Discontinuous Galerkin Method

Praveen Chandrashekar
TIFR Centre for Applicable Mathematics
Bangalore – 560065
http://cpraveen.github.io

February 10, 2019

Contents

Chapter 1

Introduction

In finite volume methods, there is one solution variable per cell, the cell average value, and the solution is assumed to be piecewise constant. To obtain high order accuracy, we reconstruct the solution by a polynomial inside each cell by making use of the cell averages in a small stencil around the current cell. The size of the stencil depends on the order of accuracy we want, with higher orders requiring larger stencils. While this is relatively easy to do on structured grids, it can become quite involved for unstructured grids. The choice of the stencil is not unique and the large stencil can lead to loss of data locality that can affect computational efficiency. In parallel computations, the variable stencil size require more intricate coding and transfer of large amount of data across partition boundaries. We also need special care at boundaries to choose the stencil in an appropriate way.

In DG methods we start with a polynomial solution in each cell which is evolved forward in time by the scheme. DG methods are thus finite element methods, where we approximate some function in terms of certain basis functions with compact support. The basis functions are taken to be polynomials and are allowed to be discontinuous across the cell boundaries. The compact support property ensures that the stencil of the scheme is small which leads to efficient methods for solution. In fact each cell only needs to exchange data with its immediate face neighbouring cells for the computation of fluxes at the cell boundaries. We have to deal with a mass matrix but the matrix for each cell is decoupled from other cells which makes it efficient to construct explicit time stepping schemes. Since the stencil of the scheme is same irrespective or scheme order, parallelization becomes easier and also requires less data transfer.

The high order nature and compact structure of DG schemes also leads to very low dissipation and dispersion errors, and these schemes are thus ideal for wave propogation problems.

Chapter 2

Scalar conservation law

2.1 Scalar conservation law

A scalar conservation law is a partial differential equation of the form

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

where

- u is called the conserved variable
- $f(u)$ is the flux of u

Such an equation is a conservation law because, if we integrate the equation on any interval $[a, b]$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_a^b u(x,t)\mathrm{d}x + f(u(b,t)) - f(u(a,t)) = 0$$

This says that

$$\text{Rate of change of total } u \text{ in } [a,b] = \text{Net flux of } u \text{ into } [a,b]$$

If the net flux is zero, then the total quantity is conserved. Moreover, these are hyperbolic conservation laws since their solutions have wave-like properties.

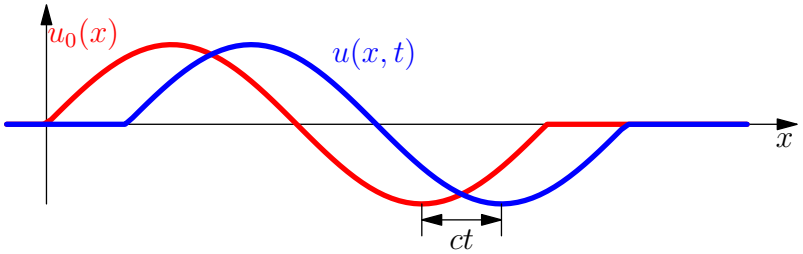
2.1.1 Linear convection equation

The flux is a linear function of the conserved variable, $f(u) = cu$, where $c =$ constant

$$u_t + cu_x = 0, \qquad u(x,0) = u_0(x)$$

The exact solution is given by

$$u(x,t) = u_0(x - ct)$$



The initial condition is transported with velocity c without change of form.

2.1.2 Burger's equation

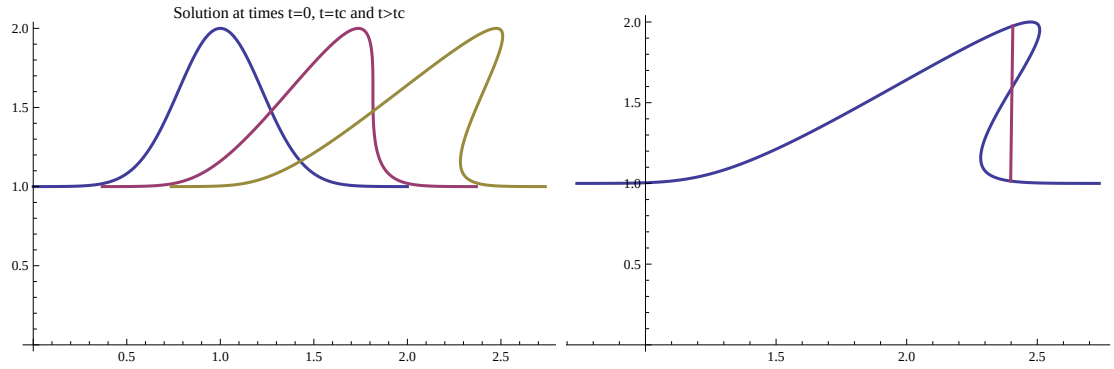
If the flux is of the form $f(u) = \frac{1}{2}u^2$, then the conservation law is

$$u_t + (\frac{1}{2}u^2)_x = 0 \qquad \text{or} \qquad u_t + uu_x = 0 \qquad \text{if } u \text{ is smooth}$$

As long as the solution is smooth, it is given by

$$u(x,t) = u_0(x - u(x,t)t)$$

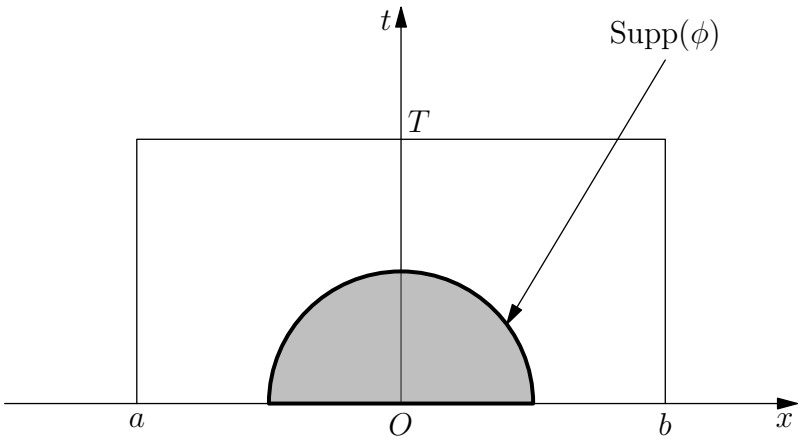
which is an implicit solution. If we use the above formula to find the solution for a smooth initial condition as shown in figure, we find that after some time, the solution becomes multi-valued, which does not make physical sense. We will modify the solution before it becomes multi-valued by introducing a discontinuity. The solution then is no longer is a classical solution since it does not satisfy the PDE. For non-linear problems, this is the general situation; even inifintitely smooth initial condition can eventually develop discontiuities. We have to now enlarge the notion of solution of the PDE by allowing discontinuities, known as weak solutions.



2.2 Weak solution

Take a smooth test function with compact support, say $\phi \in C_0^1(\mathbb{R} \times \mathbb{R}^+)$

$$\int_0^\infty \int_{\mathbb{R}} \left(\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} \right) \phi \mathrm{d}x\mathrm{d}t = 0$$



Perform an integration by parts on both terms in the equation so that the derivatives are transfered to the test function. This motivates the following definition.

Definition 2.1 (Weak solution)**.** *A function $u : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}$ is a weak solution of the IVP*

$$u_t + f(u)_x = 0, \quad (x,t) \in \mathbb{R} \times \mathbb{R}^+, \qquad u(x,0) = u_0(x)$$

together with locally integrable initial data u_0, if u is locally integrable and satisfies

$$\int_0^\infty \int_{-\infty}^\infty (u\phi_t + f(u)\phi_x)\,dx\,dt + \int_{-\infty}^\infty u_0(x)\phi(x,0)\,dx = 0, \qquad \forall \phi \in C_0^1(\mathbb{R} \times \mathbb{R}^+)$$

Lemma 2.1 (Classical solution)**.** *Let $u \in C^1(\mathbb{R} \times \mathbb{R}^+)$ be a weak solution. Then it is a classical solution.*

2.2.1 Rankine-Hugoniot condition

Suppose there is a discontinuity at $x(t)$ and let

$$s = \frac{\mathrm{d}}{\mathrm{d}t}x(t) = \text{shock speed}$$

Then the two values $u(x^-(t),t)$ and $u(x^+(t),t)$ satisfy the RH condition

$$f(u(x^+(t),t)) - f(u(x^-(t),t)) = s(u(x^+(t),t) - u(x^-(t),t))$$

Definition 2.2 (Weak solution)**.** *A weak solution u is a piecewise smooth solution which satisfies the RH condition at the points of discontinuity of u.*

2.3 Kruzkov's result

The scalar Cauchy problem for

$$u_t + f(u)_x = 0, \quad f \in C^1(\mathbb{R})$$

with initial condition

$$u(0,x) = u_0(x), \quad u_0 \in L^\infty(\mathbb{R})$$

has a **unique entropy solution**

$$u \in L^\infty(\mathbb{R}_+ \times \mathbb{R})$$

which fulfills (important for numerics)

1. Stability: $||u(t,\cdot)||_{L^\infty} \leq ||u_0||_{L^\infty}$, a.e. in $t \in \mathbb{R}_+$
2. Monotone: if $u_0 \geq v_0$ a.e. in $\mathbb{R}$, then
$$u(t,\cdot) \geq v(t,\cdot) \quad \text{a.e. in } \mathbb{R}, \text{ a.e. in t} \in \mathbb{R}_+$$
3. TV-diminishing: if $u_0 \in BV(\mathbb{R})$ then
$$u(t,\cdot) \in BV(\mathbb{R}) \quad \text{and} \quad TV(u(t,\cdot)) \leq TV(u_0)$$
4. Conservation: if $u_0 \in L^1(\mathbb{R})$ then
$$\int_{\mathbb{R}} u(t,x)\mathrm{d}x = \int_{\mathbb{R}} u_0(x)\mathrm{d}x, \quad \text{a.e. in } t \in \mathbb{R}_+$$

5. Finite domain of dependence: if u, v are two entropy solutions corresponding to $u_0, v_0 \in L^\infty$ and

$$M = \max_{\phi}\{|f'(\phi)| : |\phi| \leq \max(||u_0||_{L^\infty}, ||v_0||_{L^\infty})\}$$

then

$$\int_{|x|\leq R} |u(t,x) - v(t,x)|\mathrm{d}x \leq \int_{|x|\leq R+Mt} |u_0(x) - v_0(x)|\mathrm{d}x$$

Chapter 3

DG scheme in 1-D

Let us consider a general conservation law of the form

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0$$

and try to construct the DG scheme for this problem.

3.1 Mesh and approximation space

Divide domain $\Omega = [0, 1]$ into disjoint cells $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ using the parition

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \ldots < x_{N+\frac{1}{2}} = 1$$

as shown in figure (3.1). Define the cell center and cell size as

$$x_i = \frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}), \qquad \Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \qquad h = \max_i \Delta x_i$$

Define the space of broken polynomials

$$V_h^k = \{v \in L^2(\Omega) : v|_{I_i} \in \mathbb{P}_k(I_i),\ 1 \leq i \leq N\}$$

Note that these functions can be discontinuous on the boundary of the elements as shown in the figure (3.2). Define the left and right limits

$$v_h(x^-) = \lim_{\epsilon \searrow 0} v_h(x - \epsilon), \qquad v_h(x^+) = \lim_{\epsilon \searrow 0} v_h(x + \epsilon)$$

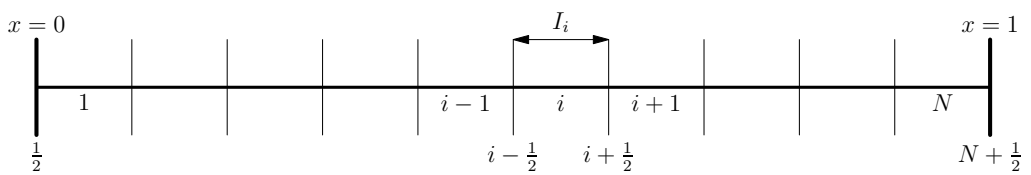


Figure 3.1: Mesh for DG scheme

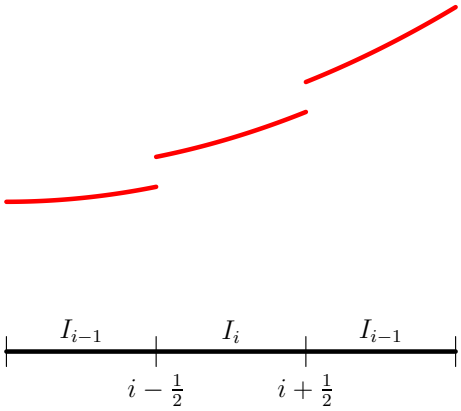


Figure 3.2: Discontinuous solution

3.2 Semi-discrete DG scheme

Multiply conservation law by a smooth test function v

$$\int_{I_i} \left(\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} \right) v \mathrm{d}x = 0$$

and integrate by parts on flux derivative term

$$\int_{I_i} \frac{\partial u}{\partial t} v \mathrm{d}x - \int_{I_i} f(u) \frac{\partial v}{\partial x} \mathrm{d}x \\ + f(x_{i+\frac{1}{2}}, t) v(x^-_{i+\frac{1}{2}}) - f(x_{i-\frac{1}{2}}, t) v(x^+_{i-\frac{1}{2}}) = 0$$

We now want to replace u with u_h and v with v_h, and u_h, v_h belong to finite dimensional spaces usually comprising of polynomials. At $x = x_{i+\frac{1}{2}}$, u_h may be discontinuous, i.e., $u_h(x^-_{i+\frac{1}{2}}, t) \neq u_h(x^+_{i+\frac{1}{2}}, t)$. In this case, how to compute the flux $f(x_{i+\frac{1}{2}}, t)$? Following the finite volume method, we will approximate this flux by a *numerical flux function* denoted by

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

leading to

Definition 3.1 (Semi-discrete DG scheme)**.** *Find $u_h(\cdot, t) \in V_h^k$ such that for all $v_h \in V_h^k$*

$$\int_{I_i} \frac{\partial u_h}{\partial t} v_h dx - \int_{I_i} f(u_h) \frac{\partial v_h}{\partial x} dx \\ + \hat{f}_{i+\frac{1}{2}}(t) v_h(x^-_{i+\frac{1}{2}}) - \hat{f}_{i-\frac{1}{2}}(t) v_h(x^+_{i-\frac{1}{2}}) = 0 \tag{3.1}$$

The initial condition is obtained by an L^2 projection onto the finite element solution space V_h^k, i.e.,

$$\int_{I_i} u_h(x, 0) v_h(x) \mathrm{d}x = \int_{I_i} u(x, 0) v_h(x) \mathrm{d}x, \qquad \forall\ v_h \in V_h^k$$

Note that the numerical flux couples the solution in I_i to those in the neighbouring elements.

3.3 Relation to finite volume scheme

If the degree is $k = 0$, then we have only the constant basis function $v_h = 1$, and the solution is piecewise constant

$$u_h(x) = u_i, \qquad x \in (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$$

The DG scheme takes the form

$$\Delta x_i \frac{\mathrm{d}u_i}{\mathrm{d}t} + \hat{f}(u_i, u_{i+1}) - \hat{f}(u_{i-1}, u_i) = 0$$

which is the finite volume scheme. Thus the DG scheme can be considered as a generalization of finite volume scheme to higher order of accuracy. In FVM only the cell average values are updated by the scheme and to achieve higher order accuracy, the solution has to be reconstructed by piecewise polynomials inside each cell using the cell average values. In DG schemes, we start with a polynomial solution representation inside each cell and evolve the entire polynomial forward in time by using the DG scheme.

3.4 Conservation property

Finite volume schemes are conservative which is necessary to compute correct weak solutions. We will now show that DG schemes are also conservative. Let degree $k > 0$. Take the test function $v_h \in V_h^k$ of the form

$$v_h = \begin{cases} 1 & x \in I_i \\ 0 & \text{otherwise} \end{cases}$$

The DG scheme (17.1) becomes

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{I_i} u_h \mathrm{d}x + \hat{f}_{i+\frac{1}{2}}(t) - \hat{f}_{i-\frac{1}{2}}(t) = 0$$

which is a statement of conservation of u in the element I_i.

Chapter 4

Numerical flux

The numerical flux is an important ingredient of the DG scheme. We can make use of the developments in finite volume methods to compute the numerical flux and there are many possible schemes for this. In any case, we desire some common properties of the numerical flux as follows.

4.1 Properties of numerical flux

1. Consistency
$$\hat{f}(u,u) = f(u)$$
2. Locally Lipschitz continuous
$$|\hat{f}(a_2,b_2) - \hat{f}(a_1,b_1)| \leq L_1|a_2 - a_1| + L_2|b_2 - b_1|$$
3. monotone
$$\hat{f}(a,b), \qquad \text{increasing in } a \text{ and decreasing in } b, \quad f(\uparrow,\downarrow)$$
In terms of the derivatives, we can state this property as
$$\frac{\partial}{\partial a}\hat{f}(a,b) \geq 0, \qquad \frac{\partial}{\partial b}\hat{f}(a,b) \leq 0$$

4.2 Riemann solver

One way to obtain a numerical flux is to solve a **Riemann problem**. To compute the flux $\hat{f}_{i+\frac{1}{2}}$, we solve the Riemann problem

$$\frac{\partial w}{\partial \tau} + \frac{\partial f(w)}{\partial x} = 0, \qquad w(x,\tau = t) = \begin{cases} u(x^-_{i+\frac{1}{2}},t) & x < x_{i+\frac{1}{2}} \\ u(x^+_{i+\frac{1}{2}},t) & x > x_{i+\frac{1}{2}} \end{cases}, \qquad \tau \geq t$$

This has a self-similar solution

$$w(x,\tau) = w_R\left(\frac{x - x_{i+\frac{1}{2}}}{t-\tau}; u(x^-_{i+\frac{1}{2}},t), u(x^+_{i+\frac{1}{2}},t)\right)$$

and the flux across $x = x_{i+\frac{1}{2}}$ is given by

$$\hat{f}_{i+\frac{1}{2}} = f(w_R(0; u(x^-_{i+\frac{1}{2}},t), u(x^+_{i+\frac{1}{2}},t)) \qquad \text{(Godunov flux)}$$

In practice, the Riemann problem is solved approximately.

4.3 Numerical flux: Linear convection equation

The flux is linear $f(u) = au$ where $a =$ constant and the equation is of the form

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0$$

The upwind numerical flux is given by

$$\hat{f}(u^-, u^+) = \begin{cases} au^- & \text{if } a \geq 0 \\ au^+ & \text{if } a < 0 \end{cases}$$

This can also be written as

$$\hat{f}(u^-, u^+) = \underbrace{\frac{1}{2}[f(u^-) + f(u^+)]}_{\text{centered flux}} - \underbrace{\frac{1}{2}|a|(u^+ - u^-)}_{\text{dissipative flux}}$$

The second part of the flux contributes to the numerical dissipation that gives rise to stability.

4.4 Numerical flux: Non-linear conservation law

Here we list some standard numerical fluxes and the reader must consult the literature on finite volume schemes for more details.

4.4.1 Godunov flux

This is obtained by exactly solving the Riemann problem. The general formula is

$$\hat{f}(u^-, u^+) = \begin{cases} \min\limits_{u \in [u^-, u^+]} f(u), & \text{if} \quad u^- \leq u^+ \\ \max\limits_{u \in [u^+, u^-]} f(u), & \text{if} \quad u^- > u^+ \end{cases}$$

For convex flux, we can give simpler expression as follows. Let u^* be the only sonic point, i.e., $f'(u^*) = 0$. Then the Godunov flux is given by

$$\hat{f} = \max\{f(\max\{u^*, u^-\}), f(\min\{u^*, u^+\})\}$$

4.4.2 Lax-Friedrich flux

This may be considered as a generalization of the upwind flux formula by using the wave speed to be $\Delta x/\Delta t$

$$\hat{f}(u^-, u^+) = \frac{1}{2}[f(u^-) + f(u^+)] - \frac{1}{2}\frac{\Delta x}{\Delta t}(u^+ - u^-)$$

4.4.3 Rusanov flux

This flux is also refered to as a local Lax-Friedrich flux and it makes use of a local wave speed estimate in the dissipative flux

$$\hat{f} = \frac{1}{2}[f(u^-) + f(u^+)] - \frac{1}{2}\lambda(u^+ - u^-)$$

where

$$\lambda = \max_{\xi \in (u^-, u^+)} |f'(\xi)|$$

A simple choice, which is exact for convex fluxes, is

$$\lambda = \max\{|f'(u^-)|, |f'(u^+)|\}$$

4.4.4 Roe flux

This flux is based on a approximate Riemann solver. The non-linear Riemann problem is replaced by the linear problem

$$\frac{\partial u}{\partial t} + \lambda \frac{\partial u}{\partial x} = 0$$

and the exact solution of this problem gives the flux

$$\hat{f} = \frac{1}{2}[f(u^-) + f(u^+)] - \frac{1}{2}\lambda(u^+ - u^-)$$

where

$$\lambda = \left| \frac{f(u^+) - f(u^-)}{u^+ - u^-} \right|$$

We have to modify the Roe scheme to satisfy entropy condition.

Remark 4.1. *Many of the numerical fluxes have the upwind property in the sense that if $f' \geq 0$ in the Riemann problem, then $\hat{f}(u^-, u^+) = f(u^-)$. This is not the case for Lax-Friedrich and Rusanov fluxes, which are hence called central fluxes.*

Chapter 5

Stability

5.1 Linear convection equation

Let us consider the linear convection equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \qquad x \in (0,1)$$

The semi-discrete DG scheme is given by

$$\int_{I_i} \frac{\partial u_h}{\partial t} v_h \mathrm{d}x - \int_{I_i} a u_h \frac{\partial v_h}{\partial x}\mathrm{d}x + \hat{f}_{i+\frac{1}{2}}(t) v_h(x^-_{i+\frac{1}{2}}) - \hat{f}_{i-\frac{1}{2}}(t) v_h(x^+_{i-\frac{1}{2}}) = 0$$

Taking $v_h = u_h$

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\int_{I_i} u_h^2 \mathrm{d}x - \frac{a}{2}\left[u_h^2(x^-_{i+\frac{1}{2}},t) - u_h^2(x^+_{i-\frac{1}{2}},t)\right] \\ + \hat{f}_{i+\frac{1}{2}}(t) u_h(x^-_{i+\frac{1}{2}},t) - \hat{f}_{i-\frac{1}{2}}(t) u_h(x^+_{i-\frac{1}{2}},t) = 0\end{aligned}$$

Summing over all the cells, we obtain the energy equation

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|^2 + \frac{a}{2}u_h^2(x_{\frac{1}{2}},t) - \hat{f}_{\frac{1}{2}}(t) u_h(x_{\frac{1}{2}},t) + \sum_{i=1}^{N-1}\left[\frac{a}{2}[\![u_h^2]\!]_{i+\frac{1}{2}} - \hat{f}_{i+\frac{1}{2}}(t)[\![u_h]\!]_{i+\frac{1}{2}}\right] \\ - \frac{a}{2}u_h^2(x_{N+\frac{1}{2}},t) + \hat{f}_{N+\frac{1}{2}}(t) u_h(x_{N+\frac{1}{2}},t) = 0\end{aligned}$$

where the jump term is defined as

$$[\![v]\!]_{i+\frac{1}{2}} = v(x^+_{i+\frac{1}{2}}) - v(x^-_{i+\frac{1}{2}})$$

For the upwind numerical flux, which is given by

$$\hat{f}_{i+\frac{1}{2}} = \frac{a}{2}[u_h(x^-_{i+\frac{1}{2}},t) + u_h(x^+_{i+\frac{1}{2}},t)] - \frac{|a|}{2}[\![u_h]\!]_{i+\frac{1}{2}}$$

we can write

$$\frac{a}{2}[\![u_h^2]\!]_{i+\frac{1}{2}} - \hat{f}_{i+\frac{1}{2}}(t)[\![u_h]\!]_{i+\frac{1}{2}} = \frac{|a|}{2}[\![u_h]\!]^2_{i+\frac{1}{2}}$$

so that the energy equation is

$$\begin{aligned}\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|^2 + \frac{a}{2}u_h^2(x_{\frac{1}{2}},t) - \hat{f}_{\frac{1}{2}}(t) u_h(x_{\frac{1}{2}},t) + \sum_{i=1}^{N-1}\frac{|a|}{2}[\![u_h]\!]^2_{i+\frac{1}{2}} \\ - \frac{a}{2}u_h^2(x_{N+\frac{1}{2}},t) + \hat{f}_{N+\frac{1}{2}}(t) u_h(x_{N+\frac{1}{2}},t) = 0\end{aligned} \tag{5.1}$$

5.1.1 Periodic boundary conditions

In this case

$$\hat{f}_{\frac{1}{2}} = \hat{f}_{N+\frac{1}{2}} = \hat{f}(u_h(x_{N+\frac{1}{2}}, t), u_h(x_{\frac{1}{2}}, t))$$

and (5.1) simpifies to

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|^2 + \frac{|a|}{2}\sum_{i=1}^{N}[\![u_h]\!]^2_{i+\frac{1}{2}} = 0$$

where

$$[\![u_h]\!]_{N+\frac{1}{2}} = u_h(x_{N+\frac{1}{2}}) - u_h(x_{\frac{1}{2}})$$

Integrating over time

$$\|u_h(T)\|^2 + |a|\sum_{i=1}^{N}\int_0^T [\![u_h]\!]^2_{i+\frac{1}{2}}\mathrm{d}t = \|u_h(0)\|^2$$

This immediately implies that

$$\|u_h(t)\| \leq \|u_h(0)\|$$

The use of upwind scheme together with discontinuous basis functions leads to L^2 stability. If we use the central flux

$$\hat{f}_{i+\frac{1}{2}} = \frac{a}{2}[u_h(x^-_{i+\frac{1}{2}}, t) + u_h(x^+_{i+\frac{1}{2}}, t)]$$

then we conclude that

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|^2 = 0$$

and in this case, the energy is conserved, $\|u_h(T)\| = \|u_h(0)\|$. The scheme does not have any dissipation and is neutrally stable. However, the central flux should not be used in practice since it gives less accurate solutions than the upwind flux and will be unstable for non-linear problems.

