Lax-Wendroff Flux Reconstruction for compressible flows

A Thesis

submitted to the
Tata Institute of Fundamental Research, Mumbai
for the degree of **Doctor of Philosophy**
in **Mathematics**

by
Arpit Babbar

Centre for Applicable Mathematics
Tata Institute of Fundamental Research
Bangalore - 560065
India

December 1, 2023

DECLARATION

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

Arpit Babbar

The work was done under the guidance of my advisor Prof. Dr. Praveen Chandrashekar at the Tata Institute of Fundamental Research, Centre For Applicable Mathematics.

Prof. Dr. Praveen Chandrashekar

Acknowledgements

I begin by thanking my advisor Praveen Chandrashekar for his mentoring, teaching and generousity with sharing his expertise made this thesis possible.

Abstract

Table of contents

Introduction

This is similar to introduction section of papers and will have a lot of literature review.

Chapter 1

CONSERVATION LAWS

Plan for this chapter

1. Derivation of Euler's equations, 10 moment problem, Navier-Stokes equations
2. Definition of weak solution, regularity of solutions of conservation laws
3. Some finite volume methods, convergence results for scalar equations

Chapter 2

Finite volume method, Discontinuous Galerkin and Flux Reconstruction

Plan for this chapter

1. Finite volume method including MUSCL, MUSCL-Hancock
2. Discontinuous Galerkin and Flux Reconstruction for 1-D

2.1. Scalar conservation law

Let us consider a scalar conservation law of the form

$$u_t + f(u)_x = 0$$

where u is some conserved quantity, $f(u)$ is the corresponding flux, together with some initial and boundary conditions. We will divide the computational domain Ω into disjoint elements Ω_e, with

$$\Omega_e = [x_{e-\frac{1}{2}}, x_{e+\frac{1}{2}}] \qquad \text{and} \qquad \Delta x_e = x_{e+\frac{1}{2}} - x_{e-\frac{1}{2}}$$

Let us map each element to a reference element, $\Omega_e \to [0,1]$, by

$$x \to \xi = \frac{x - x_{e-\frac{1}{2}}}{\Delta x_e}$$

Inside each element, we approximate the solution by degree $N \geq 0$ polynomials belonging to the set $\mathbb{P}_N$. For this, choose $N+1$ distinct nodes

$$0 \leq \xi_0 < \xi_1 < \cdots < \xi_N \leq 1$$

which will be taken to be Gauss-Legendre (GL) or Gauss-Lobatto-Legendre (GLL) nodes, and will also be referred to as *solution points*. There are associated quadrature weights w_j such that the quadrature rule is exact for polynomials of degree up to $2N+1$ for GL points and upto degree $2N-1$ for GLL points. Note that the nodes and weights we use are with respect to the interval $[0,1]$ whereas they are usually defined for the interval $[-1,+1]$. The solution inside an element is given by

$$x \in \Omega_e: \qquad u_h(\xi, t) = \sum_{j=0}^{N} u_j^e(t)\, \ell_j(\xi)$$

where each ℓ_j is a Lagrange polynomial of degree N given by

$$\ell_j(\xi) = \prod_{i=0, i\neq j}^{N} \frac{\xi - \xi_i}{\xi_j - \xi_i} \in \mathbb{P}_N, \qquad \ell_j(\xi_i) = \delta_{ij}$$

Figure (2.1a) illustrates a piecewise polynomial solution at some time t_n with discontinuities at the element boundaries. Note that the coefficients u_j^e which are the basic unknowns or *degrees of freedom* (dof), are the solution values at the solution points.

Figure 2.1. (a) Piecewise polynomial solution at time t_n, and (b) discontinuous and continuous flux.

The numerical method will require spatial derivatives of certain quantities. We can compute the spatial derivatives on the reference interval using a differentiation matrix $D = [D_{ij}]$ whose entries are given by

$$D_{ij} = \ell_j'(\xi_i), \qquad 0 \leq i, j \leq N$$

For example, we can obtain the spatial derivatives of the solution u_h at all the solution points by a matrix-vector product as follows

$$\begin{bmatrix} \partial_x u_h(\xi_0, t) \\ \vdots \\ \partial_x u_h(\xi_N, t) \end{bmatrix} = \frac{1}{\Delta x_e} D u(t), \qquad u = \begin{bmatrix} u_0^e \\ \vdots \\ u_N^e \end{bmatrix}$$

We will use symbols in sans serif font like D, u, etc. to denote matrices or vectors defined with respect to the solution points. The entries of the differentiation matrix are given by

$$D_{ij} = \frac{W_j}{W_i}\frac{1}{(\xi_i - \xi_j)}, \qquad i \neq j \qquad \text{and} \qquad D_{ii} = -\sum_{j=0, j\neq i}^{N} D_{ij}$$

where the W_i are barycentric weights given by

$$W_i = \frac{1}{\prod_{j=0, j\neq i}^{N} (\xi_i - \xi_j)}, \qquad 0 \leq i \leq N$$

Define the Vandermonde matrices corresponding to the left and right boundaries of a cell by

$$V_L = [\ell_0(0), \ell_1(0), \ldots, \ell_N(0)]^\top, \qquad V_R = [\ell_0(1), \ell_1(1), \ldots, \ell_N(1)]^\top \tag{2.1}$$

which is used to extrapolate the solution and/or flux to the cell faces for the computation of inter-cell fluxes.

2.2. RUNGE-KUTTA FR

The RKFR scheme is based on a FR spatial discretization leading to a system of ODE followed by application of an RK scheme to march forward in time. The key idea is to construct a continuous polynomial approximation of the flux which is then used in a collocation scheme to update the nodal solution values. At some time t, we have the piecewise polynomial solution defined inside each cell; the FR scheme can be described by the following steps.

Step 1. In each element, we construct the flux approximation by interpolating the flux at the solution points leading to a polynomial of degree N, given by

$$f_h^\delta(\xi, t) = \sum_{j=0}^{N} f(u_j^e(t))\, \ell_j(\xi)$$

The above flux is in general discontinuous across the elements similar to the red curve in Figure (2.1b).

Step 2. We build a continuous flux approximation by adding some correction terms at the element boundaries

$$f_h(\xi, t) = \left[f_{e-\frac{1}{2}}(t) - f_h^\delta(0, t) \right] g_L(\xi) + f_h^\delta(\xi, t) + \left[f_{e+\frac{1}{2}}(t) - f_h^\delta(1, t) \right] g_R(\xi)$$

where

$$f_{e+\frac{1}{2}}(t) = f(u_h(x^-_{e+\frac{1}{2}}, t), u_h(x^+_{e+\frac{1}{2}}, t))$$

is a numerical flux function that makes the flux unique across the cells. The continuous flux approximation is illustrated by the black curve in Figure (2.1b). The functions g_L, g_R are the correction functions which must be chosen to obtain a stable scheme.

Step 3. We obtain the system of ODE by collocating the PDE at the solution points

$$\frac{du_j^e}{dt}(t) = -\frac{1}{\Delta x_e}\frac{\partial f_h}{\partial \xi}(\xi_j, t), \qquad 0 \leq j \leq N$$

which is solved in time by a Runge-Kutta scheme.

? Correction functions. The correction functions g_L, g_R should satisfy the end point conditions

$$\begin{aligned} g_L(0) &= 1, & g_R(0) &= 0 \\ g_L(1) &= 0, & g_R(1) &= 1 \end{aligned}$$

which ensures the continuity of the flux, i.e., $f_h(x^-_{e+\frac{1}{2}}, t) = f_h(x^+_{e+\frac{1}{2}}, t) = f_{e+\frac{1}{2}}(t)$. Moreover, we want them to be close to zero inside the element. There is a wide family of correction functions available in the literature [2, ?]. A family of correction functions depending on a parameter c were developed in [?] based on stability in a Sobolev-type norm. Two of these functions, the Radau and g2 correction functions, are of major interest since they correspond to commonly used DG formulations. The Radau correction function is a polynomial of degree $N+1$ which belongs to the family of [?] corresponding to the parameter $c = 0$ and given by

$$g_L(\xi) = \frac{(-1)^N}{2}\left[L_N(2\,\xi - 1) - L_{N+1}(2\,\xi - 1)\right], \qquad g_R(\xi) = \frac{1}{2}\left[L_N(2\,\xi - 1) + L_{N+1}(2\,\xi - 1)\right]$$

where $L_N : [-1, 1] \to \mathbb{R}$ is the Legendre polynomial of degree N. The resulting RKFR scheme can be shown to be identical to the nodal RKDG scheme using Gauss-Legendre nodes for solution points and quadrature. In the general class of [?], g_2 correction function of degree $N+1$ corresponds to $c = \frac{2\,(N+1)}{(2\,N+1)\,N\,(a_N\,N!)^2}$ where a_N is the leading coefficient of L_N; they are given by

$$g_L(\xi) = \frac{(-1)^N}{2}\left[L_N\,(2\,\xi-1) - \frac{(N+1)\,L_{N-1}\,(2\,\xi-1) + N\,L_{N+1}\,(2\,\xi-1)}{2\,N+1}\right]$$

$$g_R(\xi) = \frac{1}{2}\left[L_N\,(2\,\xi-1) + \frac{(N+1)\,L_{N-1}\,(2\,\xi-1) + N\,L_{N+1}\,(2\,\xi-1)}{2\,N+1}\right]$$

The resulting RKFR scheme can be shown to be identical to the nodal RKDG scheme using Gauss-Lobatto-Legendre points as solution points and for quadrature. We will perform Fourier stability analysis of the Lax-Wendroff scheme based on these correction functions in a later section. Note that the correction functions are usually defined in the interval $[-1, 1]$ but here we have written them for our reference interval which is $[0, 1]$.

Chapter 3
Lax-Wendroff Flux Reconstruction

3.1. Lax-Wendroff FR scheme

The Lax-Wendroff scheme combines the spatial and temporal discretization into a single step. The starting point is a Taylor expansion in time following the Cauchy-Kowalewski procedure where the PDE is used to rewrite some of the time derivatives in the Taylor expansion as spatial derivatives. Using Taylor expansion in time around $t = t_n$, we can write the solution at the next time level as

$$u^{n+1} = u^n + \sum_{m=1}^{N+1} \frac{\Delta t^m}{m!} \partial_t^m u^n + O(\Delta t^{N+2})$$

Since the spatial error is expected to be of $O(\Delta x^{N+1})$, we retain terms up to $O(\Delta t^{N+1})$ in the Taylor expansion, so that the over all accuracy is of order $N+1$ both space and time. Using the PDE, $\partial_t u = -\partial_x f$, we re-write time derivatives of the solution in terms of spatial derivatives of the flux

$$\partial_t^m u = -\partial_t^{m-1} \partial_x f = -(\partial_t^{m-1} f)_x, \qquad m = 1, 2, \ldots$$

so that

$$\begin{aligned} u^{n+1} &= u^n - \sum_{m=1}^{N+1} \frac{\Delta t^m}{m!} (\partial_t^{m-1} f)_x + O(\Delta t^{N+2}) \\ &= u^n - \Delta t \left[\sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!} \partial_t^m f \right]_x + O(\Delta t^{N+2}) \\ &= u^n - \Delta t \frac{\partial F}{\partial x}(u^n) + O(\Delta t^{N+2}) \end{aligned} \tag{3.1}$$

where

$$F(u) = \sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!} \partial_t^m f(u) = f(u) + \frac{\Delta t}{2} \partial_t f(u) + \ldots + \frac{\Delta t^N}{(N+1)!} \partial_t^N f(u) \tag{3.2}$$

Note that $F(u^n)$ is an approximation to the time average flux in the interval $[t_n, t_{n+1}]$ since it can be written as

$$F(u^n) = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \left[f(u^n) + (t - t_n) \partial_t f(u^n) + \ldots + \frac{(t-t_n)^N}{N!} \partial_t^N f(u^n) \right] dt \tag{3.3}$$

where the quantity inside the square brackets is the truncated Taylor expansion of the flux f in time. Equation (3.1) is the basis for the construction of the Lax-Wendroff method. Following the ideas in the RKFR scheme, we will first reconstruct the time average flux F inside each element by a continuous polynomial $F_h(\xi)$. Then truncating equation (3.1), the solution at the nodes is updated by a collocation scheme as follows

$$(u_j^e)^{n+1} = (u_j^e)^n - \frac{\Delta t}{\Delta x_e} \frac{dF_h}{d\xi}(\xi_j), \qquad 0 \le j \le N \tag{3.4}$$

This is the single step Lax-Wendroff update scheme for any order of accuracy. The major work in the above scheme is involved in the construction of the time average flux approximation F_h which is explained in subsequent sections.

3.1.1. Conservation property

The computation of correct weak solutions for non-linear conservation laws in the presence of discontinuous solutions requires the use of conservative numerical schemes. The Lax-Wendroff theorem shows that if a consistent, conservative method converges, then the limit is a weak solution. The method (3.4) is also conservative though it is not directly apparent; to see this multiply (3.4) by the quadrature weights associated with the solution points and sum over all the points in the e'th element,

$$\sum_{j=0}^{N} w_j (u_j^e)^{n+1} = \sum_{j=0}^{N} w_j (u_j^e)^n - \frac{\Delta t}{\Delta x_e} \sum_{j=0}^{N} w_j \frac{\partial F_h}{\partial \xi}(\xi_j)$$

The correction functions are of degree $N+1$ and the flux F_h is a polynomial of degree $\le N+1$. If the quadrature is exact for polynomials of degree atleast N, which is true for both GLL and GL points, then the quadrature is exact for the flux derivative term and we can write it as an integral, which leads to

$$\int_{\Omega_e} u_h^{n+1} dx = \int_{\Omega_e} u_h^n dx - \Delta t \left[F_{e+\frac{1}{2}} - F_{e-\frac{1}{2}} \right] \tag{3.5}$$

This shows that the total mass inside the cell changes only due to the boundary fluxes and the scheme is hence conservative.

3.1.2. Reconstruction of the time average flux

To complete the description of the LW method (3.4), we must explain the method for the computation of the time average flux F_h. The flux reconstruction $F_h(\xi)$ for a time interval $[t_n, t_{n+1}]$ is performed in three steps.

Step 1. Use the approximate Lax-Wendroff procedure to compute the time average flux F at all the solution points

$$F_j^e \approx F(\xi_j), \qquad 0 \le j \le N$$

The approximate LW procedure is explained in a subsequent section.

Step 2. Build a local approximation of the time average flux inside each element by interpolating at the solution points

$$F_h^\delta(\xi) = \sum_{j=0}^{N} F_j^e \, \ell_j(\xi)$$

which however may not be continuous across the elements. This is illustrated in Figure (2.1b).

Step 3. Modify the flux approximation $F_h^\delta(\xi)$ so that it becomes continuous across the elements. Let $F_{e+\frac{1}{2}}$ be some numerical flux function that approximates the flux F at $x = x_{e+\frac{1}{2}}$. Then the continuous flux approximation is given by

$$F_h(\xi) = \left[F_{e-\frac{1}{2}} - F_h^\delta(0) \right] g_L(\xi) + F_h^\delta(\xi) + \left[F_{e+\frac{1}{2}} - F_h^\delta(1) \right] g_R(\xi)$$

which is illustrated in Figure (2.1b). The correction functions g_L, g_R are chosen from the FR literature.

Step 4. The derivatives of the continuous flux approximation at the solution points can be obtained as

$$\partial_\xi F_h = \left[F_{e-\frac{1}{2}} - V_L^\top F \right] b_L + DF + \left[F_{e+\frac{1}{2}} - V_R^\top F \right] b_R, \qquad b_L = \begin{bmatrix} g_L'(\xi_0) \\ \vdots \\ g_L'(\xi_N) \end{bmatrix}, \qquad b_R = \begin{bmatrix} g_R'(\xi_0) \\ \vdots \\ g_R'(\xi_N) \end{bmatrix}$$

which can also be written as

$$\partial_\xi F_h = F_{e-\frac{1}{2}} b_L + D_1 F + F_{e+\frac{1}{2}} b_R, \qquad D_1 = D - b_L V_L^\top - b_R V_R^\top \tag{3.6}$$

where V_L, V_R are Vandermonde matrices which are defined in (2.1). The quantities $b_L, b_R, V_L, V_R, D, D_1$ can be computed once and re-used in all subsequent computations. They do not depend on the element and are computed on the reference element. Equation (3.6) contains terms that can be computed inside a single cell (middle term) and those computed at the faces (first and third terms) where it is required to use the data from two adjacent cells. The computation of the flux derivatives can thus be performed by looping over cells and then the faces.

3.1.3. Direct flux reconstruction (DFR) scheme

An alternate approach to flux reconstruction which does not require the choice of a correction function is based on the idea of direct flux reconstruction [?], which we adopt in the Lax-Wendroff scheme as follows. Let us take the solution points to be the $N+1$ Gauss-Legendre nodes, and define

$$\xi_{-1} = 0, \qquad \xi_{N+1} = 1$$

The Lagrange polynomials corresponding to the $N+3$ points $\{\xi_i, i = -1, 0, \ldots, N+1\}$ are given by

$$\tilde{\ell}_j(\xi) = \prod_{i=-1, i\neq j}^{N+1} \frac{\xi - \xi_i}{\xi_j - \xi_i} \in \mathbb{P}_{N+2}$$

We approximate the continuous flux in terms of these polynomials

$$F_h(\xi) = F_{e-\frac{1}{2}} \tilde{\ell}_{-1}(\xi) + \sum_{j=0}^{N} F_j^e \, \tilde{\ell}_j(\xi) + F_{e+\frac{1}{2}} \tilde{\ell}_{N+1}(\xi)$$

We can compute the spatial derivatives using a differentiation matrix $\tilde{D} \in \mathbb{R}^{(N+1)\times(N+3)}$

$$\tilde{D}_{ij} = \tilde{\ell}_j'(\xi_i), \qquad 0 \le i \le N, \quad -1 \le j \le N+1$$

Define b_L, b_R to be the first and last column of the matrix $\tilde{D}$ and D_1 to be the remaining columns

$$b_L = \tilde{D}\texttt{(:,-1)}, \qquad b_R = \tilde{D}\texttt{(:,N+1)}, \qquad D_1 = \tilde{D}\texttt{(0:N, 0:N)}$$

The flux derivatives at all the solution points can be computed as follows

$$\partial_\xi F_h = F_{e-\frac{1}{2}} b_L + D_1 F + F_{e+\frac{1}{2}} b_R$$

Note that the above equation has the same structure as (3.6) from the FR procedure but b_L, b_R, D_1 are obtained using different idea. In this DFR approach, we cannot use GLL points since then the boundary points $\xi_{-1} = \xi_0$, $\xi_{N+1} = \xi_N$ would be repeated and the Lagrange interpolation is not well-defined; if we use GL points, the resulting scheme is identical to the LWFR approach using Radau correction function in combination with GL points as solution points, as shown in Appendix sec:frdfr.

