have a slight different behaviour in

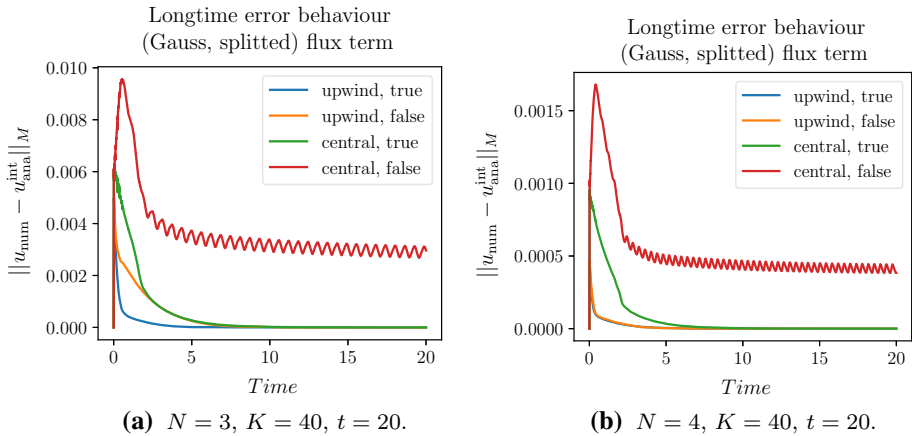


Fig. 2 Errors of numerical solutions using the split form and both split (true) and unsplit (false) numerical fluxes

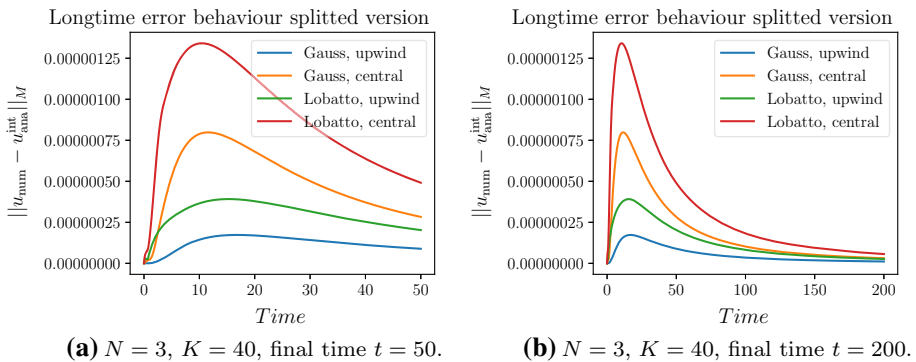


Fig. 3 Errors of discretisations for $a(x) = x^2, [x_L, x_R] = [0.1, 1]$, and $u_0(x) = \cos(\frac{\pi x}{2})$

the usage of the central fluxes. The test indicates that the split discretisation (volume/surface and numerical fluxes) should be preferred, matching our stability analysis.

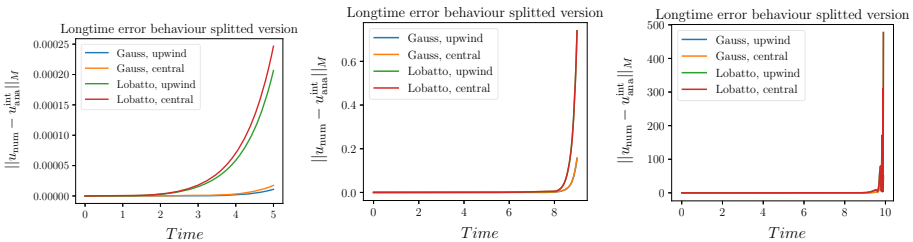
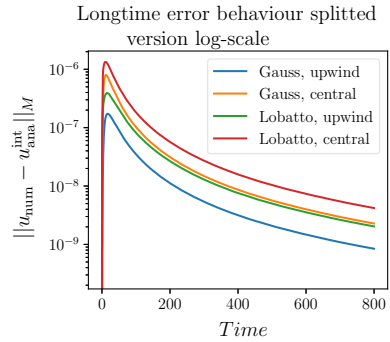
6.2 Coefficient $a(x) = x^2$

In our second experiment, we choose $a(x) = x^2$ with initial condition $u_0(x) = \cos(\frac{\pi x}{2})$. The interval is $[x_L, x_R] = [0.1, 1]$ and we choose the inflow boundary condition according to the solution

$$u(t, x) = \frac{u_0(x/(1 + tx))}{(1 + tx)^2}. \tag{52}$$

In our simulation shown in Fig. 3, we apply different numbers of time steps up to $t = 200$. First, we recognize that all errors are bounded in time, but different from the first case we do not have any noisy behavior of the central fluxes, at least we can not identify some. Simultaneously, the unsplit central flux error with Lobatto nodes increases at first rapidly before it finally tends to its asymptotic value. In all cases, the errors are small but we get always the *best* results by applying Gauß–Legendre nodes. Nevertheless, it takes a lot of time

Fig. 4 Error behaviour on a logarithmic scale for $a(x) = x^2$, $t = 800$



(a) $N = 3, K = 40, t = 5.$ **(b)** $N = 3, K = 40, t = 9.$ **(c)** $N = 3, K = 40, t = 9.9.$

Fig. 5 Errors of discretisations for $a(x) = x^2, [x_L, x_R] = [-0.1, 1],$ and $u_0(x) = \cos(\frac{\pi x}{2})$

for the errors to reach the asymptotic values. Even at time $t = 800,$ the asymptotic is still not reached, cf. Fig. 4.

In the first simulation, the interval has been chosen as $[x_L, x_R] = [0.1, 1]$ to guarantee the positivity of the derivative of a and also of its interpolation. Now, we change the interval to $[x_L, x_R] = [-0.1, 1],$ resulting in two major issues. First, the first derivative of a is not strictly positive anymore and the solution develops a pole at time $t = 10.$ Here, the solution is also not uniformly bounded in its Sobolev norm and our error bounds (46) and (51) do not hold. Nevertheless, the error behaviour can be investigated. Using only the split discretisation for different times, we see in Fig. 5 that the errors increase and will increase further. They are unbounded. Simultaneously, we also recognize that the errors using Gauß–Legendre nodes still increase slower due to the fact that the methods using these nodes are more accurate.

Furthermore, by changing the initial condition to $u_0(x) = \exp(-x^4)$ instead of $u_0(x) = \cos(\frac{\pi x}{2}),$ we are able to avoid the pole in the solution (52) since the exponential function will tend fast enough to zero compared to $(1 + tx)^2$ and we can extend the solution. Nevertheless we get further problems here. If we have a look on the error behaviour in Fig. 6, we see that we get a similar increase of the errors like in Fig. 5, but they are much smaller. Nevertheless they are still unbounded, but why do we have this behaviour? The analytical solution is for fixed times bounded, nevertheless we demand as one assumption right at the beginning at equation (1) the solution to be uniformly bounded in time. However, this is not the case anymore. This demonstrates again how essential this assumption is.

The same issue arises if we are investigate $a(x) = \cosh(x) + 1$ as in [36]. Therefore, we skip this case here.