5.1.2 Dirichlet condition

Assume that $a > 0$. In this case, we can specify boundary conditions only at $x = x_{\frac{1}{2}} = 0$, i.e.,

$$u(0, t) = g(t)$$

We take the numerical flux at the boundaries as

$$\hat{f}_{\frac{1}{2}}(t) = ag(t), \qquad \hat{f}_{N+\frac{1}{2}}(t) = au_h(x_{N+\frac{1}{2}}, t)$$

which is an upwind flux. This corresponds to a weak implementation of the boundary condition since the solution does not satisfy $u_h(x_{\frac{1}{2}}, t) = g(t)$. The energy equation is

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|^2 + \frac{a}{2}u_h^2(x_{\frac{1}{2}}, t) - ag(t)u_h(x_{\frac{1}{2}}, t) + \frac{|a|}{2}\sum_{i=1}^{N-1}[\![u_h]\!]^2_{i+\frac{1}{2}} + \frac{a}{2}u_h^2(x_{N+\frac{1}{2}}, t) = 0$$

The boundary terms can be rearranged as

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u_h\|^2 + \frac{a}{2}[g(t) - u_h(x_{\frac{1}{2}}, t)]^2 - \frac{a}{2}g^2(t) + \frac{|a|}{2}\sum_{i=1}^{N-1}[\![u_h]\!]^2_{i+\frac{1}{2}} + \frac{a}{2}u_h^2(x_{N+\frac{1}{2}}, t) = 0$$

Integrating in time, we obtain the energy equation

$$\|u_h(T)\|^2 + \int_0^T \left\{ \frac{a}{2}[g(t) - u_h(x_{\frac{1}{2}}, t)]^2 + \frac{|a|}{2} \sum_{i=1}^{N-1} [\![u_h]\!]^2_{i+\frac{1}{2}} + \frac{a}{2} u_h^2(x_{N+\frac{1}{2}}, t) \right\} \mathrm{d}t$$
$$= \|u_h(T)\|^2 + \frac{a}{2} \int_0^T g^2(t) \mathrm{d}t$$

The jump terms at the boundary and interior faces lead to dissipation of energy giving the inequality

$$\|u_h(T)\|^2 \leq \|u_h(0)\|^2 + \frac{a}{2} \int_0^T g^2(t) \mathrm{d}t \leq \|u(0)\|^2 + \frac{a}{2} \int_0^T g^2(t) \mathrm{d}t$$

The energy in the numerical solution is bounded by the energy in the exact solution. If $g(t) \equiv 0$ then the energy will decrease with time and we have L^2 stability.

5.2 Non-linear conservation law

Weak solutions of non-linear conservation laws can be non-unique. To obtain a unique weak solution, we need to impose an entropy condition.

5.2.1 Entropy condition

Let $U(u)$ be a convex entropy function and let $F(u)$ be an associated entropy flux such that

$$F'(u) = U'(u) f'(u) \tag{5.2}$$

If u is a smooth solution, then it satisfies the equation

$$\frac{\partial u}{\partial t} + f'(u) \frac{\partial u}{\partial x} = 0$$

Multiplying throughout by $U'(u)$

$$U'(u) \frac{\partial u}{\partial t} + U'(u) \underbrace{f'(u) \frac{\partial u}{\partial x}}_{\frac{\partial f}{\partial x}} = 0 \quad \Longrightarrow \quad \frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0$$

we see that the smooth solution satisfies an additional conservation law. For a discontinuous solution, we will demand that it satisfy the entropy inequality

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} \leq 0$$

in the sense of distributions with equality in smooth regions. Then Kruzkov theory shows that the weak solution is unique.

Theorem 5.1 (Cell entropy inequality)**.** *The solution u_h of the semi-discrete DG scheme satisfies*

$$\frac{d}{dt} \int_{I_i} U(u_h) dx + \hat{F}_{i+\frac{1}{2}}(t) - \hat{F}_{i-\frac{1}{2}}(t) \leq 0$$

for the square entropy $U(u) = \frac{1}{2}u^2$ with some consistent numerical entropy flux $\hat{F}_{i+\frac{1}{2}}(t) = \hat{F}(u_h(x^-_{i+\frac{1}{2}}, t), u_h(x^+_{i+\frac{1}{2}}, t))$.

Proof: Take $v_h = u_h$ in the DG scheme

$$\int_{I_i} \frac{\partial u_h}{\partial t} u_h \mathrm{d}x - \int_{I_i} f(u_h) \frac{\partial u_h}{\partial x} \mathrm{d}x + \hat{f}_{i+\frac{1}{2}}(t) u_h(x^-_{i+\frac{1}{2}}, t) - \hat{f}_{i-\frac{1}{2}}(t) u_h(x^+_{i-\frac{1}{2}}, t) = 0$$

Define

$$\tilde{F}(u) = \int_0^u f(s)\mathrm{d}s \qquad \Longrightarrow \qquad \tilde{F}'(u) = f(u)$$

Integrating the compatibility condition (5.2), we get

$$F(u) - F(0) = uf(u) - \int_0^u f(s)\mathrm{d}s = uf(u) - \tilde{F}(u)$$

Ignoring the constant term $F(0)$ we have $F(u) = uf(u) - \tilde{F}(u)$. Then

$$-\int_{I_i} f(u_h) \frac{\partial u_h}{\partial x} \mathrm{d}x = -\int_{I_i} \tilde{F}'(u_h) \frac{\partial u_h}{\partial x} \mathrm{d}x = -\tilde{F}(u_h(x^-_{i+\frac{1}{2}}, t)) + \tilde{F}(u_h(x^+_{i-\frac{1}{2}}, t))$$

so that the entropy equation becomes

$$\int_{I_i} \frac{\partial U(u_h)}{\partial t} - \tilde{F}(u_h(x^-_{i+\frac{1}{2}}, t)) + \tilde{F}(u_h(x^+_{i-\frac{1}{2}}, t)) + \hat{f}_{i+\frac{1}{2}}(t) u_h(x^-_{i+\frac{1}{2}}, t) - \hat{f}_{i-\frac{1}{2}}(t) u_h(x^+_{i-\frac{1}{2}}, t) = 0$$

This can be re-written as

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{I_i} U(u_h)\mathrm{d}x + \hat{F}_{i+\frac{1}{2}} - \hat{F}_{i-\frac{1}{2}} + \Theta_{i-\frac{1}{2}} = 0$$

with consistent numerical entropy flux

$$\hat{F}_{i+\frac{1}{2}} = -\tilde{F}(u_h(x^-_{i+\frac{1}{2}})) + \hat{f}_{i+\frac{1}{2}} u_h(x^-_{i+\frac{1}{2}})$$

and

$$\Theta_{i-\frac{1}{2}} = -\tilde{F}(u_h(x^-_{i-\frac{1}{2}})) + \hat{f}_{i-\frac{1}{2}} u_h(x^-_{i-\frac{1}{2}}) + \tilde{F}(u_h(x^+_{i-\frac{1}{2}})) - \hat{f}_{i-\frac{1}{2}} u_h(x^+_{i-\frac{1}{2}})$$

Ignoring the suffixes, the quantity Θ can be written as

$$\begin{aligned} \Theta &= \tilde{F}(u^+) - \tilde{F}(u^-) + (u^- - u^+)\hat{f} \\ &= (u^+ - u^-)(\tilde{F}'(\xi) - \hat{f}), \qquad \min\{u^-, u^+\} \le \xi \le \max\{u^-, u^+\} \\ &= (u^+ - u^-)(f(\xi) - \hat{f}) \end{aligned}$$

Now assume that $u^+ \ge \xi \ge u^-$. Then since $\hat{f}$ is a monotone flux

$$f(\xi) = \hat{f}(\xi, \xi) \ge \hat{f}(u^-, \xi) \ge \hat{f}(u^-, u^+)$$

and hence $\Theta \ge 0$. In the case $u^+ \le \xi \le u^-$ we can again show that $\Theta \ge 0$. Thus *the semi-discrete DG scheme satisfies the entropy condition for any order of the basis functions* k. □

Remark 5.1. *To obtain entropy inequality, we can also use the* **E-flux** *condition*

$$(u^+ - u^-)(f(\xi) - \hat{f}(u^-, u^+)) \ge 0, \qquad \forall \xi \text{ between } u^-, u^+$$

This condition can be extended to system of conservation laws (Barth).

Corollary 5.1 (L^2 stability)**.** *For periodic or compactly supported boundary conditions, the semi-discrete DG scheme satisfies*

$$\frac{d}{dt}\int_\Omega u_h^2\,dx \le 0$$

which implies that

$$\|u_h(t)\| \le \|u_h(0)\| \le \|u(0)\|$$

Proof: Adding the cell entropy inequality from all the cells

$$\sum_{i=1}^{N}\frac{\mathrm{d}}{\mathrm{d}t}\int_{I_i} U(u_h)\mathrm{d}x + \sum_{i=1}^{N}[\hat{F}_{i+\frac{1}{2}}(t) - \hat{F}_{i-\frac{1}{2}}(t)] \le 0$$

the internal fluxes cancel one another, leading to

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega u_h^2\mathrm{d}x + \hat{F}_{N+\frac{1}{2}} - \hat{F}_{\frac{1}{2}} \le 0$$

For periodic case, $\hat{F}_{\frac{1}{2}} = \hat{F}_{N+\frac{1}{2}}$, while for compactly supported case $\hat{F}_{\frac{1}{2}} = \hat{F}_{N+\frac{1}{2}} = 0$, we obtain desired result. □

Chapter 6

Error estimates

6.1 Error estimate: $u_t + cu_x = 0$, semi-discrete scheme

Theorem 2.1 (First L^2-error estimate) *Suppose that the initial condition u_0 belongs to $H^{k+1}(0,1)$. Let e be the approximation error $u - u_h$. Then we have,*

$$\| e(T) \|_{L^2(0,1)} \leq C \, | u_0 |_{H^{k+1}(0,1)} (\Delta x)^{k+1/2},$$

where C depends solely on k, $|c|$, and T.

Theorem 2.2 (Second L^2-error estimate) *Suppose that the initial condition u_0 belongs to $H^{k+2}(0,1)$. Let e be the approximation error $u - u_h$. Then we have,*

$$\| e(T) \|_{L^2(0,1)} \leq C \, | u_0 |_{H^{k+2}(0,1)} (\Delta x)^{k+1},$$

where C depends solely on k, $|c|$, and T.

(B. Cockburn, Lecture notes on *Discontinuous Galerkin methods for convection dominated problem*)

Chapter 7

Basis functions

We have to construct basis functions for V_h^k for which there are two approaches: **nodal** and **modal**. The DG solution has the form

$$x \in I_i : \qquad u_h(x,t) = \sum_{j=0}^{k} u_{ij}(t)\phi_{ij}(x), \qquad \dim(V_h^k) = k+1$$

and we refer to the set of values $\{u_{ij} : 0 \leq j \leq k\}$ as the *degrees of freedom* or *dof* associated with the i'th cell.

7.1 Nodal basis functions

A degree k polynomial is determined by $k+1$ values. In the **nodal** approach, we choose $k+1$ distinct nodes in each cell

$$x_{ij} \in I_i, \qquad j = 0, 1, 2, \ldots, k$$

These nodes can be used to define the Lagrange polynomials of degree k which have the interpolation property

$$\phi_{ij}(x_{il}) = \delta_{jl}, \qquad 0 \leq j, l \leq k$$

This property implies that

$$u_h(x_{ij}) = u_{ij}, \qquad 0 \leq j \leq k$$

so that the dofs in this case are the solution values at the nodes as shown in figure (7.1). The location of the $k+1$ nodal points x_{ij} can be

- uniformly distributed inside I_i
  - We may encounter Runge phenomenon for high degree polynomials
  - The mass matrix is full and ill-conditioned for large degree k
- based on Gauss-Legendre or Gauss-Lobatto integration points

Once the nodal points are chosen, the basis functions can be obtained from Lagrange interpolation.

- $\underline{k=0}$: There is only one dof per element which is the cell average value, so the basis functions are piecewise constant

$$\phi_{i,0}(x) = \begin{cases} 1 & x \in I_i \\ 0 & \text{otherwise} \end{cases}$$

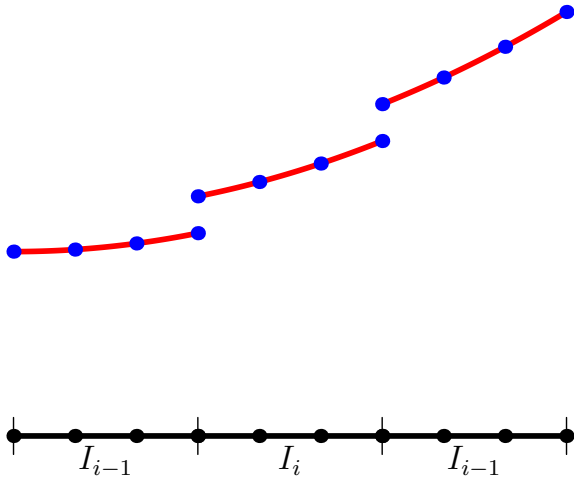


Figure 7.1: Nodal solution representation

- $\underline{k=1}$: There are two dof per element; we can choose the two end points as the nodes, $x_{i,0} = x_{i-\frac{1}{2}}$, $x_{i,1} = x_{i+\frac{1}{2}}$

$$\phi_{i,0}(x) = \begin{cases} \frac{x_{i+\frac{1}{2}} - x}{\Delta x_i} & x \in I_i \\ 0 & \text{otherwise} \end{cases}, \qquad \phi_{i,1}(x) = \begin{cases} \frac{x - x_{i-\frac{1}{2}}}{\Delta x_i} & x \in I_i \\ 0 & \text{otherwise} \end{cases}$$

- $\underline{k=2}$: There are three dof per element; we can choose the end points and the middle point as the nodes, $x_{i,0} = x_{i-\frac{1}{2}}$, $x_{i,1} = x_i$, $x_{i,2} = x_{i+\frac{1}{2}}$

$$\phi_{i,0}(x) = \frac{(x - x_{i,1})(x - x_{i,2})}{(x_{i,0} - x_{i,1})(x_{i,0} - x_{i,2})}, \quad \phi_{i,1}(x) = \frac{(x - x_{i,0})(x - x_{i,2})}{(x_{i,1} - x_{i,0})(x_{i,1} - x_{i,2})},$$

$$\phi_{i,2}(x) = \frac{(x - x_{i,0})(x - x_{i,1})}{(x_{i,2} - x_{i,0})(x_{i,2} - x_{i,1})}$$

- $\underline{\text{In general}}$: We choose $k+1$ distinct nodes $\{x_{i,0}, x_{i,1}, \ldots, x_{i,k}\} \subset I_i$

$$\phi_{ij}(x) = \frac{(x - x_{i,0}) \ldots (x - x_{i,j-1})(x - x_{i,j+1}) \ldots (x - x_{i,k})}{(x_{i,j} - x_{i,0}) \ldots (x_{i,j} - x_{i,j-1})(x_{i,j} - x_{i,j+1}) \ldots (x_{i,j} - x_{i,k})}$$

7.2 Mapped nodal basis functions

It is efficient to compute the shape functions on a reference cell instead of defining them in terms of the real coordinate x as we have done until now. Let us map cell I_i to $[-1, +1]$ by

$$\xi = \frac{x - x_i}{\frac{1}{2}\Delta x_i}, \qquad x = \frac{1 - \xi}{2} x_{i-\frac{1}{2}} + \frac{1 + \xi}{2} x_{i+\frac{1}{2}}$$

Choose a set of distinct nodes $\xi_0, \xi_1, \ldots, \xi_k \in [-1, +1]$. The j'th basis function is given by

$$\phi_{ij}(x) = \hat{\varphi}_j(\xi) = \frac{(\xi - \xi_0) \ldots (\xi - \xi_{j-1})(\xi - \xi_{j+1}) \ldots (\xi - \xi_k)}{(\xi_j - \xi_0) \ldots (\xi_j - \xi_{j-1})(\xi_j - \xi_{j+1}) \ldots (\xi_j - \xi_k)}$$

Remark 7.1. *If nodes are located at the element boundaries, then we have multiple dofs at the boundary, see Figure (7.1), since the solution is in general discontinuous.*

Remark 7.2. *The direct evaluation of Lagrange polynomials in the form given above can be costly and error prone. A better way to compute them is based on barycentric Lagrange interpolation form.*

7.3 Taylor basis functions

Here we do not use nodal basis functions, but use Taylor series to generate the basis functions. The basis functions are defined in terms of the physical spatial coordinate x. The cell average value u_i is one of the degrees freedom. The other dof are the gradient, hessian, etc.

Define the **moments**

$$m_{is} = \frac{1}{s!}\int_{I_i}\left(\frac{x-x_i}{\Delta x_i}\right)^s \mathrm{d}x, \qquad s = 1, 2, \ldots$$

Then the solution for different degree are taken as follows.
$\underline{k=1}$: Dofs are (u_i, s_i) and the linear polynomial is

$$u_h(x,t) = u_i(t) + \frac{x-x_i}{\Delta x_i} s_i(t)$$

Hence the two basis functions are

$$\phi_{i,0}(x) = \begin{cases} 1 & x \in I_i \\ 0 & \text{otherwise} \end{cases}, \qquad \phi_{i,1}(x) = \begin{cases} \frac{x-x_i}{\Delta x_i} & x \in I_i \\ 0 & \text{otherwise} \end{cases}$$

$\underline{k=2}$: Dofs are (u_i, s_i, q_i) and the quadratic polynomial is

$$u_h(x,t) = u_i(t) + \frac{x-x_i}{\Delta x_i} s_i(t) + \left[\frac{1}{2}\left(\frac{x-x_i}{\Delta x_i}\right)^2 - m_{i2}\right] q_i(t)$$

The extra basis function is

$$\phi_{i,2}(x) = \begin{cases} \frac{1}{2}\left(\frac{x-x_i}{\Delta x_i}\right)^2 - m_{i2} & x \in I_i \\ 0 & \text{otherwise} \end{cases}$$

In general, the basis functions are chosen as

$$\phi_{i,j}(x) = \frac{1}{j!}\left(\frac{x-x_i}{\Delta x_i}\right)^j - m_{ij}, \qquad j = 0, 1, 2, \ldots$$

We thus have following property for the basis functions

$$\int_{I_i} \phi_{i,0} \mathrm{d}x = \Delta x_i, \qquad \int_{I_i} \phi_{i,j} \mathrm{d}x = 0, \quad j = 1, 2, 3, \ldots$$

In each case, the first coefficient u_i is the cell average of the solution. We have a heirarchical representation of the solution and the mass matrix is diagonal for $k \leq 2$. This idea can be extended to multi-dimensions on arbitrary polygonal elements. It is also possible to generate an orthogonal set of basis functions using a Gram-Schmidt process.

7.4 Orthogonal polynomials (Modal approach)

The Legendre polynomials are the solution of Legendre's differential equation

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\left[(1-\xi^2)\frac{\mathrm{d}}{\mathrm{d}\xi}P_n(\xi)\right] + n(n+1)P_n(\xi) = 0, \qquad n = 0, 1, 2, \ldots$$

The first few polynomials are given by

$$\begin{array}{ll} P_0(\xi) = 1 & P_1(\xi) = \xi \\ P_2(\xi) = \frac{1}{2}(3\xi^2 - 1) & P_3(\xi) = \frac{1}{2}(5\xi^3 - 3\xi) \\ P_4(\xi) = \frac{1}{8}(35\xi^4 - 30\xi^2 + 3) & P_5(\xi) = \frac{1}{8}(63\xi^5 - 70\xi^3 + 15\xi) \end{array}$$

These polynomials satisfy the following recurrence relation which is useful for numerical computation

$$P_0(\xi) = 1, \qquad P_1(\xi) = \xi$$
$$P_n(\xi) = \left(\frac{2n-1}{n}\right)\xi P_{n-1}(\xi) - \left(\frac{n-1}{n}\right)P_{n-2}(\xi), \qquad n = 2, 3, \ldots$$

A very useful property of these polynomials is that they are mutually orthogonal, i.e.,

$$\int_{-1}^{+1} P_j(\xi)P_l(\xi)\mathrm{d}\xi = \begin{cases} 0 & j \neq l \\ \frac{2}{2j+1} & j = l \end{cases}$$

Using these polynomials, we can define our basis functions as: $j = 0, 1, 2, \ldots$

$$\phi_{ij}(x) = \sqrt{2j+1}P_j\left(\frac{x - x_i}{\Delta x_i/2}\right), \qquad \int_{I_i}\phi_{ij}\phi_{il}\mathrm{d}x = \begin{cases} 0 & j \neq l \\ \Delta x_i & j = l \end{cases}$$

or equivalently

$$\phi_{ij}(x) = \hat{\varphi}_j(\xi) = \sqrt{2j+1}P_j(\xi), \qquad \xi = \frac{x - x_i}{\frac{1}{2}\Delta x_i}$$

We will need derivatives of basis functions also in the DG scheme, and the derivatives of Legendre polynomials can be computed from

$$P_0'(\xi) = 0, \qquad P_1'(\xi) = 1, \qquad P_n'(\xi) = nP_{n-1}(\xi) + \xi P_{n-1}'(\xi)$$

Remark 7.3. *In* `deal.II`*, the* `FE_DGP` *space makes use of these basis functions, but the reference cell is* $[0, 1]$*, so that the definitions are*

$$\phi_{ij}(x) = \hat{\varphi}_j(\xi) = \sqrt{2j+1}P_j(2\xi - 1), \qquad \xi = \frac{x - x_{i-\frac{1}{2}}}{\Delta x_i} \in [0, 1]$$

and we have the orthogonality property

$$\begin{aligned} \int_{I_i}\phi_{ij}\phi_{il}\,dx &= \Delta x_i \int_0^1 \sqrt{(2j+1)(2l+1)}P_j(2\xi-1)P_l(2\xi-1)\,d\xi \\ &= \frac{1}{2}\Delta x_i \int_{-1}^{+1} \sqrt{(2j+1)(2l+1)}P_j(\xi)P_l(\xi)\,d\xi \\ &= \begin{cases} 0 & j \neq l \\ \Delta x_i & j = l \end{cases} \end{aligned}$$

Remark 7.4. *Many authors use the following definition for the basis functions*

$$\phi_{ij}(x) = P_j\left(\frac{x - x_i}{\Delta x_i/2}\right), \qquad \int_{I_i}\phi_{ij}\phi_{il}\,dx = \begin{cases} 0 & j \neq l \\ \frac{\Delta x_i}{2j+1} & j = l \end{cases}$$

Chapter 8

Implementation in 1-D

The solution inside cell I_i is a polynomial of degree k and is of the form

$$u_h(x,t) = \sum_{l=0}^{k} u_{il}(t)\phi_{il}(x), \qquad x \in I_i$$

The semi-discrete DG scheme is given by equation (17.1) where we can take the test functions to be equal to the basis functions, leading to the system of equations

$$\int_{I_i} \frac{\partial u_h}{\partial t}\phi_{ij}\mathrm{d}x - \int_{I_i} f(u_h)\frac{\partial \phi_{ij}}{\partial x}\mathrm{d}x + \hat{f}_{i+\frac{1}{2}}(t)\phi_{ij}(x^-_{i+\frac{1}{2}}) - \hat{f}_{i-\frac{1}{2}}(t)\phi_{ij}(x^+_{i-\frac{1}{2}}) = 0, \qquad 0 \le j \le k$$

The DG scheme requires the computation of some integrals over the cell. In simple cases like linear advection equation, we can perform these integrations exactly and explicitly but in general, it is better to use some numerical integration techniques. These are reviewed in the Appendix (A.1).

8.1 ODE system

Plugging the solution u_h in the DG scheme, the first term is of the form

$$\int_{I_i} \frac{\partial u_h}{\partial t}\phi_{ij}\mathrm{d}x = \sum_{l=0}^{k} \frac{\mathrm{d}u_{il}}{\mathrm{d}t}\int_{I_i}\phi_{ij}\phi_{il}\mathrm{d}x = \sum_{l=0}^{k} M^{(i)}_{jl}\frac{\mathrm{d}u_{il}}{\mathrm{d}t}$$

The quantities $M^{(i)}_{jl}$ form the elements of the **mass matrix** $M^{(i)} \in \mathbb{R}^{(k+1)\times(k+1)}$. This is evaluated using a q-point quadrature rule as

$$M^{(i)}_{jl} = \int_{I_i}\phi_{ij}\phi_{il}\mathrm{d}x = \sum_q \omega_{iq}\phi_{ij}(x_{iq})\phi_{il}(x_{iq}), \quad 0 \le j,l \le k$$

where $\{x_{iq}\}$ are the quadrature nodes in element I_i and ω_{iq} are the corresponding weights. The remaining terms which we will put on the right hand side are computed as

$$\begin{aligned} L^{(i)}(U(t))_j &= \int_{I_i} f(u_h)\phi'_{ij}\mathrm{d}x - \hat{f}_{i+\frac{1}{2}}(t)\phi_{ij}(x^-_{i+\frac{1}{2}}) + \hat{f}_{i-\frac{1}{2}}(t)\phi_{ij}(x^+_{i-\frac{1}{2}}) \\ &\approx \sum_q \omega_{iq} f(u_h(x_{iq},t))\phi'_{ij}(x_{iq}) - \hat{f}_{i+\frac{1}{2}}(t)\phi_{ij}(x^-_{i+\frac{1}{2}}) \\ &\quad + \hat{f}_{i-\frac{1}{2}}(t)\phi_{ij}(x^+_{i-\frac{1}{2}}) \end{aligned}$$

Let us put the degrees of freedom associated with I_i in a vector

$$U^{(i)} = [u_{i,0}, \ldots, u_{i,k}]^\top \in \mathbb{R}^{k+1}$$

Then we obtain the following system of ODE for element I_i

$$M^{(i)} \frac{\mathrm{d}U^{(i)}(t)}{\mathrm{d}t} = L^{(i)}(U(t))$$

The mass matrix is obviously symmetric, and it is also positive definite. For any vector $U = [u_0, \ldots, u_k]^\top \in \mathbb{R}^{k+1}$

$$U^\top M^{(i)} U = \int_{I_i} |\sum_{j=0}^{k} u_j \phi_{ij}|^2 \mathrm{d}x > 0, \qquad U \neq 0$$

Hence we can invert the mass matrix and write the ODE as

$$\frac{\mathrm{d}U^{(i)}(t)}{\mathrm{d}t} = [M^{(i)}]^{-1} L^{(i)}(U(t)) = R^{(i)}(U(t))$$

We have one such system of ODE from each cell. These equations from different cells are coupled through the terms involving the numerical flux function.