3.1.4. Approximate Lax-Wendroff procedure

The time average flux at the solution points F_j^e must be computed using (3.2). The usual approach is to use the PDE and replace time derivatives with spatial derivatives, but this leads to large amount of algebraic computations since we need to evaluate the flux Jacobian and its higher tensor versions. To avoid this process, we follow the ideas in [?, ?] and adopt an approximate Lax-Wendroff procedure. To present this idea in a concise and efficient form, we introduce the notation

$$u^{(m)} = \Delta t^m \partial_t^m u, \qquad f^{(m)} = \Delta t^m \partial_t^m f, \qquad m = 1, 2, \ldots$$

The time derivatives of the solution are computed using the PDE

$$u^{(m)} = -\Delta t \partial_x f^{(m-1)}, \qquad m = 1, 2, \ldots$$

The approximate Lax-Wendroff procedure uses a finite difference approximation applied at the solution points to compute the time derivatives of the fluxes. For example, a second order approximation is given by

$$f_t(\xi, t) \approx \frac{f\left(u\left(\xi, t + \Delta t\right)\right) - f\left(u\left(\xi, t - \Delta t\right)\right)}{2\,\Delta t}$$

The arguments to the flux are in turn approximated by a Taylor expansion in time

$$u\left(\xi, t \pm \Delta t\right) \approx u\left(\xi, t\right) \pm u_t\left(\xi, t\right) \Delta t$$

Using this approximation at the j'th solution point in an element, we get

$$f_j^{(1)} = f_t(\xi_j, t)\,\Delta t \approx \frac{1}{2}[f\left(u_j + u_j^{(1)}\right) - f\left(u_j - u_j^{(1)}\right)], \qquad u_j^{(1)} = u_t\left(\xi_j, t\right)\Delta t = -\frac{\Delta t}{\Delta x_e} f_\xi(\xi_j, t) \approx -\frac{\Delta t}{\Delta x_e}(Df)_j$$

It can be shown that the above approximation to f_t is second order accurate in Δt. Such approximations can be written for higher accuracy and for higher time derivatives [?, ?], and we summarize them below at different orders of accuracy which are used in the paper. The neglected term in the Taylor expansion (3.2) is of $O(\Delta t^{N+1})$, and hence the derivative approximation $\partial_t^m f$ must be computed to at least $O(\Delta t^{N+1-m})$ accuracy. The Lax-Wendroff procedure is applied in each element and so for simplicity of notation, we do not show the element index in the following sub-sections.

3.1.4.1. Second order scheme, $N = 1$

The time average flux at the solution points is given by

$$F = f + \frac{1}{2} f^{(1)}$$

where

$$\begin{aligned} u^{(1)} &= -\frac{\Delta t}{\Delta x_e} Df \\ f^{(1)} &= \frac{1}{2}\left[f\left(u + u^{(1)}\right) - f\left(u - u^{(1)}\right)\right] \end{aligned}$$

3.1.4.2. Third order scheme, $N = 2$

The time average flux at the solution points is given by

$$F = f + \frac{1}{2} f^{(1)} + \frac{1}{6} f^{(2)}$$

where

$$\begin{aligned} u^{(1)} &= -\frac{\Delta t}{\Delta x_e} Df \\ f^{(1)} &= \frac{1}{2}\left[f\left(u + u^{(1)}\right) - f\left(u - u^{(1)}\right)\right] \\ u^{(2)} &= -\frac{\Delta t}{\Delta x_e} Df^{(1)} \\ f^{(2)} &= f\left(u + u^{(1)} + \frac{1}{2}u^{(2)}\right) - 2\,f(u) + f\left(u - u^{(1)} + \frac{1}{2}u^{(2)}\right) \end{aligned}$$

3.1.4.3. Fourth order scheme, $N = 3$

For the fourth order scheme, the time average flux at the solution points reads as

$$F = f + \frac{1}{2}f^{(1)} + \frac{1}{6}f^{(2)} + \frac{1}{24}f^{(3)}$$

where

$$\begin{aligned}
u^{(1)} &= -\frac{\Delta t}{\Delta x_e} Df \\
f^{(1)} &= \frac{1}{12}\left[-f\left(u+2u^{(1)}\right) + 8\,f\left(u+u^{(1)}\right) - 8\,f\left(u-u^{(1)}\right) + f\left(u-2u^{(1)}\right)\right] \\
u^{(2)} &= -\frac{\Delta t}{\Delta x_e} Df^{(1)} \\
f^{(2)} &= f\left(u+u^{(1)}+\frac{1}{2}u^{(2)}\right) - 2\,f(u) + f\left(u-u^{(1)}+\frac{1}{2}u^{(2)}\right) \\
u^{(3)} &= -\frac{\Delta t}{\Delta x_e} Df^{(2)} \\
f^{(3)} &= \frac{1}{2}\left[f\left(u+2u^{(1)}+\frac{2^2}{2!}u^{(2)}+\frac{2^3}{3!}u^{(3)}\right) - 2\,f\left(u+u^{(1)}+\frac{1}{2!}u^{(2)}+\frac{1}{3!}u^{(3)}\right)\right. \\
&\qquad \left. +2\,f\left(u-u^{(1)}+\frac{1}{2!}u^{(2)}-\frac{1}{3!}u^{(3)}\right) - f\left(u-2u^{(1)}+\frac{2^2}{2!}u^{(2)}-\frac{2^3}{3!}u^{(3)}\right)\right]
\end{aligned}$$

3.1.4.4. Fifth order scheme, $N = 4$

The time average flux at the solution points for the fifth order scheme takes the form

$$F = f + \frac{1}{2}f^{(1)} + \frac{1}{6}f^{(2)} + \frac{1}{24}f^{(3)} + \frac{1}{120}f^{(4)}$$

where

$$\begin{aligned}
u^{(1)} &= -\frac{\Delta t}{\Delta x_e} Df \\
f^{(1)} &= \frac{1}{12}\left[-f\left(u+2u^{(1)}\right) + 8\,f\left(u+u^{(1)}\right) - 8\,f\left(u-u^{(1)}\right) + f\left(u-2u^{(1)}\right)\right] \\
u^{(2)} &= -\frac{\Delta t}{\Delta x_e} Df^{(1)} \\
f^{(2)} &= \frac{1}{12}\left[-f\left(u+2u^{(1)}+\frac{2^2}{2!}u^{(2)}\right) + 16\,f\left(u+u^{(1)}+\frac{1}{2!}u^{(2)}\right) - 30\,f(u)\right. \\
&\qquad \left. +16\,f\left(u-u^{(1)}+\frac{1}{2!}u^{(2)}\right) - f\left(u-2u^{(1)}+\frac{2^2}{2!}u^{(2)}\right)\right] \\
u^{(3)} &= -\frac{\Delta t}{\Delta x_e} Df^{(2)} \\
f^{(3)} &= \frac{1}{2}\left[f\left(u+2u^{(1)}+\frac{2^2}{2!}u^{(2)}+\frac{2^3}{3!}u^{(3)}\right) - 2\,f\left(u+u^{(1)}+\frac{1}{2!}u^{(2)}+\frac{1}{3!}u^{(3)}\right)\right. \\
&\qquad \left. +2\,f\left(u-u^{(1)}+\frac{1}{2!}u^{(2)}-\frac{1}{3!}u^{(3)}\right) - f\left(u-2u^{(1)}+\frac{2^2}{2!}u^{(2)}-\frac{2^3}{3!}u^{(3)}\right)\right] \\
u^{(4)} &= -\frac{\Delta t}{\Delta x_e} Df^{(3)} \\
f^{(4)} &= \left[f\left(u+2u^{(1)}+\frac{2^2}{2!}u^{(2)}+\frac{2^3}{3!}u^{(3)}+\frac{2^4}{4!}u^{(4)}\right)\right. \\
&\quad -4\,f\left(u+u^{(1)}+\frac{1}{2!}u^{(2)}+\frac{1}{3!}u^{(3)}+\frac{1}{4!}u^{(4)}\right) + 6\,f(u) \\
&\quad -4\,f\left(u-u^{(1)}+\frac{1}{2!}u^{(2)}-\frac{1}{3!}u^{(3)}+\frac{1}{4!}u^{(4)}\right) \\
&\quad \left. +f\left(u-2u^{(1)}+\frac{2^2}{2!}u^{(2)}-\frac{2^3}{3!}u^{(3)}+\frac{2^4}{4!}u^{(4)}\right)\right]
\end{aligned}$$

The above set of formulae shows the sequence of steps that have to be performed to compute the time average flux at various orders. The arguments of the fluxes used on the right hand side in these steps are built in a sequential manner. Note that all the equations are vectorial equations and are applied at each solution point.

3.2. Numerical flux

The numerical flux couples the solution between two neighbouring cells in a discontinuous Galerkin type method. In RK methods, the numerical flux is a function of the trace values of the solution at the faces. In the Lax-Wendroff scheme, we have constructed the time average flux at all the solution points inside the element and we want to use this information to compute the time averaged numerical flux at the element faces. The simplest numerical flux is based on Lax-Friedrich type approximation and is given by [5]

$$F_{e+\frac{1}{2}} = \frac{1}{2}[F^-_{e+\frac{1}{2}} + F^+_{e+\frac{1}{2}}] - \frac{1}{2}\lambda_{e+\frac{1}{2}}\left[u_h\left(x^+_{e+\frac{1}{2}}, t_n\right) - u_h\left(x^-_{e+\frac{1}{2}}, t_n\right)\right] \tag{3.7}$$

which consists of a central flux and a dissipative part. For linear advection equation $u_t + a\,u_x = 0$, the coefficient in the dissipative part of the flux is taken as $\lambda_{e+\frac{1}{2}} = |a|$, while for a non-linear PDE like Burger's equation, we take it to be

$$\lambda_{e+\frac{1}{2}} = \max\left\{|f'(\bar{u}^n_e)|, |f'(\bar{u}^n_{e+1})|\right\}$$

where $\bar{u}^n_e$ is the cell average solution in element Ω_e at time t_n, and will be referred to as Rusanov or local Lax-Friedrich [?] approximation. Note that the dissipation term in the above numerical flux is evaluated at time t_n whereas the central part of the flux uses the time average flux. Since the dissipation term contains solution difference at faces, we still expect to obtain optimal convergence rates, which is verified in numerical experiments. This numerical flux depends on the following quantities: $\{\bar{u}^n_e, \bar{u}^n_{e+1}, u_h\left(x^-_{e+\frac{1}{2}}, t_n\right), u_h\left(x^+_{e+\frac{1}{2}}, t_n\right), F^-_{e+\frac{1}{2}}, F^+_{e+\frac{1}{2}}\}$.

The numerical flux of the form (3.7) leads to somewhat reduced CFL numbers as shown by Fourier stability analysis in a later section, and also does not have upwind property even for linear advection equation. An alternate form of the numerical flux is obtained by evaluating the dissipation term using the time average solution, leading to the formula

$$F_{e+\frac{1}{2}} = \frac{1}{2}[F^-_{e+\frac{1}{2}} + F^+_{e+\frac{1}{2}}] - \frac{1}{2}\lambda_{e+\frac{1}{2}}[U^+_{e+\frac{1}{2}} - U^-_{e+\frac{1}{2}}] \tag{3.8}$$

where

$$U = \sum_{m=0}^{N} \frac{\Delta t^m}{(m+1)!}\partial_t^m u = u + \frac{\Delta t}{2}\partial_t u + \ldots + \frac{\Delta t^N}{(N+1)!}\partial_t^N u \tag{3.9}$$

is the time average solution. In this case, the numerical flux depends on the following quantities: $\{\bar{u}^n_e, \bar{u}^n_{e+1}, U^-_{e+\frac{1}{2}}, U^+_{e+\frac{1}{2}}, F^-_{e+\frac{1}{2}}, F^+_{e+\frac{1}{2}}\}$. We will refer to the above two forms of dissipation as D1 and D2, respectively. The dissipation model D2 is not computationally expensive compared to the D1 model since all the quantities required to compute the time average solution U are available during the Lax-Wendroff procedure. Numerical fluxes for the case of systems of hyperbolic equations are described in the Appendix. It remains to explain how to compute $F^{\pm}_{e+\frac{1}{2}}$ appearing in the central part of the numerical flux, which can be accomplished in two different ways, which we term AE and EA in the next two sub-sections.

Remark 3.1. In case of constant linear advection equation, $u_t + a\,u_x = 0$, $f^{(m)} = a\,u^{(m)}$ so that $F^e_j = a\,U^e_j$. Then, since $\lambda_{e+\frac{1}{2}} = |a|$, the numerical flux (3.8) becomes the upwind flux

$$F_{e+\frac{1}{2}} = \begin{cases} F^\delta_h(x^-_{e+\frac{1}{2}}), & a \geq 0 \\ F^\delta_h(x^+_{e+\frac{1}{2}}), & a < 0 \end{cases}$$

but the flux (3.7) does not have this upwind property. For a variable coefficient advection problem with flux $f = a(x)\,u$, we get $F^e_j = a(x_j)\,U^e_j$, the numerical flux (3.8) is

$$F_{e+\frac{1}{2}} = \frac{1}{2}[F^-_{e+\frac{1}{2}} + F^+_{e+\frac{1}{2}}] - \frac{1}{2}|a(x_{e+\frac{1}{2}})|\,[U^+_{e+\frac{1}{2}} - U^-_{e+\frac{1}{2}}] \tag{3.10}$$

which does not reduce to an upwind flux due to interpolation errors, though it will be close to it in the well resolved cases. In this case, we can define the upwind numerical flux as

$$F_{e+\frac{1}{2}} = \begin{cases} F^-_{e+\frac{1}{2}}, & a(x_{e+\frac{1}{2}}) \geq 0 \\ F^+_{e+\frac{1}{2}}, & a(x_{e+\frac{1}{2}}) < 0 \end{cases} \tag{3.11}$$

which is defined in terms of the time average flux only and does not make use of the solution.

Remark 3.2. For non-linear problems, we can also consider the global Lax-Friedrich and Roe type dissipation models which are given by

$$\lambda_{e+\frac{1}{2}} = \lambda = \max_e |f'(\bar{u}_e)|, \qquad \lambda_{e+\frac{1}{2}} = \left| f'\Big(\frac{\bar{u}_e + \bar{u}_{e+1}}{2}\Big)\right|$$

respectively. In the global Lax-Friedrich flux, the maximum is taken over the whole grid. For Burger's equation, we can consider an Osher type flux [?] which is given by

$$F_{e+\frac{1}{2}} = \begin{cases} F^-_{e+\frac{1}{2}} & \bar{u}_e, \bar{u}_{e+1} > 0 \\ F^+_{e+\frac{1}{2}} & \bar{u}_e, \bar{u}_{e+1} < 0 \\ F^-_{e+\frac{1}{2}} + F^+_{e+\frac{1}{2}} & \bar{u}_e \geq 0 \geq \bar{u}_{e+1} \\ 0 & \text{otherwise} \end{cases}$$

3.2.1. Numerical flux – average and extrapolate to face (AE)

In each element, the time average flux F^δ_h has been constructed using the Lax-Wendroff procedure. The simplest approximation that can be used for $F^\pm_{e+\frac{1}{2}}$ in the central part of the numerical flux is to extrapolate the flux F^δ_h to the faces,

$$F^\pm_{e+\frac{1}{2}} = F^\delta_h(x^\pm_{e+\frac{1}{2}})$$

We will refer to this approach with the abbreviation **AE**. However, as shown in the numerical results, this approximation can lead to sub-optimal convergence rates for some non-linear problems. Hence we propose another method for the computation of the inter-cell flux which overcomes this problem as explained next.

3.2.2. Numerical flux – extrapolate to face and average (EA)

Instead of extrapolating the time average flux from the solution points to the faces, we can instead build the time average flux at the faces directly using the approximate Lax-Wendroff procedure that is used at the solution points. The flux at the faces is constructed after the solution is evolved at all the solution points. In the following equations, α denotes either the left face (L) or the right face (R) of a cell. For $\alpha \in \{L, R\}$, we compute the time average flux at the faces of the e'th element by the following steps, where we suppress the element index since all the operations are performed inside one element.

Degree $N = 1$

$$\begin{aligned} u_\alpha &= V_\alpha^\top u \\ u^\pm_\alpha &= V_\alpha^\top (u \pm u^{(1)}) \\ f^{(1)}_\alpha &= \frac{1}{2}[f(u^+_\alpha) - f(u^-_\alpha)] \\ F_\alpha &= f(u_\alpha) + \frac{1}{2} f^{(1)}_\alpha \end{aligned}$$

Degree $N = 2$

$$\begin{aligned} u_\alpha &= V_\alpha^\top u \\ u^\pm_\alpha &= V_\alpha^\top (u \pm u^{(1)} + \frac{1}{2} u^{(2)}) \\ f^{(1)}_\alpha &= \frac{1}{2}[f(u^+_\alpha) - f(u^-_\alpha)] \\ f^{(2)}_\alpha &= f(u^+_\alpha) - 2\, f(u_\alpha) + f(u^-_\alpha) \\ F_\alpha &= f(u_\alpha) + \frac{1}{2} f^{(1)}_\alpha + \frac{1}{6} f^{(2)}_\alpha \end{aligned}$$

Degree $N = 3$

$$\begin{aligned} u_\alpha &= V_\alpha^\top u \\ u^\pm_\alpha &= V_\alpha^\top \left(u \pm u^{(1)} + \frac{1}{2!} u^{(2)} \pm \frac{1}{3!} u^{(3)} \right) \\ u^{\pm 2}_\alpha &= V_\alpha^\top \left(u \pm 2u^{(1)} + \frac{2^2}{2!} u^{(2)} \pm \frac{2^3}{3!} u^{(3)} \right) \\ f^{(1)}_\alpha &= \frac{1}{12}[-f(u^{+2}_\alpha) + 8\, f(u^+_\alpha) - 8\, f(u^-_\alpha) + f(u^{-2}_\alpha)] \\ f^{(2)}_\alpha &= f(u^-_\alpha) - 2\, f(u_\alpha) + f(u^+_\alpha) \\ f^{(3)}_\alpha &= \frac{1}{2}[f(u^{+2}_\alpha) - 2\, f(u^+_\alpha) + 2\, f(u^-_\alpha) - f(u^{-2}_\alpha)] \\ F_\alpha &= f(u_\alpha) + \frac{1}{2} f^{(1)}_\alpha + \frac{1}{6} f^{(2)}_\alpha + \frac{1}{24} f^{(3)}_\alpha \end{aligned}$$

Degree $N=4$

$$
\begin{aligned}
u_\alpha &= V_\alpha^\top u \\
u_\alpha^{\pm} &= V_\alpha^\top \left(u \pm u^{(1)} + \frac{1}{2!}u^{(2)} \pm \frac{1}{3!}u^{(3)} + \frac{1}{4!}u^{(4)} \right) \\
u_\alpha^{\pm 2} &= V_\alpha^\top \left(u \pm 2u^{(1)} + \frac{2^2}{2!}u^{(2)} \pm \frac{2^3}{3!}u^{(3)} + \frac{2^3}{3!}u^{(3)} \right) \\
f_\alpha^{(1)} &= \frac{1}{12}[-f(u_\alpha^{+2}) + 8\, f(u_\alpha^{+}) - 8\, f(u_\alpha^{-}) + f(u_\alpha^{-2})] \\
f_\alpha^{(2)} &= \frac{1}{12}[-f(u_\alpha^{+2}) + 16\, f(u_\alpha^{+}) - 30\, f(u_\alpha) + 16\, f(u_\alpha^{-}) - f(u_\alpha^{-2})] \\
f_\alpha^{(3)} &= \frac{1}{2}[f(u_\alpha^{+2}) - 2\, f(u_\alpha^{+}) + 2\, f(u_\alpha^{-}) - f(u_\alpha^{-2})] \\
f_\alpha^{(4)} &= [f(u_\alpha^{+2}) - 4\, f(u_\alpha^{+}) + 6\, f(u_\alpha) - 4\, f(u_\alpha^{-}) + f(u_\alpha^{-2})] \\
F_\alpha &= f(u_\alpha) + \frac{1}{2}\, f_\alpha^{(1)} + \frac{1}{6}\, f_\alpha^{(2)} + \frac{1}{24}\, f_\alpha^{(3)} + \frac{1}{120}\, f_\alpha^{(4)}
\end{aligned}
$$

We see that the solution is first extrapolated to the cell faces and the same finite difference formulae for the time derivatives of the flux which are used at the solution points, are also used at the faces. The numerical flux is computed using the time average flux built as above at the faces; the central part of the flux $F^{\pm}_{e+\frac{1}{2}}$ in equations (3.7), (3.8) are computed as

$$F^{-}_{e+\frac{1}{2}} = (F_R)_e, \qquad F^{+}_{e+\frac{1}{2}} = (F_L)_{e+1}$$

We will refer to this method with the abbreviation **EA**. The dissipative part of the flux is computed using either the solution at time t_n or the time average solution, which are extrapolated to the faces, leading to the D1 and D2 models, respectively.

Remark 3.3. The two methods AE and EA are different only when there are no solution points at the faces. E.g., if we use GLL solution points, then the two methods yield the same result since there is no interpolation error. For the constant coefficient advection equation, the above two schemes for the numerical flux lead to the same approximation but they differ in case of variable coefficient advection problems and when the flux is non-linear with respect to u. The effect of this non-linearity and the performance of the two methods are shown later using some numerical experiments.