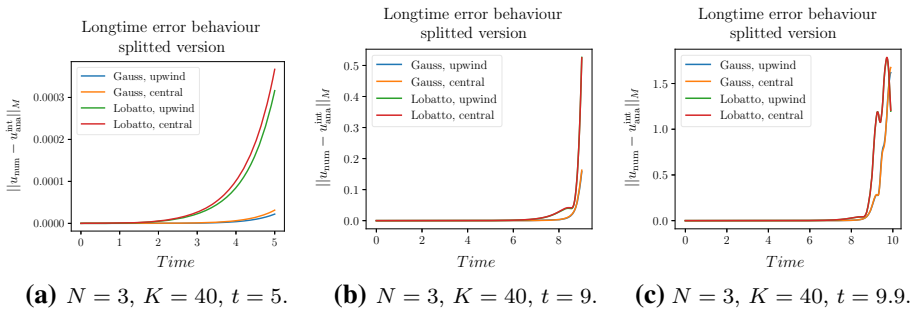
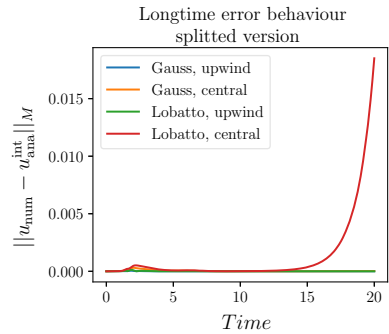


Fig. 6 Errors of discretisations for $a(x) = x^2, [x_L, x_R] = [-0.1, 1],$ and $u_0(x) = \exp(-x^4)$

Fig. 7 Error behaviour for $a(x) = 1 - x^2, t = 20$



6.3 Coefficient $a(x) = 1 - x^2$

Here, we choose the coefficient $a(x) = 1 - x^2.$ The solution of the Cauchy problem is

$$u(t, x) = \frac{u_0((-x \cosh(t) + \sinh(t))/(x \sinh(t) - \cosh(t)))}{(\cosh(t) - x \sinh(t))^2}. \tag{53}$$

Using the domain $[x_L, x_R] = [-1, 0.9]$ and the initial condition $u_0(x) = \sin(\pi x),$ the solution remains bounded but $a'(x) < 0$ for $x > 0.$

If we investigate now the long time error behaviour, we get a huge increase of the errors if we apply the central fluxes, cf. Fig. 7. This matches perfectly our theoretical investigations in Sects. 4 and 5, cf. Remark 4.1. We explain the reasons again in detail in the next test case and a physical interpretation and illustration is given afterwards.

6.4 Coefficient $a(x) = \cos(x)$

Here, we choose $a(x) = \cos(x)$ and $u_0(x) = \sin(5x).$ The solution of the Cauchy problem is

$$u(t, x) = u_0(x_0(t, x)) \frac{\cos(x_0(t, x))}{\cos(x)},$$

$$x_0(t, x) = -2 \arctan(\tanh(t/2 - \operatorname{artanh}(\tan(x/2)))). \tag{54}$$

We can find an interval for our solution (54) so that $a'(x) \leq 0$ and $u(t, x)$ does not blow up, e.g. $[x_L, x_R] = [0.1, \pi/3].$ The solution remains bounded but $a'(x) < 0.$

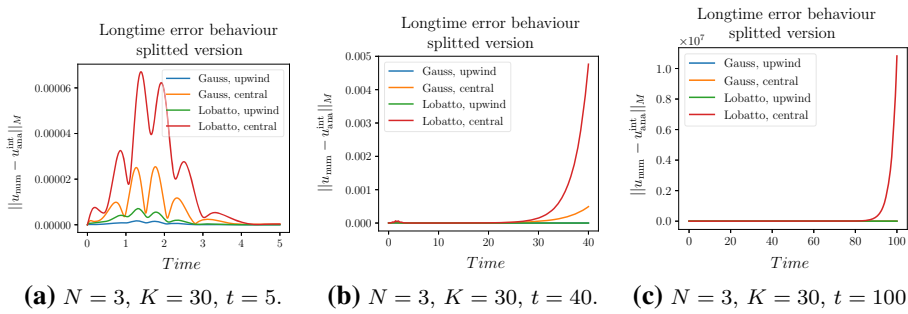


Fig. 8 For $a(x) = \cos(x)$, the slope of a is negative. The errors of numerical solutions using the central flux increase, whereas the upwind flux results in bounded errors

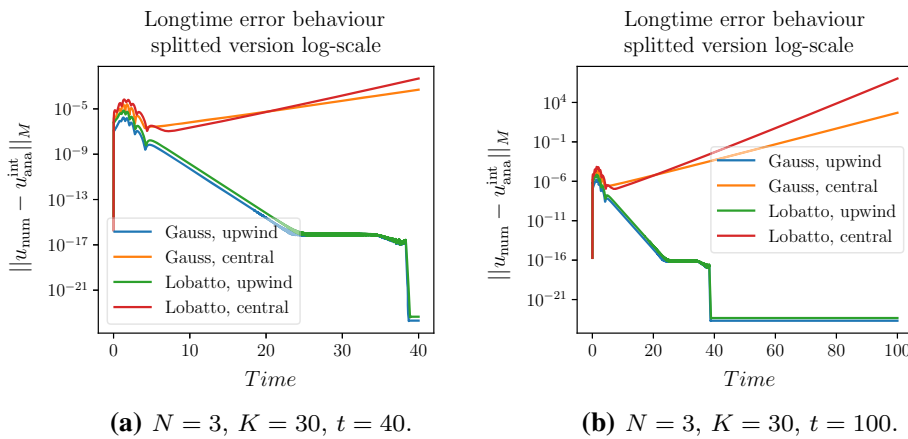


Fig. 9 Error of numerical solutions for $a(x) = \cos(x)$ in logarithmic scale

In Fig. 8, we see the behaviour of the error for different times. First, one may suppose that the error remains bounded in time, but this is not the case as can be seen stepping further in time. Using the central fluxes ($\sigma = 0$), the BT s terms are zero and we do not find an η which is bounded with a positive constant from below away from zero. One may recognize also that for Gauss–Legendre nodes, the error increases much slower (second picture). Surely, one reason for this is the smaller error in the Gauss–Legendre case. Furthermore, also the term Θ may have a positive impact of the error behaviour.

However, this example demonstrates well that the condition $a'(x) > 0$ is essential for the boundedness of the error, also in the test case of Sect. 6.3. One can rescue (46) and (51) by applying the upwind flux like it can be seen in this test case and especially in Fig. 9.

By applying an SBP-SAT finite difference scheme with one block, the internal terms BT s do not exist. Using the SBP difference operator of [23] with interior order of accuracy eight, the split form, and 100 nodes for this problem, the error is unbounded, as can be seen in Fig. 10. However, if the high-order artificial dissipation operator of [24] is applied additionally, the error remains bounded.

Comparing Figs. 9 and 10 demonstrates that stabilisation induced by upwind fluxes or artificial dissipation operators is crucial and comparable. Furthermore, Gauss–Legendre nodes not including boundary points provide some stabilisation.

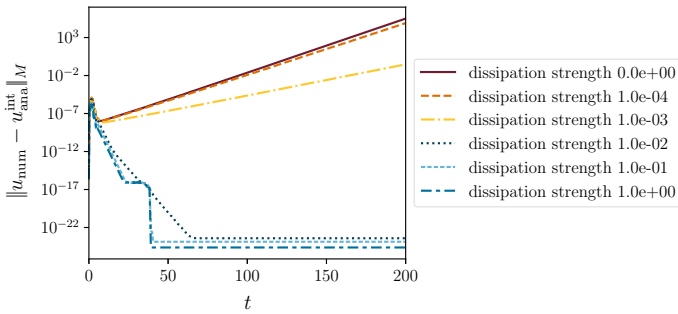


Fig. 10 Errors of numerical solutions using SBP finite difference schemes

6.5 A First Analytical Study

As can be seen in Figs. 9 and 10, if $a'(x)$ is not positive the long time errors show different behaviours depending on the dissipation which is added to the scheme by numerical fluxes or artificial dissipation terms. Here, we give a short rough analysis on this topic under what conditions we can guarantee boundedness. A more detail analysis should follow in future research with more validations.