8.2 Quadrature rules

The quadrature rule must be chosen such that mass matrix is evaluated exactly. Moreover, in order to achieve the optimal convergence rate, the integral inside the cell must be computed with a quadrature rule which is exact for polynomials of degree $2k$. These requirements are satisfied by using

- $(k+1)$-point Gauss-Legendre quadrature, which is exact for $\mathbb{P}_{2k+1}$
- $(k+2)$-point Gauss-Lobatto-Legendre quadrature, which is exact for $\mathbb{P}_{2k+1}$

The quadrature rules are explained in section (A.1). The elements of the mass matrix can be computed as

$$M_{jl}^{(i)} = \int_{I_i} \phi_{ij}\phi_{il} \mathrm{d}x = \sum_q \omega_{iq} \phi_{ij}(x_{iq}) \phi_{il}(x_{iq}), \quad 0 \leq j, l \leq k$$

where

$$\omega_{iq} = \frac{1}{2}\omega_q \Delta x_i, \qquad x_{iq} = \frac{1-\xi_q}{2} x_{i-\frac{1}{2}} + \frac{1+\xi_q}{2} x_{i+\frac{1}{2}}$$

and $\{\xi_q\}$ are the quadrature points in the reference element $[-1, +1]$.

8.3 Modal DG

If we use the orthogonal Legendre basis, then it is convenient to transform to a reference cell and then derive the equations. The solution has the form

$$u_h = \sum_{l=0}^{k} u_{il}\phi_{il}(x) = \sum_{l=0}^{k} u_{il}\hat{\varphi}_l(\xi), \qquad \hat{\varphi}_l(\xi) = \sqrt{2l+1} P_l(\xi), \qquad \xi = \frac{x - x_i}{\frac{1}{2}\Delta x_i}$$

Since

$$\mathrm{d}x = \frac{\Delta x_i}{2}\mathrm{d}\xi, \qquad \frac{\mathrm{d}\phi_{il}}{\mathrm{d}x} = \frac{\mathrm{d}\xi}{\mathrm{d}x}\frac{\mathrm{d}\hat{\varphi}_l}{\mathrm{d}\xi} = \frac{2}{\Delta x_i}\frac{\mathrm{d}\hat{\varphi}_l}{\mathrm{d}\xi}, \qquad \int_{I_i} f(u_h)\frac{\mathrm{d}\phi_{il}}{\mathrm{d}x}\mathrm{d}x = \int_{-1}^{+1} f(u_h)\frac{\mathrm{d}\hat{\varphi}_l}{\mathrm{d}\xi}\mathrm{d}\xi$$

the semi-discrete DG scheme takes the form

$$\begin{aligned}\frac{\Delta x_i}{2}\int_{-1}^{+1}\frac{\partial u_h}{\partial t}\hat{\varphi}_j\mathrm{d}\xi - \int_{-1}^{+1} f(u_h)\frac{\mathrm{d}\hat{\varphi}_j}{\mathrm{d}\xi}\mathrm{d}\xi \\ + \hat{f}_{i+\frac{1}{2}}\hat{\varphi}_j(+1) - \hat{f}_{i-\frac{1}{2}}\hat{\varphi}_j(-1) = 0, \qquad 0 \le j \le k\end{aligned}$$

The first term can be computed exactly as

$$\frac{\Delta x_i}{2}\int_{-1}^{+1}\frac{\partial u_h}{\partial t}\hat{\varphi}_j\mathrm{d}\xi = \frac{\Delta x_i}{2}\sum_{l=0}^{k}\frac{\mathrm{d}u_{il}}{\mathrm{d}t}\underbrace{\int_{-1}^{+1}\hat{\varphi}_l(\xi)\hat{\varphi}_j(\xi)\mathrm{d}\xi}_{2\delta_{jl}} = \Delta x_i\frac{\mathrm{d}u_{ij}}{\mathrm{d}t}$$

We get the system of ODE

$$\Delta x_i\frac{\mathrm{d}u_{ij}}{\mathrm{d}t} = L_j^{(i)}, \qquad 0 \le j \le k$$

The right hand side is computed using a quadrature rule with N_q points

$$\begin{aligned}L^{(i)}(U(t))_j &= \int_{-1}^{+1} f(u_h)\hat{\varphi}_j'\mathrm{d}\xi - \hat{f}_{i+\frac{1}{2}}\hat{\varphi}_j(+1) + \hat{f}_{i-\frac{1}{2}}\hat{\varphi}_j(-1) \\ &\approx \sum_{q=0}^{N_q-1}\omega_q f(u_h(\xi_q))\hat{\varphi}_j'(\xi_q) - \hat{f}_{i+\frac{1}{2}}\hat{\varphi}_j(+1) + \hat{f}_{i-\frac{1}{2}}\hat{\varphi}_j(-1)\end{aligned}$$

and if $f(u)$ is non-linear, then the quadrature is not exact. The last sum for the quadrature can be computed as a dot product of two vectors

$$\{f(u_h(\xi_q))\}_q \qquad \text{and} \qquad \{\omega_q\hat{\varphi}_j'(\xi_q)\}_q$$

or as a matrix-vector product if we compute all terms in one step

$$\begin{bmatrix}\int_{-1}^{+1} f(u_h)\hat{\varphi}_0'\mathrm{d}\xi \\ \vdots \\ \int_{-1}^{+1} f(u_h)\hat{\varphi}_k'\mathrm{d}\xi\end{bmatrix} = S\cdot\begin{bmatrix}f(u_h(\xi_0)) \\ \vdots \\ f(u_h(\xi_{N_q-1}))\end{bmatrix}$$

where the matrix S is defined as

$$S_{jq} = \omega_q\hat{\varphi}_j'(\xi_q), \quad 0 \le j \le k, \quad 0 \le q \le N_q - 1$$

The solution at the quadrature points are given by

$$u_h(\xi_q) = \sum_{j=0}^{k} u_{ij}\hat{\varphi}_j(\xi_q), \qquad 0 \le q \le N_q - 1$$

This can be written as a matrix vector product

$$\begin{bmatrix}u_h(\xi_0) \\ \vdots \\ u_h(\xi_{N_q-1})\end{bmatrix} = V\cdot\begin{bmatrix}u_{i,0} \\ \vdots \\ u_{i,k}\end{bmatrix}$$

where V is the Vandermonde matrix of size $N_q \times (k+1)$, whose elements are given by

$$V_{qj} = \hat{\varphi}_j(\xi_q), \quad 0 \le q \le N_q - 1, \quad 0 \le j \le k$$

Thus, all the computations involving basis functions and their quadrature can be performed on the reference cell which is common to all the cells. We can write the right hand side vector as

$$L^{(i)} = S[f]_q - f_{i+\frac{1}{2}} b^+ + f_{i-\frac{1}{2}} b^-$$

where $[f]_q$ is vector of flux values at the N_q quadrature points and the vectors $b^\pm$ are defined as

$$b_j^\pm = \hat{\varphi}_j(\pm 1), \qquad 0 \le j \le k$$

Numerical flux To compute the numerical flux $f_{i+\frac{1}{2}}$ we require the solution from the i'th and $(i+1)$'th cell evaluated at the face $x_{i+\frac{1}{2}}$. These are given by

$$u^-_{i+\frac{1}{2}} = \sum_{j=0}^{k} u_{ij} \hat{\varphi}_j(+1) = U^{(i)} \cdot b^+, \qquad u^+_{i+\frac{1}{2}} = \sum_{j=0}^{k} u_{i+1,j} \hat{\varphi}_j(-1) = U^{(i+1)} \cdot b^-$$

Then the numerical flux is obtained from some Riemann solver $f_{i+\frac{1}{2}} = \hat{f}(u^-_{i+\frac{1}{2}}, u^+_{i+\frac{1}{2}})$.

8.4 Setting the initial condition

Suppose the initial condition is

$$u(x,0) = u_0(x)$$

We have to set the initial condition $u_0(x)$ onto the finite element function $u_h(x)$.

- Modal basis: Do an L^2 projection, i.e.,

  $$\min \int_{I_i} (u_h(x,0) - u_0(x))^2 \mathrm{d}x \quad \text{wrt} \quad u_{i,0}, u_{i,1}, \ldots, u_{i,k}$$

  The first order optimality condition is

  $$\frac{\mathrm{d}}{\mathrm{d}u_{il}} \int_{I_i} (u_h(x,0) - u_0(x))^2 \mathrm{d}x = 0, \qquad l = 0, 1, 2, \ldots, k$$

  $$\implies \int_{I_i} u_h \phi_{il} \mathrm{d}x = \int_{I_i} u_0 \phi_{il} \mathrm{d}x, \qquad l = 0, 1, 2, \ldots, k$$

  Using the orthogonality of the basis functions, and a quadrature rule for the right hand side, we get

  $$u_{il} \Delta x_i = \sum_q u_0(x_{iq}) \phi_{il}(x_{iq}) \omega_{iq}$$

  which determines the dofs of the initial solution. This can also be written on reference cell as

  $$u_{il} = \frac{1}{2} \sum_q u_0(x_{iq}) \hat{\varphi}_l(\xi_q) \omega_q, \qquad l = 0, 1, \ldots, k$$

- Nodal basis: We can interpolate initial condition

  $$u_{ij}(0) = u_h(x_{ij}, 0) = u_0(x_{ij})$$

  Alternately, we can also set the initial condition by performing an L^2 projection.

8.5 Boundary condition

We can specify Dirichlet boundary condition at some boundary point if the characteristics are entering the domain (inflow boundary) at that point. For example, at $x = 0$ if $f' > 0$, then we can specify the boundary condition on u. In general let us take the boundary conditions

$$u(0,t) = a(t), \qquad u(1,t) = b(t)$$

The boundary conditions are incorporated in the DG scheme via the boundary fluxes

$$\hat{f}_{\frac{1}{2}}(t) = \hat{f}(a(t), u_h(0^+,t)), \qquad \hat{f}_{N+\frac{1}{2}}(t) = \hat{f}(u_h(1^-,t), b(t))$$

In case of periodic boundaries, the fluxes at the boundary are computed as

$$\hat{f}_{\frac{1}{2}}(t) = \hat{f}(u_h(x^-_{N+\frac{1}{2}},t), u_h(x^+_{\frac{1}{2}},t)), \qquad \hat{f}_{N+\frac{1}{2}}(t) = \hat{f}(u_h(x^-_{N+\frac{1}{2}},t), u_h(x^+_{\frac{1}{2}},t))$$

8.6 Strong stability preserving RK schemes

Consider an ODE of the form

$$\frac{\mathrm{d}U}{\mathrm{d}t} = R(U,t)$$

A standard method to solve ODEs is the Runge-Kutta method. We will consider the strong stability preserving RK schemes, examples of which are given below.

- 1-stage, 1-st order RK (forward Euler)

$$U^{n+1} = U^n + \Delta t\ R(U^n, t_n)$$

- 2-stage, 2-nd order RK

$$\begin{aligned} U^{(1)} &= U^n + \Delta t\ R(U^n, t_n) \\ U^{n+1} &= \frac{1}{2}U^n + \frac{1}{2}[U^{(1)} + \Delta t\ R(U^{(1)}, t_n + \Delta t)] \end{aligned}$$

- 3-stage, 3-rd order RK

$$\begin{aligned} U^{(1)} &= U^n + \Delta t\ R(U^n, t_n) \\ U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}[U^{(1)} + \Delta t\ R(U^{(1)}, t_n + \Delta t)] \\ U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}[U^{(2)} + \Delta t\ R(U^{(2)}, t_n + \tfrac{1}{2}\Delta t)] \end{aligned}$$

- 4-stage, 3-rd order RK

$$\begin{aligned} U^{(1)} &= U^n + \frac{\Delta t}{2}R(U^n, t_n) \\ U^{(2)} &= U^{(1)} + \frac{\Delta t}{2}R(U^{(1)}, t_n + \tfrac{1}{2}\Delta t) \\ U^{(3)} &= \frac{2}{3}U^n + \frac{1}{3}[U^{(2)} + \frac{\Delta t}{2}R(U^{(2)}, t_n + \Delta t)] \\ U^{n+1} &= U^{(3)} + \frac{\Delta t}{2}R(U^{(3)}, t_n + \tfrac{1}{2}\Delta t) \end{aligned}$$

- 5-stage, 4-th order RK

$$\begin{aligned} U^{(1)} &= U^n + c_{11}\Delta t R(U^n) \\ U^{(2)} &= c_{21}U^n + c_{22}U^{(1)} + c_{23}\Delta t R(U^{(1)}) \\ U^{(3)} &= c_{31}U^n + c_{32}U^{(2)} + c_{33}\Delta t R(U^{(2)}) \\ U^{(4)} &= c_{41}U^n + c_{42}U^{(3)} + c_{43}\Delta t R(U^{(3)}) \\ U^{n+1} &= c_{51}U^{(2)} + c_{52}U^{(3)} + c_{53}U^{(4)} + \Delta t[c_{54}R(U^{(3)}) + c_{55}R(U^{(4)})] \end{aligned}$$

where the constants are

$$c_{11} = 0.391752226571890$$

$$c_{21} = 0.444370493651235, \quad c_{22} = 1.0 - c21, \quad c_{23} = 0.368410593050371$$

$$c_{31} = 0.620101851488403, \quad c_{32} = 1.0 - c31, \quad c_{33} = 0.251891774271694$$

$$c_{41} = 0.178079954393132, \quad c_{42} = 1.0 - c41, \quad c_{43} = 0.544974750228521$$

$$c_{51} = 0.517231671970585, \quad c_{52} = 0.096059710526147, \quad c_{53} = 1.0 - (c51 + c52)$$

$$c_{54} = 0.063692468666290, \quad c_{55} = 0.226007483236906$$

Remark 8.1. *The SSP property requires positivity of all the coefficients. There is no 4-stage, 4-th order SSPRK scheme with positive coefficients. We need atleast 5-stages for the fourth order SSPRK [16], [13] to have all positive coefficients.*

Lemma 8.1 (SSP Runge-Kutta)**.** *If the forward Euler scheme is stable in some norm and under some time step condition, i.e.,*

$$\Delta t \leq \Delta t_{FE}(U) \qquad \Longrightarrow \qquad \|U + \Delta t\ R(U)\| \leq \|U\|$$

then the SSPRK scheme is stable under a CFL condition $\Delta t \leq \alpha \Delta t_{FE}$. For the second and third order schemes, $\alpha = 1$. The 4-stage, third order method has $\alpha = 2$. The 5-stage, 4-th order method has $\alpha = 1.508$.

Proof: Let us show this for the 2-stage scheme. Since the forward Euler scheme is stable if $\Delta t \leq \Delta t_{FE}$, and as first stage resembles a forward Euler scheme, we get

$$\left\|U^{(1)}\right\| = \|U^n + \Delta t\ R(U^n)\| \leq \|U^n\|$$

For the second stage, we apply triangle inequality

$$\left\|U^{n+1}\right\| \leq \frac{1}{2}\|U^n\| + \frac{1}{2}\left\|U^{(1)} + \Delta t\ R(U^{(1)})\right\| \leq \frac{1}{2}\|U^n\| + \frac{1}{2}\left\|U^{(1)}\right\| \leq \|U^n\|$$

since the second term has the form of a forward Euler scheme. □

Remark 8.2. *The allowed time step Δt_{FE} depends on the current solution. The above proof is correct provided*

$$\Delta t \leq \min\{\Delta t_{FE}(U^n), \Delta t_{FE}(U^{(1)})\}$$

But we do not know $U^{(1)}$ at the beginning of the time step. In practice, we can put a margin of safety and choose the time step as $\Delta t = 0.95\Delta t_{FE}(U^n)$ for example, which usually works in practice. If it turns out after the first stage, that the Δt we have chosen is larger than $\Delta t_{FE}(U^{(1)})$, then we can reduce it further and restart the time step from the first stage.

8.7 Storage requirements

The above SSPRK scheme require storage for three vectors,

- U^n
- current stage solution $U^{(s)}$
- Residual R

We anyway need to store U^n, U^{n+1} and we can use U^{n+1} to also store the intermediate stage solutions. Here is a pseudo algorithm to perform one time step of the 2-stage, 2-nd order method.

```
uold = u
// First stage
R = residual(u, t)
u = uold + dt * R
// Second stage
R = residual(u, t+dt)
u = 0.5*uold + 0.5*(u + dt * R)
```

This shows that we need to store the vectors `uold, u, R`. The 5-stage, 4-th order SSPRK scheme requires more storage.

8.8 Classical RK

The classical 4-stage, 4-th order RK scheme is not SSP. It is given by

$$\begin{aligned} k_1 &= R(U^n, t_n) \\ k_2 &= R(U^n + \tfrac{1}{2}\Delta t\ k_1, t_n + \tfrac{1}{2}\Delta t) \\ k_3 &= R(U^n + \tfrac{1}{2}\Delta t\ k_2, t_n + \tfrac{1}{2}\Delta t) \\ k_4 &= R(U^n + \Delta t\ k_3, t_n + \Delta t) \\ U^{n+1} &= U^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

A pseudo algorithm is of the following form.

```
uold = u
k1   = residual(u,t)
u    = uold + 0.5*dt*k1
k2   = residual(u,t+0.5*dt)
u    = uold + 0.5*dt*k2
k3   = residual(u,t+0.5*dt)
u    = uold + dt*k3
k4   = residual(u,t+dt)
u    = uold + (dt/6)*(k1 + 2*k2 + 2*k3 + k4)
```

The above algorithm requires storage for six vectors: `uold, u, k1, k2, k3, k4`. We can reduce the storage to four vectors by rewriting the algorithm in the following way.

```
uold = u
utmp = u
k    = residual(u,t)
u    = uold + 0.5*dt*k
utmp+= (dt/6)*k
k    = residual(u,t+0.5*dt)
```

Table 2.3. CFL Numbers for RKDG Methods of Order $k+1$

k	0	1	2
CFL_{TV}	1	1/2	1/2
CFL_{L^2}	1	1/3	1/5

```
u    = uold + 0.5*dt*k
utmp+= (dt/3)*k
k    = residual(u,t+0.5*dt)
u    = uold + dt*k
utmp+= (dt/3)*k
k    = residual(u,t+dt)
u    = utmp + (dt/6)*k
```

This requires storage for the vectors: `uold, u, utmp, k`.

8.9 CFL condition

Explicit time integration schemes are stable only under a restriction on the time step Δt. For $k = 0$, the DG scheme with an upwind flux is identical to upwind finite volume scheme which is L^2 stable under CFL number of one. For $k \geq 1$, the forward Euler scheme (RK1) is known to be unconditionally unstable in L^2 (Chavent and Cockburn, 1989) if the CFL number is of order unity. It is L^2 stable for finite time intervals if the CFL number is $\mathcal{O}\left(h^{1/2}\right)$ which means that $\Delta t = \mathcal{O}\left(h^{3/2}\right)$, which is very restrictive. For DG space discretizations using polynomials of degree k, and a $(k+1)$-stage RK method of order $k+1$, a Von-Neumann stability analysis for the one-dimensional linear case

$$f(u) = cu$$

with upwind flux gives the CFL condition

$$|c|\frac{\Delta t}{\Delta x} \leq \frac{1}{2k+1}$$

Theoretical proof of this is available only for $k = 0, 1, 2$. For $k \geq 3$ the above condition is close to the numerically determined values of CFL numbers. In a later chapter we will perform Fourier stability analysis to obtain the precise CFL numbers for different combination of polynomial degree and RK scheme.

Remark 8.3. *The CFL number for SSPRK scheme to be TVDM is higher than that required for L^2 stability. However, to control round-off errors, the smaller CFL condition from L^2 stability has to be used in practical computations.*

8.10 Algorithm

Let us now summarize the main steps in the DG scheme.

- Compute and store the mass matrix if needed.
- Determine u_h^0 from initial condition u_0 by an L^2-projection or interpolation
- Set time counter $t = 0$
- For $n = 0, 1, \ldots$
  - Compute time step Δt from CFL condition
  - Set $u_h^{n,0} = u_h^n$
  - RK stages: For $r = 0, 1, \ldots, N_{rk} - 1$
    - Compute right hand side $L_h(u_h^{n,r})$
    - Update solution to next RK stage

$$u_h^{n,r} \to u_h^{n,r+1}$$

  - Increment time counter $t = t + \Delta t$

8.11 Numerical example

Let us apply the DG scheme to linear and non-linear problems with smooth and discontinuous initial conditions. A set of Python codes are available at

http://github.com/cpraveen/fembook/tree/master/dg1d/scalar

You can see the available command line arguments

```
$ python3 ./dg.py -h
```

This code uses the three stage, third order SSPRK scheme. The time step is chosen as

$$\Delta t = \text{CFL} \frac{\Delta x}{(2k+1) \max |f'(u)|}$$

Let us try some examples using the Python code.

1. Linear advection: smooth solution

```
$ python3 ./dg.py -pde linear -ic sin2pi -ncell 50 -degree 1
$ python3 ./dg.py -pde linear -ic sin2pi -ncell 10 -degree 3
```

2. Linear advection: smooth solution

```
$ python3 ./dg.py -pde linear -ic gauss -ncell 10 -degree 3
$ python3 ./dg.py -pde linear -ic gauss -ncell 20 -degree 3
```

3. Linear advection with varying speed, smooth solution: This corresponds to Example 5.3 in [8].

```
$ python3 ./dg.py -pde varadv -ic sin4pi -Tf 10.5 -ncell 5 \
                  -degree 16 -cfl 0.5 -plot_freq 10
```

   We have to reduce the CFL since we use a third order SSPRK scheme. Compare the solution at final time with Fig. (5.1) in [8].

4. Linear advection, discontinuous solution

```
$ python3 ./dg.py -pde linear -ic hat -degree 1 -ncell 52
$ python3 ./dg.py -pde linear -ic hat -degree 2 -ncell 52
$ python3 ./dg.py -pde linear -ic hat -degree 2 -ncell 100 \
                  -plot_freq 10
```

5. Burgers equation

```
$ python3 ./dg.py -pde burger -ic sin2pi -degree 1 -ncell 50
```

Change the initial condition to $1 + \sin(2\pi x)$ and try above scheme again.

```
$ python3 ./dg.py -pde burger -ic sin2pi -degree 2 -ncell 50
$ python3 ./dg.py -pde burger -ic sin2pi -degree 2 -ncell 100 \
                  -plot_freq 10
```

Codes based on the `deal.II` library are available here

http://bitbucket.org/cpraveen/deal_ii/src/master/dg/1d_scalar_legendre
http://bitbucket.org/cpraveen/deal_ii/src/master/dg/1d_burger_legendre

- Linear convection equation: smooth initial condition with periodic boundary conditions

$$u(x,0) = \sin(\pi x), \qquad x \in [-1,+1]$$

Running the code will simulate this problem. We can perform grid convergence study by setting

```
param.nstep        = 5;
param.output_step  = 10;
```

Run the code and generate error.pdf file from error.tex file.

- Linear convection equation: continuous initial condition with periodic boundary conditions

$$u(x,0) = \begin{cases} 1+2x, & -\frac{1}{2} \le x \le 0 \\ 1-2x, & 0 \le x \le \frac{1}{2} \\ 0, & \text{otherwise} \end{cases}, \qquad x \in [-1,+1]$$

- Linear convection equation: discontinuous initial condition with periodic boundary conditions

$$u(x,0) = \begin{cases} 1, & |x| < \frac{1}{4} \\ 0, & \text{otherwise} \end{cases}, \qquad x \in [-1,+1]$$

- Burgers equation: smooth initial condition

$$u(x,0) = \sin(2\pi x), \qquad x \in [0,1]$$

8.12 Nodal DG scheme

When using nodal basis functions, certain simplifications can be made which improve the computational efficiency. The solution in the cell I_i is represented as

$$u_h = \sum_{l=0}^{k} u_{il}\phi_{il}(x) = \sum_{l=0}^{k} u_{il}\ell_l(\xi)$$

where $u_{il} = u_h(\xi_l)$ are solution values at certain $k+1$ distinct nodes inside the cell which will be taken from a quadrature rule. The DG scheme can be written as

$$\int_{I_i} \frac{\partial u_h}{\partial t}\phi_{ij}\mathrm{d}x - \int_{I_i} f(u_h)\frac{\partial \phi_{ij}}{\partial x}\mathrm{d}x + \hat{f}_{i+\frac{1}{2}}(t)\phi_{ij}(x^-_{i+\frac{1}{2}}) - \hat{f}_{i-\frac{1}{2}}(t)\phi_{ij}(x^+_{i-\frac{1}{2}}) = 0, \qquad 0 \le j \le k$$

We will now approximate the flux with the same basis functions

$$f_h(x) = \sum_{l=0}^{k} f(u_{il})\phi_{il}(x) = \sum_{l=0}^{k} f(u_{il})\ell_l(\xi)$$

i.e., we are interpolating the flux at the solution points. The actual computations are best performed on a reference cell.