3.3. Fourier stability analysis in 1-D

We now perform Fourier stability analysis of the LW schemes applied to the linear advection equation $u_t + a\, u_x = 0$ where a is the constant advection speed. We will assume that the advection speed a is positive and denote the CFL number by

$$\sigma = \frac{a\, \Delta t}{\Delta x}$$

Since $f^{(m)} = a\, u^{(m)}$, the time average flux at all the solution points is given by

$$F_e = aU_e \quad \text{where} \quad U_e = Tu_e \quad \text{and} \quad T = \sum_{m=0}^{N} \frac{1}{(m+1)!}(-\sigma D)^m$$

Then the discontinuous flux at the cell boundaries are given by

$$F_h^\delta(x^{+}_{e-\frac{1}{2}}) = V_L^\top F_e, \qquad F_h^\delta(x^{-}_{e+\frac{1}{2}}) = V_R^\top F_e$$

We can write the update equation in the form

$$u_e^{n+1} = -\sigma A_{-1} u_{e-1}^n + (I - \sigma A_0) u_e^n - \sigma A_{+1} u_{e+1}^n \tag{3.12}$$

where the matrices A_{-1}, A_0, A_{+1} depend on the choice of the dissipation model in the numerical flux. The EA and AE schemes for the flux are identical for this linear problem, and hence we do not make any distinction between them for Fourier stability analysis.

Dissipation model D1. The numerical flux is given by

$$F_{e+\frac{1}{2}} = \frac{1}{2}[V_R^\top F_e + V_L^\top F_{e+1}] - \frac{1}{2} a\, (V_L^\top u_{e+1} - V_R^\top u_e)$$

Since the flux difference at the faces is

$$F_{e-\frac{1}{2}} - F_h^\delta(x^+_{e-\frac{1}{2}}) = \frac{1}{2}aV_R^\top(T+I)u_{e-1} - \frac{1}{2}a\,V_L^\top(T+I)u_e$$

$$F_{e+\frac{1}{2}} - F_h^\delta(x^-_{e+\frac{1}{2}}) = \frac{1}{2}aV_L^\top(T-I)u_{e+1} - \frac{1}{2}a\,V_R^\top(T-I)u_e$$

the flux derivative at the solution points is given by

$$\partial_\xi F_h = \frac{1}{2}ab_LV_R^\top(T+I)u_{e-1} + \Big[aDT - \frac{1}{2}ab_LV_L^\top(T+I) - \frac{1}{2}ab_RV_R^\top(T-I)]u_e + \frac{1}{2}a\,b_RV_L^\top(T-I)u_{e+1}$$

Thus the matrices in (3.12) are given by

$$A_{-1} = \frac{1}{2}b_LV_R^\top(T+I), \qquad A_{+1} = \frac{1}{2}b_RV_L^\top(T-I), \qquad A_0 = DT - \frac{1}{2}b_LV_L^\top(T+I) - \frac{1}{2}b_RV_R^\top(T-I)$$

Dissipation model D2 . The numerical flux is given by

$$F_{e+\frac{1}{2}} = V_R^\top F_e = aV_R^\top Tu_e$$

and the flux differences at the face are

$$F_{e-\frac{1}{2}} - F_h^\delta(x^+_{e-\frac{1}{2}}) = aV_R^\top Tu_{e-1} - aV_L^\top Tu_e, \qquad F_{e+\frac{1}{2}} - F_h^\delta(x^-_{e+\frac{1}{2}}) = 0$$

so that the flux derivative at the solution points is given by

$$\partial_\xi F_h = (aV_R^\top Tu_{e-1} - aV_L^\top Tu_e)b_L + aDTu_e = ab_LV_R^\top Tu_{e-1} + a\,(DT - b_LV_L^\top T)u_e$$

Thus the matrices in (3.12) are given by

$$A_{-1} = b_LV_R^\top T, \qquad A_0 = DT - b_LV_L^\top T, \qquad A_{+1} = 0$$

The upwind character of the flux leads to $A_+ = 0$ and the right cell does not appear in the update equation.

Stability analysis. We assume a solution of the form $u_e^n = \hat{u}_k^n \exp(\mathrm{i}k\,x_e)$ where $\mathrm{i} = \sqrt{-1}$, k is the wave number which is an integer and $\hat{u}_k^n \in \mathbb{R}^{N+1}$ are the Fourier amplitudes; substituting this ansatz in (3.12), we find that the amplitudes evolve according to the equation

$$\hat{u}_k^{n+1} = H(\sigma,\kappa)\hat{u}_k^n, \qquad H = I - \sigma A_0 - \sigma\,A_{-1}\exp(-\mathrm{i}\kappa) - \sigma\,A_{+1}\exp(\mathrm{i}\kappa), \qquad \kappa = k\,\Delta x$$

where κ is the non-dimensional wave number. The eigenvalues of H depend on the CFL number σ and wave number κ, i.e., $\lambda = \lambda(\sigma,\kappa)$; for stability, all the eigenvalues of H must have magnitude less than or equal to one for all $\kappa \in [0, 2\,\pi]$, i.e.,

$$\lambda(\sigma) = \max_\kappa |\lambda(\sigma,\kappa)| \le 1$$

The CFL number is the maximum value of σ for which above stability condition is satisfied. This CFL number is determined approximately by sampling in the wave number space; we partition $\kappa \in [0, 2\,\pi]$ into a large number of uniformly spaced points κ_j and determine

$$\lambda(\sigma) = \max_j |\lambda(\sigma,\kappa_j)|$$

The values of σ are also sampled in some interval $[\sigma_{\min}, \sigma_{\max}]$ and the largest value of $\sigma_l \in [\sigma_{\min}, \sigma_{\max}]$ for which $\lambda(\sigma_l) \le 1$ is determined in a Python code. We start with a large interval $[\sigma_{\min}, \sigma_{\max}]$ and then progressively reduce the size of this interval so that the CFL number is determined to about three decimal places. The results of this numerical investigation of stability are shown in Table (3.1) for two correction functions and different polynomial degrees.

N	Radau			g_2		
	D1	D2	Ratio	D1	D2	Ratio
1	0.226	0.333	1.47	0.465	1.000	2.15
2	0.117	0.170	1.45	0.204	0.333	1.63
3	0.072	0.103	1.43	0.116	0.170	1.47
4	0.049	0.069	1.40	0.060	0.103	1.72

Table 3.1. CFL numbers for 1-D LWFR using the two dissipation models and correction functions

We see that dissipation model D2 has a higher CFL number compared to dissipation model D1. The CFL numbers for the g_2 correction function are also significantly higher than those for the Radau correction function. The LWFR scheme with Radau correction function is identical to DG scheme and the CFL numbers found here agree with those from the ADER-DG scheme [?, ?]. The optimality of these CFL numbers has been verified by experiment on the linear advection test case sec:cla, i.e., the solution eventually blows up if the time step is slightly higher than what is allowed by the CFL condition.

Remark 3.4. (Dirichlet boundary condition) The boundary conditions for hyperbolic problems are usually implemented in a weak manner through the fluxes. The boundary condition on the solution can be specified only at inflow boundaries, i.e., where the characteristics are entering the domain. For example, at the left boundary of the domain, say $x=0$, the boundary condition can be specified if $f'>0$. Assuming this is the case for our problem, let the boundary condition be given as $u(0,t)=g(t)$. It will be enforced by defining an upwind numerical flux at the boundary face, which is given by

$$F_{lb} = F_{lb}^{-} \approx \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(g(t)) dt$$

If the integral cannot be computed analytically, then it is approximated by quadrature in time. From (3.3), we see that integral must be at least accurate to $O(\Delta t^{N+1})$ which is of the same order as the neglected terms in (3.3). In the numerical tests, we use $(N+1)$-point Gauss-Legendre quadrature which ensures the required accuracy. If the right boundary is an outflow boundary, i.e., $f'>0$, then the flux across the right boundary is computed in an upwind manner using the interior solution, i.e., $F_{rb}=F_{rb}^{-}$ where F_{rb}^{-} is obtained from the Lax-Wendroff procedure.

3.4. TVD LIMITER

The computation of discontinuous solutions with high order methods can give oscillatory solutions which must be limited by some non-linear process. The a posteriori limiters developed in the context of RKDG schemes [?, ?] can be adopted in the framework of LWFR schemes. The limiter is applied in a post-processing step after the solution is updated to the new time level. The limiter is thus applied only once for each time step unlike in RKDG scheme where it has to be applied after each RK stage update. Let $u_h(x)$ denote the solution at time t_{n+1}. In element Ω_e, let the average solution be $\bar{u}_e$; define the backward and forward differences of the solution and cell averages by

$$\Delta^- u_e = \bar{u}_e - u_h(x^+_{e-\frac{1}{2}}), \qquad \Delta^+ u_e = u_h(x^-_{e+\frac{1}{2}}) - \bar{u}_e$$

$$\Delta^- \bar{u}_e = \bar{u}_e - \bar{u}_{e-1}, \qquad \Delta^+ \bar{u}_e = \bar{u}_{e+1} - \bar{u}_e$$

We limit the solution by comparing its variation within the cell with the difference of the neighbouring cell averages through a limiter function,

$$\Delta^- u_e^m = \text{minmod}(\Delta^- u_e, \Delta^- \bar{u}_e, \Delta^+ \bar{u}_e), \qquad \Delta^+ u_e^m = \text{minmod}(\Delta^+ u_e, \Delta^- \bar{u}_e, \Delta^+ \bar{u}_e)$$

where

$$\text{minmod}(a,b,c) = \begin{cases} s \min(|a|,|b|,|c|), & \text{if } s = \text{sign}(a) = \text{sign}(b) = \text{sign}(c) \\ 0, & \text{otherwise} \end{cases}$$

If $\Delta^- u_e^m \neq \Delta^- u_e$ or $\Delta^+ u_e^m \neq \Delta^+ u_e$, then the solution is deemed to be locally oscillatory and we modify the solution inside the cell by replacing it as a linear polynomial with a limited slope, which is taken to be the average limited slope. The limited solution polynomial in cell Ω_e is given by

$$u_h|_{\Omega_e} = \bar{u}_e + \frac{\Delta^- u_e^m + \Delta^+ u_e^m}{2}(2\xi - 1), \qquad \xi \in [0,1]$$

This limiter is known to clip smooth extrema since it cannot distinguish them from jump discontinuities. A small modification based on the idea of TVB limiters [?] can be used to relax the amount of limiting that is performed which leads to improved resolution of smooth extrema. The minmod function is replaced by

$$\overline{\text{minmod}}(a,b,c) = \begin{cases} a, & |a| \leq M \Delta x^2 \\ \text{minmod}(a,b,c), & \text{otherwise} \end{cases}$$

which requires the choice of a parameter M, which is an estimate of the second derivative of the solution at smooth extrema. In the case of systems of equations, the limiter is applied to the characteristic variables, which is known to yield better control on the spurious numerical oscillations [?]. The limiters used in this work are not able to provide high order accuracy and the development of better limiters for LW schemes is deferred to future work. Here we would like to show that with the use of TVD-type limiters, the LW and RK schemes provides similar resolution.

3.5. NUMERICAL RESULTS IN 1-D: SCALAR PROBLEMS

In this section, we present some numerical results to show the convergence rates and the comparison of different scheme parameters like correction function, solution points and dissipation model. For each problem in this section, the corresponding CFL number is chosen from Table 3.1. Here after, when we use the CFL numbers obtained using the Fourier stability analysis, we multiply it with a safety factor of 0.95. When D1 and D2 schemes are compared together, the CFL numbers of D1 schemes are used as these are lower; the same CFL numbers are used for the RKFR schemes. Up to degree $N = 3$, RKFR schemes use Runge-Kutta time integration of order $N+1$ with $N+1$ stages. In the $N = 4$ case, for non-linear problems there is no five stage Runge-Kutta method of order 5, see Chapter 32 of [?]. However, for linear, autonomous problems, the five stage SSPRK method in [?] is fifth order accurate, and we make use of it for the constant advection test cases with periodic boundary conditions and refer to it as SSPRK55. For non-linear or non-periodic problems, to make a fair comparison of LW and RK, we make use of the six stage, fifth order Runge-Kutta (RK65) time integration introduced in [?].

The RKFR and LWFR schemes are illustrated at a high level in Algorithm 1 and Algorithm 2, respectively, for solving a hyperbolic conservation law in a time interval $[0, T]$. Here we assume that an a posteriori limiter like a TVD/TVB limiter and a positivity limiter are applied in a post-processing step after the solution is updated. The LWFR scheme requires the application of the limiter only once per time step while the RKFR scheme requires multiple applications of the limiter depending on the number of RK stages. The limiter can be costly to apply for systems of equations where a characteristic approach and/or WENO-type limiter is used. In the present work, we use a simple TVD/TVB limiter but use characteristic limiting for systems.

Algorithm 3.1
RKFR scheme

$t = 0$
while $t < T$ **do**
  for each RK stage **do**
    Loop over cells and assemble rhs Loop over faces and assemble rhs Update solution to next RK stage Apply a posteriori limiter Apply positivity limiter
  end for
  $t = t + \Delta t$
end while

Algorithm 3.2
LWFR scheme

$t = 0$
while $t < T$ **do**
  Loop over cells and assemble rhs Loop over faces and assemble rhs Update solution to next time level Apply a posteriori limiter Apply positivity limiter $t = t + \Delta t$
end while

Chapter 4

Admissibility preserving subcell based blending limiter for Lax-Wendroff Flux Reconstruction

Plan for this chapter

1. Everything in the second paper
2. WCT performance improvement by going for higher order methods

Chapter 5

Multi-derivative Runge-Kutta schemes in Flux Reconstruction framework

1. Fourier stability analysis of MDRK methods
2. MDRK methods performance in the blending scheme, results compared

Chapter 6

10 Moment Problem

1. HLLC fluxes for 10 moment problem with admissibility preservation

Chapter 7

Lax-Wendroff Flux Reconstruction on curved geometries

Plan for this chapter

1. LWFR on curved meshes
2. Free stream preservation
3. 2 different kind of FR-DG schemes in literature
4. Error based time stepping

NOT READY TO BE READ YET

7.1. Transformation of conservation law

We are working with the conservation law

$$\boldsymbol{u}_t + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{0} \tag{7.1}$$

where

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f} = \sum_{n=1}^{3} \partial_{x_i} \boldsymbol{f}_i. \tag{7.2}$$

on a domain Ω which is partitioned into M non-overlapping quadrilateral elements Ω_e which can be deformed or curved

$$\Omega = \bigcup_{e=1}^{M} \Omega_e. \tag{7.3}$$

Each quadrilateral element can be mapped into a reference quadrilateral $\Omega_s = [-1,1]^3$ in the transformed space

$$\boldsymbol{\xi} = (\xi, \eta, \zeta) = (\xi^1, \xi^2, \xi^3),$$

with the generic mapping

$$\boldsymbol{x} = \Theta_e(\boldsymbol{\xi}), \tag{7.4}$$

where the e would usually be suppressed. For an arbitrary shaped element, the mapping will look like

$$\Theta(\boldsymbol{\xi}) = \sum_{p,q,r=0}^{P} \hat{\boldsymbol{x}}_{p,q,r}\, \phi_{pqr}(\boldsymbol{\xi}), \tag{7.5}$$

where $\{\phi_{pqr}\}$ is the Lagrange basis corresponding to the points $\{\hat{\boldsymbol{x}}_{p,q,r}\}$, explicitly written in terms of tensor product polynomials

$$\phi_{pqr}(\boldsymbol{\xi}) = \phi_p(\xi_1)\, \phi_q(\xi_2)\, \phi_r(\xi_3).$$

That is, the mapping is a degree P polynomial in each direction.

Define the covariant and contravariant basis

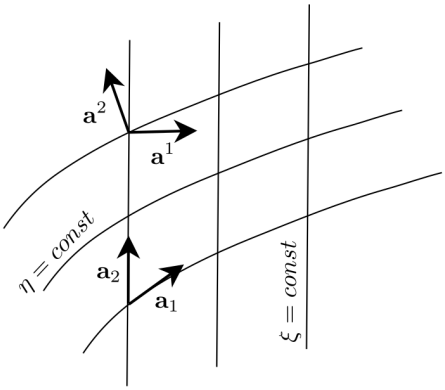

Figure 7.1. Covariant and contravariant coordinate vectors in relation to the coordinate lines

DEFINITION 7.1. *Covariant basis*

The covariant basis $\boldsymbol{a}_i$, $i=1,2,3$, *varies along a coordinate line. It is the tangent to the coordinate line*

$$\boldsymbol{a}_i = \lim_{\Delta\xi^i \to 0} \frac{\Delta \boldsymbol{x}}{\Delta \xi^i} = \frac{\partial \boldsymbol{x}}{\partial \xi^i}, \quad i=1,2,3. \tag{7.6}$$

Covariant basis vectors are the vectors that we can directly compute from the mapping function $\boldsymbol{x}(\xi)$. By its definition, covariant basis vector $\boldsymbol{a}_i$ is tangent to the curve where $\boldsymbol{x}$ varies along ξ_i.

Remark 7.2. Here we prove that $\boldsymbol{a}_1, \boldsymbol{a}_2$ will be tangent to the plane $\xi_3 = \xi_{30}$. For any point $\boldsymbol{X}^0 = \boldsymbol{x}(\xi_{10}, \xi_{20}, \xi_{30}) = \Theta(\xi_{10}, \xi_{20}, \xi_{30})$ on $\xi_3 = \xi_{30}$, we can find curves

$$\begin{aligned} \boldsymbol{X}^1(t) &= \boldsymbol{x}(\xi_{10}+t, \xi_{20}, \xi_{30}) = \Theta(\xi_{10}+t, \xi_{20}, \xi_{30}), \\ \boldsymbol{X}^2(t) &= \boldsymbol{x}(\xi_{10}, \xi_{20}+t, \xi_{30}) = \Theta(\xi_{10}, \xi_{20}+t, \xi_{30}), \end{aligned}$$

which lie on $\xi_3 = \xi_{30}$. Thus, the tangent vectors of $\xi_3 = \text{const}$ are clearly given by

$$\left.\frac{\partial \boldsymbol{x}}{\partial \xi_1}\right|_{\boldsymbol{x}=\boldsymbol{X}^0}, \quad \left.\frac{\partial \boldsymbol{x}}{\partial \xi_2}\right|_{\boldsymbol{x}=\boldsymbol{X}^0}.$$

DEFINITION 7.3. *Contravariant basis*

The contravariant basis vectors are normal to the ***coordinate lines*** *and are defined by the gradients*

$$\boldsymbol{a}^i = \nabla_{\boldsymbol{x}} \xi^i \quad i=1,2,3. \tag{7.7}$$

Remark 7.4. Formally, it appears that we need the inverse transformation $\boldsymbol{\xi} = \boldsymbol{X}^{-1}(\boldsymbol{x})$ to compute the contravariant basis vectors, but we will later prove that

$$\nabla_{\boldsymbol{x}} \xi^k = J^{-1}\, \boldsymbol{x}_i \times \boldsymbol{x}_j.$$

This will show that $\boldsymbol{a}^k = \nabla_{\boldsymbol{x}} \xi^k$ is perpendicular to $\boldsymbol{x}_i, \boldsymbol{x}_j$ which are the tangent vectors on $\xi_3 = \text{const}$, as proven in the previous remark.

The first step to transforming equations is to write how derivative operators transform under a mapping. First, we see that we can write a differential element in terms of the covariant basis vectors

$$d\boldsymbol{x} = \frac{\partial \boldsymbol{x}}{\partial \xi} d\xi + \frac{\partial \boldsymbol{x}}{\partial \eta} d\eta + \frac{\partial \boldsymbol{x}}{\partial \zeta} d\zeta = \sum_{i=1}^{3} \boldsymbol{a}_i\, d\xi^i. \tag{7.8}$$

The arc length is the magnitude of this vector

$$(ds)^2 = |d\boldsymbol{x}|^2 = \sum_{i=1}^{3}\sum_{j=1}^{3} \boldsymbol{a}_i \cdot \boldsymbol{a}_j\, d\xi^i\, d\xi^j = \sum_{i=1}^{3}\sum_{j=1}^{3} g_{ij}\, d\xi^i\, d\xi^j, \tag{7.9}$$

where we have used the definition of the *covariant metric tensor* $g_{ij} \equiv \boldsymbol{a}_i \cdot \boldsymbol{a}_j = g_{ji}$ to simplify the sums.