We are starting considering $\eta_G(t)$ from (50). It is

$$\eta(t) := \frac{BTs + Int_d + \Theta_2}{\|\varepsilon_1\|_N^2} \tag{55}$$

with $Int_d := \frac{1}{2} \sum_{k=1}^K \frac{\Delta x_k}{2} \left(\partial_x a^k, \underline{\varepsilon}_1^k \underline{\varepsilon}_1^k \right)_N$. A sufficient condition for the mean of $\eta(t)$ to be positive is that every value of $\eta(t)$ is positive. Therefore, we require

$$\frac{BTs + Int_d + \Theta_2}{\|\varepsilon_1\|_N^2} > 0.$$

If the derivative of a is negative, we can reformulate the inequality above as

$$(BTs + \Theta_2) \frac{1}{\|\varepsilon_1\|_N^2} > \frac{1}{2\|\varepsilon_1\|_N^2} \left(\sum_{k=1}^K \frac{\Delta x_k}{2} \left(|\partial_x a^k|, \underline{\varepsilon}_1^k \underline{\varepsilon}_1^k \right)_N \right),$$

and even strengthen our assumptions by requiring

$$\begin{aligned} (BTs + \Theta_2) \frac{1}{\|\varepsilon_1\|_N^2} &> \frac{\max_{x \in (x_0, x_K)} |\partial_x a|}{2\|\varepsilon_1\|_N^2} \left(\sum_{k=1}^K \frac{\Delta x_k}{2} \left(\underline{1}, \underline{\varepsilon}_1^k \underline{\varepsilon}_1^k \right)_N \right) \\ \text{or } (BTs + \Theta_2) \frac{1}{\|\varepsilon_1\|_N^2} &> \frac{\max_{x \in (x_0, x_K)} |a'(x)|}{2}. \end{aligned} \tag{56}$$

From (56) we realize the BTs -terms are responsible to guarantee that this sufficient condition is fulfilled. In case of a central numerical flux, $BTs \equiv 0$ and we have to add additional dissipation to the scheme as it is done in the SBP-SAT schemes in Fig. 10. However, also the dependence of the error is important and we may also realize that in case of using Gauß–Legendre we rather get the condition (56) fulfilled. However, this estimation is rough and should be improved in further research.

6.6 Physical Interpretation and Illustration

In order to understand some results better, a physical interpretation of the advection equation can be used. This serves also as illustration and explains the rational behind some of the choices regarding for example the numerical experiments.

The advection equation $\partial_t u + \partial_x (au) = 0$ with non-negative velocity $a(x)$ is a conservation law with varying coefficients. Thus, the total mass $\int u$ is conserved and u is transported from left to right due to $a(x) \geq 0$. In order to compute analytical solutions of the Cauchy problem, the method of characteristics can be used, cf. [4, Chapter 3].

- Solve the ODE $x'(t) = a(x(t))$, $x(0) = x_0$, for $x(t) = x(t; x_0)$. Compute also the inverse function $x_0 = x_0(t; x)$.
- Solve the ODE $z'(t) = -a'(x(t; x_0))z(t)$, $z(0) = z_0$, for $z(t) = z(t; z_0, x_0)$.
- Set $z_0 = z_0(x_0) = u_0(x_0) = u_0(x_0(t; x))$ and obtain the analytical solution $u(t, x) = z(t; z_0, x_0) = z(t; z_0(x_0(t; x)), x_0(t; x))$.

In the second step, if $a' > 0$, the absolute value of $z(t)$ decreases. Contrary, if $a' < 0$, the absolute value of $z(t)$ increases. This corresponds directly to the physical interpretation as transport problem. Since u is conserved and transported with velocity $a(x)$, there is a loss of u if $a' > 0$, since there is less new mass coming from the left than going to the right. Similarly, $a' < 0$ yields an increase of u , since more mass is coming from the left than transported to the right. This explains also the critical role of $a'(x)$. If $a' < 0$, there can be blow-up phenomena in the solution u , resulting in possibly finite life spans and increasing energies and errors of numerical solutions. If $a' > 0$, this cannot happen.

If one wants to investigate a situation with $a'(x) > 0$ in some parts and $a'(x) < 0$ in other parts of the domain, there are basically two possibilities. Firstly, there can be a local minimum of $a(x)$, e.g. for $a(x) = x^2$. In this case, there can be a blow-up of the solution u , since more mass is coming from the left than transported to the right at this minimum. However, this blow-up phenomenon caused by the varying transport velocity $a(x)$ can be balanced by the initial condition u_0 . If there is simply not enough mass on the left, than the higher transport speed there can not cause a blow-up of the solution u . This explains our choice of the intervals and the initial conditions for these cases.

Secondly, there can be a local maximum of $a(x)$, e.g. for $a(x) = 1 - x^2$ or $a(x) = \cos(x)$. Now, there is no blow-up at the critical point, since more mass is transported to the right. However, both examples have stagnation points with $a(x) = 0$. At such points, there will be a blow-up of the solution, since mass is coming from the left but not transported to the right. In order to avoid this phenomenon of the Cauchy problem, the interval can be chosen adequately, i.e. bounded away at the right from the point with $a(x) = 0$. Then, the blow-up of the solution of the Cauchy problem does not cause any problems for the corresponding solution of the initial value problem. This explains our choices of the domains for these cases.

7 Possible Generalisation and Examples

As has been demonstrated hitherto, the error of numerical solutions of scalar hyperbolic conservation laws with varying coefficients does not necessarily remain bounded in finite domains, contrary to the expectation for linear systems with constant coefficients. Here, some further remarks concerning generalisations of this result are given.

7.1 Linearized Euler Equations

We start by considering the theory for the linearized Euler Equations which are one of the most—if not the most—investigated system in computational fluid dynamics. The one-dimensional compressible Euler equations in conservation form are

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = 0, \tag{57}$$

where \mathbf{U} is the state vector of the conserved quantities and \mathbf{F} is the flux. Thus,

$$\mathbf{U} = \begin{pmatrix} \rho \\ m \\ E \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} m \\ \rho u^2 + p \\ u(E + p) \end{pmatrix}, \tag{58}$$

where ρ is the mass density, $m = \rho u$ is the momentum, E is the total energy, u is the velocity and p is the pressure related to \mathbf{U} by the equation of state $p = (\gamma - 1)(E - \frac{\rho u^2}{2})$ using γ for the specific heat capacities. We can rewrite (57) as

$$\partial_t \mathbf{U} + \mathbf{A}(\mathbf{U}) \partial_x \mathbf{U} = 0,$$

where $\mathbf{A} = \partial_{\mathbf{U}} \mathbf{F}$ is the Jacobian matrix which has only real eigenvalues and can be diagonalized by the matrix \mathbf{R} of eigenvectors. Indeed, $\mathbf{A} = \mathbf{R} \mathbf{\Lambda} \mathbf{R}^{-1}$, where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \lambda_3) = \text{diag}(u + c, u, u - c)$. Here, c is the speed of sound which satisfies

$$c(\rho)^2 = p'(\rho) > 0. \tag{59}$$

As mentioned before, a lot of investigations of (57) can be found in the literature where also different linearization techniques were used depending on the numerical schemes [9,42,43, 45]. Here, we will focus on this topic and the problems which can appear. This yields us to some outlook for future research.