8.12.1 Weak form DG

Using the projected flux in the weak formulation and transforming to reference cell, we get

$$\frac{\Delta x_i}{2}\int_{-1}^{+1} \frac{\partial u_h}{\partial t}\ell_j \mathrm{d}\xi - \int_{-1}^{+1} f_h(\xi)\frac{\partial \ell_j}{\partial \xi}\mathrm{d}\xi + \hat{f}_{i+\frac{1}{2}}(t)\ell_j(+1) - \hat{f}_{i-\frac{1}{2}}(t)\ell_j(-1) = 0, \qquad 0 \le j \le k$$

We will now use the nodes used to define the solution to perform the integrals.

$$\frac{\Delta x_i}{2}\sum_{q=0}^{k} \frac{\partial u_h}{\partial t}(\xi_q)\ell_j(\xi_q)\omega_q - \sum_{q=0}^{k} f_h(\xi_q)\frac{\partial \ell_j}{\partial \xi}(\xi_q)\omega_q + \hat{f}_{i+\frac{1}{2}}(t)\ell_j(+1) - \hat{f}_{i-\frac{1}{2}}(t)\ell_j(-1) = 0$$

Since $\ell_j(\xi_q) = \delta_{jq}$ and

$$f_h(\xi_q) = \sum_{l=0}^{k} f(u_{il})\ell_l(\xi_q) = f(u_{iq})$$

the DG scheme reduces to

$$\frac{\Delta x_i}{2}\omega_j \frac{\mathrm{d}u_{ij}}{\mathrm{d}t} - \sum_{q=0}^{k} f(u_{iq})\frac{\mathrm{d}\ell_j}{\mathrm{d}\xi}(\xi_q)\omega_q + \hat{f}_{i+\frac{1}{2}}(t)\ell_j(+1) - \hat{f}_{i-\frac{1}{2}}(t)\ell_j(-1) = 0$$

This can be written in matrix-vector form as

$$\frac{\Delta x_i}{2} M \frac{\mathrm{d}U^{(i)}}{\mathrm{d}t} - D^\top M^\top F^{(i)} + \hat{f}_{i+\frac{1}{2}} b^+ - \hat{f}_{i-\frac{1}{2}} b^- = 0$$

where M is diagonal matrix containing the quadrature weigths, D is differentiation matrix with components

$$D_{qj} = \frac{\mathrm{d}\ell_j}{\mathrm{d}\xi}(\xi_q), \qquad 0 \le q, j \le k$$

and the vectors $b^\pm$ are given by

$$b^\pm_j = \ell_j(\pm 1), \qquad 0 \le j \le k$$

For actual computer implementation, we can define the stiffness matrix $S = D^\top M^\top$ whose elements are given by

$$S_{jq} = \frac{\mathrm{d}\ell_j}{\mathrm{d}\xi}(\xi_q)\omega_q, \qquad 0 \le j, q \le k$$

8.12.2 Strong form DG

Performing another integration by parts in the second term we get

$$\int_{I_i} \frac{\partial u_h}{\partial t}\phi_{ij}\mathrm{d}x + \int_{I_i} \frac{\partial f(u_h)}{\partial x}\phi_{ij}\mathrm{d}x + [\hat{f}_{i+\frac{1}{2}}(t) - f(u_h(x^-_{i+\frac{1}{2}}))]\phi_{ij}(x^-_{i+\frac{1}{2}}) - [\hat{f}_{i-\frac{1}{2}}(t) - f(u_h(x^+_{i-\frac{1}{2}}))]\phi_{ij}(x^+_{i-\frac{1}{2}}) = 0, \qquad 0 \le j \le k$$

and using the projected flux approximation in the DG scheme to get

$$\frac{\Delta x_i}{2}\int_{-1}^{+1} \frac{\partial u_h}{\partial t}\ell_j\mathrm{d}\xi + \int_{-1}^{+1} \frac{\partial f_h}{\partial \xi}\ell_j\mathrm{d}\xi + [\hat{f}_{i+\frac{1}{2}}(t) - f_h(+1)]\ell_j(+1) - [\hat{f}_{i-\frac{1}{2}}(t) - f_h(-1)]\ell_j(-1) = 0, \qquad 0 \le j \le k$$

We will also approximate the two cell integrals using the solution nodes as the quadrature nodes. Then we get

$$\frac{\Delta x_i}{2}\omega_j\frac{\mathrm{d}u_{ij}}{\mathrm{d}t} + \omega_j\frac{\partial f_h}{\partial \xi}(\xi_j) + [\hat{f}_{i+\frac{1}{2}}(t) - f_h(+1)]\ell_j(+1) - [\hat{f}_{i-\frac{1}{2}}(t) - f_h(-1)]\ell_j(-1) = 0, \qquad 0 \le j \le k$$

This looks like a collocation scheme with some coupling terms coming from the numerical fluxes at the cell faces. The derivative of the flux can be computed as a matrix-vector product

$$\left\{\frac{\partial f_h}{\partial \xi}(\xi_q)\right\}_q = D \cdot \{f(u_{iq})\}_q$$

where the differentiation matrix D is of size $(k+1)\times(k+1)$ and given by

$$D_{ql} = \frac{\mathrm{d}\ell_l}{\mathrm{d}\xi}(\xi_q), \qquad 0 \le q, l \le k$$

We can write the scheme in compact form

$$\frac{\Delta x_i}{2}M\frac{\mathrm{d}U^{(i)}}{\mathrm{d}t} + MDF^{(i)} - [\hat{f}_{i-\frac{1}{2}}(t) - f_h(-1)]b^- + [\hat{f}_{i+\frac{1}{2}}(t) - f_h(+1)]b^+ = 0$$

where M is diagonal matrix with quadrature weights on the diagonal and the vectors $b^\pm$ are given by

$$b^\pm_j = \ell_j(\pm 1), \qquad 0 \le j \le k$$

Gauss-Lobatto nodes In this case $\xi_0 = -1$ and $\xi_k = +1$ so that

$$\ell_0(-1) = 1, \quad \ell_j(-1) = 0, \quad 1 \le j \le k$$

$$\ell_k(+1) = 1, \quad \ell_j(+1) = 0, \quad 0 \le j \le k-1$$

and the set of equations becomes

$$\begin{aligned}
&\frac{\Delta x_i}{2}\frac{\mathrm{d}u_{i0}}{\mathrm{d}t} + \frac{\partial f_h}{\partial \xi}(-1) - \frac{1}{\omega_0}[\hat{f}_{i-\frac{1}{2}}(t) - f_h(-1)] = 0 \\
&\frac{\Delta x_i}{2}\frac{\mathrm{d}u_{ij}}{\mathrm{d}t} + \frac{\partial f_h}{\partial \xi}(\xi_j) \qquad\qquad\qquad\qquad = 0, \quad 1 \le j \le k-1 \\
&\frac{\Delta x_i}{2}\frac{\mathrm{d}u_{ik}}{\mathrm{d}t} + \frac{\partial f_h}{\partial \xi}(+1) + \frac{1}{\omega_k}[\hat{f}_{i+\frac{1}{2}}(t) - f_h(+1)] = 0
\end{aligned}$$

The coupling terms are present only in the first and last equations which correspond to solution nodes located at the boundaries of the cell.

Remark 8.4. *Since we used the same nodes for quadrature as are used to define the solution, we directly have access to the solution values at the quadrature points. This was not the case when we used orthogonal polynomials; there we have to evaluate the solution at the quadrature points by computing the sum over all basis functions. This adds to the computational cost expecially when we go to three dimensional. Some optimizations can be performed to evaluate the sums more efficiently using factorization techniques.*

Remark 8.5. *Because of the choice of the same nodes to represent the solution and to perform the quadrature, the quadrature sums become simplified and essentially dissapear from the scheme. For this reason, such schemes are also said to be* quadrature-free.

Remark 8.6. *The efficient computation of Lagrange polynomials and their derivatives is explained in the Appendix.*

8.12.3 Numerical fluxes

The numerical flux $\hat{f}_{i+\frac{1}{2}}$ is computed using your favourite Riemann solver. One approach is to first evaluate the two states on either side of the face and pass them to the Riemann solver to compute the flux

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}(u_h(x^-_{i+\frac{1}{2}}), u_h(x^+_{i+\frac{1}{2}}))$$

Note that the state $u_h(x^-_{i+\frac{1}{2}})$ is obtained using the polynomial in cell I_i and $u_h(x^+_{i+\frac{1}{2}})$ is obtained using the polynomial in cell I_{i+1}

$$u_h(x^-_{i+\frac{1}{2}}) = \sum_{j=0}^{k} u_{i,j}\ell_j(+1) = b^+ \cdot U^{(i)}, \qquad u_h(x^+_{i+\frac{1}{2}}) = \sum_{j=0}^{k} u_{i+1,j}\ell_j(-1) = b^- \cdot U^{(i+1)}$$

If we use GLL nodes, then this simplifies to

$$u_h(x^-_{i+\frac{1}{2}}) = u_{i,k}, \qquad u_h(x^+_{i+\frac{1}{2}}) = u_{i+1,0}$$

Chapter 9

Limiters and TVD property

When the solution is discontinuous or has large gradients, the higher order DG scheme produces oscillatory solution. This situation is similar to high order finite volume schemes and is related to loss of TVD property. The oscillatory numerical solution has more total variation than the initial condition. In the case of finite volume schemes, this problem is resolved by reducing the slope of the reconstructed solution by appropriate limiter functions so that the scheme becomes TVD. In DG schemes we do not have to perform any reconstruction since we have a polynomial inside each cell. But we can borrow the limiter idea and reduce the slope of the solution in each cell to achieve TVD property. The approach we will take is to construct a limiter so that the DG scheme with forward Euler discretization is TVD. The use of an SSPRK scheme then automatically gives TVD property for higher order versions of the DG scheme.

9.1 Limiter for DG scheme

Consider the forward Euler time discretization, i.e., find w_h^{n+1} such that

$$\int_{I_i} \frac{w_h^{n+1} - u_h^n}{\Delta t} v_h \mathrm{d}x - \int_{I_i} f(u_h^n) \frac{\partial v_h}{\partial x} \mathrm{d}x + \hat{f}^n_{i+\frac{1}{2}} v_h(x^-_{i+\frac{1}{2}}) - \hat{f}^n_{i-\frac{1}{2}} v_h(x^+_{i-\frac{1}{2}}) = 0$$

The solution w_h^{n+1} may be oscillatory. We will treat this as a provisional solution and limit it to obtain the solution at the next time step

$$u_h^{n+1} = \Lambda\Pi_h(w_h^{n+1})$$

We require that the limiter $\Lambda\Pi_h()$ satisfy some basic properties as follows.
Properties of $\Lambda\Pi_h(\cdot)$

1. It should not change the cell average value.
2. It should not affect the accuracy in smooth regions.

Define the **cell average value**

$$\bar{u}_i = \frac{1}{\Delta x_i} \int_{I_i} u_h \mathrm{d}x$$

the forward and backward differences

$$\hat{u}_i = u_h(x^-_{i+\frac{1}{2}}) - \bar{u}_i, \qquad \check{u}_i = \bar{u}_i - u_h(x^+_{i-\frac{1}{2}})$$

$$\Delta_+ \bar{u}_i = \bar{u}_{i+1} - \bar{u}_i, \qquad \Delta_- \bar{u}_i = \bar{u}_i - \bar{u}_{i-1}$$

For $k = 1$ we have $\hat{u}_i = \check{u}_i$. We cannot modify the cell average value but we can modify the slopes

$$\hat{u}_i^{(m)} = m(\hat{u}_i, \Delta_+ \bar{u}_i, \Delta_- \bar{u}_i), \qquad \check{u}_i^{(m)} = m(\check{u}_i, \Delta_+ \bar{u}_i, \Delta_- \bar{u}_i)$$

where m is the minmod function

$$m(a_1, \ldots, a_l) = \begin{cases} s \min(|a_1|, \ldots, |a_l|) & s = \text{sign}(a_1) = \ldots = \text{sign}(a_l) \\ 0 & \text{otherwise} \end{cases}$$

If all arguments have same sign, the minmod function returns the one with smallest magnitude, otherwise it returns zero. The trace values are recomputed using the limited slopes

$$u_h^{(m)}(x^-_{i+\frac{1}{2}}) = \bar{u}_i + \hat{u}_i^{(m)}, \qquad u_h^{(m)}(x^+_{i-\frac{1}{2}}) = \bar{u}_i - \check{u}_i^{(m)}$$

For $k = 1$, we have $\hat{u}_i^{(m)} = \check{u}_i^{(m)}$. The effect of the limiter is to reduce the slope of the

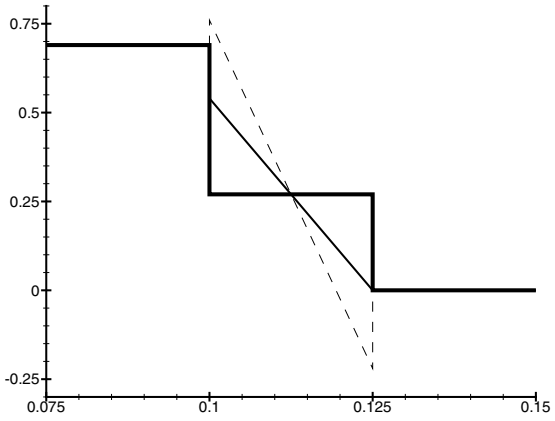


Figure 9.1: Effect of applying TVD limiter is to reduce the slope of the solution so that it is less oscillatory

solution in case it is larger than the finite difference slope of the cell average values as shown in figure (9.1). If the cell averages are monotone around cell i, then the limited linear polynomial satisfies

$$\min\{\bar{u}_{i-1}, \bar{u}_{i+1}\} \leq u_h^{(m)}(x) \leq \max\{\bar{u}_{i-1}, \bar{u}_{i+1}\}, \qquad x \in I_i$$

For $k = 0, 1, 2$, this procedure uniquely determines a new polynomial of degree k. For $k = 1$, let us denote the limited function by $\Lambda\Pi_h^1(u_h)$. For $k \geq 3$ there is more freedom since the cell average and the two trace values do not completely determine the polynomial. One approach is to determine the remaining dofs by an L^2 projection. A more simple approach that works well in practice is the following.

1. If the limiter does not modify the trace values, i.e.,

$$u_h^{(m)}(x^+_{i-\frac{1}{2}}) = u_h(x^+_{i-\frac{1}{2}}) \qquad \text{and} \qquad u_h^{(m)}(x^-_{i+\frac{1}{2}}) = u_h(x^-_{i+\frac{1}{2}})$$

   then take $u_h^{(m)} = u_h$ for $x \in I_i$.

2. Otherwise, let $u_h^1 \in \mathbb{P}_1(I_i)$ be the L^2 projection of $u_h|_{I_i}$. Take $u_h^{(m)}|_{I_i} = \Lambda\Pi_h^1(u_h^1)$.

If the DG solution has been processed by the above limiter, then we can prove a TVD property. Let us first recall Harten's sufficient conditions for a finite volume scheme to be total variation diminishing.

Lemma 9.1. *(Harten) If a scheme can be written as*

$$\bar{u}_i^{n+1} = \bar{u}_i^n + C_{i+\frac{1}{2}}^n \Delta_+ \bar{u}_i^n - D_{i-\frac{1}{2}}^n \Delta_- \bar{u}_i^n$$

Assume that boundary conditions are periodic or compactly supported. If

$$C_{i+\frac{1}{2}} \geq 0, \qquad D_{i+\frac{1}{2}} \geq 0, \qquad C_{i+\frac{1}{2}} + D_{i+\frac{1}{2}} \leq 1$$

then the scheme is TVD

$$TV(u_h^{n+1}) \leq TV(u_h^n)$$

where the total variation is defined as

$$TV(u_h) = \sum_i |\Delta_+ \bar{u}_i|$$

Proof: Take the update equation at i and $i+1$

$$\begin{aligned} \bar{u}_{i+1}^{n+1} &= \bar{u}_{i+1}^n + C_{i+\frac{3}{2}}^n \Delta_+ \bar{u}_{i+1}^n - D_{i+\frac{1}{2}}^n \Delta_- \bar{u}_{i+1}^n \\ \bar{u}_i^{n+1} &= \bar{u}_i^n + C_{i+\frac{1}{2}}^n \Delta_+ \bar{u}_i^n - D_{i-\frac{1}{2}}^n \Delta_- \bar{u}_i^n \end{aligned}$$

and subtracting them yields

$$\begin{aligned} \Delta_+ \bar{u}_i^{n+1} &= \Delta_+ \bar{u}_i^n + C_{i+\frac{3}{2}}^n \Delta_+ \bar{u}_{i+1}^n - D_{i+\frac{1}{2}}^n \Delta_- \bar{u}_{i+1}^n - C_{i+\frac{1}{2}}^n \Delta_+ \bar{u}_i^n + D_{i-\frac{1}{2}}^n \Delta_- \bar{u}_i^n \\ &= \Delta_+ \bar{u}_i^n + C_{i+\frac{3}{2}}^n \Delta_+ \bar{u}_{i+1}^n - D_{i+\frac{1}{2}}^n \Delta_+ \bar{u}_i^n - C_{i+\frac{1}{2}}^n \Delta_+ \bar{u}_i^n + D_{i-\frac{1}{2}}^n \Delta_+ \bar{u}_{i-1}^n \\ &= (1 - C_{i+\frac{1}{2}} - D_{i+\frac{1}{2}}) \Delta_+ \bar{u}_i^n + C_{i+\frac{3}{2}}^n \Delta_+ \bar{u}_{i+1}^n + D_{i-\frac{1}{2}}^n \Delta_+ \bar{u}_{i-1}^n \end{aligned}$$

Applying triangle inequality and noting that all the coefficients are positive according to our assumption

$$|\Delta_+ \bar{u}_i^{n+1}| \leq (1 - C_{i+\frac{1}{2}} - D_{i+\frac{1}{2}}) |\Delta_+ \bar{u}_i^n| + C_{i+\frac{3}{2}}^n |\Delta_+ \bar{u}_{i+1}^n| + D_{i-\frac{1}{2}}^n |\Delta_+ \bar{u}_{i-1}^n|$$

Summing over all cells, all the terms on the right cancel except the first one, yielding

$$\sum_i |\Delta_+ \bar{u}_i^{n+1}| \leq \sum_i |\Delta_+ \bar{u}_i^n|$$

which proves the lemma. □

Remark 9.1. *Consider the first order upwind finite volume scheme for $u_t + a u_x = 0$ which is identical to the DG scheme for degree $k = 0$. This scheme can be written in the incremental form with*

$$C_{i+\frac{1}{2}} = \frac{\Delta t}{\Delta x} \frac{|a| - a}{2}, \qquad D_{i+\frac{1}{2}} = \frac{\Delta t}{\Delta x} \frac{|a| + a}{2}$$

The coefficients and the condition

$$C_{i+\frac{1}{2}} + D_{i+\frac{1}{2}} = \frac{|a| \Delta t}{\Delta x} \leq 1$$

leads to the CFL condition.

Definition 9.1 (TVM)**.** *For DG solution u_h, let us define the* **total variation of the means** *by*

$$TVM(u_h) = \sum_i |\Delta_+ \bar{u}_i|$$

We will show that the TVM does not increase with time if the limiter is applied. For this we make use of the monotone flux property. Since $\hat{f}(u,v)$ is increasing function of u and decreasing function of v, we have

$$\frac{\hat{f}(a,b) - \hat{f}(c,b)}{a-c} \geq 0, \qquad \frac{\hat{f}(a,b) - \hat{f}(a,c)}{b-c} \leq 0$$

We can bound the above ratios as follows. Since

$$\frac{\hat{f}(a,b) - \hat{f}(c,b)}{a-c} = \frac{\partial}{\partial u}\hat{f}(\xi,b), \qquad \xi \in I(a,c)$$

we have

$$\frac{\hat{f}(a,b) - \hat{f}(c,b)}{a-c} \leq \max_{\xi \in I(b,c)} \frac{\partial}{\partial u}\hat{f}(\xi,b) =: L_1(b)$$

Similarly

$$-\frac{\hat{f}(a,b) - \hat{f}(a,c)}{b-c} = -\frac{\partial}{\partial v}\hat{f}(a,\eta), \qquad \eta \in I(b,c)$$

so that

$$-\frac{\hat{f}(a,b) - \hat{f}(a,c)}{b-c} \leq \max_{\eta \in I(b,c)} \left| \frac{\partial}{\partial v}\hat{f}(a,\eta) \right| =: L_2(a)$$

Theorem 9.1. *For periodic or compactly supported boundary conditions, the DG scheme with the minmod limiter is TVD in the means, i.e.,*

$$TVM(u_h^{n+1}) \leq TVM(u_h^n)$$

Proof: Taking $v_h = 1$ for $x \in I_i$

$$\begin{aligned} w_i &= \bar{u}_i - \lambda_i[\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}], \qquad \lambda_i = \frac{\Delta t}{\Delta x_i} \\ &= \bar{u}_i - \lambda_i[\hat{f}(\bar{u}_i + \hat{u}_i, \bar{u}_{i+1} - \check{u}_{i+1}) - \hat{f}(\bar{u}_{i-1} + \hat{u}_{i-1}, \bar{u}_i - \check{u}_i)] \end{aligned}$$

We can write this in incremental form with

$$C_{i+\frac{1}{2}} = -\lambda_i \frac{\hat{f}(\bar{u}_i + \hat{u}_i, \bar{u}_{i+1} - \check{u}_{i+1}) - \hat{f}(\bar{u}_i + \hat{u}_i, \bar{u}_i - \check{u}_i)}{\Delta_+ \bar{u}_i}$$

$$D_{i-\frac{1}{2}} = \lambda_i \frac{\hat{f}(\bar{u}_i + \hat{u}_i, \bar{u}_i - \check{u}_i) - \hat{f}(\bar{u}_{i-1} + \hat{u}_{i-1}, \bar{u}_i - \check{u}_i)}{\Delta_- \bar{u}_i}$$

Rewrite the coefficient

$$C_{i+\frac{1}{2}} = \underbrace{-\lambda_i \hat{f}_2}_{\geq 0} \left(1 - \frac{\check{u}_{i+1}}{\Delta_+ \bar{u}_i} + \frac{\check{u}_i}{\Delta_+ \bar{u}_i} \right)$$

where

$$0 \leq -\hat{f}_2 = -\frac{\hat{f}(\bar{u}_i + \hat{u}_i, \bar{u}_{i+1} - \check{u}_{i+1}) - \hat{f}(\bar{u}_i + \hat{u}_i, \bar{u}_i - \check{u}_i)}{(\bar{u}_{i+1} - \check{u}_{i+1}) - (\bar{u}_i - \check{u}_i)} \leq L_2$$

Since u_h^n has been pre-processed by the minmod limiter, we have

$$0 \le \frac{\check{u}_{i+1}}{\Delta_+ \bar{u}_i} \le 1, \qquad 0 \le \frac{\check{u}_i}{\Delta_+ \bar{u}_i} \le 1 \quad \Longrightarrow \quad 1 - \frac{\check{u}_{i+1}}{\Delta_+ \bar{u}_i} + \frac{\check{u}_i}{\Delta_+ \bar{u}_i} \le 2$$

and hence

$$0 \le C_{i+\frac{1}{2}} \le 2\lambda_i L_2$$

and similarly

$$0 \le D_{i+\frac{1}{2}} \le 2\lambda_{i+1} L_1$$

If the time step satisfies the condition

$$C_{i+\frac{1}{2}} + D_{i+\frac{1}{2}} = 2(\lambda_i L_2 + \lambda_{i+1} L_1) \le 1 \qquad \text{or} \qquad \Delta t \le \frac{1}{2(\frac{L_1}{\Delta x_{i+1}} + \frac{L_2}{\Delta x_i})}$$

then from Hartens's Lemma, we have

$$\text{TVM}(w_h^{n+1}) \le \text{TVM}(u_h^n)$$

We now apply the limiter to obtain $u_h^{n+1} = \Lambda\Pi_h(w_h^{n+1})$, and we know that u_h^{n+1} and w_h^{n+1} have the same cell average values. Hence

$$\text{TVM}(u_h^{n+1}) = \text{TVM}(w_h^{n+1}) \le \text{TVM}(u_h^n)$$

which proves the theorem. □

Remark 9.2. *In the case of linear convection equation and upwind flux, we have* $L_1 = \frac{1}{2}(|a| + a)$, $L_2 = \frac{1}{2}(|a| - a)$ *so that the time step on a uniform mesh should satisfy the condition* $\frac{|a|\Delta t}{\Delta x} \le \frac{1}{2}$.