Next, we define the surface area element, which we compute from the cross product (Fig. 7.2)

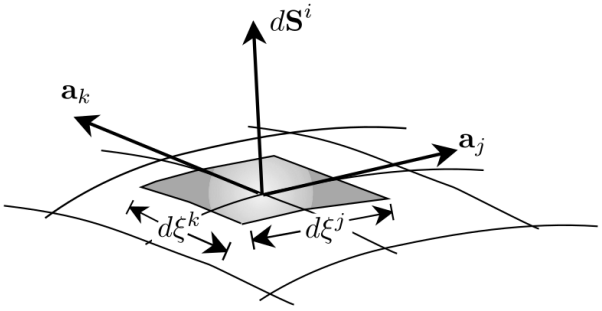

Figure 7.2. Surface element vector computed from covariant basis vectors

$$d\boldsymbol{S}^i = \boldsymbol{a}_j\, d\xi^j \times \boldsymbol{a}_k\, d\xi^k = (\boldsymbol{a}_j \times \boldsymbol{a}_k)\, d\xi^j\, d\xi^k. \tag{7.10}$$

Here, i, j, k are considered to be *cyclic*, so knowing one will fix the rest. Thus, a surface element in Cartesian space with size Δx in the $\hat{x}$ direction and Δy in $\hat{y}$ direction has the surface element

$$d\boldsymbol{S}^{(z)} = \Delta x\, \Delta y\, \hat{z}$$

according to this relation.

Finally the volume element extends the surface element in normal direction

$$dV = \boldsymbol{a}_i \cdot (\boldsymbol{a}_j \times \boldsymbol{a}_k)\, d\xi^i\, d\xi^j\, d\xi^k. \tag{7.11}$$

By definition of the covariant metric tensor g

$$\boldsymbol{a}_i \cdot (\boldsymbol{a}_j \times \boldsymbol{a}_k) = \sqrt{\det(g)} =: J, \tag{7.12}$$

giving the well-known result from calculus

$$dV = J d\xi^i\, d\xi^j\, d\xi^k. \tag{7.13}$$

We can now derive how different derivative operators transform under the mapping. First, recall that the divergence of a flux $\boldsymbol{f}$ is also given by

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f} = \lim_{\Delta V \to 0} \frac{1}{\Delta V} \int_{\partial \Delta V} \boldsymbol{f} \cdot d\boldsymbol{S}. \tag{7.14}$$

If we take the volume to be a differential pillbox, see Fig. 7.3, the surface integral can be broken into

$$\sum_{i=1}^{3} \{\boldsymbol{f} \cdot (\boldsymbol{a}_j \times \boldsymbol{a}_k) \Delta\xi^j \Delta\xi^k\,|^+ - \boldsymbol{f} \cdot (\boldsymbol{a}_j \times \boldsymbol{a}_k) \Delta\xi^j \Delta\xi^k\,|^- \}. \tag{7.15}$$

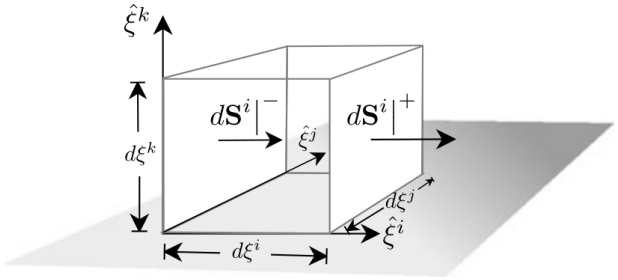

Figure 7.3. Differential volume element for divergence

With $\Delta V = J\, \Delta\xi^i\, \Delta\xi^j\, \Delta\xi^k$,

$$\frac{1}{\Delta V} \int_{\partial V} \boldsymbol{f} \cdot dS = \frac{1}{J} \sum_{i=1}^{3} \left\{ \frac{\boldsymbol{f} \cdot (\boldsymbol{a}_j \times \boldsymbol{a}_k) \Delta\xi^j \Delta\xi^k\,|^+ - \boldsymbol{f} \cdot (\boldsymbol{a}_j \times \boldsymbol{a}_k) \Delta\xi^j \Delta\xi^k\,|^-}{\Delta\xi^i\, \Delta\xi^j\, \Delta\xi^k} \right\}. \tag{7.16}$$

The limit as $\Delta V \to 0$ gives the divergence,

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}(\boldsymbol{x}) = \frac{1}{J} \sum_{i=1}^{3} \frac{\partial}{\partial \xi^i} (\boldsymbol{f} \cdot (\boldsymbol{a}_j \times \boldsymbol{a}_k))(\boldsymbol{\xi}). \tag{7.17}$$

This is known as the *conservative form* of the divergence because of its relation to the differential form of a conservation law.

We derive an alternative, called *non-conservative* form. By using the two expressions for divergence (7.2,7.17), supposing that $\boldsymbol{f} = \boldsymbol{c}$ gives

$$\nabla \cdot \boldsymbol{f} = \frac{1}{J} \sum_{i=1}^{3} \frac{\partial}{\partial \xi^i} (\boldsymbol{c} \cdot (\boldsymbol{a}_j \times \boldsymbol{a}_k)) = 0. \tag{7.18}$$

Since $\boldsymbol{c}$ is arbitrary, we must have

$$\sum_{i=1}^{3} \frac{\partial}{\partial \xi^i} (\boldsymbol{a}_j \times \boldsymbol{a}_k) = \boldsymbol{0}. \tag{7.19}$$

The vecotr identities (7.19) are known as *metric identities*. The metric identities primarily say that the divergence in computational space of specific products of the covariant basis must vanish.

Using the metric identities, we rewrite the divergence (7.17) in non-conservative form

$$\nabla \cdot \boldsymbol{f} = \frac{1}{J} \sum_{i=1}^{3} (\boldsymbol{a}_j \times \boldsymbol{a}_k) \cdot \frac{\partial \boldsymbol{f}}{\partial \xi^i}. \tag{7.20}$$

We now use the representation of divergence operator to write the representation of gradient operator in computational space. If we write out the terms in nonconservative form of the divergence, we get

$$\frac{\partial \boldsymbol{f}_1}{\partial x}+\frac{\partial \boldsymbol{f}_2}{\partial y}+\frac{\partial \boldsymbol{f}_3}{\partial z}=\nabla\cdot\boldsymbol{f}=\frac{1}{J}\sum_{i=1}^{3}(\boldsymbol{a}_j\times\boldsymbol{a}_k)\cdot\left(\frac{\partial \boldsymbol{f}_1}{\partial \xi^i}\hat{x}+\frac{\partial \boldsymbol{f}_2}{\partial \xi^i}\hat{y}+\frac{\partial \boldsymbol{f}_3}{\partial \xi^i}\hat{z}\right).$$

Matching the terms on either side, we will get, for a scalar, f,

$$\nabla f=\frac{1}{J}\sum_{i=1}^{3}(\boldsymbol{a}_j\times\boldsymbol{a}_k)\frac{\partial f}{\partial \xi^i}. \tag{7.21}$$

We use the conservative identities, (7.19) again, to derive the conservative form of gradient

$$\nabla f=\frac{1}{J}\sum_{i=1}^{3}\frac{\partial}{\partial \xi^i}[(\boldsymbol{a}_j\times\boldsymbol{a}_k)\,f]. \tag{7.22}$$

Note that all the transformations of the differential operators are in terms of quantitites that we can compute directly from the mapping, namely the covariant basis vectors, $\boldsymbol{a}_i$.

We use the gradient relations to find the contravariant basis vectors. To relate the contravariant and covariant basis vectors, we set $f=\xi^i$ in (7.22). Then, the gradient of ξ^i is

$$\nabla_{\boldsymbol{x}}\xi^i=\frac{1}{J}\sum_{l=1}^{3}(\boldsymbol{a}_j\times\boldsymbol{a}_k)\frac{\partial \xi^i}{\partial \xi^l}, \tag{7.23}$$

with (l,j,k) cyclic. Since $\partial\xi^i/\partial\xi^l=\delta_{i,l}$,

$$\nabla\xi^i=\boldsymbol{a}^i=\frac{1}{J}(\boldsymbol{a}_j\times\boldsymbol{a}_k), \tag{7.24}$$

or

$$J\boldsymbol{a}^i=\boldsymbol{a}_j\times\boldsymbol{a}_k. \tag{7.25}$$

Recall that our transformations from computational to physical domains are constructed so that physical boundaries correspond to either a $\xi=\text{const}$ or $\eta=\text{const}$ line, we can compute boundary normals from the contravariant basis vectors. Boundary normals are needed to set Neumann or normal flux boundary conditions. Since the contravariant basis vectors are normal to a coordinate line, a normal in the positive direction of the i^{th} coordinate variable is in the direction of $\boldsymbol{a}^i$. Thus, we have a normal in direction of increasing ξ^i to be

$$\hat{n}^i=\frac{|J|}{J}\frac{\boldsymbol{a}_j\times\boldsymbol{a}_k}{\|\boldsymbol{a}_j\times\boldsymbol{a}_k\|}, \tag{7.26}$$

where $\|\cdot\|$ denotes the Euclidean norm of the vector. Here i corresponds to the i^{th} direction boundary normals in reference coordinates.

The relationship between the covariant and contravariant basis vectors

$$J\boldsymbol{a}^i=\boldsymbol{a}_j\times\boldsymbol{a}_k,$$

allows us to rewrite the conservative forms of the differential operators that we have derived so far. We can write the divergence in the more compact form

$$\nabla\cdot\boldsymbol{F}=\frac{1}{J}\sum_{i=1}^{3}\frac{\partial}{\partial \xi^i}(J\boldsymbol{a}^i\cdot\boldsymbol{F}). \tag{7.27}$$

Similarly, the gradient becomes

$$\nabla f=\frac{1}{J}\sum_{i=1}^{3}J\boldsymbol{a}^i\frac{\partial f}{\partial \xi^i}. \tag{7.28}$$

Finally, the metric identitites (7.19) are equivalent to

$$\sum_{i=1}^{3}\frac{\partial J\boldsymbol{a}^i}{\partial \xi^i}=\boldsymbol{0}. \tag{7.29}$$

In practise, we will have to interpolate the metric terms to the grid be used in numerical scheme. We will denote the interpolation operator as $\mathcal{I}_N$ so that

$$\mathcal{I}_N(J\boldsymbol{a}^i) \tag{7.30}$$

denotes the interpolation of i^{th} contravariant to the mesh. The choice of interpolation operator will affect the free stream condition (Section 7.2.4) and its various choices are discussed in Section 7.2.5. Outside of these section, the metric terms will be written without the operator $\mathcal{I}_N$.

We explain the construction of grid and its reference maps to master cube $[0,1]^3$ in Section ???. Within each element Ω_e, performing change of variables with the reference map, the transformed conservation law is given by

$$\tilde{\boldsymbol{u}}_t + \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}} = \boldsymbol{0}, \tag{7.31}$$

where

$$\begin{aligned} \tilde{\boldsymbol{u}} &= J\boldsymbol{u}, \\ \tilde{\boldsymbol{f}}^i &= J\boldsymbol{a}^i \cdot \boldsymbol{f} = \sum_{n=1}^{3} J a_n^i \boldsymbol{f}_n. \end{aligned} \tag{7.32}$$

For the discretization, we will interpret the PDE as

$$J\boldsymbol{u}_t + \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}} = \boldsymbol{0}, \tag{7.33}$$

or

$$\boldsymbol{u}_t + \frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}} = \boldsymbol{0}. \tag{7.34}$$

Remark 7.5. The above analysis specializes to 2-D by setting $x_3 = \xi_3 = \zeta$ so that $\boldsymbol{a}_3 = (0,0,1)$.

7.2. Conservative Lax-Wendroff Flux Reconstruction (LWFR) on curvilinear grids

7.2.1. Discontinuous Galerkin

We define degree N Lagrange polynomial basis $\{\ell_{ijk}\}$ on the reference cell $\Omega_s = [-1,1]^3$. Let $\boldsymbol{u}^\delta, \boldsymbol{f}^\delta$ be the approximate solution and flux in the physical space, which need not be polynomials in the physical space. Corresponding to each Ω_e, we define the reference maps

$$\Theta_e : \Omega_s \to \Omega_e.$$

Thus, we can define degree N approximate solution and flux in the reference space Ω_s as

$$\begin{aligned} \hat{\boldsymbol{u}}_e^\delta(\boldsymbol{\xi}) &= \sum_{i,j,k=1}^{N+1} \hat{\boldsymbol{u}}_{e,ijk} \ell_{ijk}(\boldsymbol{\xi}), \\ \hat{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi}) &= \sum_{i,j,k=1}^{N+1} \boldsymbol{f}(\hat{\boldsymbol{u}}_{e,ijk}) \ell_{ijk}(\boldsymbol{\xi}), \end{aligned} \tag{7.35}$$

where $\hat{\boldsymbol{u}}_{e,ijk}$ are the unknowns we solve for so that

$$\hat{\boldsymbol{u}}_e^\delta(\boldsymbol{\xi}) = \boldsymbol{u}^\delta(\Theta_e(\boldsymbol{\xi})), \qquad \hat{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi}) = \boldsymbol{f}^\delta(\Theta_e(\boldsymbol{\xi})).$$

We can either formulate the DG scheme for the transformed PDE (7.33) or we can construct a weak formulation in the physical space and transform it to the reference cell. We will show that both can be done so that the obtained schemes are equivalent. We first derive the DG scheme for the tranformed conservation law. The first step is to multiply the transformed conservation law (7.33) with a test function φ which is a degree N polynomial in reference space

$$\int_{\Omega_s} J \frac{\partial \boldsymbol{u}_e^\delta}{\partial t} \varphi(\boldsymbol{\xi})\, d\boldsymbol{\xi} + \int_{\Omega_s} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e^\delta \, d\boldsymbol{\xi} = \boldsymbol{0}$$

where

$$\tilde{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi}) = J\boldsymbol{a}^i \cdot \hat{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi}).$$

Performing a formal integration by parts gives us

$$\int_{\Omega_s} J \frac{\partial \boldsymbol{u}_e^\delta}{\partial t} \varphi(\boldsymbol{\xi})\, d\boldsymbol{\xi} - \int_{\Omega_s} \tilde{\boldsymbol{f}}_e^\delta \cdot (\nabla_{\boldsymbol{\xi}} \varphi)\, d\boldsymbol{\xi} + \sum_{i=1}^{3} \int_{\partial\Omega_{s,i}} (\tilde{\boldsymbol{f}}_e^\delta \cdot \boldsymbol{n}_{s,i})^* \varphi \, dS_{\boldsymbol{\xi}} = \boldsymbol{0},$$

(s,i) corresponds to the two faces in i^{th} direction, $\boldsymbol{n}_{s,i}$ is the outward unit normal in the reference cell and $(\tilde{\boldsymbol{f}}_e^\delta \cdot \boldsymbol{n}_{s,i})^*$ is the numerical flux function. We now derive the other approach where the DG procedure is performed in the physical cell. Consider a test function φ which is a polynomial in the reference coordinates. After multiplying (7.1) with φ and performing a formal integration by parts, we get

$$\int_{\Omega_e} \frac{\partial \boldsymbol{u}_e^\delta}{\partial t} \varphi(\boldsymbol{\xi})\, d\boldsymbol{x} - \int_{\Omega_e} \boldsymbol{f}_e^\delta \cdot (\nabla_{\boldsymbol{x}} \varphi)\, d\boldsymbol{x} + \sum_{i=1}^{3} \int_{\partial\Omega_{e,i}} (\boldsymbol{f}_e^\delta \cdot \boldsymbol{n}_i)^* \varphi \, dS_{\boldsymbol{x}} = \boldsymbol{0},$$

where (e,i) corresponds to the image of the two faces of Ω_s in i^{th} direction, under the reference map Θ_e, $\boldsymbol{n}_i$ is the outward unit normal. The volume and surface change of variables are (7.13, 7.10)

$$d\boldsymbol{x} = J\, d\boldsymbol{\xi}, \qquad d S^i_{\boldsymbol{x}} = \|J\boldsymbol{a}^i\|\, d S_{\boldsymbol{\xi}},$$

where i indicates the two faces of Ω_e mapped by the two i direction faces of Ω_s under Θ_e.

The transformations of gradient and normal are (7.28, 7.26)

$$\nabla_{\boldsymbol{x}}\,\varphi = \sum_{i=1}^{3} \boldsymbol{a}^i \frac{\partial \varphi}{\partial \xi^i}, \qquad \hat{\boldsymbol{n}}^i = \frac{\boldsymbol{a}^i}{\|\boldsymbol{a}^i\|}.$$

Thus, performing a change of variables to map to Ω_s, we get

$$\int_{\Omega_s} \frac{\partial \boldsymbol{u}^\delta_e}{\partial t}\,\varphi(\boldsymbol{\xi})\, J\, d\boldsymbol{\xi} - \int_{\Omega_s} \boldsymbol{f}^\delta_e \cdot \left(\sum_{i=1}^{3} \boldsymbol{a}^i \frac{\partial \varphi}{\partial \xi^i}\right) J\, d\boldsymbol{\xi} + \sum_{i=1}^{3} \int_{\partial\Omega_{s,i}} (\boldsymbol{f}^\delta_e \cdot \boldsymbol{n}_i)^*\, \varphi\, \|J\boldsymbol{a}^i\|\, d S_{\boldsymbol{\xi}} = \boldsymbol{0}.$$

Now,

$$\begin{aligned}
\boldsymbol{f} \cdot \left(\sum_{i=1}^{3} \boldsymbol{a}^i \frac{\partial \varphi}{\partial \xi^i}\right) &= \sum_{n=1}^{3} \boldsymbol{f}_n \cdot \left(\sum_{i=1}^{3} a^i_n \frac{\partial \varphi}{\partial \xi^i}\right) J \\
&= \sum_{i=1}^{3} \left(\sum_{n=1}^{3} J a^i_n \boldsymbol{f}_n\right) \frac{\partial \varphi}{\partial \xi^i} \\
&= \sum_{i=1}^{3} (J\boldsymbol{a}^i \cdot \boldsymbol{f})\, (\nabla_{\boldsymbol{\xi}}\,\varphi)_i \\
&= \tilde{\boldsymbol{f}} \cdot (\nabla_{\boldsymbol{\xi}}\,\varphi)
\end{aligned}$$

Thus, the volume terms match. We now compare the surface terms. That is, we compare

$$(\boldsymbol{f}_e \cdot \boldsymbol{n}_i)^*\, \|J\boldsymbol{a}^i\|$$

and

$$(\tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i})^*$$

on a face. Now,

$$\tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i} \;=\; \boldsymbol{f} \cdot J\boldsymbol{a}^i \;=\; \|J\boldsymbol{a}^i\|\, \boldsymbol{f} \cdot \boldsymbol{n}_i.$$

Thus,

$$(\tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i})^* = (\boldsymbol{f}_e \cdot \boldsymbol{n}_i)^*\, \|J\boldsymbol{a}^i\|,$$

and we get our claim that the two approaches are equivalent.