Linerization Around a Smooth Solution: An Outlook

We are not considering the full system (58) for a smooth solution, but the truncated/simplified/shortened version [14]

$$\begin{aligned} \partial_t \rho + \partial_x(\rho u) &= 0, \\ \partial_t u + u \partial_x u + \frac{1}{\rho} \partial_x p(\rho) &= 0, \end{aligned} \tag{60}$$

to explain the problem. Using a Taylor series approach for the linearization around a smooth solution $(\hat{\rho}, \hat{u})$ yields a linear system with variable coefficients of the form

$$\partial_t \begin{pmatrix} \rho \\ u \end{pmatrix} + \begin{pmatrix} \hat{u} & \hat{\rho} \\ \frac{c(\hat{\rho})^2}{\hat{\rho}} & \hat{u} \end{pmatrix} \partial_x \begin{pmatrix} \rho \\ u \end{pmatrix} + C \begin{pmatrix} \rho \\ \hat{u} \end{pmatrix} = 0,$$

where C depends on $(\hat{\rho}, \hat{u})$ and their derivatives such that $C = 0$ if $\hat{\rho}$ and \hat{u} are constant. This system can be symmetrized using $\rho_S := \frac{c(\hat{\rho})}{\hat{\rho}} \rho$, resulting in

$$\partial_t \begin{pmatrix} \rho_S \\ u \end{pmatrix} + \begin{pmatrix} \hat{u} & c(\hat{\rho}) \\ c(\hat{\rho}) & \hat{u} \end{pmatrix} \partial_x \begin{pmatrix} \rho_S \\ u \end{pmatrix} + \tilde{C} \begin{pmatrix} \rho_S \\ u \end{pmatrix} = 0, \tag{61}$$

where \tilde{C} depends on $(\hat{\rho}, \hat{u})$ and their derivatives such that $\tilde{C} = 0$ if $\hat{\rho}$ and \hat{u} are constant. If we have constant coefficients, this investigation belongs to the case which was already studied in [20,27,31] and the error remains bounded under the conditions give there. Otherwise, all entries of \tilde{C} are non-trivial and the equations cannot be decoupled. Already for symmetric systems, we get further problems depending on the estimation of the energy growth, as described in Sect. 7.2. The investigation of the error behaviour for this problem is not straightforward and should be considered in more detail in future work.

Remark 7.1 As mentioned above, there are different techniques for linearizing the Euler equations. They depend on the numerical schemes which are used/constructed for these system. Here, we only mention the approach by Roe [42] about flux difference splitting or the flux vector splitting in [43]. The linearization is used in the construction of the numerical schemes in some sense. To follow their ideas together with our analysis about the long time error behaviour is an alternative ansatz and will also be considered in future research.

7.2 Multidimensional Systems

We consider the linear magnetic induction equation

$$\begin{aligned} \partial_t B(t, x) &= \nabla \times (u(t, x) \times B(t, x)), \quad t \in (0, 50), x \in (0, 1)^3, \\ B(0, x) &= u(t, x) = \begin{pmatrix} \sin(\pi x) \cos(\pi y) \cos(\pi z) \\ \cos(\pi x) \sin(\pi y) \cos(\pi z) \\ -2 \cos(\pi x) \cos(\pi y) \sin(\pi z) \end{pmatrix}, \quad x \in [0, 1]^3, \end{aligned} \quad (62)$$

supplemented with the divergence constraint $\operatorname{div} B(t, x) = 0$, cf. [18,25]. This specific example is taken from [41]. Here, B is the magnetic field and u the particle velocity. Since u vanishes at the boundary of the domain, no boundary condition is specified. In order to get a symmetric hyperbolic system, the nonconservative source term $-u \operatorname{div} B$ is added to the right hand side, resulting in an energy estimate if a splitting is used as described in the references listed above. There are several discrete forms of the equation allowing an energy estimate [41]. Using the terminology introduced there, the most obvious one uses the same split form as applied at the continuous level and is called (product, central, split). Another choice described there is (central, central, central). The implementations of [40] are used in the following.

Applying both discretisations, SBP FD operators of interior order of accuracy 4, and 40^3 nodes to discretize the domain yields the results visualized in Fig. 11. As can be seen there, the form (product, central, split) results in an exponential growth of both the energy and the error while the other form yields a bounded error. Adding artificial dissipation does not change the result significantly.

These results are in accordance with the energy estimates (an exponential growth is allowed as worst case estimate) and the investigations in this article. The main complications for (product, central, split) are presumably a combination of

- The velocity u vanishes at the boundary and errors cannot be transported out of the domain; instead, they accumulate.
- While the analytical solution has a bounded energy, the worst case estimate allows an exponential growth.
- The analytical solution is a steady state which is not necessarily represented exactly by the discretisation.

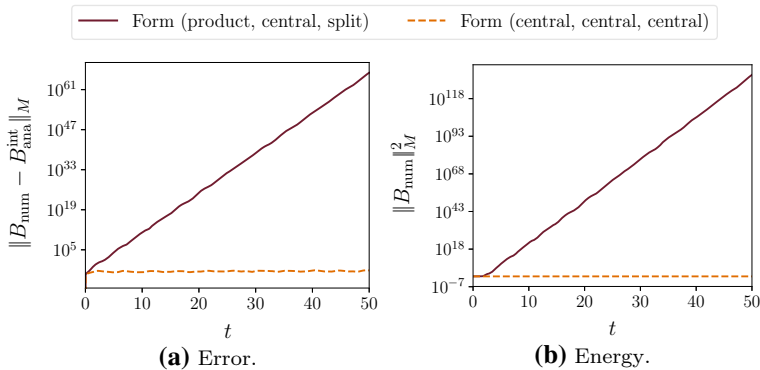


Fig. 11 Errors and energies of numerical solutions of the induction equation

This shows that severe problems can be expected for general symmetric hyperbolic systems with varying coefficients in multiple space dimensions.

8 Summary and Discussion

In this article, we have conducted an analysis of the long-time behaviour of the error of numerical solutions to the linear advection equation with variable coefficients in bounded domains. Using flux reconstruction schemes/discontinuous Galerkin methods with summation-by-parts operators, we provide a detailed analysis of the influence of both the choice of the numerical flux and the polynomial basis. If boundary conditions are imposed in a provably stable way using numerical fluxes, the error can be bounded uniformly in time, depending on the variable coefficient $a(x)$ and the numerical fluxes at the interior boundaries. However, there can be also an unbounded growth of the error if certain conditions are not satisfied.

Firstly, if the varying coefficient $a(x)$ behaves nicely, inducing a decay of the analytical solution, the long time behaviour of the numerical error is comparable to the case of constant coefficients. The application of upwind fluxes at interior boundaries results in a smaller asymptotic value of the error and this value is also attained faster. Using Gauß–Legendre nodes results in smaller errors compared to Gauß–Lobatto–Legendre nodes.

However, if the varying coefficient $a(x)$ induces a possible growth or blow-up of the analytical solution, the situation is totally different. Of course, if the solutions blows up in finite time, so does the error. This behaviour is not possible for constant coefficients. Moreover, there can still be some problems, even if the solution does not blow up. Indeed, the variable coefficients can trigger a growth of the error that has to be balanced by additional stabilisation such as upwind numerical fluxes compared to central ones or artificial dissipation, e.g. in finite difference methods. We have explained this behaviour and have presented several numerical examples, where upwind numerical fluxes or artificial dissipation result in uniformly bounded errors while the errors increase without bound if central numerical fluxes or no additional dissipation operators are applied.

Finally, in the last section we have extended our analysis of the long time error behaviour to systems. Here, several problems emerge and we have given an outlook for further research

topics in this context focussing on coupled symmetric systems with variable coefficients such as the linearized Euler or magnetic induction equations. As can be seen there, further problems can arise for general symmetric hyperbolic systems in multiple space dimensions, even if energy stable discretizations are used.

Acknowledgements Philipp Öffner was supported by SNF Project (Number 175784) “Solving advection dominated problems with high order schemes with polygonal meshes: application to compressible and incompressible flow problems” and Hendrik Ranocha was supported by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) under Grant SO 363/14-1.