Remark 9.3. *On non-uniform grids, we can define the limiter as*

$$\begin{aligned}
\hat{u}_i^{(m)} &= \Delta x_i m\left(\frac{\hat{u}_i}{\Delta x_i}, \frac{\Delta_+ \bar{u}_i}{\frac{1}{2}(\Delta x_i + \Delta x_{i+1})}, \frac{\Delta_- \bar{u}_i}{\frac{1}{2}(\Delta x_i + \Delta x_{i-1})}\right) \\
\check{u}_i^{(m)} &= \Delta x_i m\left(\frac{\check{u}_i}{\Delta x_i}, \frac{\Delta_+ \bar{u}_i}{\frac{1}{2}(\Delta x_i + \Delta x_{i+1})}, \frac{\Delta_- \bar{u}_i}{\frac{1}{2}(\Delta x_i + \Delta x_{i-1})}\right)
\end{aligned}$$

The factor in $C_{i+\frac{1}{2}}$ *is*

$$1 - \frac{\check{u}_{i+1}}{\Delta_+ \bar{u}_i} + \frac{\check{u}_i}{\Delta_+ \bar{u}_i} \le 1 + \frac{\Delta x_i + \Delta x_{i+1}}{2\Delta x_i}$$

and in $D_{i+\frac{1}{2}}$ *is*

$$1 - \frac{\hat{u}_i}{\Delta_- \bar{u}_{i+1}} + \frac{\hat{u}_{i+1}}{\Delta_- \bar{u}_{i+1}} \le 1 + \frac{\Delta x_i + \Delta x_{i+1}}{2\Delta x_{i+1}}$$

The CFL condition is

$$C_{i+\frac{1}{2}} + D_{i+\frac{1}{2}} \le \left(1 + \frac{\Delta x_i + \Delta x_{i+1}}{2\Delta x_i}\right)\lambda_i L_1 + \left(1 + \frac{\Delta x_i + \Delta x_{i+1}}{2\Delta x_{i+1}}\right)\lambda_{i+1} L_2 \le 1$$

9.2 Limiters: Implementation

Degree $k = 1$: Let us write the solution in terms of Taylor or Legendre basis

$$u_h = \bar{u}_i + \left(\frac{x - x_i}{\frac{1}{2}\Delta x_i} \right) s_i$$

Note that $u_h(x^-_{i+\frac{1}{2}}) = \bar{u}_i + s_i$ and $u_h(x^+_{i-\frac{1}{2}}) = \bar{u}_i - s_i$. We limit the slope with the minmod function

$$s_i^{(m)} = \text{minmod}\left(s_i, \bar{u}_i - \bar{u}_{i-1}, \bar{u}_{i+1} - \bar{u}_i\right)$$

The limited solution is

$$\Lambda\Pi_h^1(u_h) = \bar{u}_i + \left(\frac{x - x_i}{\frac{1}{2}\Delta x_i} \right) s_i^{(m)}$$

Degree $k > 1$: Let us write the solution in terms of Taylor or Legendre basis

$$u_h = \bar{u}_i + \left(\frac{x - x_i}{\frac{1}{2}\Delta x_i} \right) s_i + HOT$$

We obtain a limited slope

$$s_i^{(m)} = \text{minmod}\left(s_i, \bar{u}_i - \bar{u}_{i-1}, \bar{u}_{i+1} - \bar{u}_i\right)$$

If $s_i^{(m)} = s_i$ then

$$\Lambda\Pi_h(u_h) = u_h \in \mathbb{P}_k$$

else

$$\Lambda\Pi_h(u_h) = \bar{u}_i + \left(\frac{x - x_i}{\frac{1}{2}\Delta x_i} \right) s_i^{(m)} \in \mathbb{P}_1$$

Remark 9.4. *If we use the Legendre polynomials to form the basis functions as* $\hat{\varphi}_j(\xi) = \sqrt{2j+1}P_j(\xi)$ *which is used in* `deal.II`*, then the linear solution is of the form*

$$u_h = \bar{u}_i + \sqrt{3}\left(\frac{x - x_i}{\frac{1}{2}\Delta x_i} \right) s_i$$

The limited slope is given by

$$s_i^{(m)} = \frac{1}{\sqrt{3}} minmod\left(\sqrt{3}s_i, \bar{u}_i - \bar{u}_{i-1}, \bar{u}_{i+1} - \bar{u}_i\right)$$

9.3 Numerical example

Discontinuous solution and smooth solution

$\boxed{Numerical\ example}$

If the solution in cell I_i has an extremum, see figure (xxx), then the slope in the cell has a different sign compared to $\Delta_- \bar{u}_i$, $\Delta_+ \bar{u}_i$ and the limiter returns zero slope. The limited solution becomes constant and equal to the cell average value. Thus there is loss of accuracy at smooth extrema due to TVD limiter. We have to modify the limiter so that it does not change the solution at smooth extrema.

9.4 TVB Limiter

In smooth regions of the solution, the differences inside the cell are

$$\hat{u}_i = \frac{1}{2}u_x(x_i)\Delta x_i + \mathcal{O}\left(h^2\right), \qquad \check{u}_i = \frac{1}{2}u_x(x_i)\Delta x_i + \mathcal{O}\left(h^2\right)$$

while the differences of cell averages are

$$\Delta_+ \bar{u}_i = \frac{1}{2}u_x(x_i)(\Delta x_i + \Delta x_{i+1}) + \mathcal{O}\left(h^2\right)$$

$$\Delta_- \bar{u}_i = \frac{1}{2}u_x(x_i)(\Delta x_i + \Delta x_{i-1}) + \mathcal{O}\left(h^2\right)$$

If the solution is smooth and monotone around I_i, and since the $\mathcal{O}\left(h^2\right)$ term is small, all the above quantities have the same sign, so that the limiter yields

$$\hat{u}_i^{(m)} = \hat{u}_i, \qquad \check{u}_i^{(m)} = \check{u}_i$$

Since the solution is not modified by the limiter, we obtain the full accuracy of the scheme. However, if there is a smooth extremum in cell I_i, then $u_x(x_i) \approx 0$ and the sign is determined by the second derivative. If M is the magnitude of the second derivative at smooth extrema, then $\hat{u}_i = \mathcal{O}\left(Mh^2\right)$. This motivates the definition of the TVB limiter function as

$$\tilde{m}(a_1, a_2, \ldots, a_l) = \begin{cases} a_1 & \text{if } |a_1| \leq Mh^2 \\ m(a_1, a_2, \ldots, a_l) & \text{otherwise} \end{cases}$$

If we are near a smooth extremum, the TVB limiter returns the original slope and the solution is not modified in that cell. With the above limiter, the scheme is no longer TVDM and the TVM can increase. However this violation is small and of the other of the mesh size.

Lemma 9.2 (TVB property)**.** *With the TVB limiter, if the CFL condition*

$$\Delta t \leq \frac{1}{2(\frac{L_1}{\Delta x_{i+1}} + \frac{L_2}{\Delta x_i})}, \qquad \forall i$$

is satisfied, then

$$TVM(u_h^{n+1}) \leq TVM(u_h^n) + CMh$$

Proof: See [6]

Remark 9.5. *If we are interested in the solution in a finite time interval $[0, T]$, then the TVB limiter ensures that*

$$TVM(u_h^n) \leq TV(u_0) + CMT, \qquad \forall\ n\Delta t \leq T$$

so that the numerical solutions have bounded variation. This is sufficient to prove convergence to a weak solution.

Remark 9.6. *The quantity M is an estimate of the second derivative of the solution at smooth extrema. This can be based on the initial condition, e.g.,*

$$M = \max_x \{|u_0''(x)| : u_0'(x) = 0\}$$

Ideally M should be estimated from the numerical solution but there is no reliable way to do this. The solution may have several extrema with different magnitude of the second derivatives. In practice people choose the value of M by some trial and error. But this is still a weak point of the TVB limiter.

Numerical example

9.5 Algorithm

The DG scheme together with the TVD/TVB limiter is as follows.

- Compute and store the mass matrix
- Determine w_h^0 from initial condition u_0 by an L^2-projection
- Find u_h^0 from w_h^0 by applying the limiter, $u_h^0 = \Lambda\Pi_h(w_h^0)$
- For $n = 0, 1, \ldots$
  - Compute time step from CFL condition
  - Set $u_h^{n,0} = u_h^n$
  - RK stages: For $r = 0, 1, \ldots, N_{rk} - 1$
    - Compute right hand side $L_h(u_h^{n,r})$
    - Update solution to next RK stage

      $$u_h^{n,r} \to w_h^{n,r+1}$$

    - Apply limiter

      $$u_h^{n,r+1} = \Lambda\Pi_h(w_h^{n,r+1})$$

Chapter 10

Fourier analysis

We will perform Fourier stability analysis of the DG scheme applied to the linear advection equation

$$u_t + au_x = 0, \qquad a > 0$$

We first derive the semi-discrete equations in matrix form, then apply a time integration scheme to obtain the fully discrete scheme. Finally, we plug in one Fourier mode and derive the amplification matrix. An examination of the eigenvalues of the amplification matrix yields the CFL condition.

10.1 Semi-discrete scheme

Let us take a uniform grid of size Δx and approximate the solution in cell I_i in terms of some modal basis functions

$$x \in I_i : \qquad u_h = \sum_{l=0}^{N} u_{il}\hat{\varphi}_l(\xi), \qquad -1 \le \xi \le +1$$

which is a polynomial of degree $N \ge 0$. Then, using the upwind flux, the semi-discrete DG scheme is given by: for $j = 0, 1, 2, \ldots, N$

$$\frac{\Delta x}{2}\int_{-1}^{+1} \frac{\partial u_h}{\partial t}\hat{\varphi}_j \mathrm{d}\xi = a\int_{-1}^{+1} u_h \frac{\mathrm{d}\hat{\varphi}_j}{\mathrm{d}\xi}\mathrm{d}\xi + au_h(x^-_{i-\frac{1}{2}})\hat{\varphi}_j(-1) - au_h(x^-_{i+\frac{1}{2}})\hat{\varphi}_j(+1)$$

Define

$$M_{jl} = \frac{1}{2}\int_{-1}^{+1} \hat{\varphi}_j\hat{\varphi}_l \mathrm{d}\xi, \qquad A_{jl} = \int_{-1}^{+1} \frac{\mathrm{d}\hat{\varphi}_j}{\mathrm{d}\xi}\hat{\varphi}_l \mathrm{d}\xi$$

$$B^-_{jl} = \hat{\varphi}_j(-1)\hat{\varphi}_l(+1), \qquad B^+_{jl} = \hat{\varphi}_j(+1)\hat{\varphi}_l(+1)$$

Then the semi-discrete scheme can be written as an ODE system

$$M\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{a}{\Delta x}(Au_i + B^- u_{i-1} - B^+ u_i)$$

where

$$u_i = [u_{i0},\ u_{i1},\ \ldots,\ u_{iN}]^\top \in \mathbb{R}^{N+1}$$

We assume that the modal coefficients have a Fourier representation, so that

$$u_i(t) = U(t)\mathrm{e}^{\mathrm{i}kx_i}$$

where $U \in \mathbb{R}^{N+1}$ is the vector of Fourier amplitudes which depends on the wavenumber k. Then the semi-discrete scheme takes the form

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \frac{a}{\Delta x} M^{-1}(A + \mathrm{e}^{-\mathrm{i}k\Delta x} B^- - B^+)U = \frac{a}{\Delta x} CU$$

where

$$C = M^{-1}(A + \mathrm{e}^{-\mathrm{i}k\Delta x} B^- - B^+)$$

is a $(N+1) \times (N+1)$ matrix. Since we use orthogonal polynomials, the mass matrix M is diagonal and equal to identity matrix with appropriate choice of basis functions, i.e., $\hat{\varphi}_j(\xi) = \sqrt{2j+1} P_j(\xi)$.

10.2 Dissipation and dispersion property

Assume that the Fourier amplitude is of the form $U(t) = \hat{U} \exp(-\mathrm{i}\omega t)$ and consider the eigenvalue problem

$$\frac{a}{\Delta x} C\hat{U} = -\mathrm{i}\omega \hat{U} \quad \Longrightarrow \quad C\hat{U} = -\mathrm{i}\frac{\omega \Delta x}{a}\hat{U} = -\mathrm{i}\Omega \hat{U}, \qquad \Omega = \frac{\omega \Delta x}{a}$$

The eigenvalue Ω which could be complex depends on $k\Delta x$ since the matrix C depends on $k\Delta x$. The exact dispersion relation is $\omega = ak$. Define

$$K = \frac{k\Delta x}{N+1}$$

The wave number k is related to the wavelength λ by $k = 2\pi/\lambda$ so that

$$K = \frac{2\pi}{\lambda}\frac{\Delta x}{N+1} = \frac{2\pi}{p}, \qquad p = \frac{\lambda}{h/(N+1)}$$

Note that $h/(N+1)$ is the effective spacing and p is the points per wave-length. The minimum value of p is 2 in order to be able to identify a wave and $K \in [0, \pi]$ while $k\Delta x \in [0, (N+1)\pi]$. We obtain $N+1$ eigenvalues but only one of them corresponds to the physically correct mode while the remaining modes are parasitic modes. For the case $N = 2$, Figure (10.1) shows all three modes. The imaginary part of Ω represents dissipation and note that it is negative for all modes indicating the stability of the DG scheme. The physical mode is the only one with zero dissipation at zero wavenumber and all the parasitic modes are heavily damped at the smaller wavenumbers, which are the ones which are well resolved on the mesh. The real part of Ω gives rise to phase errors. Figure (10.2) shows the real and imaginary parts of Ω for the physical mode at various orders of approximation. With increasing degree, the scheme becomes accurate over increasing range of wavenumbers. For more information, see [10], [9], [1].

10.3 Time integration schemes

The forward Euler scheme is given by

$$U^{n+1} = (I + \nu C)U^n = GU^n, \qquad G := I + \nu C, \qquad \nu := \frac{a\Delta t}{\Delta x}$$

The 2-stage, 2-nd order SSPRK scheme is

$$U^1 = GU^n, \qquad U^{n+1} = \frac{1}{2}U^n + \frac{1}{2}GU^1$$

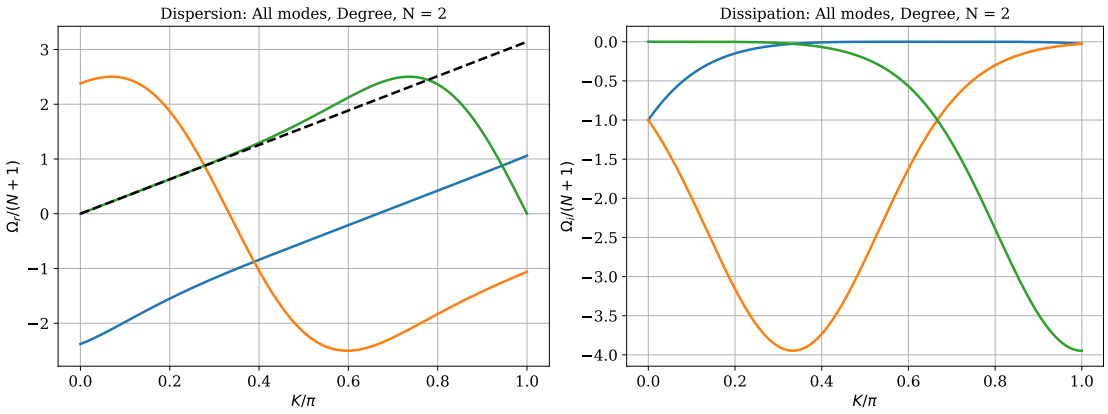


Figure 10.1: Real and imaginary parts of eigenvalues for $N = 2$. The physical mode is in green.

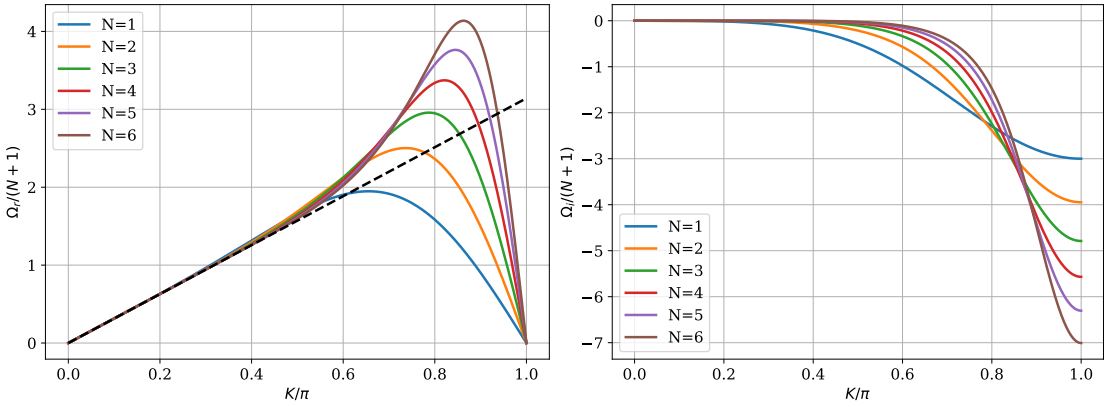


Figure 10.2: Real and imaginary parts of eigenvalues of physical mode for various degree N.

so that combining both steps, we get

$$U^{n+1} = \frac{1}{2}[I + G^2]U^n$$

The 3-stage, 3-rd order SSPRK scheme is

$$U^1 = GU^n, \quad U^2 = \frac{3}{4}U^n + \frac{1}{4}GU^1, \quad U^{n+1} = \frac{1}{3}U^n + \frac{2}{3}GU^2$$

and combining all the steps we obtain

$$U^{n+1} = \left(\frac{1}{3}I + \frac{1}{2}G + \frac{1}{6}G^3\right)U^n$$

The 4-stage, 3-rd order SSPRK scheme is (now $G = I + \frac{1}{2}\nu C$)

$$U^1 = GU^n, \quad U^2 = GU^1, \quad U^3 = \frac{2}{3}U^n + \frac{1}{3}GU^2, \quad U^{n+1} = GU^3$$

and combining all the steps, we get

$$U^{n+1} = \left(\frac{2}{3}G + \frac{1}{3}G^4\right)U^n$$

The 5-stage, 4-th order SSPRK scheme can be written as

$$U^{n+1} = (c_{51}G_2 + G_5G_3 + G_6G_4)U^n$$

where

$$G_1 = I + c_{11}\nu C, \qquad G_2 = c_{21}I + (c_{22}I + c_{23}\nu C)G_1$$
$$G_3 = c_{31}I + (c_{32}I + c_{33}\nu C)G_2, \qquad G_4 = c_{41}I + (c_{42}I + c_{43}\nu C)G_3$$
$$G_5 = c_{52} + c_{54}\nu C, \qquad G_6 = c_{53}I + c_{55}\nu C$$

The 4-stage, 4-order classical RK scheme is

$$U^{n+1} = \left(I + \frac{1}{6}G_1 + \frac{1}{3}G_2 + \frac{1}{3}G_3 + \frac{1}{6}G_4\right)U^n$$

where

$$G_1 = \nu C, \quad G_2 = G_1\left(I + \frac{1}{2}G_1\right), \quad G_3 = G_1\left(I + \frac{1}{2}G_2\right), \quad G_4 = G_1(I + G_3)$$

10.3.1 CFL condition

Each of the schemes has the generic form

$$U^{n+1} = H(k\Delta x, \nu)U^n$$

where $H(k\Delta x, \nu)$ is the amplification matrix which depends on the CFL number ν. After n time steps, the amplitude is

$$U^n = [H(k\Delta x, \nu)]^n U^0$$

where U^0 is the initial Fourier amplitude. If we want the solution to remain bounded then all the eigenvalues of the matrix $H(k\Delta x, \nu)$ must lie inside the unit circle in the complex plane for all $k\Delta x \in [0, 2\pi]$. The smallest value of $\nu > 0$ for which atleast one eigenvalue crosses the boundary of the unit circle is the limiting value, beyond which we lose stability. This critical value of ν is determined numerically and the results are shown in Table (10.1).

Scheme	$N = 0$	$N = 1,$	$N = 2$	$N = 3$
FE	1.000	-	-	-
SSPRK22	1.000	0.333	-	-
SSPRK33	1.256	0.409	0.209	0.130
SSPRK43	2.000	0.590	0.306	0.191
SSPRK54	2.190	0.662	0.343	0.215
RK4	1.392	0.464	0.235	0.145

Table 10.1: CFL numbers from Fourier stability analysis

Remark 10.1. *There are some schemes whose order matches the number of stages, but this is not true beyond third order accuracy for SSP schemes. Schemes with higher number of stages can have higher CFL number and may be efficient due to this reason. E.g., at third order ($N = 2$) spatial accuracy, SSPRK33 has a smaller CFL than SSPRK43 though both are third order in time. Suppose we want to reach a final time T; then the number of time steps ($T/\Delta t$) is proportional to T/ν and the amount of work required is $W = O(Ts/\nu)$ where s is number of stages. We can compare the relative work of two schemes (s_1, ν_1) and (s_2, ν_2) as*

$$\frac{W_1}{W_2} = \frac{s_1}{s_2}\frac{\nu_2}{\nu_1}$$

For the third order spatial DG scheme

$$\frac{W(SSPRK33)}{W(SSPRK43)} = \frac{3}{4}\frac{0.306}{0.209} \approx 1.1$$

Hence using SSPRK43 can be about 11% faster than SSPRK33. However the accuracy of the solutions can be poor when run with higher CFL. For more analysis on this kind of issues, see [14].

Chapter 11

Hyperbolic systems in 1-D

11.1 Linear hyperbolic system

Consider the linear system of conservation laws

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}}{\partial x} = 0, \qquad \boldsymbol{f} = A\boldsymbol{u}, \qquad A = \text{constant matrix}$$

Let

$$R = \text{matrix of right eigenvectors}, \qquad L = \text{matrix of left eigenvectors}$$

Then we can diagonalize A

$$A = R\Lambda R^{-1} = R\Lambda L, \qquad \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m)$$

Define the characteristic variables

$$\boldsymbol{c} = R^{-1}\boldsymbol{u} = L\boldsymbol{u}$$

Then we get

$$L\frac{\partial \boldsymbol{u}}{\partial t} + LR\Lambda L\frac{\partial \boldsymbol{u}}{\partial x} = 0 \quad \Longrightarrow \quad \frac{\partial \boldsymbol{c}}{\partial t} + \Lambda\frac{\partial \boldsymbol{c}}{\partial x} = 0$$

The is a set of decoupled scalar convection equations

$$\frac{\partial c_i}{\partial t} + \lambda_i\frac{\partial c_i}{\partial x} = 0, \qquad i = 1, 2, \ldots, m$$

We can solve for $c_i(x,t)$

$$c_i(x,t) = c_i(x - \lambda_i t, 0)$$

and then obtain the solution as

$$\boldsymbol{u}(x,t) = R\boldsymbol{c}(x,t)$$

The solution $\boldsymbol{u}$ is a linear combination of the waves c_i. This motivates the definition of non-linear hyperbolic systems.

11.2 Non-linear hyperbolic system of conservation laws

For a set of conserved quantities $\boldsymbol{u} \in \mathbb{R}^m$ with flux $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^m$, the system of conservation laws can be written as

$$\boldsymbol{u}_t + \boldsymbol{f}(\boldsymbol{u})_x = 0$$

We assume that this system is hyperbolic. This means that the Jacobian of the flux

$$A(\boldsymbol{u}) = \boldsymbol{f}'(\boldsymbol{u}) \in \mathbb{R}^{m \times m}$$

1. has only real eigenvalues

2. and the eigenvectors are linearly independent so that

$$\text{span of eigenvectors } = \mathbb{R}^m$$

11.3 Euler equations

An important example of non-linear, hyperbolic system of conservation laws is the Euler equations which govern the flow of inviscid compressible fluids. The conserved variables and flux vector are given by

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho v \\ E \end{bmatrix}, \qquad \boldsymbol{f} = \begin{bmatrix} \rho v \\ p + \rho v^2 \\ (E+p)u \end{bmatrix}$$

Here we have defined the internal and total energy per unit volume

$$e = \frac{p}{\gamma - 1}, \qquad E = e + \frac{1}{2}\rho u^2, \qquad \gamma = \frac{C_p}{C_v} > 1$$

Since density and pressure (internal energy) must be positive, the solution must lie in the physically admissible set of states

$$\mathcal{U}_{\text{ad}} = \{\boldsymbol{u} \in \mathbb{R}^3 : \rho(\boldsymbol{u}) > 0,\ p(\boldsymbol{u}) > 0\}$$

where

$$\rho(\boldsymbol{u}) = u_1, \qquad p(\boldsymbol{u}) = (\gamma - 1)\left[u_3 - \frac{u_2^2}{2u_1}\right]$$

The set $\mathcal{U}_{\text{ad}}$ is convex.

The Euler equations are hyperbolic. The flux Jacobian $A(\boldsymbol{u}) = \boldsymbol{f}'(\boldsymbol{u})$ has eigenvalues

$$\lambda_1 = v - a, \qquad \lambda_2 = v, \qquad \lambda_3 = v + a$$

where

$$a = \sqrt{\frac{\gamma p}{\rho}} = \text{speed of sound}$$

11.4 DG scheme

Divide domain $\Omega = [0,1]$ into cells $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$.

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \ldots < x_{N+\frac{1}{2}} = 1$$

$$x_i = \frac{1}{2}(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}), \qquad \Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \qquad h = \max_i \Delta x_i$$

Space of broken polynomials

$$\boldsymbol{V}_h^k = \{\boldsymbol{v} \ : \ \boldsymbol{v}|_{I_i} \in \boldsymbol{P}_k(I_i), \ 1 \le i \le N\}$$

Note that these functions can be discontinuous on the boundary of the elements. Define the left and right limits

$$\boldsymbol{v}_h(x^-) = \lim_{\epsilon \searrow 0} \boldsymbol{v}_h(x-\epsilon), \qquad \boldsymbol{v}_h(x^+) = \lim_{\epsilon \searrow 0} \boldsymbol{v}_h(x+\epsilon)$$

Multiply conservation law by $\boldsymbol{v}_h \in \boldsymbol{V}_h^k$ and integrate by parts

$$\int_{I_i} \frac{\partial \boldsymbol{u}}{\partial t} \cdot \boldsymbol{v}_h \mathrm{d}x - \int_{I_i} \boldsymbol{f}(\boldsymbol{u}) \cdot \frac{\partial \boldsymbol{v}_h}{\partial x} \mathrm{d}x + \boldsymbol{f}(\boldsymbol{u}(x^-_{i+\frac{1}{2}}, t)) \cdot \boldsymbol{v}_h(x^-_{i+\frac{1}{2}}) - \boldsymbol{f}(\boldsymbol{u}(x^+_{i-\frac{1}{2}}, t)) \cdot \boldsymbol{v}_h(x^+_{i-\frac{1}{2}}) = 0$$

We approximate the inter-element flux using a numerical flux function

$$\hat{\boldsymbol{f}}_{i+\frac{1}{2}}(t) = \hat{\boldsymbol{f}}(\boldsymbol{u}(x^-_{i+\frac{1}{2}}, t), \boldsymbol{u}(x^+_{i+\frac{1}{2}}, t))$$

This couples the solution in I_i to those in the neighbouring elements I_{i-1} and I_{i+1}.