7.2.2. Flux Reconstruction

We derive the collocation based Flux Reconstruction scheme directly from the DG scheme. We define the multi-index $\boldsymbol{p} = p_1 p_2 p_3$ and take the test function to be

$$\ell_{\boldsymbol{p}}(\boldsymbol{\xi}) = \ell_{p_1 p_2 p_3}(\boldsymbol{\xi}) = \ell_{p_1}(\xi^1)\, \ell_{p_2}(\xi^2)\, \ell_{p_3}(\xi^3).$$

The DG scheme in reference cell is given by

$$\int_{\Omega_s} J \frac{\partial \boldsymbol{u}^\delta_e}{\partial t} \ell_{\boldsymbol{p}}\, d\boldsymbol{\xi} - \int_{\Omega_s} \tilde{\boldsymbol{f}}_e \cdot (\nabla_{\boldsymbol{\xi}}\,\ell_{\boldsymbol{p}})\, d\boldsymbol{\xi} + \sum_{i=1}^{3} \int_{\partial\Omega^+_{s,i}} (\tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i})^*\, \ell_{\boldsymbol{p}}\, dS_{\boldsymbol{\xi}} - \sum_{i=1}^{3} \int_{\partial\Omega^-_{s,i}} (\tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i})^*\, \ell_{\boldsymbol{p}}\, dS_{\boldsymbol{\xi}} = \boldsymbol{0},$$

where $\partial\Omega^+_{s,i}$ denotes face where outward normal is so that the direction where i^{th} coordinate is one, i.e., $\boldsymbol{n}_{s,i} = \boldsymbol{e}_i$ and $\partial\Omega^-_{s,i}$ has outward normal $-\boldsymbol{n}_{s,i}$. We have also used

$$(\tilde{\boldsymbol{f}}_e \cdot (-\boldsymbol{n}_{s,i}))^* = -(\tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i})^*.$$

We can perform integration by parts if we use Gauss-Legendre quadrature points (integrals will be exact) or Gauss-Lobatto quadrature points (integrals will be exact along the relevant direction). Thus, the scheme is equivalent to

$$\begin{aligned}
&\int_{\Omega_s} J \frac{\partial \boldsymbol{u}^\delta_e}{\partial t} \ell_{\boldsymbol{p}}\, d\boldsymbol{\xi} + \int_{\Omega_s} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e\, \ell_{\boldsymbol{p}}\, d\boldsymbol{\xi} \\
&\quad + \sum_{i=1}^{3} \int_{\partial\Omega^+_{s,i}} ((\tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i})^* - \tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i})\, \ell_{\boldsymbol{p}}\, dS_{\boldsymbol{\xi}} - \sum_{i=1}^{3} \int_{\partial\Omega^-_{s,i}} ((\tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i})^* - \tilde{\boldsymbol{f}}_e \cdot \boldsymbol{n}_{s,i})\, \ell_{\boldsymbol{p}}\, dS_{\boldsymbol{\xi}} = \boldsymbol{0}.
\end{aligned}$$

Now, we collocate the scheme at solution points $\{\boldsymbol{\xi_p}=(\xi_{p_1},\xi_{p_2},\xi_{p_3}), p_i=1,\dots,N+1\}$ and use the following notations for a fixed $\boldsymbol{\xi_p}$

$$\begin{aligned} \boldsymbol{w} &= w_{p_1} w_{p_2} w_{p_3}, & \boldsymbol{w}_i &= \boldsymbol{w}/w_{p_i}, \\ \boldsymbol{\ell} &= \ell_{p_1 p_2 p_3}, & \boldsymbol{\ell}_{p_i}(\boldsymbol{\xi}) &= \ell_{p_j}(\xi^j)\,\ell_{p_k}(\xi^k), \quad i,j,k \text{ are in a cycle,} \\ \boldsymbol{\xi}_i^{L/R} &\text{is the point that agrees with } \boldsymbol{\xi} \text{ at } j,k \text{ and equals 0 or 1 for } L \text{ or } R \text{ at } i. \end{aligned}$$

Performing quadrature at solution points, the collocation scheme at the fixed $\boldsymbol{\xi_p}$ will be

$$\begin{aligned} &J\frac{\mathrm{d}\boldsymbol{u}_{e,\boldsymbol{p}}^{\delta}}{\mathrm{d}t}\boldsymbol{w}+\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{f}}_e(\boldsymbol{\xi_p})\,\boldsymbol{w} \\ &\quad+\sum_{i=1}^{3}\left((\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i})^*-\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i}\right)(\boldsymbol{\xi}_i^R)\,\ell_{p_i}(1)\,\boldsymbol{w}_i-\sum_{i=1}^{3}\left((\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i})^*-\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i}\right)(\boldsymbol{\xi}_i^L)\,\ell_{p_i}(0)\,\boldsymbol{w}_i=\boldsymbol{0}. \end{aligned}$$

Dividing by $J\boldsymbol{w}$ gives

$$\begin{aligned} &\frac{\mathrm{d}\boldsymbol{u}_{e,\boldsymbol{p}}^{\delta}}{\mathrm{d}t}+\frac{1}{J}\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{f}}_e(\boldsymbol{\xi_p}) \\ &\quad+\frac{1}{J}\sum_{i=1}^{3}\left((\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i})^*-\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i}\right)(\boldsymbol{\xi}_i^R)\,\frac{\ell_{p_i}(1)}{w_{p_i}}-\frac{1}{J}\sum_{i=1}^{3}\left((\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i})^*-\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i}\right)(\boldsymbol{\xi}_i^L)\,\frac{\ell_{p_i}(0)}{w_{p_i}}=\boldsymbol{0}. \end{aligned}$$

Then, using

$$\frac{\ell_{p_i}(-1)}{w_{p_i}},\frac{\ell_{p_i}(1)}{w_{p_i}}=\begin{cases} -g'_{\mathrm{Radau},L}(\xi_{p_i}),\, g'_{\mathrm{Radau},R}(\xi_{p_i}), & \text{GL solution points and quadrature,} \\ -g'_{\mathrm{Hu},L}(\xi_{p_i}),\, g'_{\mathrm{Hu},R}(\xi_{p_i}), & \text{GLL solution points and quadrature,} \end{cases}$$

we obtain

$$\begin{aligned} &\frac{\mathrm{d}\boldsymbol{u}_{e,\boldsymbol{p}}^{\delta}}{\mathrm{d}t}+\frac{1}{J}\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{f}}_e(\boldsymbol{\xi_p}) \\ &\quad+\frac{1}{J}\sum_{i=1}^{3}\left((\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i})^*-\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i}\right)(\boldsymbol{\xi}_i^R)\,g'_R(\xi_{p_i})+\frac{1}{J}\sum_{i=1}^{3}\left((\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i})^*-\tilde{\boldsymbol{f}}_e\cdot\boldsymbol{n}_{s,i}\right)(\boldsymbol{\xi}_i^L)\,g'_L(\xi_{p_i})=\boldsymbol{0}. \end{aligned} \tag{7.36}$$

7.2.3. Lax-Wendroff Flux Reconstruction (LWFR)

We perform the Lax-Wendroff procedure on the equation (7.34). We will discretize in time and do a polynomial approximation in space. For a degree N polynomial approximation, we will use the update

$$\boldsymbol{u}^{n+1}(\boldsymbol{\xi})=\boldsymbol{u}^n(\boldsymbol{\xi})+\sum_{k=1}^{N}\frac{\Delta t^k}{k!}\,\partial_t^{(k)}\,\boldsymbol{u}^n(\boldsymbol{\xi}).$$

Using the transformed PDE (7.34), we get

$$\boldsymbol{u}^{n+1}(\boldsymbol{\xi})=\boldsymbol{u}^n(\boldsymbol{\xi})-\frac{1}{J}\sum_{k=1}^{N}\frac{\Delta t^k}{k!}\,\partial_t^{(k-1)}\,(\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{f}}).$$

Shifting indices and writing in conservative form, we get

$$\boldsymbol{u}^{n+1}(\boldsymbol{\xi})=\boldsymbol{u}^n(\boldsymbol{\xi})-\frac{\Delta t}{J}\,\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{F}},$$

where the time averaged flux is

$$\tilde{\boldsymbol{F}}=\sum_{k=0}^{N}\frac{\Delta t^k}{(k+1)!}\,\partial_t^k\,\tilde{\boldsymbol{f}}.$$

We will find a local order $N+1$ approximation $\tilde{\boldsymbol{F}}_e^{\delta}$ to $\tilde{\boldsymbol{F}}$ and then, following (7.36), the LWFR update will be

$$\begin{aligned} &\boldsymbol{u}^{n+1}-\boldsymbol{u}^n+\frac{1}{J}\Delta t\,\nabla_{\boldsymbol{\xi}}\cdot\tilde{\boldsymbol{F}}_e^{\delta}(\boldsymbol{\xi}_{p_1p_2p_3}) \\ &\quad+\frac{1}{J}\Delta t\sum_{i=1}^{3}\left((\tilde{\boldsymbol{F}}_e^{\delta}\cdot\boldsymbol{n}_{s,i})^*-\tilde{\boldsymbol{F}}_e^{\delta}\cdot\boldsymbol{n}_{s,i}\right)(\boldsymbol{\xi}_i^R)\,g'_R(\xi_{p_i}) \\ &\quad+\frac{1}{J}\Delta t\sum_{i=1}^{3}\left((\tilde{\boldsymbol{F}}_e^{\delta}\cdot\boldsymbol{n}_{s,i})^*-\tilde{\boldsymbol{F}}_e^{\delta}\cdot\boldsymbol{n}_{s,i}\right)(\boldsymbol{\xi}_i^L)\,g'_L(\xi_{p_i})=\boldsymbol{0}. \end{aligned} \tag{7.37}$$

We now illustrate how to approximate $\tilde{\boldsymbol{F}}$ for various degrees. For $N=1$, we need to approximate

$$\tilde{\boldsymbol{F}}^{\delta}=\tilde{\boldsymbol{f}}^{\delta}+\frac{\Delta t}{2}\,\partial_t\,\tilde{\boldsymbol{f}}^{\delta},$$

so we need to approximate $\partial_t \tilde{\boldsymbol{f}}$, which we do as

$$\partial_t \tilde{\boldsymbol{f}} \approx \frac{\tilde{\boldsymbol{f}}^{n+1} - \tilde{\boldsymbol{f}}^{n-1}}{2\,\Delta t} \approx \frac{\tilde{\boldsymbol{f}}(\boldsymbol{u} + \Delta t\,\boldsymbol{u}_t) - \tilde{\boldsymbol{f}}(\boldsymbol{u} - \Delta t\,\boldsymbol{u}_t)}{2\,\Delta t} =: \partial_t \tilde{\boldsymbol{f}}^{\delta}. \tag{7.38}$$

We approximate $\boldsymbol{u}_t$ as

$$\boldsymbol{u}_t = -\frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e^{\delta}, \tag{7.39}$$

where $\tilde{\boldsymbol{f}}_e^{\delta}$ is the cell local approximation to the flux $\tilde{\boldsymbol{f}}_e^{\delta}$. For $N = 2$, we additionally need to approximate $\partial_{tt} \tilde{\boldsymbol{f}}$, which we do as

$$\begin{aligned} \partial_{tt} \tilde{\boldsymbol{f}} &\approx \frac{1}{\Delta t^2} (\tilde{\boldsymbol{f}}^{n+1} - 2\,\tilde{\boldsymbol{f}}^{n} + \tilde{\boldsymbol{f}}^{n-1}) \\ &\approx \frac{1}{\Delta t^2} \left(\tilde{\boldsymbol{f}}\left(\boldsymbol{u} + \Delta t\,\boldsymbol{u}_t + \frac{\Delta t}{2}\,\boldsymbol{u}_{tt} \right) - 2\,\tilde{\boldsymbol{f}}(\boldsymbol{u}) + \tilde{\boldsymbol{f}}\left(\boldsymbol{u} - \Delta t\,\boldsymbol{u}_t + \frac{\Delta t}{2}\,\boldsymbol{u}_{tt} \right) \right) \\ &=: \partial_{tt} \tilde{\boldsymbol{f}}^{\delta}, \end{aligned}$$

where we approximate $\boldsymbol{u}_{tt}$ as

$$\boldsymbol{u}_{tt} = -\frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot \partial_t \tilde{\boldsymbol{f}}^{\delta}. \tag{7.40}$$

The procedure for other degrees will be similar.

7.2.4. Free stream preservation for LWFR

We will show that the usual metric identities

$$\sum_{i=1}^{3} \partial_{\xi^i} (\mathcal{I}_N(J \boldsymbol{a}^i)) = \boldsymbol{0},$$

where $\mathcal{I}_N$ is an interpolation operator (7.30) used to approximate the metric terms, are sufficient for free stream preservation of LWFR. Assuming $\boldsymbol{u}^n = \underline{\boldsymbol{c}}$ and $\boldsymbol{f}(\underline{\boldsymbol{c}}) = \boldsymbol{c}$, we will begin by proving that

$$\tilde{\boldsymbol{F}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta}.$$

For the constant flux, $\tilde{\boldsymbol{f}}$ will be

$$\tilde{\boldsymbol{f}}_i = \mathcal{I}_N(J \boldsymbol{a}^i) \cdot \boldsymbol{c} = \sum_{n=1}^{3} \mathcal{I}_N(J a_n^i)\, \boldsymbol{c}_n.$$

This will give, using (7.39),

$$\begin{aligned} \boldsymbol{u}_t &= -\frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e^{\delta} = -\frac{1}{J} \sum_{i=1}^{3} \partial_{\xi_i} \tilde{\boldsymbol{f}}_{e,i}^{\delta} \\ &= -\frac{1}{J} \sum_{i=1}^{3} \partial_{\xi_i} \left(\sum_{n=1}^{3} \mathcal{I}(J a_n^i)\, \boldsymbol{c}_n \right) \\ &= -\sum_{n=1}^{3} \left(\sum_{i=1}^{3} \partial_{\xi_i} \mathcal{I}(J a_n^i) \right) \boldsymbol{c}_n = \boldsymbol{0}, \end{aligned}$$

For $N = 1$, by (7.38), this proves that

$$\partial_t \tilde{\boldsymbol{f}}^{\delta} = \frac{\tilde{\boldsymbol{f}}(\boldsymbol{u} + \Delta t\,\boldsymbol{u}_t) - \tilde{\boldsymbol{f}}(\boldsymbol{u} - \Delta t\,\boldsymbol{u}_t)}{2\,\Delta t} = \boldsymbol{0}.$$

Thus, we obtain

$$\tilde{\boldsymbol{F}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta} + \frac{\Delta t}{2}\, \partial_t \tilde{\boldsymbol{f}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta}.$$

Building on this, for $N = 2$, by (7.40),

$$\boldsymbol{u}_{tt} = -\frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot \partial_t \tilde{\boldsymbol{f}}^{\delta} = \boldsymbol{0},$$

which will prove

$$\partial_{tt} \tilde{\boldsymbol{f}}^{\delta} = \boldsymbol{0},$$

and we get

$$\tilde{\boldsymbol{F}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta} + \frac{\Delta t}{2}\, \partial_t \tilde{\boldsymbol{f}}^{\delta} + \frac{\Delta t^2}{3!}\, \partial_{tt} \tilde{\boldsymbol{f}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta}.$$

The claim shall similarly hold for all degrees. Thus, substituting

$$\tilde{\boldsymbol{F}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta} = \{ J \boldsymbol{a}^i \cdot \boldsymbol{c} \}_{i=1}^{3}$$

in (7.37) gives

$$\begin{aligned}\boldsymbol{u}^{n+1}-\boldsymbol{u}^{n}+\frac{1}{J}\Delta t\left(\sum_{i=1}^{3}\partial_{\xi^i}\mathcal{I}_N(J\boldsymbol{a}^i)\right)\cdot\boldsymbol{c}\\ +\frac{1}{J}\Delta t\sum_{i=1}^{3}\left(((J\boldsymbol{a}^i\cdot\boldsymbol{c})\cdot\boldsymbol{n}_{s,i})^*-(J\boldsymbol{a}^i\cdot\boldsymbol{c})\cdot\boldsymbol{n}_{s,i})(\boldsymbol{\xi}_i^R)\,g_R'(\xi_{p_i})\right.\\ +\frac{1}{J}\Delta t\sum_{i=1}^{3}\left(((J\boldsymbol{a}^i\cdot\boldsymbol{c})\cdot\boldsymbol{n}_{s,i})^*-(J\boldsymbol{a}^i\cdot\boldsymbol{c})\cdot\boldsymbol{n}_{s,i})(\boldsymbol{\xi}_i^L)\,g_L'(\xi_{p_i})=\boldsymbol{0}.\end{aligned}\tag{7.41}$$

The volume term vanishes by the metric identities and the surface terms vanish by conformality of the mesh.

7.2.5. Satisfying metric identities

Now that we have reduced metric identities to

$$\sum_{i=1}^{3}\partial_{\xi^i}\mathcal{I}_N(J\boldsymbol{a}^i)=\boldsymbol{0},$$

we show conditions under which they are satisfied for various choices of $\mathcal{I}_N$.

Lemma 7.6. $\nabla u\times\nabla v=-\nabla\times(v\,\nabla u)=\nabla\times(u\,\nabla v)$

Proof. The result is a direct consequence of the basic vector calculus identity

$$\nabla\times(v\,\nabla u)=\nabla v\times\nabla u+v\,\nabla\times\nabla u$$

and $\nabla\times\nabla u=0$. The identity is proven as

$$\begin{aligned}\nabla\times(v\,\nabla u) &= (v\,\nabla u)_{j,i}\,\epsilon_{ijk}\,\boldsymbol{e}_k\\ &= (v\,u_{,j})_{,i}\,\epsilon_{ijk}\,\boldsymbol{e}_k\\ &= v_{,i}\,u_{,j}\,\epsilon_{ijk}\,\boldsymbol{e}_k+v\,u_{,ij}\,\epsilon_{ijk}\,\boldsymbol{e}_k\\ &= \nabla v\times\nabla u+v\,\nabla\times\nabla u\end{aligned}$$

Since $\nabla\times\nabla u=0$, we get the first identity. Swapping u and v, we get the second identity. □

The lemma gives us

$$\nabla_{\boldsymbol{\xi}}\,x_m\times\nabla_{\boldsymbol{\xi}}\,x_l=-\nabla_{\boldsymbol{\xi}}\times(x_l\,\nabla_{\boldsymbol{\xi}}\,x_m).$$

Writing the left-hand side of this explicitly gives

$$\begin{aligned}\nabla_{\boldsymbol{\xi}}\,x_m\times\nabla_{\boldsymbol{\xi}}\,x_l \;=\; &\left[\frac{\partial x_m}{\partial\eta}\frac{\partial x_l}{\partial\zeta}-\frac{\partial x_m}{\partial\zeta}\frac{\partial x_l}{\partial\eta}\right]\hat{x}+\left[\frac{\partial x_m}{\partial\zeta}\frac{\partial x_l}{\partial\xi}-\frac{\partial x_m}{\partial\xi}\frac{\partial x_l}{\partial\zeta}\right]\hat{y}\\ &+\left[\frac{\partial x_m}{\partial\xi}\frac{\partial x_l}{\partial\eta}-\frac{\partial x_m}{\partial\eta}\frac{\partial x_l}{\partial\xi}\right]\hat{z}.\end{aligned}\tag{7.42}$$

Recall that

$$J\boldsymbol{a}^i=\boldsymbol{a}_j\times\boldsymbol{a}_k=\frac{\partial\boldsymbol{x}}{\partial\xi_j}\times\frac{\partial\boldsymbol{x}}{\partial\xi_k}$$

Defining $\boldsymbol{e}_i$ to be the Cartesian unit vector in the i^{th} derection, we will show

$$Ja_n^i=J\boldsymbol{a}^i\cdot\boldsymbol{e}_n=-\boldsymbol{e}_i\cdot\nabla_\xi\times(x_l\,\nabla_{\boldsymbol{\xi}}\,x_m),\quad i=1,2,3,\quad n=1,2,3,\quad (n,m,l)\,\text{cyclic}.$$

We first illustrate this for $i=1$

$$\begin{aligned}-\boldsymbol{e}_1\cdot\nabla_\xi\times(x_l\,\nabla_\xi x_m) &= \frac{\partial x_m}{\partial\xi^2}\frac{\partial x_l}{\partial\xi^3}-\frac{\partial x_m}{\partial\xi^3}\frac{\partial x_l}{\partial\xi^2}\\ &= \frac{\partial\boldsymbol{x}}{\partial\xi_2}\times\frac{\partial\boldsymbol{x}}{\partial\xi_3}\cdot\boldsymbol{e}_1\end{aligned}$$

To see the general case, write

$$\begin{aligned}\nabla_\xi\,x_m\times\nabla_\xi\,x_l &= \sum_{i=1}^{3}\left[\frac{\partial x_m}{\partial\xi^j}\frac{\partial x_l}{\partial\xi^k}-\frac{\partial x_m}{\partial\xi^k}\frac{\partial x_l}{\partial\xi^j}\right]\boldsymbol{e}_i \qquad (i,j,k)\,\text{cycle},\\ &= \sum_{i=1}^{3}\left(\frac{\partial\boldsymbol{x}}{\partial\xi_j}\times\frac{\partial\boldsymbol{x}}{\partial\xi_k}\cdot\boldsymbol{e}_n\right)\boldsymbol{e}_i \qquad (n,m,l)\,\text{cycle}.\end{aligned}\tag{7.43}$$

Thus, we get our claim and we write it in the following theorem

THEOREM 7.7.

$$Ja_n^i = -\hat{x}_i \cdot \nabla_{\boldsymbol{\xi}} \times (x_l \nabla_{\boldsymbol{\xi}} x_m), \quad i=1,2,3, \quad n=1,2,3, \quad (n,m,l)\,\text{cyclic}. \tag{7.44}$$

This is the curl form of metric terms. Notice that metric identities follow from these immediately since

$$\sum_{i=1}^{3} \frac{\partial (Ja_n^i)}{\partial \xi^i} = -\nabla_{\boldsymbol{\xi}} \cdot (\nabla_{\boldsymbol{\xi}} \times (x_l \nabla_{\boldsymbol{\xi}} x_m)) = 0, \quad n=1,2,3 \quad (n,m,l)\,\text{cyclic}.$$

We will denote I_N as the operator which will interpolate a function to a degree N polynomial using the solution point values. We are now ready to see three approaches for computing the metric terms

1. **Cross product form**

$$\mathcal{I}_N(J\vec{a}^i) = I_N\left(\frac{\partial \boldsymbol{x}}{\partial \xi^j} \times \frac{\partial \boldsymbol{x}}{\partial \xi^k}\right), \quad i=1,2,3, \quad (i,j,k) \quad \text{cyclic}. \tag{7.45}$$

   Here, $\boldsymbol{x}$ represents the mapping from the element to the reference element.