Appendix

Technical Explanation of the Investiagtion in Sect. 4

We presented the ideas how to reach (35) from (31). Applying the interpolation operator together with discrete norms results in¹²

$$\begin{aligned} \langle \partial_t \mathbb{I}^N(u^k), \varphi^k \rangle &= \left(\partial_t \underline{\mathbb{I}}^N(u^k), \underline{\varphi}^k \right)_N \\ &\quad + \left\{ \langle \partial_t \mathbb{I}^N(u^k), \varphi^k \rangle - \left(\partial_t \underline{\mathbb{I}}^N(u^k), \underline{\varphi}^k \right)_N \right\}, \end{aligned} \tag{32}$$

$$\begin{aligned} \frac{1}{2} \langle a^k \mathbb{I}^N(u^k), \partial_\xi \varphi^k \rangle &= \frac{1}{2} \left(\underline{a}^k \underline{\mathbb{I}}^N(u^k), \partial_\xi \underline{\varphi}^k \right)_N \\ &\quad + \frac{1}{2} \left\{ \langle a^k \mathbb{I}^N(u^k), \partial_\xi \varphi^k \rangle - \left(\underline{a}^k \underline{\mathbb{I}}^N(u^k), \partial_\xi \underline{\varphi}^k \right)_N \right\}, \end{aligned} \tag{63}$$

$$\begin{aligned} \frac{1}{2} \langle a^k \partial_\xi \mathbb{I}^N(u^k), \varphi^k \rangle &= \frac{1}{2} \left(\underline{a}^k \partial_\xi \underline{\mathbb{I}}^N(u^k), \underline{\varphi}^k \right)_N \\ &\quad + \frac{1}{2} \left\{ \langle a^k \partial_\xi \mathbb{I}^N(u^k), \varphi^k \rangle - \left(\underline{a}^k \partial_\xi \underline{\mathbb{I}}^N(u^k), \underline{\varphi}^k \right)_N \right\}, \end{aligned} \tag{64}$$

$$\begin{aligned} \frac{1}{2} \langle \mathbb{I}^N(u^k) \partial_\xi a^k, \varphi^k \rangle &= \frac{1}{2} \left(\underline{\mathbb{I}}^N(u^k) \partial_\xi \underline{a}^k, \underline{\varphi}^k \right)_N \\ &\quad + \frac{1}{2} \left\{ \langle \mathbb{I}^N(u^k) \partial_\xi a^k, \varphi^k \rangle - \left(\underline{\mathbb{I}}^N(u^k) \partial_\xi \underline{a}^k, \underline{\varphi}^k \right)_N \right\}. \end{aligned} \tag{65}$$

It is well known [5, Section 5.4.3] that the integration error arising from the use of Gauß quadrature (Gauß–Legendre and Gauß–Lobatto–Legendre) decays spectrally fast. Indeed, for all $\varphi \in \mathbb{P}^N$ and $m \geq 1$,

$$\left| \langle u, \varphi \rangle - \left(\underline{u}, \underline{\varphi} \right)_N \right| \leq C N^{-m} |u|_{H^{m,N-1}(-1,1)} \|\varphi\|_{L^2(-1,1)},$$

where C is a constant independent of m and u . The curly brackets of (32), (63)–(65) have to be reformulated. Using

¹² Since $\varphi \in \mathbb{P}^N$ and if $a \equiv 1$, the volume term is

$$\left(\mathbb{I}^N(u^k), \partial_\xi \varphi^k \right) = \left(\underline{\mathbb{I}}^N(u^k), \partial_\xi \underline{\varphi}^{k,T} \right)_N = \underline{\varphi}^k \underline{D}^T \underline{M} \underline{\mathbb{I}}^N(u^k)$$

and also the terms (63)–(65) simplify and can be brought together, see inter alia [20] for details.

$$\begin{aligned} \left\langle \partial_t \mathbb{I}^N(u^k), \varphi^k \right\rangle - \left(\partial_t \mathbb{I}^N(u^k), \varphi^k \right)_N &= \left\langle \underbrace{\partial_t \left(\mathbb{I}^N(u^k) - P_{N-1}^m \left(\mathbb{I}^N(u^k) \right) \right)}_{=: Q(u^k)}, \varphi^k \right\rangle \\ &\quad - \left(\partial_t \left(\mathbb{I}^N(u^k) - P_{N-1}^m \left(\mathbb{I}^N(u^k) \right) \right), \varphi^k \right)_N, \end{aligned} \tag{66}$$

where P_{N-1}^m is the orthogonal projection of u onto \mathbb{P}^{N-1} using the inner product of $H^m(e^k)$, gives a new formulation for (32). The projection operator is defined by the classical truncated Fourier series $P^{N-1}u = \sum_{k=0}^{N-1} \hat{u}_k \Phi_k$ up to order $N - 1$ where Sobolev type orthogonal polynomials $\{\Phi_k\}$ are used as basis functions in the Hilbert space $H^m(e^k)$. The coefficients are calculated using the inner product of $H^m(e^k)$ given by

$$\langle u, v \rangle_m = \sum_{k=0}^m \int_{e_k} \frac{d^k u}{d x^k}(x) \frac{d^k v}{d x^k}(x) dx.$$

For more details about the projection operator and about approximation results, we strongly recommend [5, Section 5] and also [2,3]. An analogous approach as (66) leads to terms with Q_1 for (63), Q_2 for (64) and Q_3 for (65). The Q_j measure the projection error of a polynomial of degree N to a polynomial of degree $N - 1$. Since u and a are bounded, also these values have to be bounded. This values can be introduced and finally one obtains (35). Later, in this section the error of the fluxes has to be calculated. We obtain for the left and right boundary:

$$\begin{aligned} \text{left: } -\mathbf{E}_L^1 \left(f_L^{\text{num},1} - \frac{1}{2} a_L^1 \mathbf{E}_L^1 \right) &= -\mathbf{E}_L^1 \left(\left(a_L^1 \frac{0 + \mathbf{E}_L^1}{2} - \sigma a_L^1 \frac{\mathbf{E}_L^1}{2} \right) - \frac{a_L^1 \mathbf{E}_L^1}{2} \right) \\ &= \frac{\sigma a_L^1}{2} \left(\mathbf{E}_L^1 \right)^2, \\ \text{right: } \mathbf{E}_R^K \left(f_R^{\text{num},K} - \frac{1}{2} a_R^K \mathbf{E}_R^K \right) &= \mathbf{E}_R^K \left(\left(a_R^K \frac{0 + \mathbf{E}_R^K}{2} + \frac{1}{2} \sigma a_R^K \mathbf{E}_R^K \right) - \frac{\mathbf{E}_R^K a_R^K}{2} \right) \\ &= \frac{\sigma a_R^K}{2} \left(\mathbf{E}_R^K \right)^2. \end{aligned}$$

Technical Steps of the Development in Sect. 5

Here, we are presenting the main steps to reach (48).

$$\begin{aligned} \frac{\Delta x_k}{2} \left\langle \partial_t \mathbb{I}^N(u^k), \varphi^k \right\rangle + \frac{1}{2} \left(a^k \mathbb{I}^N(u^k) \varphi^k \Big|_{-1}^1 - \left\langle a^k \mathbb{I}^N(u^k), \partial_\xi \varphi^k \right\rangle + \left\langle \partial_\xi \mathbb{I}^N(u^k), a^k \varphi^k \right\rangle \right. \\ \left. + \left\langle \partial_\xi a^k, \varphi^k \mathbb{I}^N(u^k) \right\rangle \right) &= -\frac{\Delta x_k}{2} \left\langle \partial_t \varepsilon_p^k, \varphi^k \right\rangle - \frac{1}{2} \left(a^k \varepsilon_p^k \varphi^k \Big|_{-1}^1 \right) + \frac{1}{2} \left\langle a^k \varepsilon_p^k, \partial_\xi \varphi^k \right\rangle \\ &\quad - \frac{1}{2} \left\langle \partial_\xi \varepsilon_p^k, a^k \varphi^k \right\rangle - \frac{1}{2} \left\langle \varphi^k \varepsilon_p^k, \partial_\xi a^k \right\rangle. \end{aligned}$$