The semi-discrete DG scheme is: Find $\boldsymbol{u}_h(\cdot, t) \in \boldsymbol{V}_h^k$ such that for all $\boldsymbol{v}_h \in \boldsymbol{V}_h^k$

$$\int_{I_i} \frac{\partial \boldsymbol{u}_h}{\partial t} \cdot \boldsymbol{v}_h \mathrm{d}x - \int_{I_i} \boldsymbol{f}(\boldsymbol{u}_h) \cdot \frac{\partial \boldsymbol{v}_h}{\partial x} \mathrm{d}x + \hat{\boldsymbol{f}}_{i+\frac{1}{2}}(t) \cdot \boldsymbol{v}_h(x^-_{i+\frac{1}{2}}) - \hat{\boldsymbol{f}}_{i-\frac{1}{2}}(t) \cdot \boldsymbol{v}_h(x^+_{i-\frac{1}{2}}) = 0$$

11.5 Numerical flux functions

We would like the numerical flux to satisfy some basic requirements.

1. Consistency
$$\hat{\boldsymbol{f}}(\boldsymbol{u}, \boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{u})$$

2. Lipschitz continuous
$$\|\hat{\boldsymbol{f}}(\boldsymbol{u}_2, \boldsymbol{v}_2) - \hat{\boldsymbol{f}}(\boldsymbol{u}_1, \boldsymbol{v}_1)\| \le L_1\|\boldsymbol{u}_2 - \boldsymbol{u}_1\| + L_2\|\boldsymbol{v}_2 - \boldsymbol{v}_1\|$$

Some numerical fluxes

- Rusanov flux
$$\hat{\boldsymbol{f}}(\boldsymbol{u},\boldsymbol{v}) = \frac{1}{2}[\boldsymbol{f}(\boldsymbol{u}) + \boldsymbol{f}(\boldsymbol{v})] - \frac{1}{2}\lambda(\boldsymbol{u},\boldsymbol{v})(\boldsymbol{v}-\boldsymbol{u})$$
where λ is an estimate of the maximum wave speed in the Riemann problem, e.g.,
$$\lambda(\boldsymbol{u},\boldsymbol{v}) = \max\{\lambda_{max}(\boldsymbol{u}), \lambda_{max}(\boldsymbol{v})\}$$
where
$$\lambda_{max}(\boldsymbol{u}) = \max_i |\lambda_i(A(\boldsymbol{u}))|$$

- Roe flux
$$\hat{\boldsymbol{f}}(\boldsymbol{u},\boldsymbol{v}) = \frac{1}{2}[\boldsymbol{f}(\boldsymbol{u}) + \boldsymbol{f}(\boldsymbol{v})] - \frac{1}{2}|A(\boldsymbol{u},\boldsymbol{v})|(\boldsymbol{v}-\boldsymbol{u})$$
where $|A| = R|\Lambda|L$ is computed at the Roe-average state.

11.6 Limiters and TVD property

Forward difference in time: Find $\boldsymbol{u}_h^{n+1}$ such that

$$\int_{I_i} \frac{\tilde{\boldsymbol{u}}_h^{n+1} - \boldsymbol{u}_h^n}{\Delta t} \cdot \boldsymbol{v}_h \mathrm{d}x - \int_{I_i} \boldsymbol{f}(\boldsymbol{u}_h^n) \cdot \frac{\partial \boldsymbol{v}_h}{\partial x} \mathrm{d}x + \hat{\boldsymbol{f}}^n_{i+\frac{1}{2}} \cdot \boldsymbol{v}_h(x^-_{i+\frac{1}{2}}) - \hat{\boldsymbol{f}}^n_{i-\frac{1}{2}} \cdot \boldsymbol{v}_h(x^+_{i-\frac{1}{2}}) = 0$$

The value $\tilde{\boldsymbol{u}}_h^{n+1}$ may be oscillatory. We limit it to obtain the new solution

$$\boldsymbol{u}_h^{n+1} = \Lambda\Pi_h(\tilde{\boldsymbol{u}}_h^{n+1})$$

Properties of $\Lambda\Pi_h(\cdot)$

1. It should not change the cell average value.
2. It should not affect the accuracy in smooth regions.

There are two possibilities to applying the limiter.

- Apply the limiter componentwise to the conserved variables.
- Apply the limiter componentwise to the characteristic variables.

The first approach is easy to implement and we can follow the procedure outlined for the scalar problems in previous chapters.

11.6.1 Characteristic limiter

Let $\overline{\boldsymbol{u}}_i$ denote the cell average value in cell I_i. The solution inside this cell has the form

$$\boldsymbol{u}_h = \overline{\boldsymbol{u}}_i + \left(\frac{x - x_i}{\frac{1}{2}\Delta x_i}\right) \boldsymbol{u}_x + HOT$$

Let

$$R_i = R(\overline{\boldsymbol{u}}_i) = \text{matrix of right eigenvectors of the flux Jacobian } \boldsymbol{f}'(\overline{\boldsymbol{u}}_i)$$

We define the local characteristic variables by

$$\boldsymbol{c}_i = R_i^{-1}\overline{\boldsymbol{u}}_i, \quad \boldsymbol{c}_{i\pm1} = R_i^{-1}\overline{\boldsymbol{u}}_{i\pm1}, \quad \boldsymbol{c}_x = R_i^{-1}\boldsymbol{u}_x$$

Now compute a limited estimate of the derivative

$$\boldsymbol{c}_x^{(m)} = m(\boldsymbol{c}_x, \boldsymbol{c}_i - \boldsymbol{c}_{i-1}, \boldsymbol{c}_{i+1} - \boldsymbol{c}_i)$$

If $\boldsymbol{c}_x^{(m)} = \boldsymbol{c}_x$ then

$$\Lambda\Pi_h(\boldsymbol{u}_h) = \boldsymbol{u}_h$$

else

$$\Lambda\Pi_h(\boldsymbol{u}_h) = \overline{\boldsymbol{u}}_i + \left(\frac{x - x_i}{\frac{1}{2}\Delta x_i}\right)\boldsymbol{u}_x^{(m)}, \qquad \boldsymbol{u}_x^{(m)} = R_i\boldsymbol{c}_x^{(m)}$$

For better acccuracy at smooth extrema, we can use the TVB version of the minmod limiter.

Another version

$$\boldsymbol{c}_i = R_i^{-1}\overline{\boldsymbol{u}}_i, \quad \boldsymbol{c}_{i\pm1} = R_i^{-1}\overline{\boldsymbol{u}}_{i\pm1}, \quad \boldsymbol{c}^+_{i-\frac{1}{2}} = R_i^{-1}\boldsymbol{u}^+_{i-\frac{1}{2}}, \quad \boldsymbol{c}^-_{i+\frac{1}{2}} = R_i^{-1}\boldsymbol{u}^-_{i+\frac{1}{2}}$$

and the differences

$$\hat{\boldsymbol{c}}_i = \boldsymbol{c}^-_{i+\frac{1}{2}} - \boldsymbol{c}_i, \qquad \check{\boldsymbol{c}}_i = \boldsymbol{c}_i - \boldsymbol{c}^+_{i-\frac{1}{2}}$$

$$\Delta_+\boldsymbol{c}_i = \boldsymbol{c}_{i+1} - \boldsymbol{c}_i, \qquad \Delta_-\boldsymbol{c}_i = \boldsymbol{c}_i - \boldsymbol{c}_{i-1}$$

We cannot modify the cell average value but we can modify the slopes

$$\hat{\boldsymbol{c}}_i^{(m)} = m(\hat{\boldsymbol{c}}_i, \Delta_+\boldsymbol{c}_i, \Delta_-\boldsymbol{c}_i), \qquad \check{\boldsymbol{c}}_i^{(m)} = m(\check{\boldsymbol{c}}_i, \Delta_+\boldsymbol{c}_i, \Delta_-\boldsymbol{c}_i)$$

where m is the minmod function

$$m(a_1, \ldots, a_l) = \begin{cases} s\min(|a_1|, \ldots, |a_l|) & s = \text{sign}(a_1) = \ldots = \text{sign}(a_l) \\ 0 & \text{otherwise} \end{cases}$$

For $k = 1$: We have $\hat{\boldsymbol{c}}_i = \check{\boldsymbol{c}}_i$. The modified trace values are recomputed using the limited slopes

$$\boldsymbol{u}_h^{(m)}(x^-_{i+\frac{1}{2}}) = \overline{\boldsymbol{u}}_i + R_i\hat{\boldsymbol{c}}_i^{(m)}, \qquad \boldsymbol{u}_h^{(m)}(x^+_{i-\frac{1}{2}}) = \overline{\boldsymbol{u}}_i - R_i\check{\boldsymbol{c}}_i^{(m)}$$

This completely specifies the limited polynomial solution inside the cell.
For $k > 1$: If $\hat{\boldsymbol{c}}_i^{(m)} \neq \hat{\boldsymbol{c}}_i$ or $\check{\boldsymbol{c}}_i^{(m)} \neq \check{\boldsymbol{c}}$ then the limiter is active which indicates that we are probably near a discontinuity. In this case we retain only the linear part of the solution. The trace values of the linear part are taken as

$$\boldsymbol{u}_h^{(m)}(x^-_{i+\frac{1}{2}}) = \overline{\boldsymbol{u}}_i + \frac{1}{2}R_i(\check{\boldsymbol{c}}_i^{(m)} + \hat{\boldsymbol{c}}_i^{(m)}), \qquad \boldsymbol{u}_h^{(m)}(x^+_{i-\frac{1}{2}}) = \overline{\boldsymbol{u}}_i - \frac{1}{2}R_i(\check{\boldsymbol{c}}_i^{(m)} + \hat{\boldsymbol{c}}_i^{(m)})$$

Remark 11.1. *We can use the TVB version of the minmod limiter which leads to a less restrictive limiter.*

Linear system In the case of the linear system

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}}{\partial x} = 0, \qquad \boldsymbol{f} = A\boldsymbol{u}, \qquad A = \text{constant matrix}$$

we have the following result

Theorem 11.1 (TVD property [6])**.** *The DG scheme with characteristic limiter is total variation diminishing in the means with the TVD limiter and total variation bounded with the TVB limiter, where the total variation is defined as*

$$TVM(\boldsymbol{u}_h) = \sum_i \sum_{j=1}^{m} |\overline{u}_{i,j} - \overline{u}_{i-1,j}|$$

11.7 Some implementation details

We have a vector of unknowns $\boldsymbol{u}_h$. We write each component of $\boldsymbol{u}_h$ in terms of the usual basis functions, e.g., Legendre polynomials. The j'th component of $\boldsymbol{u}_h$ has the form

$$x \in I_i : \qquad w_j(x,t) = \sum_{s=1}^{k+1} w_{i,j,s}(t)\phi_{i,s}(x), \qquad j = 1, \ldots, m$$

where k = degree of polynomial space. We have $(k+1)m$ dofs. The test functions are of the form

$$\boldsymbol{v}_h = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \phi_{i,r} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

There are $(k+1)m$ test functions.

The DG scheme is given by

$$\int_{I_i} \frac{\partial w_j}{\partial t}\phi_{i,r}\mathrm{d}x - \int_{I_i} f_j(\boldsymbol{u}_h)\frac{\partial \phi_{i,r}}{\partial x}\mathrm{d}x + (\hat{\boldsymbol{f}}_{i+\frac{1}{2}})_j \phi_{i,r}(x^-_{i+\frac{1}{2}}) - (\hat{\boldsymbol{f}}_{i-\frac{1}{2}})_j \phi_{i,r}(x^+_{i-\frac{1}{2}}) = 0$$
$$r = 1, 2, \ldots, k+1, \qquad j = 1, 2, \ldots, m$$

For component j define its dofs

$$\vec{w}_{i,j} = [w_{i,j,1}, \ldots, w_{i,j,k+1}]^\top \in \mathbb{R}^{k+1}$$

Then the DG scheme in element I_i can be written as

$$M_i \frac{\mathrm{d}\vec{w}_{i,j}}{\mathrm{d}t} = L_{i,j}(\boldsymbol{u}_h), \qquad j = 1, 2, \ldots, m$$

where

$$M_i \in \mathbb{R}^{(k+1)\times(k+1)} = \text{mass matrix}$$

with components

$$(M_i)_{rs} = \int_{I_i} \phi_{i,r}\phi_{i,s}\mathrm{d}x, \qquad r,s = 1,2,\ldots,k+1$$

This is a symmetric, positive definite matrix, and usually also diagonal. The integrals are evaluated using some Gauss quadrature rule. Then the resulting system of ODE can be integrated in time using a Runge-Kutta scheme.

Remark 11.2. *We have to choose a time step to apply a Runge-Kutta scheme to the ODE system. For degree k polynomials and $(k+1)$-stage Runge-Kutta scheme, we can choose*

$$\Delta t_i = CFL\frac{(|v_i| + c_i)\Delta x_i}{2k+1}, \qquad 0 < CFL \leq 1$$

for the Euler equations, where we can use cell average value to estimate the velocity and sound speed. The global time step is given by

$$\Delta t = \min_i \Delta t_i$$

which is required for time accurate computations.

Chapter 12

Mesh in 2-D

12.1 Basic element types

In 2-D the possible elements are triangles and quadrilaterals. These can have either straight sides or can be curved in order to represent curved boundaries. We can map the element from the physical domain to the computational domain where we have a reference element $\hat{K}$. The reference triangle can be taken to be the right angled triangle with unit sides and the hypotenuse of length $\sqrt{2}$ as shown in Figure (12.1). The mapping for a linear triangle is given by an affine map

$$\vec{x} = \vec{X}(\xi, \eta) = (1 - \xi - \eta)\vec{x}_1 + \xi\vec{x}_2 + \eta\vec{x}_3$$

This can also be written in a matrix-vector form as

$$\begin{bmatrix} x \\ y \end{bmatrix} = J_K \begin{bmatrix} \xi \\ \eta \end{bmatrix} + b_K, \qquad J_K =, \qquad b_K =$$

The inverse map is easy to compute and is given by

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = J_K^{-1}$$

For a linear quadrilateral, we can map to the reference quadrilateral $\hat{K} = [0, 1] \times [0, 1]$, see Figure (12.2) and the map is bi-linear and given by

$$\vec{X}(\xi, \eta) = (1 - \xi)(1 - \eta)\vec{x}_1 + \xi(1 - \eta)\vec{x}_2 + \xi\eta\vec{x}_3 + (1 - \xi)\eta\vec{x}_4$$

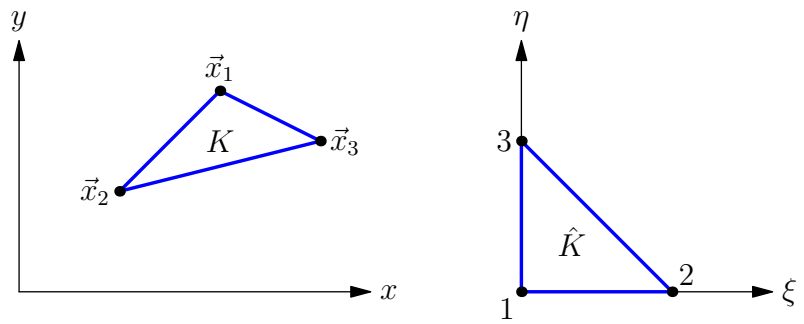


Figure 12.1: Affine mapping of a linear triangle

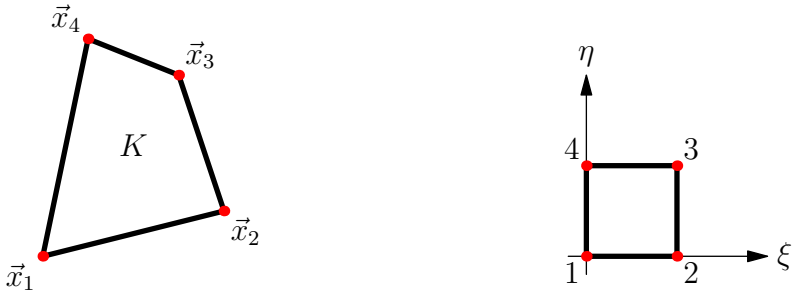


Figure 12.2: Affine mapping of a linear quadrilateral

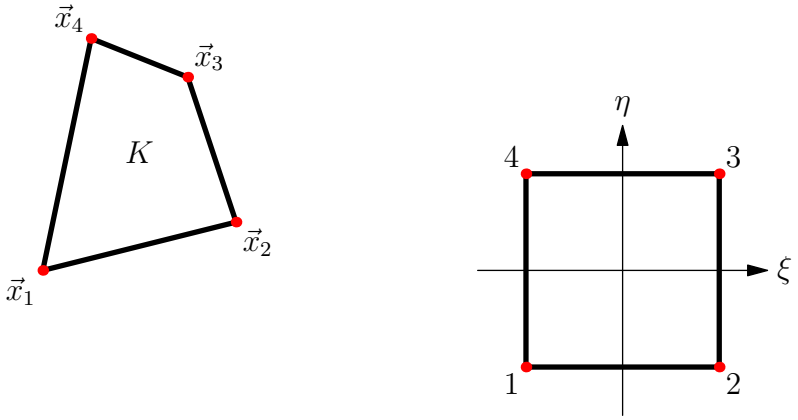


Figure 12.3: Affine mapping of a linear quadrilateral

Instead, we take the reference element to be $\hat{K} = [-1, +1] \times [-1, +1]$, see Figure (12.3), then the bi-linear map is given by

$$\begin{aligned} \vec{X}(\xi, \eta) = \quad & \frac{1}{4}(1-\xi)(1-\eta)\vec{x}_1 + \frac{1}{4}(1+\xi)(1-\eta)\vec{x}_2 \\ & + \frac{1}{4}(1+\xi)(1+\eta)\vec{x}_3 + \frac{1}{4}(1-\xi)(1+\eta)\vec{x}_4 \end{aligned} \tag{12.1}$$

In this case the inverse map cannot be written down explicitly, but see [11]. We can compute the inverse using Newton method; however it is rarely required in practice to know the inverse map.

Remark 12.1. *If the quadrilateral is a rectangle, then the bi-linear map reduces to an affine map. If the reference cell is $\hat{K} = [-1, +1]^2$, then the map is given by*

$$x = x_K + \frac{\Delta x}{2}\xi, \qquad y = y_K + \frac{\Delta y}{2}\eta$$

where (x_K, y_K) is the center of the rectangle.

Remark 12.2. *In* `deal.II`*, the reference cell is taken to be the unit square $\hat{K} = [0, 1]^d$.*

12.2 Mesh data structures

The mesh is basically specified by a list of coordinates of the vertices and the list of cells. The vertices are numbered in some manner and a cell is specified by provided the list

of vertex numbers that form the cell. A mesh of quadrilateral cells may be specified as follows

```
double coord[nvert][2];
int cell[ncell][4];
```

The order in which the vertex coordinates are listed gives them an implicit global numbering which is used in the `cell` array. We will assume that the vertices for each cell, for the i'th cell, the vertices are

```
cell[i][0], cell[i][1], cell[i][2], cell[i][3]
```

are arranged in some fixed order, say counter-clockwise. For a DG method, it is also useful to have a list of faces since we have to compute fluxes across faces and also implement some boundary conditions on faces which lie on the boundary of the domain. We need to associate certain information for each face

- The two vertices forming the face
- The two cells adjacent to the face
- Local face numbering of adjacent cells
- A unique indicator to tell us if the face is on the domain boundary and if yes, then the value of the indicator is useful to specify boundary condition.

We can encapsulate this information in a class as follows.

```
class Face
{
   public:
      int vertex[2];
      int lcell, rcell;
      int lface, rface;
      int bd_id;
};
```

By specifying the vertices, we endow an orientation to the face. When we move from vertex 0 to vertex 1, the cell `lcell` is to your left and `rcell` is to your right. For a boundary edge, we can take the convention that `lcell` is present but there is no `rcell`, which requires arranging the two vertices in a particular order. Or, we can set a value of -1 to indicate that there is no cell on one side of the face.

For example, for the face shown in Figure (12.4) we have `lface=-2` and `rface=3`. TODO We setup a local coordinate system on the face going from vertex 0 to vertex 1 and let $s \in [-1, +1]$ be a point on the face. We want to evaluate the solution at this point on the face from the two cells adjacent to the face. From the left cell, the solution is

$$u_h^{lcell}(-s, +1)$$

while from the right cell, the solution is

$$u_h^{rcell}(-1, s)$$

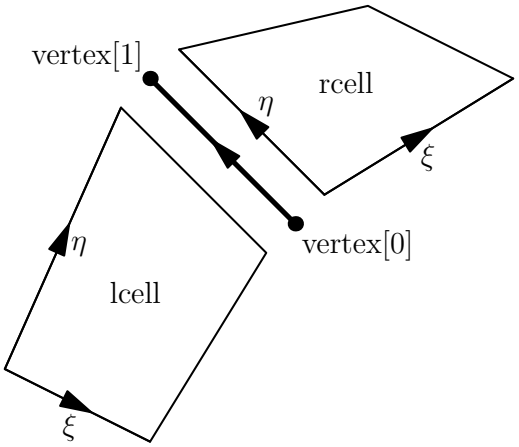


Figure 12.4: Connectivity of face to cells

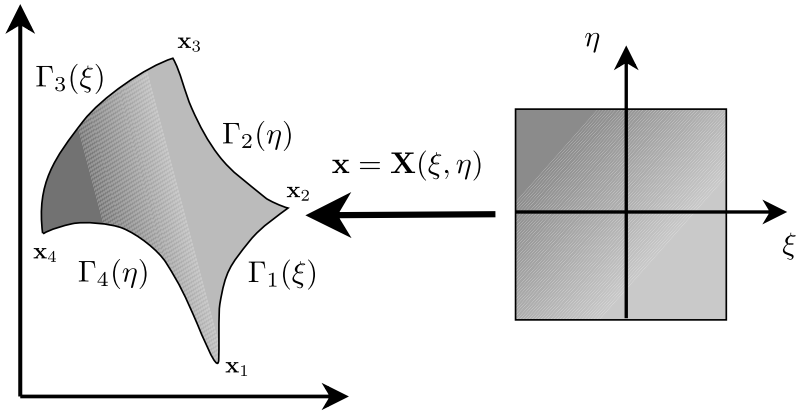


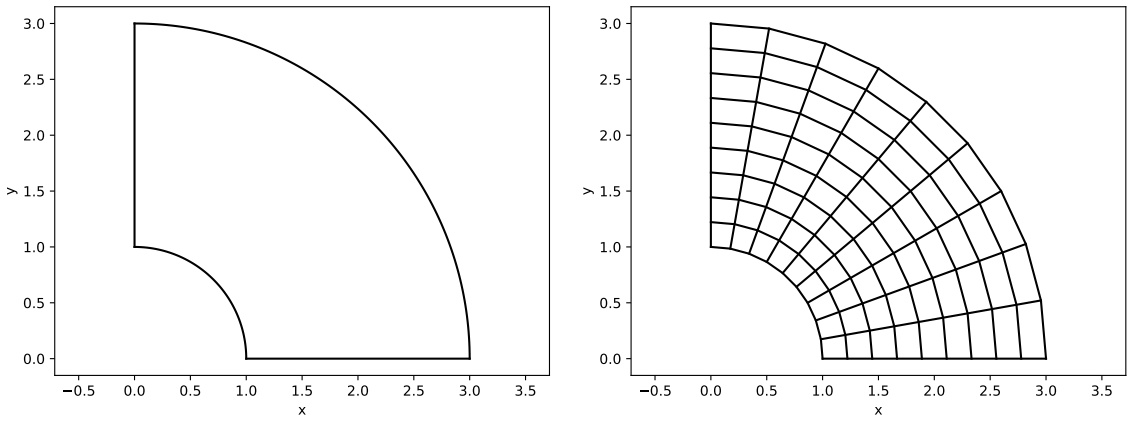
Figure 12.5: Transfinite mapping of a curved domain (from Kopriva)

12.3 Transfinite map

Suppose we are given a domain by specifying the four bounding curves $\vec{\Gamma}_1$, $\vec{\Gamma}_2$, $\vec{\Gamma}_3$, $\vec{\Gamma}_4$ as shown in Figure (12.5). Let us assume that the parameters ξ, η are proportional to the arc length and take values in $[-1, +1]$. The transfinite map is given by

$$\begin{aligned}\vec{X}(\xi,\eta) = &\frac{1-\xi}{2}\vec{\Gamma}_4(\eta) + \frac{1+\xi}{2}\vec{\Gamma}_2(\eta) + \frac{1-\eta}{2}\vec{\Gamma}_1(\xi) + \frac{1+\eta}{2}\vec{\Gamma}_3(\xi) \\ &- \frac{(1-\xi)(1-\eta)}{4}\vec{x}_1 - \frac{(1+\xi)(1-\eta)}{4}\vec{x}_2 \\ &- \frac{(1+\xi)(1+\eta)}{4}\vec{x}_3 - \frac{(1-\xi)(1+\eta)}{4}\vec{x}_4\end{aligned}$$

We can use the transfinite map to generate a structured mesh. An example is shown in Figure (12.6) which is obtained by uniformly dividing the (ξ, η) space and the sides are



(a) (b)

Figure 12.6: Structured mesh using transfinite mapping

given by the curves

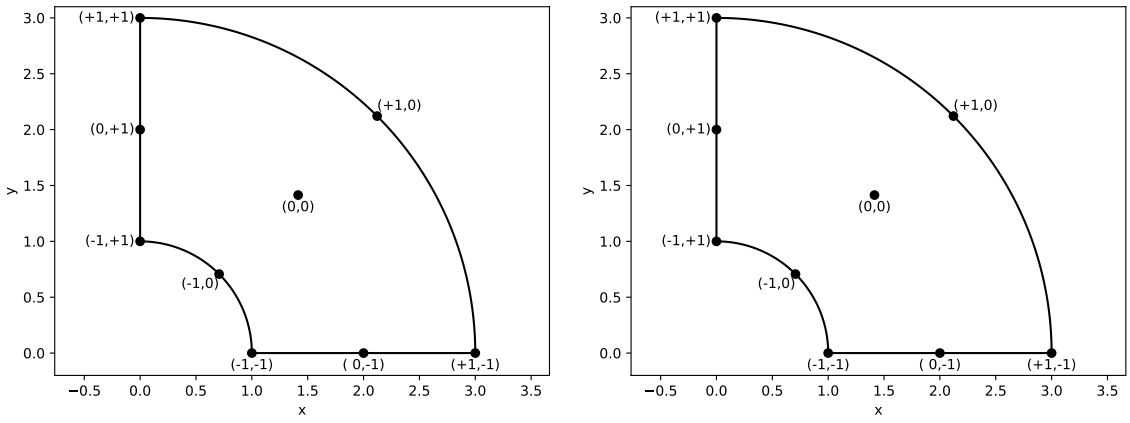
$$\begin{aligned}\vec{\Gamma}_1 &= \\ \vec{\Gamma}_2 &= \\ \vec{\Gamma}_3 &= \\ \vec{\Gamma}_4 &=\end{aligned}$$

This mesh is composed of straight sided quadrilaterals. If we want better representation of curved boundaries, we can put extra points in between the elements and use them to construct a polynomial approximation of curved sides that is more accurate than straight line approximation.