2. **Conservative curl form**

$$\begin{aligned} \mathcal{I}_N(Ja_n^i) &= -\boldsymbol{e}_i \cdot \nabla_{\boldsymbol{\xi}} \times (I_N(x_l \nabla_{\boldsymbol{\xi}} x_m)), \\ i &= 1,2,3, \quad n=1,2,3; (n,m,l)\,\text{cyclic}. \end{aligned} \tag{7.46}$$

3. **Invariant curl form**

$$\begin{aligned} Ja_n^i &= -\frac{1}{2}\boldsymbol{e}_i \cdot \nabla_{\boldsymbol{\xi}} \times [I_N(x_l \nabla_{\boldsymbol{\xi}} x_m - x_m \nabla_{\boldsymbol{\xi}} x_l)], \\ i &= 1,2,3 \quad n=1,2,3, \quad (n,m,l)\,\text{cyclic}. \end{aligned} \tag{7.47}$$

We will consider the evaluation of the metric terms in two and three space dimensions separately.

7.2.5.1. Evaluating metrics in two space dimensions

In two space dimensions, the cross-product form (7.45) of $J\boldsymbol{a}^i$ reduces to

$$\begin{aligned} J\boldsymbol{a}^1 &= \boldsymbol{a}^2 \times \boldsymbol{a}^3 = y_\eta \boldsymbol{e}_1 - x_\eta \boldsymbol{e}_2, \\ J\boldsymbol{a}^2 &= \boldsymbol{a}^3 \times \boldsymbol{a}^1 = -y_\xi \boldsymbol{e}_1 + x_\xi \boldsymbol{e}_2, \end{aligned} \tag{7.48}$$

where $\boldsymbol{x}(\xi,\eta) = x(\xi,\eta)\,\boldsymbol{e}_1 + y(\xi,\eta)\,\boldsymbol{e}_2$. In that case, we have the following result:

THEOREM 7.8. *On a well-constructed mesh in two space dimensions, the cross product form (7.45) of the metric terms satisfies the metric identity if $\boldsymbol{x} \in P_N$.*

Proof. If $\boldsymbol{x} \in P_N$, the 2D cross product form expression (7.48) gives us

$$J\boldsymbol{a}^i \in P_N \quad i=1,2.$$

Thus,

$$\mathcal{I}_N(Ja_n^i) = (Ja_n^i) = Ja_n^i.$$

Since the divergence of $J\vec{a}^i$ is zero, we get the metric identity.

On the other hand, if the element faces are not in P_N (i.e., of degree greated than N), then

$$I_N(Ja_n^i) \neq Ja_n^i.$$

Since, in general, we can't commute differentiation and interpolation, we won't be able to obtain the zero divergence of $I_N(Ja_n^i)$. Thus, with the cross product form where mappings are not in P_N, the metric identitites are not satisfied. □

Remark 7.9. Theorem 7.8 may be non-inituitive as one might expect a higher resolution boundary to improve the accuracy.

Remark 7.10. Finally, if we interpolate the mapping before differentiating, i.e., compute

$$\begin{aligned} \mathcal{I}_N J\boldsymbol{a}^1 &= (I_N y)_\eta \boldsymbol{e}_1 - (I_N x)_\eta \boldsymbol{e}_2, \\ \mathcal{I}_N J\boldsymbol{a}^2 &= -(I_N y)_\xi \boldsymbol{e}_1 + (I_N x)_\xi \boldsymbol{e}_2, \end{aligned}$$

then Theorem 7.8 holds and metric identies are satisfied. This is analytically equivalent to computing the curl form in 2D as

$$\begin{aligned} \mathcal{I}_N(Ja_n^i) &= -\boldsymbol{e}_i \cdot \nabla_{\boldsymbol{\xi}} \times (I_N(x_l \nabla_{\boldsymbol{\xi}} (I_N x_m))), \\ i &= 1,2, \quad n=1,2, \quad (n,m,l)\,\text{cyclic}. \end{aligned}$$

7.2.5.2. Evaluating metrics in three space dimensions

In 3D, the cross product form (7.45) does not satisfy the metric identities except in very special cases:

Theorem 7.11. *We consider well-constructed mesh in three space dimensions with the cross product form (7.45). The metric identities are not satisfied if $q > N/2$. The metric identities are satisfied, however, if $q \le N/2$.*

Proof. Unlike 2D, the cross products will now involve multiplication of degree N polynomials. So, the condition that $q \le N/2$ is natural. □

However, the other t[2]wo forms - conservative curl form (7.46) and invariant curl form (7.47), always satsify the metric identities, since the interpolation is performed before the curl is computed

Theorem 7.12. *On a well-constructed mesh in three space dimensions, the conservative curl form (7.46) and invariant curl form (7.47) of the metric terms satisfy the metric identities for all mappings $\boldsymbol{x}$.*

Proof. If the conservative curl form is used, then

$$\begin{aligned} \sum_{i=1}^{3} \frac{\partial \mathcal{I}_N(J a_n^i)}{\partial \xi^i} &= -\nabla_{\boldsymbol{\xi}} \cdot \nabla_{\boldsymbol{\xi}} \times (I_N(x_l \nabla_{\boldsymbol{\xi}} x_m)) = 0, \\ n &= 1,2,3, \quad (n,m,l) \quad \text{cyclic}, \end{aligned}$$

so the result is estabilished. Similarly, for the invariant curl form

$$\begin{aligned} \sum_{i=1}^{3} \frac{\partial \mathcal{I}_N(J a_n^i)}{\partial \xi^i} &= -\frac{1}{2} \nabla \cdot \nabla_{\xi} \times [I_N (x_l \nabla_{\boldsymbol{\xi}} x_m - x_m \nabla_{\boldsymbol{\xi}} x_l)] = 0, \\ n &= 1,2,3, \quad (n,m,l) \,\text{cyclic}. \end{aligned}$$

□

7.3. Non-conservative Lax-Wendroff Flux Reconstruction (FR) on curvilinear grids

This scheme has been labeled non-conservative by [1] because the flux derivative on the scheme obtained after quadrature is not the flux derivative in (7.31), but the one in (6.39) of [3].

7.3.1. Non-conservative Discontinuous Galerkin (DG) method

We define degree N Lagrange polynomial basis $\{\ell_{ijk}\}$ on the reference cell $\Omega_s = [0,1]^3$. Let $\boldsymbol{u}^\delta$, $\boldsymbol{f}^\delta$ be the approximate solution and flux in the physical space, which need not be polynomials. Corresponding to each Ω_e, we define as reference map

$$\Theta_e \colon \Omega_s \to \Omega_e,$$

using which we can define degree N approximate solution and flux in the reference space

$$\begin{aligned} \hat{\boldsymbol{u}}_e^\delta(\boldsymbol{\xi}) &= \sum_{i,j,k=0}^{N} \hat{\boldsymbol{u}}_{e,ijk} \, \ell_{ijk}(\boldsymbol{\xi}), \\ \hat{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi}) &= \sum_{i,j,k=0}^{N} \boldsymbol{f}(\boldsymbol{u}_{ijk}) \, \ell_{ijk}(\boldsymbol{\xi}), \end{aligned}$$

and the physical quantities are then defined implicitly

$$\begin{aligned} \boldsymbol{u}^\delta(\Theta_e^{-1}(\boldsymbol{\xi})) &= \hat{\boldsymbol{u}}_e^\delta(\boldsymbol{\xi}), \\ \hat{\boldsymbol{f}}_e^\delta(\Theta_e^{-1}(\boldsymbol{\xi})) &= \hat{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi}). \end{aligned}$$

To obtain the non-conservative form, we take the DG scheme to be

$$\int\int_{\Omega_e} \left(\frac{\partial \boldsymbol{u}^\delta}{\partial t} + \nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}^\delta \right) \ell_{ijk}(\boldsymbol{\xi}) \, d\boldsymbol{x} + \int_{\partial\Omega_e} \boldsymbol{n} \cdot [\boldsymbol{f}^* - \boldsymbol{f}^\delta] \, \ell_{ij}(\boldsymbol{\xi}) \, ds = 0, \quad i,j,k = 0,\ldots,N, \tag{7.49}$$

where $\ell_{ijk}(\boldsymbol{\xi})$ is a Lagrange polynomial polynomial in the reference element. We cannot compute a derivative in the physical variable, so the above $\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}^\delta$ actually refers to

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{f}^\delta = (\nabla_{\boldsymbol{x}} \xi_i) \cdot (\partial_{\xi_i} \hat{\boldsymbol{f}}_e^\delta) = (\boldsymbol{a}_j \times \boldsymbol{a}_k) \cdot (\partial_{\xi_i} \hat{\boldsymbol{f}}_e^\delta),$$

where i, j, k form a cycle, which is a quantity we can compute.

Thus, transforming to the reference coordinates, the DG scheme is

$$\begin{aligned}\int\int_{\Omega_s} \frac{\partial \hat{\boldsymbol{u}}_e^{\delta}}{\partial t} \ell_{p_1p_2p_3}(\boldsymbol{\xi})\, J_e\, d\boldsymbol{x} + \int\int_{\Omega_s} \sum_{i=1}^{3} J_e\, (\nabla_{\boldsymbol{x}}\, \xi_i) \cdot (\partial_{\xi_i} \hat{\boldsymbol{f}}_e^{\delta})\, \ell_{p_1p_2p_3}(\boldsymbol{\xi})\, d\boldsymbol{x} \\ + \hat{\boldsymbol{b}}_{\mathrm{DG}_e} = 0, \qquad p_1, p_2, p_3 = 0, \ldots, N,\end{aligned} \tag{7.50}$$

where the boundary term $\hat{\boldsymbol{b}}_{\mathrm{DG}_e}$ is given as

$$\hat{\boldsymbol{b}}_{\mathrm{DG}_e} = \sum_{i=1}^{3} \hat{\boldsymbol{b}}^{L}_{\mathrm{DG}_e,i} + \hat{\boldsymbol{b}}^{R}_{\mathrm{DG}_e,i},$$

where i, L corresponds to the boundary face along i^{th} direction on face with 0 on i^{th} coordinate and i, R corresponds to boundary face along i^{th} direction on face with 1 on i^{th} coordinate.

$$\hat{\boldsymbol{b}}^{L/R}_{\mathrm{DG}_{e,i}} = \int_{\partial\Omega^{L/R}_{s,i}} \boldsymbol{n}_i^{L/R} \cdot (\boldsymbol{f}^{*,L/R}_{e,i} - \boldsymbol{f}^{\delta,L/R}_{e,i})\, \ell_{p_1p_2p_3}\, \bar{J}^{L/R}_{e,i}\, ds,$$

where

$$\boldsymbol{n}_i^{L/R} = \boldsymbol{a}_j \times \boldsymbol{a}_k / |\boldsymbol{a}_j \times \boldsymbol{a}_k|, \qquad \bar{J}^{L/R}_{e,i} = |\boldsymbol{a}_j \times \boldsymbol{a}_k|,$$

where i, j, k form a cycle.

Thus, the boundary term is given by

$$\hat{\boldsymbol{b}}^{L/R}_{\mathrm{DG}_{e,i}} = \int_{\partial\Omega^{L/R}_{s,i}} (\boldsymbol{a}_j \times \boldsymbol{a}_k) \cdot (\boldsymbol{f}^{*,L/R}_{e,i} - \boldsymbol{f}^{\delta,L/R}_{e,i})\, \ell_{p_1p_2p_3}\, ds.$$

We choose the solution points to be Gauss-Legendre or Gauss-Lobattoo points and the quadrature points to be the same. Now, we collocate the scheme at $\{\boldsymbol{\xi}_{p_1p_2p_3}\}_{p_i=0}^{N}$ and use the following notations for a fixed $\boldsymbol{\xi}_{p_1p_2p_3}$

$$\begin{aligned} \boldsymbol{w} &= w_{p_1} w_{p_2} w_{p_3}, & \boldsymbol{w}_i &= \boldsymbol{w}/w_{p_i}, \\ \boldsymbol{\ell} &= \ell_{p_1p_2p_3}, & \boldsymbol{\ell}_i(\boldsymbol{\xi}) &= \ell_{p_j}(\xi^j)\ell_{p_k}(\xi^k), \quad i, j, k \text{ are in a cycle.}\end{aligned}$$

Thus, the collocation scheme at a fixed $\boldsymbol{\xi}_{p_1p_2p_3}$ would be

$$\begin{aligned}\boldsymbol{w} \frac{\mathrm{d}\hat{\boldsymbol{u}}_{p_1p_2p_3}}{\mathrm{d}t} + \boldsymbol{w} \sum_{i=1}^{3} (\boldsymbol{a}_j \times \boldsymbol{a}_k) \cdot (\partial_{\xi_i} \hat{\boldsymbol{f}}_e^{\delta})(\boldsymbol{\xi}_{p_1p_2p_3}) \\ + \sum_{i=1}^{3} \boldsymbol{w}_i\, (\boldsymbol{a}_j \times \boldsymbol{a}_k)_{i,L} \cdot (\boldsymbol{f}^{*,L}_{e,i} - \boldsymbol{f}^{\delta,L}_{e,i})\, \ell_{p_i}(0)\, \boldsymbol{\ell}_i(\boldsymbol{\xi}_{p_1p_2p_3}) \\ + \sum_{i=1}^{3} \boldsymbol{w}_i\, (\boldsymbol{a}_j \times \boldsymbol{a}_k)_{i,R} \cdot (\boldsymbol{f}^{*,R}_{e,i} - \boldsymbol{f}^{\delta,R}_{e,i})\, \ell_{p_i}(1)\, \boldsymbol{\ell}_i(\boldsymbol{\xi}_{p_1p_2p_3}) = 0.\end{aligned}$$

where $\{w_p\}_{p=1}^{N+1}$ are the quadrature weights. Dividing by $\boldsymbol{w}$, we get

$$\begin{aligned}\frac{\mathrm{d}\hat{\boldsymbol{u}}_{p_1p_2p_3}}{\mathrm{d}t} + \sum_{i=1}^{3} (\boldsymbol{a}_j \times \boldsymbol{a}_k) \cdot (\partial_{\xi_i} \hat{\boldsymbol{f}}_e^{\delta})(\boldsymbol{\xi}_{p_1p_2p_3}) \\ + \sum_{i=1}^{3} (\boldsymbol{a}_j \times \boldsymbol{a}_k)_{i,L} \cdot (\boldsymbol{f}^{*,L}_{e,i} - \boldsymbol{f}^{\delta,L}_{e,i}) \frac{\ell_{p_i}(0)}{w_{p_i}}\, \boldsymbol{\ell}_i(\boldsymbol{\xi}_{p_1p_2p_3}) \\ + \sum_{i=1}^{3} (\boldsymbol{a}_j \times \boldsymbol{a}_k)_{i,R} \cdot (\boldsymbol{f}^{*,R}_{e,i} - \boldsymbol{f}^{\delta,R}_{e,i}) \frac{\ell_{p_i}(1)}{w_{p_i}}\, \boldsymbol{\ell}_i(\boldsymbol{\xi}_{p_1p_2p_3}) = 0.\end{aligned}$$

It is known that

$$\frac{\ell_{p_i}(-1)}{w_{p_i}}, \frac{\ell_{p_i}(1)}{w_{p_i}} = \begin{cases} g'_{\mathrm{Radau},L}(\xi_{p_i}),\, g'_{\mathrm{Radau},L}(\xi_{p_i}), & \text{Gauss-Legendre solution points and quadrature,} \\ g'_{\mathrm{Hu},L}(\xi_{p_i}),\, g'_{\mathrm{Hu},L}(\xi_{p_i}), & \text{Gauss-Lobatto solution points and quadrature.}\end{cases}$$

With $\frac{\ell_{p_i}(0)}{w_{p_i}}$, there must be a minus to obtain correction functions.