Integration-by-parts yields

$$-\frac{1}{2} \left(a^k \varepsilon_p^k \varphi^k \Big|_{-1}^1 - \left\langle a^k \varepsilon_p^k, \partial_\xi \varphi^k \right\rangle \right) = -\frac{1}{2} \left\langle \partial_\xi (a^k \varepsilon_p^k), \varphi^k \right\rangle.$$

With (32),(63)– (65), one obtains

$$\begin{aligned}
 & \frac{\Delta x_k}{2} \left(\partial_t \mathbb{I}^N(u^k), \underline{\varphi}^k \right)_N + \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left(\mathbb{I}^N(u^k)^-, \mathbb{I}^N(u^k)^+ \right) - \frac{1}{2} \left(\underline{Ra}^k \right) \cdot \left(\underline{Ru} \right) \right) \\
 & + \underbrace{\left(\frac{1}{2} a^k \mathbb{I}^N(u^k) \varphi^k \Big|_{-1}^1 - \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left(\mathbb{I}^N(u^k)^-, \mathbb{I}^N(u^k)^+ \right) - \frac{1}{2} \left(\underline{Ra}^k \right) \cdot \left(\underline{Ru} \right) \right) \right)}_{=:\varepsilon_2^k(a^k)} \\
 & - \frac{1}{2} \left(\underline{a}^k \mathbb{I}^N(u^k), \partial_\xi \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\partial_\xi \mathbb{I}^N(u^k), \underline{a}^k \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\partial_\xi \underline{a}^k, \mathbb{I}^N(u^k) \underline{\varphi}^k \right)_N \\
 & = \frac{\Delta x_k}{2} \left\langle \hat{T}^k(u^k), \varphi^k \right\rangle + \frac{\Delta x_k}{4} \left\langle \mathcal{Q}_1(u^k), \partial_x \varphi^k \right\rangle \\
 & + \frac{\Delta x_k}{4} \left\{ \left(\underline{\mathcal{Q}}(u^k), \underline{\varphi}^k \right)_N - \left(\underline{\mathcal{Q}}_1(u^k), \partial_x \varphi^k \right)_N + \left(\underline{\mathcal{Q}}_2(u^k), \underline{a}^k \underline{\varphi}^k \right)_N \right. \\
 & \left. + \left(\partial_x \underline{a}^k, \underline{\mathcal{Q}}_3(u^k) \varphi^k \right)_N \right\} \tag{47}
 \end{aligned}$$

with

$$\begin{aligned}
 \hat{T}(u^k) := & - \left\{ \partial_t \varepsilon_p^k + \frac{1}{2} \left(\partial_x \left(a^k \varepsilon_p^k \right) + \varepsilon_p^k \partial_x a^k \partial_x \varepsilon_p^k \right) \right. \\
 & \left. + \frac{1}{2} \left(\mathcal{Q}(u^k) + a^k \mathcal{Q}_2(u^k) + \mathcal{Q}_3(u^k) \partial_x a^k \right) \right\}.
 \end{aligned}$$

We transposed every term in (29) and subtracted it from equation (47). Using $\varepsilon_1^k = \mathbb{I}^N(u^k) - U^k$ yields

$$\begin{aligned}
 & \frac{\Delta x_k}{2} \left(\partial_t \underline{\varepsilon}_1^k, \underline{\varphi}^k \right) + \underline{\varphi}^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \left(\underline{Ra}^k \right) \cdot \left(\underline{R\varepsilon}_1^k \right) \right) \\
 & + \varepsilon_2^k(a^k) - \frac{1}{2} \left(\underline{a}^k \underline{\varepsilon}_1^k, \partial_\xi \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\partial_\xi \underline{\varepsilon}_1^k, \underline{a}^k \underline{\varphi}^k \right)_N + \frac{1}{2} \left(\partial_\xi \underline{a}^k, \underline{\varepsilon}_1^k \underline{\varphi}^k \right)_N \\
 & = \frac{\Delta x_k}{2} \left\langle \hat{T}^k(u^k), \varphi^k \right\rangle + \frac{\Delta x_k}{4} \left\langle \mathcal{Q}_1(u^k), \partial_x \varphi^k \right\rangle + \frac{\Delta x_k}{4} \left\{ \left(\underline{\mathcal{Q}}(u^k), \underline{\varphi}^k \right)_N \right. \\
 & \left. - \left(\underline{\mathcal{Q}}_1(u^k), \partial_x \varphi^k \right)_N + \left(\underline{\mathcal{Q}}_2(u^k), \underline{a}^k \underline{\varphi}^k \right)_N + \left(\partial_x \underline{a}^k, \underline{\mathcal{Q}}_3(u^k) \varphi^k \right)_N \right\}.
 \end{aligned}$$

Putting $\varphi^k = \varepsilon_1^k$ results in the energy equation similar to (37):

$$\begin{aligned}
 & \frac{\Delta x_k}{4} \frac{d}{dt} \|\varepsilon_1^k\|_N^2 + \underline{\varepsilon}_1^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \left(\underline{Ra}^k \right) \cdot \left(\underline{R\varepsilon}_1^k \right) \right) \\
 & + \varepsilon_2^k(a^k) - \frac{1}{2} \left(\underline{a}^k \underline{\varepsilon}_1^k, \partial_\xi \underline{\varepsilon}_1^k \right)_N + \frac{1}{2} \left(\partial_\xi \underline{\varepsilon}_1^k, \underline{a}^k \underline{\varepsilon}_1^k \right)_N + \frac{1}{2} \left(\partial_\xi \underline{a}^k, \underline{\varepsilon}_1^k \underline{\varepsilon}_1^k \right)_N \\
 & = \frac{\Delta x_k}{2} \left\langle \hat{T}^k(u^k), \varepsilon_1^k \right\rangle + \frac{\Delta x_k}{4} \left\langle \mathcal{Q}_1(u^k), \partial_x \varepsilon_1^k \right\rangle \\
 & + \underbrace{\frac{\Delta x_k}{4} \left\{ \left(\underline{\mathcal{Q}}(u^k), \underline{\varepsilon}_1^k \right)_N - \left(\underline{\mathcal{Q}}_1(u^k), \partial_x \varepsilon_1^k \right)_N + \left(\underline{\mathcal{Q}}_2(u^k), \underline{a}^k \underline{\varepsilon}_1^k \right)_N + \left(\partial_x \underline{a}^k, \underline{\mathcal{Q}}_3(u^k) \varepsilon_1^k \right)_N \right\}}_{\hat{Q}^k}.
 \end{aligned}$$

Together with (38), one obtains

$$\begin{aligned}
 & \frac{\Delta x_k}{4} \frac{d}{dt} \|\varepsilon_1^k\|_N^2 + \underline{\varepsilon}_1^{k,T} \underline{R}^T \underline{B} \left(\underline{f}^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) - \frac{1}{2} \left(\underline{Ra}^k \right) \cdot \left(\underline{R\varepsilon}_1^k \right) \right) \\
 & + \varepsilon_2^k(a^k) + \frac{1}{2} \left(\partial_\xi \underline{a}^k, \underline{\varepsilon}_1^k \underline{\varepsilon}_1^k \right)_N = \frac{\Delta x_k}{2} \left\langle \hat{T}^k(u^k), \varepsilon_1^k \right\rangle + \frac{\Delta x_k}{4} \left\langle \mathcal{Q}_1(u^k), \partial_x \varepsilon_1^k \right\rangle + \hat{Q}^k. \tag{67}
 \end{aligned}$$

Summing this up over all elements results in

$$\begin{aligned} & \frac{1}{2} \frac{d}{dt} \sum_{k=1}^K \frac{\Delta x_k}{2} \|\varepsilon_1^k\|_N^2 + \sum_{k=1}^K \varepsilon_1^{k,T} \underline{R}^T \underline{B} \left(f^{\text{num},k} \left((\varepsilon_1^k)^-, (\varepsilon_1^k)^+ \right) \right. \\ & \quad \left. - \frac{1}{2} \left(\underline{R} a^k \right) \left(\underline{R} \varepsilon_1^k \right) \right) + \frac{1}{2} \sum_{k=1}^K \frac{\Delta x_k}{2} \left(\partial_x a^k, \underline{\varepsilon}_1^k \varepsilon_1^k \right)_N + \sum_{k=1}^K \frac{\Delta x_k}{4} \varepsilon_2^k(a^k) \\ & = \sum_{k=1}^K \frac{\Delta x_k}{2} \left\langle \hat{T}^k(u^k), \varepsilon_1^k \right\rangle + \sum_{k=1}^K \frac{\Delta x_k}{4} \left\langle Q_1(u^k), \partial_x \varepsilon_1^k \right\rangle + \sum_{k=1}^K \hat{Q}^k. \end{aligned}$$

Applying the same approach like in Eqs. (41)–(42) and the fact that $\varepsilon_1 \in \mathbb{P}^N$, it is $\|\partial_x \underline{\varepsilon}_1^k\|_N^2 \leq c_1 N^2 \|\varepsilon_1^k\|_N^2$ and we get finally (48).