12.4 Isoparametric elements

Suppose Figure (12.6a) shows one element in our mesh. We can of course use the transfinite map in our numerical scheme to represent this curved element. The other option is to construct a polynomial approximation of the curved sides. For example if we want quadratic approximation of the curved sides, we can put extra points on each of the four curves, say at their midpoint, and one point at the center, as shown in Figure (12.7a). Then the quadratic approximation is

$$\begin{aligned}\vec{X}_2(\xi,\eta) = \quad & \frac{1}{4}\xi(1-\xi)\eta(1-\eta)\vec{x}_{00} - \frac{1}{2}(1-\xi)(1+\xi)\eta(1-\eta)\vec{x}_{10} \\ & - \frac{1}{4}\xi(1+\xi)\eta(1-\eta)\vec{x}_{20} \\ & - \frac{1}{2}\xi(1-\xi)(1-\eta)(1+\eta)\vec{x}_{01} + (1-\xi)(1+\xi)(1-\eta)(1+\eta)\vec{x}_{11} \\ & + \frac{1}{2}\xi(1+\xi)(1-\eta)(1+\eta)\vec{x}_{21} \\ & - \frac{1}{4}\xi(1-\xi)\eta(1+\eta)\vec{x}_{02} + \frac{1}{2}(1-\xi)(1+\xi)\eta(1+\eta)\vec{x}_{12} \\ & + \frac{1}{4}\xi(1+\xi)\eta(1+\eta)\vec{x}_{22}\end{aligned}$$



(a) (b)

Figure 12.7: Example of quadratic element

E.g., setting $\xi = -1$ we get

$$\vec{X}_2(-1,\eta) = -\frac{1}{2}\eta(1-\eta)\vec{x}_{00} + (1-\eta)(1+\eta)\vec{x}_{01} + \frac{1}{2}\eta(1+\eta)\vec{x}_{02}$$

which is a quadratic curve wrt η and passes through the three points. Figure (12.7b) shows the original element and its quadratic approximation. An element represented by such a polynomial interpolation is usually called *isoparametric*. The term isoparametric elements arose from continuous Galerkin methods where the same basis functions used to approximate the solution are used to approximate the domain. This allows accurate representation of curved boundaries so that optimal convergence rates of the error can be achieved. For example, a general quadrilateral element can be represented in terms of tensor product of 1-D Lagrange polynomials of degree k based on GLL nodes as

$$\vec{X}(\xi,\eta) = \sum_{i=0}^{k}\sum_{j=0}^{k} \vec{x}_{ij}\ell_i(\xi)\ell_j(\eta)$$

The use of GLL nodes is essential since we want the elements to patched up in a continuous manner.

The map $\vec{X} : \hat{K} \to K$ is clearly one-to-one and it must also be onto, so that the inverse exists and is unique. This requires that the Jacobian matrix

$$J = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

must be non-singular everywhere inside the reference element, i.e.,

$$\det J(\xi,\eta) \neq 0 \qquad (\xi,\eta) \in \hat{K}$$

Since

$$|K| = \int_K \mathrm{d}x\mathrm{d}y = \int_{\hat{K}} \det(J_K)\mathrm{d}\xi\mathrm{d}\eta > 0$$

and J_K is continuous, we actually require that

$$\det J(\xi,\eta) > 0 \qquad \forall(\xi,\eta) \in \hat{K}$$

Remark 12.3. *When we want to refine the mesh by dividing the elements, the new points must be introduced using the exact boundary definitions and not using the isoparametric representation.*

12.4.1 Jacobians of transformation

In assembling the weak formulation of finite element methods, the derivatives between physical and reference cell coordinates may be required. The map $\vec{X}$ is available as a polynomial function of ξ, η so that quantities like

$$\frac{\partial x}{\partial \xi}, \quad \frac{\partial x}{\partial \eta}, \quad \frac{\partial y}{\partial \xi}, \quad \frac{\partial y}{\partial \eta}$$

are easy to compute from the known map. The derivatives of ξ, η wrt x, y are not directly available. From the two relations

$$\begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \mathrm{d}\xi \\ \mathrm{d}\eta \end{bmatrix}, \qquad \begin{bmatrix} \mathrm{d}\xi \\ \mathrm{d}\eta \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \mathrm{d}\xi \\ \mathrm{d}\eta \end{bmatrix}$$

Hence

$$\begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} = I \quad \Longrightarrow \quad \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}^{-1} = \frac{1}{\det J} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial x}{\partial \eta} \\ -\frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{bmatrix}$$

and last equation gives the required derivatives, where

$$\det J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}$$

12.4.2 Normal vector

In DG schemes we require the unit normal vector to element faces to compute the numerical flux. This can be computed using the knowledge of the mapping function. Let us take the reference cell to be $[-1,+1] \times [-1,+1]$ and consider the face $\xi = +1$. The vector $\frac{\partial}{\partial \eta}\vec{X}(+1,\eta)$ is tangential to this face and we get an outward normal vector by rotating this by 90 degrees

$$\left(\frac{\partial y}{\partial \eta}(+1,\eta), -\frac{\partial x}{\partial \eta}(+1,\eta) \right)$$

and we can normalize this to get unit outward normal vector. The restriction of the map to any face is same whether we take it from one cell or the neighbouring cell, and hence the normal vector is also same, except we have to be careful about the orientation.

12.4.3 Cell integral

An integral over a cell is performed by doing the change of variable so that the area measure transform as

$$\mathrm{d}x\mathrm{d}y = J\mathrm{d}\xi\mathrm{d}\eta$$

so that

$$\int_K \phi(x,y)\mathrm{d}x\mathrm{d}y = \int_{\hat{K}} \phi(\xi,\eta) \det J(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta$$

and then applying a numerical quadrature on the reference cell.

12.4.4 Surface integrals

To compute a surface integral

$$\int_e \phi(x,y)\mathrm{d}s$$

we have to transform the measure $\mathrm{d}s$ to the reference element so that we can perform numerical quadrature. Since

$$\mathrm{d}s = \sqrt{\mathrm{d}x^2 + \mathrm{d}y^2}$$

we can then use the map $\vec{X}$. E.g., on the face $\xi = +1$, only the η coordinates is independent and we have

$$\frac{\mathrm{d}s}{\mathrm{d}\eta}(\eta) = \sqrt{\left[\frac{\partial x}{\partial \eta}(+1,\eta)\right]^2 + \left[\frac{\partial y}{\partial \eta}(+1,\eta)\right]^2}$$

Hence

$$\int_e \phi(x,y)\mathrm{d}s = \int_{-1}^{+1} \phi(x(+1,\eta), y(+1,\eta))\frac{\mathrm{d}s}{\mathrm{d}\eta}(\eta)\mathrm{d}\eta$$

and then we can apply a 1-D numerical quadrature in the η variable.

Chapter 13

Basis functions in 2-D

In two spatial dimensions, there are mainly two types of polynomial spaces we can use for function approximation.

1. Complete polynomials of degree $k \geq 0$ denoted by $\mathbb{P}_k$
2. Tensor product polynomials of degree $k \geq 0$ denoted by $\mathbb{Q}_k$

13.1 Complete polynomials $\mathbb{P}_k$

These are usual polynomials where the total exponent of each term is $\leq k$. Some examples are:

Constant polynomial: $\mathbb{P}_0$

$$p(x, y) = a_{00}$$

Linear polynomial: $\mathbb{P}_1$

$$p(x, y) = a_{00} + a_{10}x + a_{01}y$$

Quadratic polynomial: $\mathbb{P}_2$

$$p(x, y) = a_{00} + a_{10}x + a_{01}y + a_{20}x^2 + a_{11}xy + a_{02}y^2$$

The space $\mathbb{P}_k$ has dimension

$$N(k) = \frac{1}{2}(k+1)(k+2)$$

In the above, we wrote the polynomials using physical coordinates (x, y). But in finite element methods, it is comman to use mapped coordinates (ξ, η) in a reference element in which case the polynomial has the form

$$p(\xi, \eta) = \sum_{i=0}^{k} \sum_{j=0}^{k-i} a_{ij} \xi^i \eta^j$$

Note that the total power of each term is $i + j \leq k$. We have written the polynomial in terms of monomials but this is rarely used.

13.1.1 Modal basis for $\mathbb{P}_k$ on triangles

The names of Proriol, Koornwinder [12] and Dubiner [7] are associate with this type of polynomials. The presentation below closely follows [8]. Consider the reference triangle

$$\hat{K} = \{(r, s) : r, s \geq -1, r + s \leq 0\}$$

The transformation from reference triangle to real triangle is given by

$$\vec{x} = -\frac{1}{2}(r + s)\vec{x}_1 + \frac{1}{2}(r + 1)\vec{x}_2 + \frac{1}{2}(s + 1)\vec{x}_3$$

where $\vec{x}_1, \vec{x}_2, \vec{x}_3$ are the three vertices of the triangle ordered counter-clockwise. The orthonormal polynomials on this triangle are given by: for $m = 0, 1, \ldots, N(k) - 1$,

$$\hat{\varphi}_m(r, s) = \sqrt{2} P_i(a) P_j^{(2i+1,0)}(b)(1 - b)^i, \qquad 0 \leq i, j \leq k,\ i + j \leq k$$

where P_i is the i'th Legendre polynomial, $P_j^{(\alpha,\beta)}$ is the j'th order Jacobi polynomial, and

$$a = 2\frac{1 + r}{1 - s} - 1, \qquad b = s$$

The mapping from the indices (i, j) to m can be taken as

$$m = j + (k + 1)i - \frac{1}{2}i(i - 1)$$

Then each component of the solution in a triangle is of the form

$$u_h(\xi, \eta) = \sum_{m=0}^{N(k)-1} \tilde{u}_m \hat{\varphi}_m(\xi, \eta) \tag{13.1}$$

Recurrence relations The first two Legendre polynomials are

$$P_0(a) = 1, \qquad P_1(a) = a$$

Then we have the recurrence relation

$$P_i(a) = \frac{2i - 1}{i} a P_{i-1}(a) - \frac{i - 1}{i} P_{i-2}(a), \qquad i = 2, 3, \ldots$$

The Jacobi polynomials $P_j^{(\alpha,0)}$ satisfy the recurrence relation (CHECK)

$$\begin{aligned} 2j(j + \alpha)(2j + \alpha - 2) P_j^{(\alpha,0)}(b) =& (2j + \alpha - 1)((2j + \alpha)(2j + \alpha - 2)b + \alpha^2) P_{j-1}^{(\alpha,0)}(b) \\ & - 2(j + \alpha - 1)(j - 1)(2j + \alpha) P_{j-2}^{(\alpha,0)}(b) \end{aligned}$$

Stable evaluation of basis functions The value a becomes infinite at $s = 1$ so some care must be taken while evaluating these polynomials. Define

$$Q_i(a, b) = P_i(a)(1 - b)^i$$

The first two functions are

$$Q_0(a, b) = 1, \qquad Q_1(a, b) = a(1 - b)$$

Note that

$$a(1-b) = 2(1+r) - (1-s)$$

which is well defined for all values of (r, s). Then, using the recurrence relation of P_i we get

$$\begin{aligned} Q_i(a,b) &= \frac{2i-1}{i} a P_{i-1}(a)(1-b)^i - \frac{i-1}{i} P_{i-2}(a)(1-b)^i \\ &= \frac{2i-1}{i} a(1-b) Q_{i-1}(a,b) - \frac{i-1}{i}(1-b)^2 Q_{i-2}(a,b) \end{aligned}$$

Again the factor $a(1-b)$ appears, which is well defined for all values of (r, s). Given (r, s) we can compute $Q_i(a, b)$ without having to compute (a, b), thus avoiding the singularity at $s = 1$.

13.1.2 Modal basis for $\mathbb{P}_k$ on rectangles

We have introduced Legendre polynomials $P_j(\xi)$ which are mutually orthogonal on $[-1, +1]$ and using these we can define mutually orthonormal functions $\hat{\varphi}_j(\xi) = \sqrt{2j+1} P_j(\xi)$. A function in $\mathbb{P}_k$ can be written as

$$p(\xi, \eta) = \sum_{i=0}^{k} \sum_{j=0}^{k-i} a_{ij} \hat{\varphi}_i(\xi) \hat{\varphi}_j(\eta)$$

where the dofs a_{ij} are not nodal values but some modal coefficients which are moments of the polynomial

$$a_{ij} = \frac{1}{4} \int_{-1}^{+1} \int_{-1}^{+1} p(\xi, \eta) \hat{\varphi}_i(\xi) \hat{\varphi}_j(\eta) \mathrm{d}\xi \mathrm{d}\eta$$

We provide more details on this approach in Chapter (xxx).

13.1.3 Taylor basis for $\mathbb{P}_k$ on polygons

This type of functions are constructed directly in physical space and can be used on any type of polygonal cell. As the name suggests, they are inspired by the form of the terms in a Taylor expansion. They are essentially modal in character since the dofs do not represent nodal values. We provide more details on this approach in Chapter (xxx).

13.1.4 Nodal basis for $\mathbb{P}_k$ on triangles

Let us take the reference triangle $\hat{K}$ to be the right isosceles triangle. Let (ξ_i, η_i), $i = 0, 1, \ldots, N(k) - 1$ be a set of distinct nodes in the reference triangle and let $\ell_i(\xi, \eta) \in \mathbb{P}_k$ such that

$$\ell_i(\xi_j, \eta_j) = \delta_{ij}, \qquad 0 \leq i, j \leq N(k) - 1$$

Then the set $\{\ell_i, i = 0, 1, \ldots, N(k)-1\}$ forms a basis for $\mathbb{P}_k$ and any polynomial $p(\xi, \eta) \in \mathbb{P}_k$ can be written as

$$p(\xi, \eta) = \sum_{i=0}^{N(k)-1} p(\xi_i, \eta_i) \ell_i(\xi, \eta)$$

The functions $\ell_i(\xi, \eta)$ are the Lagrange polynomials and the question is how to find a set of nodes and the corresponding Lagrange polynomials. There are many possible ways to make this choice [18], [3], [17], [5].

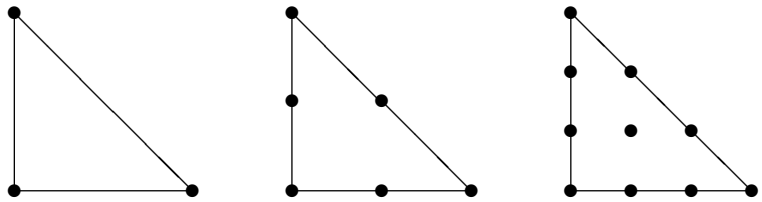


Figure 13.1: Uniform nodes for $\mathbb{P}_k$ Lagrange polynomials on triangle

Suppose we have chosen a set of $N(k)$ nodes. Let us find the Lagrange polynomials corresponding to these nodes. We first express each Lagrange polynomial is terms of the orthogonal polynomials

$$\ell_i(\xi,\eta) = \sum_{j=0}^{N(k)-1} c_{ij}\hat{\varphi}_j(\xi,\eta) \tag{13.2}$$

The coefficients c_{ij} can be found from the interpolation conditions

$$\ell_i(\xi_n,\eta_n) = \delta_{in} = \sum_{j=0}^{N(k)-1} c_{ij}\hat{\varphi}_j(\xi_n,\eta_n), \qquad 0 \le n \le N(k)-1$$

which leads to a matrix problem where we have to invert the *Vandermonde matrix* which is defined as

$$V_{ij} = \hat{\varphi}_j(\xi_i,\eta_i), \qquad 0 \le i,j \le N(k)-1$$

Clearly, for the node set to be admissible, the Vandermonde matrix must be non-singular. Moreover, it would be desirable that the determinant of the Vandermonde matrix is as large as possible or its condition number to be as large as possible. Using the inverse of Vandermonde matrix, the expansion coefficients are given by

$$c_{ij} = (V^{-1})_{ji} \tag{13.3}$$

In general, the evaluation of Lagrange polynomials is performed in terms of their expansion wrt orthogonal polynomials, i.e., using (13.2) and (13.3).

Another important consideration while performing interpolation is the *Lebesgue constant* which is defined as

$$\Lambda = \max_{(\xi,\eta)\in\hat{K}} \sum_{i=0}^{N(k)-1} |\ell_i(\xi,\eta)|$$

The value of this constant depends on the node distribution and affects the stability of the interpolation process to errors in data, e.g., round-off errors. It is desirable to find a node distribution which has a small Lebesgue constant.

Uniform nodes One possibility is to put $k+1$ uniformly spaced points on each unit side of the triangle and form a half tensor product as shown in Figure (13.1). For $k=1$, the nodes can be located at the vertices $(0,0)$, $(1,0)$, $(1,0)$, and the basis functions are

$$\phi_1 = 1-\xi-\eta, \qquad \phi_2 = \xi, \qquad \phi_3 = \eta$$

For $k=2$, the nodes are the three vertices and the mid-points of the three sides.

$$\phi_1 = 2(\tfrac{1}{2}-\xi-\eta)(1-\xi-\eta), \quad \phi_2 = -2\xi(\tfrac{1}{2}-\xi), \quad \phi_3 = -2\eta(\tfrac{1}{2}-\eta)$$

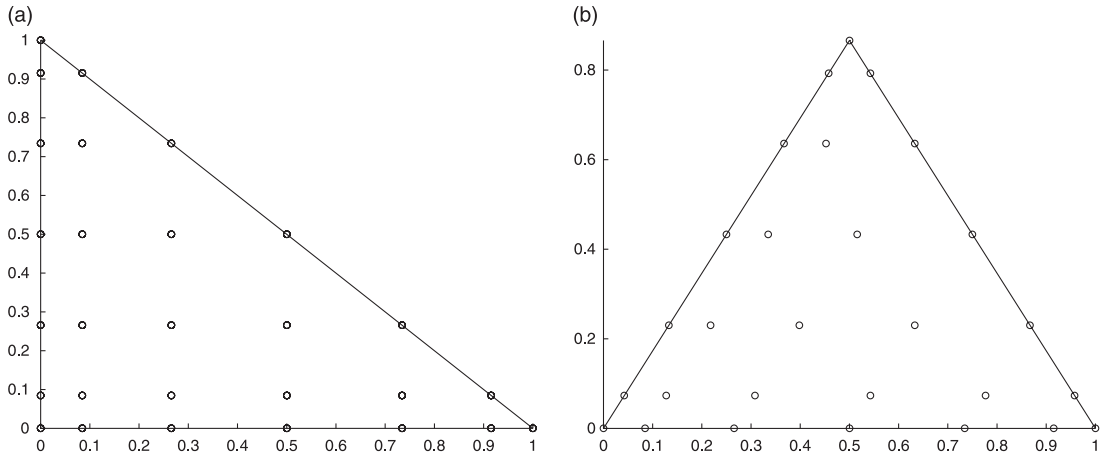


Figure 13.2: GLL nodes for $\mathbb{P}_k$ Lagrange polynomials on triangle (a) $k = 6$ (b) mapped to equilateral triangle (taken from [3])

$$\phi_4 = 4\xi(1-\xi-\eta), \quad \phi_5 = 4\xi\eta, \quad \phi_6 = 4\eta(1-\xi-\eta)$$

For any degree $k \geq 1$, we can find k straight lines on which all points except the i'th point lie; the Lagrange polynomial is the product of the k straight lines together with some constant factor to make it unity at i'th point. While these basis are simple and easy to generate[1], they suffer from the Runge phenomenon especially at high polynomial degrees. Bos [4] showed that they oscillate badly in between the nodes and have a large Lebesgue constant. Hence they should be used only when the degree is small, say $k \leq 3$.

GLL nodes Instead of uniform nodes, we can use GLL nodes on each side of the triangle. The case of $k = 6$ is shown in Figure (13.2a). The basis functions can be constructed following the ideas explained for uniform nodes. If the real triangle is equilateral, then the nodes are not symmetrically placed, as shown in the example in Figure (13.2b), which is a deficiency of this method.

Fekete points

Blyth and Pozridikis In [3], a node set which is derived from a simple modification of GLL nodes is proposed. On the unit sides of the triangle, the nodes are taken as GLL nodes and the interior nodes are obtained by an averaging process. An example is shown in Figure (13.3) for $k = 6$.

Remark 13.1. *On mapped quadrilaterals, the $\mathbb{P}_k$ basis are not complete and hence must not be used. To see this issue, consider the case $k = 1$ and a function in $\mathbb{P}_1$ is of the form*

$$p(\xi,\eta) = a_{00} + a_{10}\xi + a_{01}\eta$$

and the mapping is given by (12.1). If we want to exactly represent the linear function $f(x, y) = x$ by the above polynomial, then we require that

$$x = a_{00} + a_{10}\xi + a_{01}\eta$$

[1] There is no need to use the Vandermonde approach.

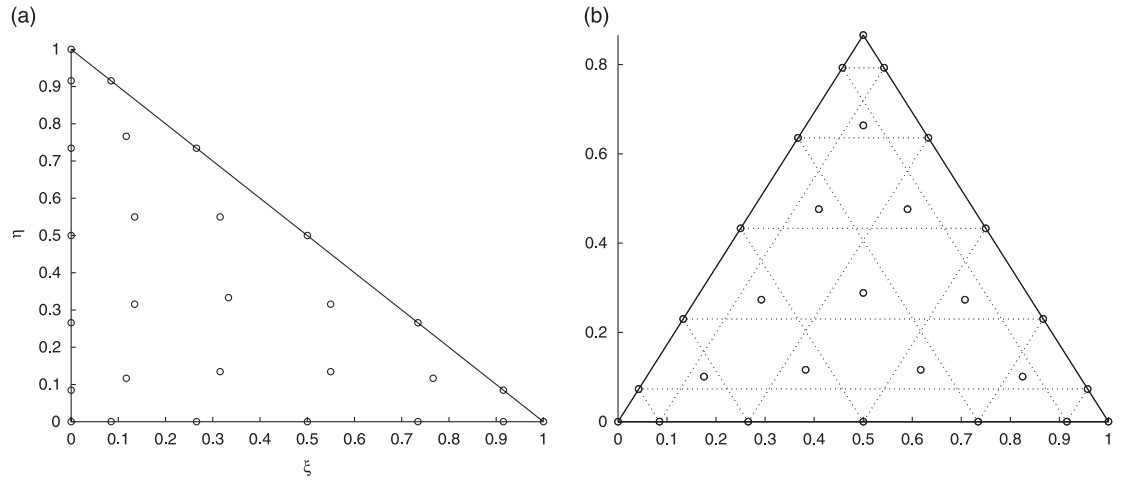


Figure 13.3: Example of Blyth and Pozridikis nodes (a) $k = 6$ (b) mapped to equilateral triangle (taken from [3])

However the map tells us that

$$x = \frac{1}{4}(1-\xi)(1-\eta)x_1 + \frac{1}{4}(1+\xi)(1-\eta)x_2 + \frac{1}{4}(1+\xi)(1+\eta)x_3 + \frac{1}{4}(1-\xi)(1+\eta)x_4$$

and the last two expressions can coincide only if the element is a rectangle. But we can use Taylor basis for $\mathbb{P}_k$ since they do not utilize the mapping for their definition and are complete on any type of polygon.

13.2 Tensor product polynomials $\mathbb{Q}_k$

These are used on quadrilateral elements and are defined in the mapped space.

Constant polynomial: $\mathbb{Q}_0$

$$p(\xi, \eta) = a_{00}$$

Linear polynomial: $\mathbb{Q}_1$

$$p(\xi, \eta) = a_{00} + a_{10}\xi + a_{01}\eta + a_{11}\xi\eta$$

Quadratic polynomial: $\mathbb{Q}_2$

$$p(\xi, \eta) = a_{00} + a_{10}\xi + a_{01}\eta + a_{11}\xi\eta + a_{20}\xi^2 + a_{02}\eta^2 + a_{21}\xi^2\eta + a_{12}\xi\eta^2 + a_{22}\xi^2\eta^2$$

The general form is given by

$$p(\xi, \eta) = \sum_{i=0}^{k}\sum_{j=0}^{k} a_{ij}\xi^i\eta^j$$

and the dimensional of this space is $N(k) = (k+1)^2$. The powers of each coordinate is $\leq k$ but the total power in some terms can exceed k and can be as high as $2k$. In practice, the above monomial basis is not a good choice.

13.2.1 Nodal basis for $\mathbb{Q}_k$

We can use Lagrange polynomials based on a set of $k+1$ distinct nodes in $[-1,+1]$

$$-1 \leq \xi_0 < \xi_1 < \ldots < \xi_k \leq +1$$

and write any function in $\mathbb{Q}_k$ as

$$p(\xi,\eta) = \sum_{i=0}^{k}\sum_{j=0}^{k} a_{ij}\ell_i(\xi)\ell_j(\eta)$$

Note that in this case, the dofs u_{ij} are actually the value of the function at the nodes, i.e.,

$$p(\xi_i,\xi_j) = a_{ij}$$

and hence the name nodal basis.