Thus, we obtain the scheme

$$\begin{aligned}\frac{\mathrm{d}\hat{\boldsymbol{u}}_{e,p_1p_2p_3}}{\mathrm{d}t} + \sum_{i=1}^{3} (\boldsymbol{a}_j \times \boldsymbol{a}_k) \cdot (\partial_{\xi_i} \hat{\boldsymbol{f}}_e^{\delta})(\boldsymbol{\xi}_{p_1p_2p_3}) \\ + \sum_{i=1}^{3} (\boldsymbol{a}_j \times \boldsymbol{a}_k)_{i,L} \cdot (\boldsymbol{f}^{*,L}_{e,i} - \boldsymbol{f}^{\delta,L}_{e,i})\, g'_L(\xi_{p_i})\, \boldsymbol{\ell}_i(\boldsymbol{\xi}_{p_1p_2p_3}) \\ + \sum_{i=1}^{3} (\boldsymbol{a}_j \times \boldsymbol{a}_k)_{i,R} \cdot (\boldsymbol{f}^{*,R}_{e,i} - \boldsymbol{f}^{\delta,R}_{e,i})\, g'_R(\xi_{p_i})\, \boldsymbol{\ell}_i(\boldsymbol{\xi}_{p_1p_2p_3}) = 0.\end{aligned} \tag{7.51}$$

7.3.2. Flux Reconstruction

We explain degree N Flux Reconstruction (FR) on a general curvilinear mesh defined in Section 7.1. We first introduce the interpolation operator I_N. For 1-D Lagrange polynomial basis $\{\ell_p(\xi)\}_{p:1}^{N+1}$ for solution points $\{\xi_p\}_{p:1}^{N+1}$ in $[-1,1]$, we define the interpolation operator I_N for any $\boldsymbol{u}$ to be

$$I_N(\boldsymbol{u})(\boldsymbol{\xi}) = \sum_{p,q,r=1}^{N+1} \boldsymbol{u}(\xi_p,\xi_q,\xi_r)\,\ell_p(\xi^1)\,\ell_q(\xi^2)\,\ell_r(\xi^3).$$

Corresponding to the element Ω_e, we define approximate polynomial and discontinuous flux polynomials to solve the transformed conservation law (7.31)

$$\begin{aligned} \tilde{\boldsymbol{u}}_e^\delta &= \tilde{\boldsymbol{u}}_e^\delta(\boldsymbol{\xi},t) &&= J_e\,\boldsymbol{u}_e^\delta(\boldsymbol{\xi},t), \\ \tilde{\boldsymbol{f}}_e^\delta &= \tilde{\boldsymbol{f}}_e^\delta(\boldsymbol{\xi},t) &&= (\tilde{f}_{e,1}^\delta, \tilde{f}_{e,2}^\delta, \tilde{f}_{e,3}^\delta) \end{aligned}$$

where $\tilde{\boldsymbol{u}}_e^\delta$ is defined as

$$\tilde{\boldsymbol{u}}_e^\delta(\boldsymbol{\xi},t) = \sum_{i,j,k=1}^{N+1} \tilde{\boldsymbol{u}}_{ijk}\,\ell_i(\xi^i)\,\ell_j(\xi^j)\,\ell_k(\xi^k),$$

and we solve for the unknowns $\tilde{\boldsymbol{u}}_{ijk}$, and

$$\boldsymbol{u}_{ijk} = J_e^{-1}\,\tilde{\boldsymbol{u}}_{ijk}.$$

The discontinous flux on reference cell is defined as the polynomial

The discontinuous flux is defined to be the polynomial

$$\boldsymbol{f}_e^\delta(\boldsymbol{\xi},t) = \sum_{i,j,k=1}^{N+1} \boldsymbol{f}(\boldsymbol{u}_{ijk})\,\ell_i(\xi^i)\,\ell_j(\xi^j)\,\ell_k(\xi^k),$$

and the contravariant flux to be

$$\tilde{\boldsymbol{f}}_e^\delta = I_N(J\boldsymbol{a}^i)\cdot\tilde{\boldsymbol{f}}_e^\delta.$$

Remark 7.13. For the later derivation, we will not be treating $\tilde{\boldsymbol{f}}_e^\delta$ as a degree N polynomial, but as a degree $2N$ polynomial. Thus, we are not doing standard FR on the transformed conservation law (7.31). However, we choose to do it this way because it turns out to be the one equivalent to non-conservative form of DG, as defined by Cinchio.

The quantities in physical cell are defined implicitly as

$$\begin{aligned} \boldsymbol{u}^\delta(\boldsymbol{x},t) &= \tilde{\boldsymbol{u}}_e^\delta(\Theta_e^{-1}(\boldsymbol{x}),t), \\ \boldsymbol{f}^\delta(\boldsymbol{x},t) &= \tilde{\boldsymbol{f}}_e^\delta(\Theta_e^{-1}(\boldsymbol{x}),t). \end{aligned}$$

We shall usually suppress the subscript e indicating the physical element. With this, we get that FR scheme in reference cell given by

$$\frac{\partial\hat{u}^\delta}{\partial t} + \nabla_{\boldsymbol{\xi}}\cdot\hat{\boldsymbol{f}}^\delta + \hat{\boldsymbol{b}}_{\mathrm{FR}_n} = 0, \tag{7.52}$$

where $\hat{\boldsymbol{b}}_{\mathrm{FR}_n}$ is the correction term defined as

$$\hat{\boldsymbol{b}}_{\mathrm{FR}_n}(\boldsymbol{\xi}) = \sum_{i=1}^{3} g_L'(\xi_i)\,\Delta\hat{\boldsymbol{f}}_{i,L}^\delta + g_R'(\xi_i)\,\Delta\hat{\boldsymbol{f}}_{i,R}^\delta,$$

where i,L denotes the direction along ξ_i where the i^{th} coordinate is -1 and i,R is where i^{th} coordinate is $+1$, and

$$\Delta\hat{\boldsymbol{f}}_{i,L/R}^\delta = \hat{\boldsymbol{f}}_{i,L/R}^{*,\delta} - \hat{\boldsymbol{f}}_{i,L/R}^\delta.$$

Write this better. It is not direction, but location.

$$\Delta\hat{f}_{i,L/R}^\delta = \Delta((\boldsymbol{a}_j\times\boldsymbol{a}_k)\cdot\boldsymbol{f}^\delta)_{i,L/R} = (\boldsymbol{a}_j\times\boldsymbol{a}_k)_{i,L/R}\cdot\Delta\boldsymbol{f}_{i,L/R}^\delta, \tag{7.53}$$

where we have used conformality of the mesh.

Remark 7.14. This is where we have used the fact that

$$\tilde{\boldsymbol{f}}_e^\delta = I^N(J\boldsymbol{a}^i)\cdot\boldsymbol{F}_e^\delta \neq I^N(I^N(J\boldsymbol{a}^i)\cdot I^N(\boldsymbol{F}_e^\delta)).$$

Using the metric identities

$$\begin{aligned}\nabla_{\boldsymbol{\xi}}\cdot\hat{\boldsymbol{f}}^{\delta} &= \sum_{i=1}^{3}\partial_{\xi_i}[(\boldsymbol{a}_j\times\boldsymbol{a}_k)\cdot\boldsymbol{f}^{\delta}] = \underbrace{\left[\sum_{i=1}^{3}\partial_{\xi_i}(\boldsymbol{a}_j\times\boldsymbol{a}_k)\right]}_{=\boldsymbol{0}}\cdot\boldsymbol{f}^{\delta}+\sum_{i=1}^{3}[(\boldsymbol{a}_j\times\boldsymbol{a}_k)\cdot\partial_{\xi_i}\boldsymbol{f}^{\delta}]\\ &= \sum_{i=1}^{3}[(\boldsymbol{a}_j\times\boldsymbol{a}_k)\cdot\partial_{\xi_i}\boldsymbol{f}^{\delta}]\end{aligned}$$

$$\begin{aligned}&\frac{\partial\hat{u}^{\delta}}{\partial t}+\sum_{i=1}^{3}(\boldsymbol{a}_j\times\boldsymbol{a}_k)\cdot[\partial_{\xi_i}\boldsymbol{f}^{\delta}]\\ &\quad+\sum_{i=1}^{3}[g_L'(\xi_i)\,(\boldsymbol{a}_j\times\boldsymbol{a}_k)_{i,L}\cdot\Delta\boldsymbol{f}^{\delta}_{i,L}+g_R'(\xi_i)\,(\boldsymbol{a}_j\times\boldsymbol{a}_k)_{i,R}\cdot\Delta\boldsymbol{f}^{\delta}_{i,R}]=0\end{aligned},$$

where, for $\boldsymbol{x}\in\Omega_e$,

$$\boldsymbol{f}^{\delta}(\boldsymbol{x})=\boldsymbol{f}^{\delta}(\Theta_e(\boldsymbol{\xi})).$$

Since we would know the function as a polynomial mapping from $\boldsymbol{\xi}\mapsto\boldsymbol{f}^{\delta}(\Theta_e^{-1}(\boldsymbol{\xi}))$, we can easily differentiate it without needing the chain rule. This scheme is clearly equivalent to the collocated DG scheme (7.51).

It is clear that this is constant state preserving.

We will show that the non-conservative form of FR in previous section is equivalent to non-conservative form of DG, following the proof of [4]. To prove that, we multiply

7.3.3. Free stream preservation for Lax-Wendroff scheme

For a Lax-Wendroff, the update would be replaced by

$$\hat{\boldsymbol{u}}^{n+1}-\hat{\boldsymbol{u}}^{n}+\sum_{i=1}^{3}(\boldsymbol{a}_j\times\boldsymbol{a}_k)\cdot[\partial_{\xi_i}\boldsymbol{F}^{\delta}+g_L'(\xi_i)\,\Delta\boldsymbol{F}^{\delta}_{i,L}+g_R'(\xi_i)\,\Delta\boldsymbol{F}^{\delta}_{i,R}]=0,$$

where we have discretized the scheme in time and replaced the spatial quantities by polynomials.

Now, we assume $\boldsymbol{u}=\underline{\boldsymbol{c}}$, and $\boldsymbol{f}(\underline{\boldsymbol{c}})=\boldsymbol{c}$. In the FR scheme, we only need to compute

$$\boldsymbol{F}^{\delta} = \sum_{m=0}^{N}\frac{\Delta t^m}{(m+1)!}\partial_t^m\boldsymbol{f}$$

Remark 7.15. It is because of the non-conservative form that we can work with temporal derivatives of $\boldsymbol{f}$ instead of $\tilde{\boldsymbol{f}}$.

Let us illustrate this for $N=1$, where we approximate

$$\partial_t\boldsymbol{f} \approx \frac{\boldsymbol{f}(\boldsymbol{u}+\Delta t\,\boldsymbol{u}_t)+\boldsymbol{f}(\boldsymbol{u}-\Delta t\,\boldsymbol{u}_t)}{2\,\Delta t},$$

where we approximate

$$\begin{aligned}\boldsymbol{u}_t &= -\nabla_{\boldsymbol{x}}\cdot\boldsymbol{f}\\ &= \boldsymbol{0},\end{aligned}$$

because we assumed that $\boldsymbol{f}=\boldsymbol{c}$. The same proof shall go forward for $N>1$.

7.4. ADAPTIVE MESH REFINEMENT

Refinement and coarsening

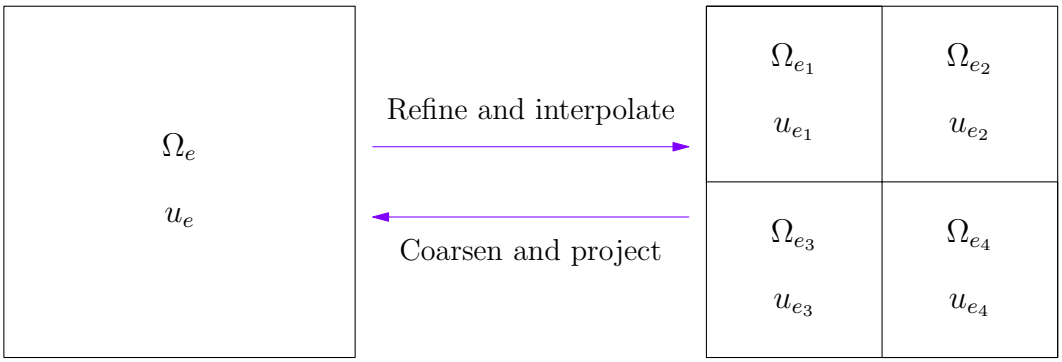

Figure 7.4. Illustration of refinement and coarsening

The refinement and coarsening are done by Löhner's smoothness indicator

$$\alpha_e = \max_{1\le i\le N-1} \frac{|u_{i+1} - 2\,u_i + u_{i-1}|}{|u_{i+1} - u_i| + |u_i - u_{i-1}| + f_{\text{wave}}\,(|u_{i+1}| + 2\,|u_i| + |u_{i-1}|)}$$

where $0 \le i \le N$ are the degrees of freedom in cell e. The value $f_{\text{wave}} = 0.2$ has been chosen in all the tests.

Source - 12 Adaptive mesh refinement · Trixi.jl (trixi-framework.github.io)

The mesh is created with two thresholds `med_threshold` and `max_threshold` and three refinement levels `base_level`, `med_level` and `max_level`. The `base_level` is the minimum refinement level, `med_level` is the minimum refinement level if the indicator is above `med_level` and `max_level` is the maximum refinement level which is automatically assigned for indicator above `max_threshold`. The levels in (`base_level`, `med_level`) and (`med_level`, `max_level`) come when trying to maintain that refinement level of neighbours only vary by 2. We always refine to match neighbour's values, never coarsen.

Projection for coarsening

For $\{\ell_i\}_{i:0}^{N}$ being Lagrange polynomials of degree N, projection of $\{\boldsymbol{u}_{e_s}\}_{s:1}^{4}$ to $\boldsymbol{u}_e$ is

$$\sum_{s:1}^{4} \int_{\Omega_{e_s}} \boldsymbol{u}_{e_s}\,\ell_i(\xi)\,\ell_j(\eta) = \int_{\Omega_e} \boldsymbol{u}_e\,\ell_i(\xi)\,\ell_j(\eta) \tag{7.54}$$

In reference coordinates

$$\boldsymbol{u}_{e_1}(\xi,\eta) = \sum_{k,l=0}^{N} \ell_k(2\,\xi+1)\,\ell_l(2\,\eta-1)\,\boldsymbol{u}_{e_1,kl}, \qquad \boldsymbol{u}_{e_2}(\xi,\eta) = \sum_{k,l=0}^{N} \ell_k(2\,\xi-1)\,\ell_l(2\,\eta-1)\,\boldsymbol{u}_{e_2,kl}$$
$$\boldsymbol{u}_{e_3}(\xi,\eta) = \sum_{k,l=0}^{N} \ell_k(2\,\xi+1)\,\ell_l(2\,\eta+1)\,\boldsymbol{u}_{e_3,kl} \qquad \boldsymbol{u}_{e_4}(\xi,\eta) = \sum_{k,l=0}^{N} \ell_k(2\,\xi-1)\,\ell_l(2\,\eta+1)\,\boldsymbol{u}_{e_4,kl}$$

Substituting these into (7.54) gives, in reference coordinates

$$\begin{aligned}
&\sum_{k,l=0}^{N} \int_0^1\int_{-1}^0 \ell_k(2\,\xi+1)\,\ell_i(\xi)\,\ell_l(2\,\eta-1)\,\ell_j(\eta)\,\boldsymbol{u}_{e_1,kl}\,d\xi\,d\eta \\
&+\sum_{k,l=0}^{N} \int_0^1\int_{0}^1 \ell_k(2\,\xi+1)\,\ell_i(\xi)\,\ell_l(2\,\eta-1)\,\ell_j(\eta)\,\boldsymbol{u}_{e_2,kl}\,d\xi\,d\eta \\
&+\sum_{k,l=0}^{N} \int_{-1}^0\int_{-1}^0 \ell_k(2\,\xi+1)\,\ell_i(\xi)\,\ell_l(2\,\eta-1)\,\ell_j(\eta)\,\boldsymbol{u}_{e_3,kl}\,d\xi\,d\eta \\
&+\sum_{k,l=0}^{N} \int_0^1\int_{-1}^0 \ell_k(2\,\xi+1)\,\ell_i(\xi)\,\ell_l(2\,\eta-1)\,\ell_j(\eta)\,\boldsymbol{u}_{e_4,kl}\,d\xi\,d\eta
\end{aligned} = \sum_{k,l=0}^{N} \int_{\Omega_e} \boldsymbol{u}_{e,ij}\,\ell_i(\xi)\,\ell_j(\eta)\,\ell_k(\eta)\,\ell_l(\eta)$$

We now use

$$\begin{aligned}
\int_{-1}^{1} \ell_q(\xi)\,\ell_p(\xi) &= w_p\,\delta_{pq} \\
\int_{0}^{1} \ell_q(2\,\xi-1)\,\ell_p(\xi) &= \frac{1}{2}\int_{-1}^{1} \ell_q(\xi)\,\ell_p\!\left(\frac{\xi+1}{2}\right) = \frac{1}{2}\,w_q\,\ell_p\!\left(\frac{\xi_q+1}{2}\right) =: w_p\,\mathcal{P}_{pq}^{1} \\
\int_{-1}^{0} \ell_q(2\,\xi+1)\,\ell_p(\xi) &= \frac{1}{2}\int_{-1}^{1} \ell_q(\xi)\,\ell_p\!\left(\frac{\xi-1}{2}\right) d\,\eta = \frac{1}{2}\,w_q\,\ell_p\!\left(\frac{\xi_q-1}{2}\right) =: w_p\,\mathcal{P}_{pq}^{2}
\end{aligned}$$

to get

$$\begin{aligned}
&\sum_{k,l=0}^{N} \int_0^1\int_{-1}^0 w_i\,w_k\,\mathcal{P}_{ik}^{2}\,\mathcal{P}_{jl}^{1}\,\boldsymbol{u}_{e_1,kl}\,d\xi\,d\eta \\
&+\sum_{k,l=0}^{N} \int_0^1\int_{0}^1 w_i\,w_k\,\mathcal{P}_{ik}^{1}\,\mathcal{P}_{jl}^{1}\,\boldsymbol{u}_{e_2,kl}\,d\xi\,d\eta \\
&+\sum_{k,l=0}^{N} \int_{-1}^0\int_{-1}^0 w_i\,w_k\,\mathcal{P}_{ik}^{2}\,\mathcal{P}_{jl}^{2}\,\boldsymbol{u}_{e_3,kl}\,d\xi\,d\eta \\
&+\sum_{k,l=0}^{N} \int_0^1\int_{-1}^0 w_i\,w_k\,\mathcal{P}_{ik}^{1}\,\mathcal{P}_{jl}^{2}\,\boldsymbol{u}_{e_4,kl}\,d\xi\,d\eta
\end{aligned} = \boldsymbol{u}_{e,ij}\,w_i\,w_j,$$

so that we get

$$\boldsymbol{u}_{e,ij} = \sum_{k,l=0}^{N} \mathcal{P}^2_{ik}\mathcal{P}^1_{jl}\,\boldsymbol{u}_{e_1,kl} + \mathcal{P}^1_{ik}\mathcal{P}^1_{jl}\,\boldsymbol{u}_{e_2,kl} + \mathcal{P}^2_{ik}\mathcal{P}^2_{jl}\,\boldsymbol{u}_{e_3,kl} + \mathcal{P}^1_{ik}\mathcal{P}^2_{jl}\,\boldsymbol{u}_{e_4,kl}.$$

Handling mortars

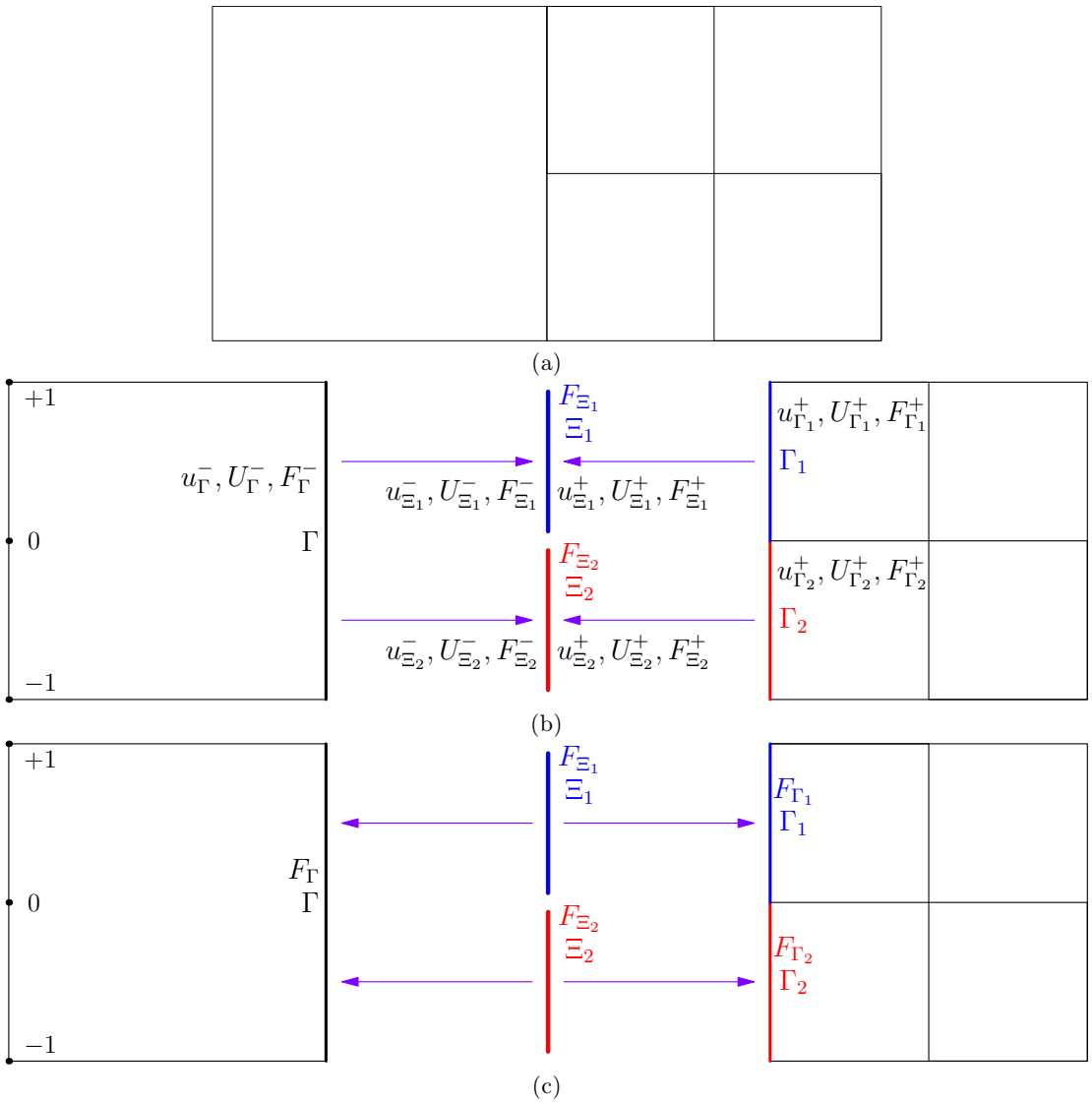

Figure 7.5. (a) Prolongation to mortar and computation of numerical flux F_{Ξ_1}, F_{Ξ_2}, (b) Projection of numerical flux to interfaces

We assume that the refinement level between elements only differs by 2. Interfaces where neighbours have different refinement levels are called **mortars**. For these non-conformal meshes, understanding AMR simply requires us to understand how to compute the numerical flux at the mortars. We explain that for the FR scheme.