Calculating the Fluxes from Table 2

– Split central flux $f^{\text{num}}(u_-, u_+) = \frac{a_- u_- + a_+ u_+}{2}$: One obtains

$$\begin{aligned} & \frac{1}{2} \left(a_R^{k-1} \mathbf{E}_R^{k-1} + a_L^k \mathbf{E}_L^k \right) \left(\mathbf{E}_R^{k-1} - \mathbf{E}_L^k \right) - \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) \\ & = \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^k \right)^2 - a_R^{k-1} \mathbf{E}_L^k \mathbf{E}_R^{k-1} + a_L^k \mathbf{E}_L^k \mathbf{E}_R^{k-1} - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) \\ & \quad - \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) = \frac{1}{2} \mathbf{E}_L^k \mathbf{E}_R^{k-1} \left(a_L^k - a_R^{k-1} \right) = 0 \end{aligned}$$

and

$$\begin{aligned} \text{left: } & -\mathbf{E}_L^1 \left(f_L^{\text{num},1} - \frac{1}{2} a_L^1 \mathbf{E}_L^1 \right) = -\mathbf{E}_L^1 \left(\frac{a_L^1}{2} \mathbf{E}_L^1 - \frac{a_L^1}{2} \mathbf{E}_L^1 \right) = 0, \\ \text{right: } & \mathbf{E}_R^K \left(f_R^{\text{num},K} - \frac{1}{2} a_R^K \mathbf{E}_R^K \right) = \frac{1}{2} \left(\mathbf{E}_R^K \right)^2 \left(a_R^K - a_R^K \right) = 0. \end{aligned}$$

– Edge based upwind flux $f^{\text{num}}(u_-, u_-) = a(x)u_-$: It is

$$\begin{aligned} & a^{k-1}(x_R) \mathbf{E}_R^{k-1} \left(\mathbf{E}_R^{k-1} - \mathbf{E}_L^k \right) - \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) \\ & = a^{k-1}(x_R) \left(\mathbf{E}_R^{k-1} \right)^2 - a^{k-1}(x_R) \mathbf{E}_L^k \mathbf{E}_R^{k-1} - \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) \\ & = \left(\mathbf{E}_R^{k-1} \right)^2 \left(a^{k-1}(x_R) - \frac{1}{2} a_R^{k-1} \right) + \frac{\mathbf{E}_L^k}{2} \left(a_L^k \mathbf{E}_L^k - a_R^{k-1} \mathbf{E}_R^{k-1} \right) \\ & = \frac{1}{2} a_R^{k-1} \left[\mathbf{E}_R^{k-1} \right]^2 \end{aligned}$$

and

$$\begin{aligned} \text{left: } & -\mathbf{E}_L^1 \left(f_L^{\text{num},1} - \frac{1}{2} a_L^1 \mathbf{E}_L^1 \right) = \frac{1}{2} \left(\mathbf{E}_L^1 \right) a_L^1, \\ \text{right: } & \mathbf{E}_R^K \left(f_R^{\text{num},K} - \frac{1}{2} a_R^K \mathbf{E}_R^K \right) = \left(\mathbf{E}_R^K \right)^2 \left(a^K(x_R) - \frac{a_R^K}{2} \right) = \left(\mathbf{E}_R^K \right)^2 \left(\frac{a_R^K}{2} \right). \end{aligned}$$

– Split upwind flux $f^{\text{num}}(u_-, u_-) = a_- u_-$: It is

$$\begin{aligned} & a_R^{k-1} \mathbf{E}_R^{k-1} \left(\mathbf{E}_R^{k-1} - \mathbf{E}_L^k \right) - \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - a_L^k \left(\mathbf{E}_L^k \right)^2 \right) \\ &= \frac{1}{2} \left(a_R^{k-1} \left(\mathbf{E}_R^{k-1} \right)^2 - 2a_R^{k-1} \mathbf{E}_L^k \mathbf{E}_R^{k-1} + a_L^k \left(\mathbf{E}_L^k \right)^2 \right) = \frac{1}{2} a_R^{k-1} \left[\left[\mathbf{E}_R^{k-1} \right] \right]^2, \end{aligned}$$

where we used in the last step the assumption about the exactness of the interpolation and the continuity of a . At the boundaries we get

$$\begin{aligned} \text{left:} & \quad \frac{a_L^1}{2} \left(\mathbf{E}_L^1 \right)^2, \\ \text{right:} & \quad \frac{a_R^K}{2} \left(\mathbf{E}_R^K \right)^2. \end{aligned}$$

References

1. Abarbanel, S., Ditkowski, A., Gustafsson, B.: On error bounds of finite difference approximations to partial differential equations—temporal behavior and rate of convergence. *J. Sci. Comput.* **15**(1), 79–116 (2000)
2. Bernardi, C., Maday, Y.: Properties of some weighted Sobolev spaces and application to spectral approximations. *SIAM J. Numer. Anal.* **26**(4), 769–829 (1989)
3. Bernardi, C., Maday, Y.: Polynomial interpolation results in Sobolev spaces. *J. Comput. Appl. Math.* **43**(1–2), 53–80 (1992)
4. Bressan, A.: *Hyperbolic Systems of Conservation Laws: The One-Dimensional Cauchy Problem*. Oxford University Press, Oxford (2000)
5. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: *Spectral Methods: Fundamentals in Single Domains*. Springer, Berlin (2006). <https://doi.org/10.1007/978-3-540-30726-6>
6. Carpenter, M.H., Nordström, J., Gottlieb, D.: A stable and conservative interface treatment of arbitrary spatial accuracy. *J. Comput. Phys.* **148**(2), 341–365 (1999)
7. Cohen, G., Ferrieres, X., Pernet, S.: A spatial high-order hexahedral discontinuous Galerkin method to solve Maxwell’s equations in time domain. *J. Comput. Phys.* **217**(2), 340–363 (2006)
8. Fernández, D.C.D.R., Hicken, J.E., Zingg, D.W.: Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Comput. Fluids* **95**, 171–196 (2014)
9. Fey, M.: Multidimensional upwinding. Part II: Decomposition of the Euler equations into advection equations. *J. Comput. Phys.* **143**(1), 181–199 (1998)
10. Fisher, T.C., Carpenter, M.H., Nordström, J., Yamaleev, N.K., Swanson, C.: Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: theory and boundary conditions. *J. Comput. Phys.* **234**, 353–375 (2013)
11. Funaro, D.: *Polynomial Approximation of Differential Equations*, vol. 8. Springer, Berlin (2008)
12. Gassner, G.J.: A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. *SIAM J. Sci. Comput.* **35**(3), A1233–A1253 (2013). <https://doi.org/10.1137/120890144>
13. Govil, N., Mohapatra, R.: Markov and Bernstein type inequalities for polynomials. *J. Inequal. Appl.* **3**(4), 349–387 (1999)
14. Gustafsson, B., Kreiss, H.O., Olinger, J.: *Time-Dependent Problems and Difference Methods*. Wiley, Hoboken (2013)
15. Hesthaven, J., Kirby, R.: Filtering in Legendre spectral methods. *Math. Comput.* **77**(263), 1425–1452 (2008). <https://doi.org/10.1090/S0025-5718-08-02110-8>
16. Hesthaven, J.S., Warburton, T.: Nodal high-order methods on unstructured grids: I. Time-domain solution of Maxwell’s equations. *J. Comput. Phys.* **181**(1), 186–221 (2002)
17. Ketcheson, D.I.: Highly efficient strong stability-preserving Runge–Kutta methods with low-storage implementations. *SIAM J. Sci. Comput.* **30**(4), 2113–2136 (2008). <https://doi.org/10.1137/07070485X>

18. Koley, U., Mishra, S., Risebro, N.H., Svård, M.: Higher order finite difference schemes for the magnetic induction equations. *BIT Numer. Math.* **49**(2), 375–395 (2009). <https://doi.org/10.1007/s10543-009-0219-y>
19. Kopriva, D.A., Gassner, G.J.: On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods. *J. Sci. Comput.* **44**(2), 136–155 (2010). <https://doi.org/10.1007/s10915-010-9372-3>
20. Kopriva, D.A., Nordström, J., Gassner, G.J.: Error boundedness of discontinuous Galerkin spectral element approximations of hyperbolic problems. *J. Sci. Comput.* **72**(1), 314–330 (2017). <https://doi.org/10.1007/s10915-017-0358-2>
21. Kreiss, H.O., Scherer, G.: Finite element and finite difference methods for hyperbolic partial differential equations. In: de Boor, C. (ed.) *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pp. 195–212. Academic Press, New York (1974)
22. Manzanero, J., Rubio, G., Ferrer, E., Valero, E., Kopriva, D.A.: Insights on aliasing driven instabilities for advection equations with application to Gauss–Lobatto discontinuous Galerkin methods. *J. Sci. Comput.* **75**, 1262–1281 (2017). <https://doi.org/10.1007/s10915-017-0585-6>
23. Mattsson, K., Nordström, J.: Summation by parts operators for finite difference approximations of second derivatives. *J. Comput. Phys.* **199**(2), 503–540 (2004)
24. Mattsson, K., Svård, M., Nordström, J.: Stable and accurate artificial dissipation. *J. Sci. Comput.* **21**(1), 57–79 (2004)
25. Mishra, S., Svård, M.: On stability of numerical schemes via frozen coefficients and the magnetic induction equations. *BIT Numer. Math.* **50**(1), 85–108 (2010). <https://doi.org/10.1007/s10543-010-0249-5>
26. Nordström, J.: Conservative finite difference formulations, variable coefficients, energy estimates and artificial dissipation. *J. Sci. Comput.* **29**(3), 375–404 (2006)
27. Nordström, J.: Error bounded schemes for time-dependent hyperbolic problems. *SIAM J. Sci. Comput.* **30**(1), 46–59 (2007). <https://doi.org/10.1137/060654943>
28. Nordström, J., Gustafsson, R.: High order finite difference approximations of electromagnetic wave propagation close to material discontinuities. *J. Sci. Comput.* **18**(2), 215–234 (2003)
29. Nordström, J., Ruggiu, A.A.: On conservation and stability properties for summation-by-parts schemes. *J. Comput. Phys.* **344**, 451–464 (2017). <https://doi.org/10.1016/j.jcp.2017.05.002>
30. Öffner, P.: Zweidimensionale klassische und diskrete orthogonale Polynome und ihre Anwendung auf spektrale Methoden zur Lösung hyperbolischer Erhaltungsgleichungen. Ph.D. thesis, TU Braunschweig (2015)
31. Öffner, P.: Error boundedness of correction procedure via reconstruction/flux reconstruction (2018). [arXiv:1806.01575](https://arxiv.org/abs/1806.01575) [math.NA] (**submitted**)
32. Öffner, P., Sonar, T.: Spectral convergence for orthogonal polynomials on triangles. *Numer. Math.* **124**(4), 701–721 (2013). <https://doi.org/10.1007/s00211-013-0530-z>
33. Ranocha, H.: Comparison of some entropy conservative numerical fluxes for the Euler equations. *J. Sci. Comput.* **76**, 216–242 (2017). <https://doi.org/10.1007/s10915-017-0618-1>
34. Ranocha, H.: Shallow water equations: split-form, entropy stable, well-balanced, and positivity preserving numerical methods. *GEM Int. J. Geomath.* **8**(1), 85–133 (2017). <https://doi.org/10.1007/s13137-016-0089-9>
35. Ranocha, H.: Generalised summation-by-parts operators and entropy stability of numerical methods for hyperbolic balance laws. Ph.D. thesis, TU Braunschweig (2018)
36. Ranocha, H.: Generalised summation-by-parts operators and variable coefficients. *J. Comput. Phys.* **362**, 20–48 (2018). <https://doi.org/10.1016/j.jcp.2018.02.021>
37. Ranocha, H., Öffner, P.: L_2 stability of explicit Runge–Kutta schemes. *J. Sci. Comput.* **75**(2), 1040–1056 (2018). <https://doi.org/10.1007/s10915-017-0595-4>
38. Ranocha, H., Öffner, P., Sonar, T.: Summation-by-parts operators for correction procedure via reconstruction. *J. Comput. Phys.* **311**, 299–328 (2016). <https://doi.org/10.1016/j.jcp.2016.02.009>
39. Ranocha, H., Öffner, P., Sonar, T.: Extended skew-symmetric form for summation-by-parts operators and varying Jacobians. *J. Comput. Phys.* **342**, 13–28 (2017). <https://doi.org/10.1016/j.jcp.2017.04.044>
40. Ranocha, H., Ostaszewski, K., Heinisch, P.: InductionEq. A set of tools for numerically solving the nonlinear magnetic induction equation with Hall effect in OpenCL (2018). <https://doi.org/10.5281/zenodo.1434409>. <https://github.com/MuMPLaCL/InductionEq>
41. Ranocha, H., Ostaszewski, K., Heinisch, P.: Numerical methods for the magnetic induction equation with Hall effect and projections onto divergence-free vector fields (2018). [arXiv:1810.01397](https://arxiv.org/abs/1810.01397) [math.NA] (**submitted**)
42. Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **43**(2), 357–372 (1981)

43. Steger, J.L., Warming, R.: Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods. *J. Comput. Phys.* **40**(2), 263–293 (1981)
44. Svärd, M., Nordström, J.: Review of summation-by-parts schemes for initial-boundary-value problems. *J. Comput. Phys.* **268**, 17–38 (2014)
45. Van Leer, B.: Flux-vector splitting for the Euler equation. In: *Upwind and High-Resolution Schemes*, pp. 80–89. Springer (1997)
46. Vincent, P.E., Castonguay, P., Jameson, A.: A new class of high-order energy stable flux reconstruction schemes. *J. Sci. Comput.* **47**(1), 50–72 (2011). <https://doi.org/10.1007/s10915-010-9420-z>
47. Vincent, P.E., Farrington, A.M., Witherden, F.D., Jameson, A.: An extended range of stable-symmetric-conservative flux reconstruction correction functions. *Comput. Methods Appl. Mech. Eng.* **296**, 248–272 (2015). <https://doi.org/10.1016/j.cma.2015.07.023>
48. Zhang, Q., Shu, C.W.: Error estimates to smooth solutions of Runge–Kutta discontinuous Galerkin methods for scalar conservation laws. *SIAM J. Numer. Anal.* **42**(2), 641–666 (2004)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.