The idea is that we will compute the numerical flux **in the smaller elements** and **project it back to the larger elements**.

Refinement and coarsening should be explained. Conservation can be violated if they are not done correctly. Refinement can be interpolation but coarsening must be projection.

Loffner's indicator and how exactly it is used, with what tolerances, must also be explained.

Prolongation to mortars

The prolongation of $\boldsymbol{u}_{\Gamma_1}, \boldsymbol{u}_{\Gamma_2}$ to Ξ_1, Ξ_2 is the identity map. The prolongation from Γ to Ξ_1, Ξ_2 is performed by an interpolation operator. The boundar Γ can be written as a continuous curve $y = \gamma(\eta) = \Theta_e(1, \eta)$ for $-1 \leq \eta \leq 1$ so that by conformality $\Gamma_1 = \{\gamma(\eta) : 0 \leq \eta \leq 1\}$ and $\Gamma_2 = \{\gamma(\eta) : -1 \leq \eta \leq 0\}$. Thus, in the 1-D reference coordinates $\eta = \gamma^{-1}(y) \in [-1, 1]$ for $y \in \Gamma$, the solution points of Γ are $\{\eta_j\}_{j:0}^N$ and the solution points in Γ_1, Γ_2 are given by $\left\{\frac{\eta_j + 1}{2}\right\}_{j:0}^N, \left\{\frac{\eta_j - 1}{2}\right\}_{j:0}^N$ respectively so that if $\{\ell_j(\eta)\}_{j:0}^N$ are Lagrange polynomials in Γ, the Lagrange polynomials in Γ_1, Γ_2 (or Ξ_1, Ξ_2) are given by $\{\ell_j(2\,\eta - 1)\}_{j:0}^N, \{\ell_j(2\,\eta + 1)\}_{j:0}^N$ respectively. Finally, the interpolation of $\boldsymbol{u}_\Gamma$ to mortars is given by $\boldsymbol{u}_{\Xi_1}^-(\eta), \boldsymbol{u}_{\Xi_2}^-(\eta)$ which are defined as

$$\begin{aligned} \boldsymbol{u}_{\Xi_1}^-(\eta) &= \sum_{i=0}^N \ell_i(2\,\eta - 1)\, \boldsymbol{u}_{\Xi_1, i}^-, \qquad \eta \in [0, 1], \\ \boldsymbol{u}_{\Xi_2}^-(\eta) &= \sum_{i=0}^N \ell_i(2\,\eta + 1)\, \boldsymbol{u}_{\Xi_2, i}^-, \qquad \eta \in [-1, 0], \end{aligned} \tag{7.55}$$

The coefficients are given by

$$\begin{aligned} \boldsymbol{u}_{\Xi_1, i}^- = \boldsymbol{u}_\Gamma^-\left(\frac{\eta_i + 1}{2}\right) &= \sum_{j=0}^N (\mathtt{V}_{\Xi_1})_{ij}\, \boldsymbol{u}_\Gamma(\eta_j), \\ \boldsymbol{u}_{\Xi_2, i}^- = \boldsymbol{u}_\Gamma^-\left(\frac{\eta_i - 1}{2}\right) &= \sum_{j=0}^N (\mathtt{V}_{\Xi_2})_{ij}\, \boldsymbol{u}_\Gamma(\eta_j), \end{aligned}$$

where

$$(\mathtt{V}_{\Xi_1})_{ij} = \ell_j\left(\frac{\eta_i + 1}{2}\right), \qquad (\mathtt{V}_{\Xi_2})_{ij} = \ell_j\left(\frac{\eta_j - 1}{2}\right).$$

Calculation of mortar flux

Once quantities are available on $\Xi_1^\pm, \Xi_2^\pm$, the numerical fluxes $\boldsymbol{F}_{\Xi_1}, \boldsymbol{F}_{\Xi_2}$ are computed.

Projection of numerical fluxes from mortars to faces

The numerical flux $\boldsymbol{F}_{\Xi_1}, \boldsymbol{F}_{\Xi_2}$ are mapped to $\boldsymbol{F}_{\Gamma_1}, \boldsymbol{F}_{\Gamma_2}$ on Γ_1, Γ_2 respectively by using the identity map. To map these to one flux $\boldsymbol{F}_\Gamma$ on Γ, we perform an L^2 projection by defining $\boldsymbol{F}_\Gamma$ so that

$$\int_{-1}^0 (\boldsymbol{F}_\Gamma - \boldsymbol{F}_{\Xi_1})\, p + \int_0^1 (\boldsymbol{F}_\Gamma - \boldsymbol{F}_{\Xi_2})\, p = 0, \qquad \forall p \in P_N.$$

where integrals are computed with quadrature at solution points. Simplifying and taking $p = \ell_i$, we get that above is equivalent to

$$\int_{-1}^1 \boldsymbol{F}_\Gamma\, \ell_i = \int_{-1}^0 \boldsymbol{F}_{\Xi_1}\, \ell_i + \int_0^1 \boldsymbol{F}_{\Xi_2}\, \ell_i \qquad 0 \leq i \leq N.$$

As in (7.5), we write the mortar fluxes as

$$\begin{aligned} \boldsymbol{F}_{\Xi_1}(\eta) &= \sum_{i=0}^N \ell_j(2\,\eta - 1)\, \boldsymbol{F}_{\Xi_1, i}, \quad \eta \in [0, 1], \\ \boldsymbol{F}_{\Xi_2}(\eta) &= \sum_{i=0}^N \ell_j(2\,\eta + 1)\, \boldsymbol{F}_{\Xi_2, i}, \quad \eta \in [-1, 0], \\ \boldsymbol{F}_\Gamma(\eta) &= \sum_{i=0}^N \ell_j(\eta)\, \boldsymbol{F}_{\Gamma, i}, \quad \eta \in [-1, 1]. \end{aligned}$$

Thus, the integral identity can be written as

$$\sum_{i=0}^N \int_{-1}^1 \boldsymbol{F}_{\Gamma, i}\, \ell_j(\eta)\, \ell_i(\eta) = \sum_{i=0}^N \int_0^1 \boldsymbol{F}_{\Xi_1, i}\, \ell_j(2\,\eta - 1)\, \ell_i(\eta) + \sum_{i=0}^N \int_{-1}^0 \boldsymbol{F}_{\Xi_2, i}\, \ell_j(2\,\eta + 1)\, \ell_i(\eta). \tag{7.56}$$

We have, performing quadrature,

$$\begin{aligned} \int_{-1}^1 \ell_j(\eta)\, \ell_i(\eta) &= w_i\, \delta_{ij} \\ \int_0^1 \ell_j(2\,\eta - 1)\, \ell_i(\eta) &= \frac{1}{2} \int_{-1}^1 \ell_j(\eta)\, \ell_i\left(\frac{\eta + 1}{2}\right) = \frac{1}{2}\, w_j\, \ell_i\left(\frac{\eta_j - 1}{2}\right) \\ \int_{-1}^0 \ell_j(2\,\eta + 1)\, \ell_i(\eta) &= \frac{1}{2} \int_{-1}^1 \ell_j(\eta)\, \ell_i\left(\frac{\eta - 1}{2}\right) d\,\eta = \frac{1}{2}\, w_j\, \ell_i\left(\frac{\eta_j - 1}{2}\right). \end{aligned}$$

Thus, the above integral equation (7.56) becomes

$$\boldsymbol{F}_{\Gamma,i} = \frac{1}{2}\sum_{j=0}^{N}\left(\boldsymbol{F}_{\Xi_1,i}\frac{w_j}{w_i}\ell_i\Big(\frac{\xi_j+1}{2}\Big)+\boldsymbol{F}_{\Xi_2,i}\frac{w_j}{w_i}\ell_i\Big(\frac{\xi_j-1}{2}\Big)\right), \qquad 0\le i\le N.$$

Thus, we define the projection operators $\mathcal{P}^{\Xi_1\to\Gamma}, \mathcal{P}^{\Xi_2\to\Gamma}$ written as matrices

$$\mathcal{P}^{\Xi_1\to\Gamma}_{ij} = \frac{1}{2}\frac{w_j}{w_i}\ell_i\Big(\frac{\xi_j+1}{2}\Big), \qquad \mathcal{P}^{\Xi_2\to\Gamma}_{ij} = \frac{1}{2}\frac{w_j}{w_i}\ell_i\Big(\frac{\xi_j-1}{2}\Big),$$

so that

$$\boldsymbol{F}_\Gamma = \mathcal{P}^{\Xi_1\to\Gamma}\boldsymbol{F}_{\Xi_1} + \mathcal{P}^{\Xi_2\to\Gamma}\boldsymbol{F}_{\Xi_2}.$$

Conservation property

The conservation property is about showing that the flux used by the cell averages is the same. Thus, we must show that

$$\int_{-1}^{1}\boldsymbol{F}_\Gamma = \int_{-1}^{1}\mathcal{P}^{\Gamma_1\to\Gamma}\boldsymbol{F}_{\Xi_1} + \mathcal{P}^{\Gamma_2\to\Gamma}\boldsymbol{F}_{\Xi_2} = \int_0^1\boldsymbol{F}_{\Xi_1} + \int_{-1}^{0}\boldsymbol{F}_{\Xi_2}.$$

By definition of the projection operation and exactness of quadrature for degree less than $2N$, we have

$$\int_{-1}^{1}\boldsymbol{F}_\Gamma\, p = \int_0^1\boldsymbol{F}_{\Xi_1}\, p + \int_{-1}^{0}\boldsymbol{F}_{\Xi_2}\, p$$

for polynomials p of degree $\le N-1$. Thus, choosing $p=1$, we get

$$\int_{-1}^{1}\boldsymbol{F}_\Gamma = \int_0^1\boldsymbol{F}_{\Xi_1} + \int_{-1}^{0}\boldsymbol{F}_{\Xi_2},$$

which was our claim. Thus, the conservation property holds.

Upwind / outflow property

If the characteristics are flowing from the refined region to the coarse region, the refined region's fluxes will be plugged directly into the flux function and the upwinding will use fluxes from only that direction. And then, those refined region fluxes will be our fluxes. The question is about what happens when there are characteristic flow from the coarse region. In that case, the upwinding will cause us to use the fluxes that have come after interpolation in Γ^1, Γ^2. But is that same as using the flux from Γ directly? Thus, we need to show that for Γ^1, Γ^2 being the mortars,

$$\mathcal{P}^{\Xi_1\to\Gamma}\mathtt{V}_{\Xi_1} + \mathcal{P}^{\Xi_2\to\Gamma}\mathtt{V}_{\Xi_2} = \mathrm{Id},$$

where I_N denotes the interpolation operator and $\mathbb{I}$ is the identity operator. We do have, with quadrature,

$$\begin{aligned}&\int_0^1((\mathcal{P}^{\Xi_1\to\Gamma}\mathtt{V}_{\Xi_1} + \mathcal{P}^{\Xi_2\to\Gamma}\mathtt{V}_{\Xi_2})\, f - \mathtt{V}_{\Xi_1} f)\, p\\ &+\int_{-1}^0((\mathcal{P}^{\Xi_1\to\Gamma}\mathtt{V}_{\Xi_1} + \mathcal{P}^{\Xi_2\to\Gamma}\mathtt{V}_{\Xi_2})\, f - \mathtt{V}_{\Xi_2} f)\, p = 0,\end{aligned}$$

for all polynomials $p\in P_N$. That is,

$$\int_{-1}^{1}(\mathcal{P}^{\Xi_1\to\Gamma}\mathtt{V}_{\Xi_1} + \mathcal{P}^{\Xi_2\to\Gamma}\mathtt{V}_{\Xi_2})\, fp = \int_0^1(\mathtt{V}_{\Xi_1} f)\, p + \int_{-1}^0(\mathtt{V}_{\Xi_2} f)\, p.$$

Since $\mathtt{V}_{\Xi_1}, \mathtt{V}_{\Xi_2}$ are interpolation operators preserving the degree, we will have

$$\int_{-1}^0(\mathtt{V}_{\Xi_1} f)\, p = \int_{-1}^0 fp, \qquad \int_0^1(\mathtt{V}_{\Xi_2} f)\, p = \int_{-1}^0 fp.$$

Thus, we have

$$\int_{-1}^{1}(\mathcal{P}^{\Xi_1\to\Gamma}\mathtt{V}_{\Xi_1} + \mathcal{P}^{\Xi_2\to\Gamma}\mathtt{V}_{\Xi_2})\, fp = \int_{-1}^0 fp + \int_0^1 fp = \int_{-1}^1 fp.$$

Since this is true for all p of degree $\le N$, we will have

$$(\mathcal{P}^{\Xi_1\to\Gamma}\mathtt{V}_{\Xi_1} + \mathcal{P}^{\Xi_2\to\Gamma}\mathtt{V}_{\Xi_2})\, f = f.$$

Thus, we have our claim.

Free stream preservation

Assuming that $\boldsymbol{f} = \boldsymbol{c}$, the contravariant flux is given by

$$\tilde{\boldsymbol{f}}_i \;=\; \sum_{n=1}^{d} \mathcal{I}_N(J a_n^i)\, \boldsymbol{c}_n, \tag{7.57}$$

where $\mathcal{I}_N$ is the interpolation operator used to compute the metric terms. The interpolation operator and degree of mesh will be chosen so that $\mathcal{I}_N(J a_n^i)$ terms are identical along both sides of the mortars. Thus, interpolation from larger element to mortar and projection of the flux from mortar to larger element will give the same polynomial. Thus, numerical flux at the interface will be given by metric terms at the interface.

As in (7.39),

$$\begin{aligned}
\boldsymbol{u}_t \;&=\; -\frac{1}{J}\nabla_{\boldsymbol{\xi}} \cdot \tilde{\boldsymbol{f}}_e^{\delta} = -\frac{1}{J}\sum_{i=1}^{3} \partial_{\xi_i} \tilde{\boldsymbol{f}}_{e,i}^{\delta} \\
&=\; -\frac{1}{J}\sum_{i=1}^{3} \partial_{\xi_i} \left(\sum_{n=1}^{3} \mathcal{I}_N(J a_n^i)\, \boldsymbol{c}_n \right) \\
&=\; -\sum_{n=1}^{3} \left(\sum_{i=1}^{3} \partial_{\xi_i} \mathcal{I}_N(J a_n^i) \right) \boldsymbol{c}_n = 0,
\end{aligned}$$

where the final metric identity continues to hold because interpolation keeps the polynomial degree same.

For $N = 1$, by (7.38), this proves that

$$\partial_t \tilde{\boldsymbol{f}}^{\delta} = \frac{\tilde{\boldsymbol{f}}(\boldsymbol{u} + \Delta t\, \boldsymbol{u}_t) - \tilde{\boldsymbol{f}}(\boldsymbol{u} - \Delta t\, \boldsymbol{u}_t)}{2\,\Delta t} = \boldsymbol{0}.$$

Thus, we obtain

$$\tilde{\boldsymbol{F}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta} + \frac{\Delta t}{2}\partial_t \tilde{\boldsymbol{f}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta}.$$

Building on this, for $N = 2$, by (7.40),

$$\boldsymbol{u}_{tt} = -\frac{1}{J}\nabla_{\boldsymbol{\xi}} \cdot \partial_t \tilde{\boldsymbol{f}}^{\delta} = \boldsymbol{0},$$

which will prove

$$\partial_{tt} \tilde{\boldsymbol{f}}^{\delta} = \boldsymbol{0},$$

and we get

$$\tilde{\boldsymbol{F}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta} + \frac{\Delta t}{2}\partial_t \tilde{\boldsymbol{f}}^{\delta} + \frac{\Delta t^2}{3!}\partial_{tt} \tilde{\boldsymbol{f}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta}.$$

The claim shall similarly hold for all degrees. Thus, substituting

$$\tilde{\boldsymbol{F}}^{\delta} = \tilde{\boldsymbol{f}}^{\delta} = \{J\boldsymbol{a}^i \cdot \boldsymbol{c}\}_{i=1}^{3}$$

in (7.37) gives

$$\begin{aligned}
&\boldsymbol{u}^{n+1} - \boldsymbol{u}^n + \frac{1}{J}\Delta t \left(\sum_{i=1}^{3} \partial_{\xi^i} (J\boldsymbol{a}^i) \right) \cdot \boldsymbol{c} \\
&\quad + \frac{1}{J}\Delta t \sum_{i=1}^{3} \int_{\partial\Omega_{s,i}} ((J\boldsymbol{a}^i \cdot \boldsymbol{c})^* - J\boldsymbol{a}^i \cdot \boldsymbol{c})(\boldsymbol{\xi}_i^R)\, g_R'(\xi_{p_i})\, dS_{\boldsymbol{\xi}} \\
&\quad + \frac{1}{J}\Delta t \sum_{i=1}^{3} \int_{\partial\Omega_{s,i}} ((J\boldsymbol{a}^i \cdot \boldsymbol{c})^* - J\boldsymbol{a}^i \cdot \boldsymbol{c})(\boldsymbol{\xi}_i^L)\, g_L'(\xi_{p_i})\, dS_{\boldsymbol{\xi}} = \boldsymbol{0}.
\end{aligned} \tag{7.58}$$

The volume term vanishes by the metric identities as metric terms are preserved under refinement (interpolation) and coarsening (projection) and the surface terms vanish because the $J\boldsymbol{a}^i$ terms are identical across interfaces.

Bibliography

[1] Alexander Cicchino, David C. Del Rey Fernández, Siva Nadarajah, Jesse Chan, and Mark H. Carpenter. Provably stable flux reconstruction high-order methods on curvilinear elements. *Journal of Computational Physics*, 463:111259, 2022.

[2] H. T. Huynh. A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods. Miami, FL, jun 2007. AIAA.

[3] David A Kopriva. *Implementing spectral methods for partial differential equations*. Scientific Computation. Springer, Dordrecht, Netherlands, oct 2010.

[4] G. Mengaldo, D. De Grazia, P. E. Vincent, and S. J. Sherwin. On the connections between discontinuous galerkin and flux reconstruction schemes: extension to curvilinear meshes. *Journal of Scientific Computing*, 67(3):1272–1292, oct 2015.

[5] Jianxian Qiu, Michael Dumbser, and Chi-Wang Shu. The discontinuous Galerkin method with Lax–Wendroff type time discretizations. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4528–4543, oct 2005.