13.2.2 Modal basis for $\mathbb{Q}_k$

We have introduced Legendre polynomials $P_j(\xi)$ which are mutually orthogonal on $[-1,+1]$ and using these we can define mutually orthonormal functions $\hat{\varphi}_j(\xi) = \sqrt{2j+1}P_j(\xi)$. A function in $\mathbb{P}_k$ can be written as

$$p(\xi,\eta) = \sum_{i=0}^{k}\sum_{j=0}^{k} a_{ij}\hat{\varphi}_i(\xi)\hat{\varphi}_j(\eta)$$

Now the dofs a_{ij} no longer represent nodal values of the function, but are certain moments

$$a_{ij} = \frac{1}{4}\int_{-1}^{+1}\int_{-1}^{+1} p(\xi,\eta)\hat{\varphi}_i(\xi)\hat{\varphi}_j(\eta)\mathrm{d}\xi\mathrm{d}\eta$$

Chapter 14

DG in 2-D

In this Chapter, we introduce the DG scheme for a scalar conservation law in a general and abstract setting, i.e., without explicitly specifying the type of element or basis functions. We will assume that each element is mapped to a reference element $\hat{K}$ and the basis functions are constructed on this reference element with coordinates (ξ, η). The mapping will be denoted by $\vec{F}_K : \hat{K} \to K$ so that $(x, y) = \vec{F}_K(\xi, \eta)$. Let us also denote the Jacobian of this mapping by

$$J_K(\xi, \eta) = \frac{\partial(x, y)}{\partial \xi, \eta)}$$

14.1 Semi-discrete DG scheme

Consider the conservation law

$$\frac{\partial u}{\partial t} + \nabla \cdot \vec{f}(u) = 0$$

and we will approximate the solution inside element K by a polynomial of degree k in terms of some local basis functions

$$u_h^K = \sum_{j=0}^{N-1} u_j^K \phi_j$$

where N is the dimension of the polynomial space. The semi-discrete DG scheme is obtained by multiplying the conservation by the test function ϕ_i and performing an integration by parts on the flux divergence term

$$\int_K \frac{\partial u_h}{\partial t} \phi_i \mathrm{d}x\mathrm{d}y - \int_K \vec{f}(u_h) \cdot \nabla_x \phi_i \mathrm{d}x\mathrm{d}y + \sum_{e \in \partial K} \int_e H(u_h^K, u_h^{K_e}, n_{K_e}) \phi_i \mathrm{d}s = 0$$

where $H(\cdot, \cdot, \cdot)$ is a numerical flux function. The time derivative term can be written as

$$\int_K \frac{\partial u_h}{\partial t} \phi_i \mathrm{d}x\mathrm{d}y = \sum_j \frac{\mathrm{d}u_j^K}{\mathrm{d}t} \int_K \phi_i \phi_j \mathrm{d}x\mathrm{d}y = \sum_j M_{ij}^K \frac{\mathrm{d}u_j^K}{\mathrm{d}t}$$

so that we obtain the semi-discrete scheme as a set of coupled ODE which can be written as

$$M^K \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} u_0^K \\ \vdots \\ u_{N-1}^K \end{bmatrix} = \begin{bmatrix} R_0^K \\ \vdots \\ R_{N-1}^K \end{bmatrix}$$

where M^K is the symmetric mass matrix with entries

$$M_{ij}^K = \int_K \phi_i \phi_j \mathrm{d}x\mathrm{d}y$$

and the right hand side is

$$R_i^K = \int_K \vec{f}(u_h) \cdot \nabla_x \phi_i \mathrm{d}x\mathrm{d}y - \sum_{e \in \partial K} \int_e H(u_h^K, u_h^{K_e}, n_{K_e}) \phi_i \mathrm{d}s$$

14.2 Quadrature and assembly

All the integrals appearing in the DG scheme are computed using numerical quadrature including the components of the mass matrix. Sometimes the mass matrix can be exactly and/or analytically integrated. The quadrature rules are usually defined on a reference interval/element and we are already making use of such a mapping to define the basis functions.

The cell integral can be computed as

$$\begin{aligned}
&\int_K \vec{f}(u_h) \cdot \nabla_x \phi_i \mathrm{d}x\mathrm{d}y \\
=&\int_{\hat{K}} \vec{f}(u_h) \cdot \nabla_x \phi_i \det(J_K) \mathrm{d}\xi\mathrm{d}\eta \\
\approx&\sum_q \vec{f}(u_h(\xi_q, \eta_q)) \cdot \nabla_x \phi_i(\xi_q, \eta_q) \det(J_K(\xi_q, \eta_q)) \omega_q
\end{aligned}$$

The solution at the quadrature points required to compute the flux are obtained by computing the sum over all basis functions

$$u_h(\xi_q, \eta_q) = \sum_{j=0}^{N-1} u_j^K \phi_j(\xi_q, \eta_q)$$

The gradient of the test functions can be computed using the chain rule of differentiation

$$\frac{\partial \phi_i}{\partial x} = \frac{\partial \phi_i}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \phi_i}{\partial \eta}\frac{\partial \eta}{\partial x}, \qquad \frac{\partial \phi_i}{\partial y} = \frac{\partial \phi_i}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \phi_i}{\partial \eta}\frac{\partial \eta}{\partial y}$$

and the derivatives of the coordinates can be obtained from the map $\vec{F}_K$ as explained in (xxx).

The integrals on the faces of cells is a 1-D integral and is computed by mapping each face to a reference interval. This map is just the restriction of $\vec{F}_K$ to a particular face. E.g., for the face $\xi = +1$, the integral is computed as

$$\begin{aligned}
&\int_e H(u_h^K, u_h^{K_e}, n_{K,e}) \phi_i \mathrm{d}s \\
&\int_e H(u_h^K, u_h^{K_e}, n_{K,e}) \phi_i \frac{\mathrm{d}s}{\mathrm{d}\eta} \mathrm{d}\eta \\
\approx&\sum_q H(u_h^K(+1, \eta_q), u_h^{K_e}(\xi(\eta_q), \eta(\eta_q)), n_{K,e}(\eta_q)) \phi_i(+1, \eta_q) \frac{\mathrm{d}s}{\mathrm{d}\eta}(\eta_q) \tilde{\omega}_q
\end{aligned}$$

and the derivative $\frac{\mathrm{d}s}{\mathrm{d}\eta}$ is obtained from the map $\vec{F}_K$, see (12.4.4). The value from the neighbouring cell $u_h^{K_e}$ must be computed taking account of the orientation of the local coordinate system in each cell.

14.3 Assembly using deal.II

Chapter 15

DG in physical space

The construction of basis functions and test functions is usually performed on a reference cell. However it is also possible to do this directly in physical space [15]. This approach can be used on all types of meshes and even on hybrid unstructured grids also. In this Chapter, let us assume that the mesh $\mathcal{T}_h$ is composed of polygonal cells which could even be curved. This type of DG scheme has been developed in [15].

15.1 Basis functions

The approximating space we will use is basically the complete polynomials $\mathbb{P}_k$ for some $k \geq 0$. For any element $K \in \mathcal{T}_h$, let (x_K, y_K) denote the centroid of the element, i.e.,

$$x_K = \frac{1}{|K|} \int_K x \mathrm{d}x\mathrm{d}y, \qquad y_K = \frac{1}{|K|} \int_K y \mathrm{d}x\mathrm{d}y$$

and let h_K denote some length scale of the element. At first order, there is only one basis function

$$\phi_0^K(x, y) = 1$$

At second order we have in addition

$$\phi_1^K(x, y) = \frac{x - x_K}{h_K}, \qquad \phi_2^K(x, y) = \frac{y - y_K}{h_K}$$

At third order, we have in addition

$$\begin{aligned}
\phi_3^K(x, y) &= \frac{1}{2}\left[\left(\frac{x - x_K}{h_K}\right)^2 - m_{20}^K\right] \\
\phi_4^K(x, y) &= \left(\frac{x - x_K}{h_K}\right)\left(\frac{y - y_K}{h_K}\right) - m_{11}^K \\
\phi_5^K(x, y) &= \frac{1}{2}\left[\left(\frac{y - y_K}{h_K}\right)^2 - m_{02}^K\right]
\end{aligned}$$

where

$$m_{ij}^K = \frac{1}{|K|} \int_K \left(\frac{x - x_K}{h_K}\right)^i \left(\frac{y - y_K}{h_K}\right)^j \mathrm{d}x\mathrm{d}y$$

Note that these functions are similar to the terms in a Taylor expansion with some shift applied to them. We can define higher degree functions by following the same methodology.

This shift ensures the following normalization

$$\frac{1}{|K|}\int_K \phi_0^K(x,y)\mathrm{d}x\mathrm{d}y = 1, \qquad \frac{1}{|K|}\int_K \phi_j^K(x,y)\mathrm{d}x\mathrm{d}y = 0, \quad j = 1,2,\ldots$$

Also, as is common in DG methods, these functions are supported only in one element, so that

$$\phi_j^K(x,y) = 0 \qquad \text{if } (x,y) \notin K$$

15.2 DG scheme

Let us consider a scalar conservation law of the form

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0$$

though the extension to systems is trivial and is performed component by component. The solution $u_h \in \mathbb{P}_k$ inside element K is of the form

$$u_h(x,y,t) = \sum_{j=0}^{N-1} u_j^K(t)\phi_j^K(x,y), \qquad (x,y) \in K$$

where the number of degrees of freedom is

$$N = \frac{1}{2}(k+1)(k+2)$$

Due to the way we have defined the basis functions, the cell average of this function is

$$\frac{1}{|K|}\int_K u_h(x,y)\mathrm{d}x\mathrm{d}y = \sum_{j=0}^{N-1} u_j^K \frac{1}{|K|}\int_K \phi_j^K(x,y)\mathrm{d}x\mathrm{d}y = u_0^K$$

Multiply the conservation law by the test function ϕ_i^K and perform integration by parts in the flux divergence term to obtain

$$\int_K \frac{\partial u_h}{\partial t}\phi_i^K \mathrm{d}x\mathrm{d}y - \int_K \left[f(u_h)\frac{\partial \phi_i^K}{\partial x} + g(u_h)\frac{\partial \phi_i^K}{\partial y}\right]\mathrm{d}x\mathrm{d}y + \sum_{e\in\partial K}\int_e H(u^K, u^{K_e}, n_{K_e})\phi_i^K \mathrm{d}s = 0, \qquad 0 \le i \le N-1$$

where $H(\cdot,\cdot,\cdot)$ is a numerical flux function. For $i = 0$, the test function is constant and the time derivative term can be written as

$$\int_K \frac{\partial u_h}{\partial t}\phi_0^K \mathrm{d}x\mathrm{d}y = |K|\frac{\mathrm{d}u_0^K}{\mathrm{d}t}$$

so that the equation for the first dof takes the form

$$|K|\frac{\mathrm{d}u_0^K}{\mathrm{d}t} + \sum_{e\in\partial K}\int_e H(u^K, u^{K_e}, n_{K_e})\mathrm{d}s = 0$$

This looks like a finite volume scheme and shows the conservation property of the DG scheme. For $i \ge 1$, we have

$$\int_K \frac{\partial u_h}{\partial t}\phi_i^K \mathrm{d}x\mathrm{d}y = \sum_{j=0}^{N-1}\frac{\mathrm{d}u_j^K}{\mathrm{d}t}\int_K \phi_i^K\phi_j^K \mathrm{d}x\mathrm{d}y = \sum_{j=1}^{N-1}\frac{\mathrm{d}u_j^K}{\mathrm{d}t}M_{ij}^K$$

where

$$M_{ij}^K = \int_K \phi_i^K \phi_j^K \mathrm{d}x\mathrm{d}y, \qquad 1 \leq i,j \leq N-1$$

This matrix is not diagonal which means that the modes $i \geq 1$ are coupled and we have to invert the mass matrix to evolve these modes. The set of semi-discrete equations for the dofs can be written as

$$|K|\frac{\mathrm{d}u_0^K}{\mathrm{d}t} = R_0^K := -\sum_{e\in\partial K}\int_e H(u^K, u^{K_e}, n_{K_e})\mathrm{d}s$$

and

$$M^K \frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} u_1^K \\ \vdots \\ u_{N-1}^K \end{bmatrix} = \begin{bmatrix} R_1^K \\ \vdots \\ R_{N-1}^K \end{bmatrix}$$

where

$$R_i^K = \int_K \left[f(u_h)\frac{\partial \phi_i^K}{\partial x} + g(u_h)\frac{\partial \phi_i^K}{\partial y}\right]\mathrm{d}x\mathrm{d}y - \sum_{e\in\partial K}\int_e H(u^K, u^{K_e}, n_{K_e})\phi_i^K \mathrm{d}s$$

15.3 Quadrature and assembly

We have integrals defined on the cells and on the faces, and these are computed using some numerical quadrature. To perform the quadrature, we have to map the cell K to a reference cell $\hat{K}$

$$(x,y) = \vec{F}_K(\xi,\eta)$$

since quadrature rules are usually developed for reference cells. Let (x_q^K, y_q^K) denote a set of quadrature points which have been mapped back to physical space

$$(x_q^K, y_q^K) = \vec{F}_K(\xi_q, \eta_q)$$

Define the Jacobian of the mapping by

$$J_K(\xi,\eta) = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}$$

The elements of the mass matrix M^K can be computed as

$$\begin{aligned} M_{ij}^K &= \int_{\hat{K}} \phi_i^K(x,y)\phi_j^K(x,y)\det(J_K(\xi,\eta))\mathrm{d}\xi\mathrm{d}\eta \\ &\approx \sum_q \phi_i^K(x_q^K, y_q^K)\phi_j(x_q^K, y_q^K)\det(J_K(\xi_q,\eta_q))\omega_q \end{aligned}$$

These quadratures are exact only on linear triangles. The mass matrix depends on the element K and they can computed once and stored for later use. For explicit time stepping, the inverse of the mass matrix can be computed once and stored. Moreover since this is symmetric, only half of it needs to be stored.

The shape functions are defined directly in physical space and hence $\frac{\partial \phi_i^K}{\partial x}$, $\frac{\partial \phi_i^K}{\partial y}$ are easy to compute and do not require the use of the mapping function $\vec{F}_K$. Performing a

quadrature, the cell integral term is given by

$$\begin{aligned}
&\int_K \left[f(u_h)\frac{\partial \phi_i^K}{\partial x} + g(u_h)\frac{\partial \phi_i^K}{\partial y} \right] \mathrm{d}x\mathrm{d}y \\
\approx \quad &\sum_q \left[f(u_h(x_q^K, y_q^K))\frac{\partial \phi_i^K}{\partial x}(x_q^K, y_q^K) + g(u_h(x_q^K, y_q^K))\frac{\partial \phi_i^K}{\partial y}(x_q^K, y_q^K) \right] \det(J_K(\xi_q, \eta_q))\omega_q
\end{aligned}$$

The solution at the quadrature points is evaluated from the expansion

$$u_h(x_q^K, y_q^K) = \sum_{j=0}^{K} u_j^K \phi_j^K(x_q^K, y_q^K)$$

and these are used to compute the fluxes required in the cell integral.

To compute the flux integral on the faces, we map each face to a reference interval $[0, 1]$ or $[-1, +1]$ and then perform numerical quadrature. This mapping is of course just the restriction of the mapping $\vec{F}_K$ to a particular face of K. Denoting the mapped coordinate on the face by s, the face integral is computed as

$$\begin{aligned}
&\sum_{e \in \partial K} \int_e H(u^K, u^{K_e}, n_{K_e})\phi_i^K \mathrm{d}s \\
\approx \quad &\sum_q H(u^K(x_q^e, y_q^e), u^{K_e}(x_q^e, y_q^e), n_{K_e})\phi_i^K(x_q^e, y_q^e)\tilde{\omega}_q
\end{aligned}$$

The assembly of the residual and one stage of an RK scheme can be accomplished by the following steps.

1. Set $R = 0$.
2. Loop over faces
   - assemble flux integral into R for cells adjacent to each face.
3. Loop over cells
   - assemble cell integral into R.
   - Multiply by inverse mass matrix.
4. Update solution to next RK stage.

Remark 15.1. *It is possible to take the shape functions defined in this Chapter and apply a Gram-Schmidt orthogonalization process to generate a set of mutually orthogonal shape functions which are still defined in physical space. Such an approach has been used in (xxx).*

Chapter 16

Nodal DG on quadrilaterals

In this Chapter, we will consider meshes composed of curvilinear quadrilateral elements. The solution on quadrilaterals can be approximated by tensor product of Lagrange polynomials which are defined using Gauss-Legendre or Gauss-Lobatto-Legendre nodes. Let us denote the $k+1$ nodes by

$$-1 \leq \xi_0 < \xi_1 < \ldots < \xi_k \leq +1$$

and the Lagrange polynomials are given by

$$\ell_i(\xi) = \prod_{j=0, j\neq i}^{k} \frac{\xi - \xi_j}{\xi_i - \xi_j}$$

The Lagrange polynomials have the interpolation property such that

$$\ell_i(\xi_j) = \delta_{ij}, \qquad 0 \leq i, j \leq k$$

Let each cell be mapped to a reference cell $\hat{K}$ by the map $\vec{F}_K : \hat{K} \to K$ so that $(x, y) = \vec{F}_K(\xi, \eta)$. The solution is a piecewise polynomial in $\mathbb{Q}_k$ and in each element K is given by an expression of the form

$$u_h = \sum_{r=0}^{k} \sum_{s=0}^{k} u_{rs}^K \ell_r(\xi) \ell_s(\eta)$$

Consider the conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0$$

To construct the DG scheme, we can map the PDE to the reference cell coordinates, but this is not strictly necessary. Let us proceed without explicitly mapping the PDE but we will in the end obtain the same scheme as we would with mapped PDE.

16.1 DG scheme

Let us multiply the PDE by the test function $\ell_i(\xi)\ell_j(\eta)$ and integrate over cell K

$$\int_K \left(\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} \right) \ell_i(\xi) \ell_j(\eta) \mathrm{d}x\mathrm{d}y = 0$$

Performing an integration by parts on the flux divergence term, we get

$$\int_K \frac{\partial u_h}{\partial t}\ell_i(\xi)\ell_j(\eta)\mathrm{d}x\mathrm{d}y - \int_K \left[f(u_h)\frac{\partial}{\partial x}[\ell_i(\xi)\ell_j(\eta)] + g(u_h)\frac{\partial}{\partial y}[\ell_i(\xi)\ell_j(\eta)\right]\mathrm{d}x\mathrm{d}y + \sum_{e\in\partial K}\int_e H(u_h^K, u_h^{K_e}, n_{K,e})\ell_i(\xi)\ell_j(\eta)\mathrm{d}s = 0$$

We will use the same nodes to perform quadrature for the cell integrals as are used to define the Lagrange polynomials.

Time derivative The time derivative term is

$$\begin{aligned}
\int_K \frac{\partial u_h}{\partial t}\ell_i(\xi)\ell_j(\eta)\mathrm{d}x\mathrm{d}y &= \int_{\hat{K}} \frac{\partial u_h}{\partial t}\ell_i(\xi)\ell_j(\eta)\det J_K \mathrm{d}\xi\mathrm{d}\eta \\
&\approx \sum_{p=0}^{k}\sum_{q=0}^{k} \frac{\partial u_h}{\partial t}(\xi_p,\xi_q)\ell_i(\xi_p)\ell_j(\xi_q)\det J_K(\xi_p,\xi_q)\omega_p\omega_q \\
&\approx \sum_{p=0}^{k}\sum_{q=0}^{k} \frac{\partial u_h}{\partial t}(\xi_p,\xi_q)\delta_{ip}\delta_{jq}\det J_K(\xi_p,\xi_q)\omega_p\omega_q \\
&= \frac{\partial u_h}{\partial t}(\xi_i,\xi_j)\det J_K(\xi_i,\xi_j)\omega_i\omega_j
\end{aligned}$$

But

$$\frac{\partial u_h}{\partial t}(\xi_i,\xi_j) = \sum_{r=0}^{k}\sum_{s=0}^{k}\frac{\mathrm{d}u_{rs}^K}{\mathrm{d}t}\ell_r(\xi_i)\ell_s(\xi_j) = \sum_{r=0}^{k}\sum_{s=0}^{k} u_{rs}^K\delta_{ri}\delta_{sj} = \frac{\mathrm{d}u_{ij}^K}{\mathrm{d}t}$$

and hence

$$\int_K \frac{\partial u_h}{\partial t}\ell_i(\xi)\ell_j(\eta)\mathrm{d}x\mathrm{d}y \approx \frac{\mathrm{d}u_{ij}^K}{\mathrm{d}t}\det J_K(\xi_i,\xi_j)\omega_i\omega_j$$

This quadrature is exact if $\det J_K$ is constant in the element, which is the case for Cartesian grids and linear triangles.

Cell integral Now let us consider the second cell integral term involving the fluxes. We first transform the derivatives by applying chain rule of differentiation,

$$\begin{aligned}
\frac{\partial}{\partial x}[\ell_i(\xi)\ell_j(\eta)] &= \frac{\partial}{\partial \xi}[\ell_i(\xi)\ell_j(\eta)]\frac{\partial \xi}{\partial x} + \frac{\partial}{\partial \eta}[\ell_i(\xi)\ell_j(\eta)]\frac{\partial \eta}{\partial x} \\
&= \ell_i'(\xi)\ell_j(\eta)\frac{\partial \xi}{\partial x} + \ell_i(\xi)\ell_j'(\eta)\frac{\partial \eta}{\partial x}
\end{aligned}$$

and similarly

$$\begin{aligned}
\frac{\partial}{\partial y}[\ell_i(\xi)\ell_j(\eta)] &= \frac{\partial}{\partial \xi}[\ell_i(\xi)\ell_j(\eta)]\frac{\partial \xi}{\partial y} + \frac{\partial}{\partial \eta}[\ell_i(\xi)\ell_j(\eta)]\frac{\partial \eta}{\partial y} \\
&= \ell_i'(\xi)\ell_j(\eta)\frac{\partial \xi}{\partial y} + \ell_i(\xi)\ell_j'(\eta)\frac{\partial \eta}{\partial y}
\end{aligned}$$

Using the above two relations, the flux integral can be written as

$$\begin{aligned}
&\int_K \left[f(u_h)\frac{\partial}{\partial x}[\ell_i(\xi)\ell_j(\eta)] + g(u_h)\frac{\partial}{\partial y}[\ell_i(\xi)\ell_j(\eta)\right]\mathrm{d}x\mathrm{d}y \\
= &\int_{\hat{K}} \left[f(u_h)\frac{\partial \xi}{\partial x} + g(u_h)\frac{\partial \xi}{\partial y}\right]\ell_i'(\xi)\ell_j(\eta)\det J_K(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta \\
&+ \int_{\hat{K}} \left[f(u_h)\frac{\partial \eta}{\partial x} + g(u_h)\frac{\partial \eta}{\partial y}\right]\ell_i(\xi)\ell_j'(\eta)\det J_K(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta
\end{aligned}$$

Define the transformed fluxes

$$\begin{aligned}\tilde{f}(\xi,\eta) &= \left[f(u_h(\xi,\eta))\frac{\partial \xi}{\partial x} + g(u_h(\xi,\eta))\frac{\partial \xi}{\partial y}\right]\det J_K \\ \tilde{g}(\xi,\eta) &= \left[f(u_h(\xi,\eta))\frac{\partial \eta}{\partial x} + g(u_h(\xi,\eta))\frac{\partial \eta}{\partial y}\right]\det J_K\end{aligned}$$

We now apply numerical quadrature using the solution nodes themselves which leads to

$$\begin{aligned}&\int_{\hat{K}} [\tilde{f}(\xi,\eta)\ell_i'(\xi)\ell_j(\eta) + \tilde{g}(\xi,\eta)\ell_i(\xi)\ell_j'(\eta)]\mathrm{d}\xi\mathrm{d}\eta \\ \approx\ & \sum_{p=0}^{k}\sum_{q=0}^{k}[\tilde{f}(\xi_p,\xi_q)\ell_i'(\xi_p)\ell_j(\xi_q) + \tilde{g}(\xi_p,\xi_q)\ell_i(\xi_p)\ell_j'(\xi_q)]\omega_p\omega_q \\ =\ & \sum_{p=0}^{k}\sum_{q=0}^{k}[\tilde{f}(\xi_p,\xi_q)\ell_i'(\xi_p)\delta_{jq} + \tilde{g}(\xi_p,\xi_q)\delta_{ip}\ell_j'(\xi_q)]\omega_p\omega_q \\ =\ & \omega_j\sum_{p=0}^{k}\tilde{f}(\xi_p,\xi_j)\ell_i'(\xi_p)\omega_p + \omega_i\sum_{q=0}^{k}\tilde{g}(\xi_i,\xi_q)\ell_j'(\xi_q)\omega_q\end{aligned}$$

We only need the fluxes at the nodes where the solution is already available in the dofs, so that

$$\begin{aligned}\tilde{f}_{ij} &= \tilde{f}(\xi_i,\xi_j) = \left[f(u_{ij})\frac{\partial \xi}{\partial x}(\xi_i,\xi_j) + g(u_{ij})\frac{\partial \xi}{\partial y}(\xi_i,\xi_j)\right]\det J_K(\xi_i,\xi_j) \\ \tilde{g}_{ij} &= \tilde{g}(\xi_i,\xi_j) = \left[f(u_{ij})\frac{\partial \eta}{\partial x}(\xi_i,\xi_j) + g(u_{ij})\frac{\partial \eta}{\partial y}(\xi_i,\xi_j)\right]\det J_K(\xi_i,\xi_j)\end{aligned}$$

and the computation of the coordinate derivatives is explained in Section (xxx). For example

$$\frac{\partial \xi}{\partial x}\det J_K = \frac{\partial y}{\partial \eta}$$

and the determinant does not appear in the expression on the right.

Face integral Consider the face $\xi = +1$ on which we need the integral of the numerical flux. We will compute this 1-D integral using the same nodes used to define the Lagrange polynomials. Hence

$$\begin{aligned}&\int_e H(u_h^K(+1,\eta), u_h^{K_e}(\cdot,\cdot), n_{K,e})\ell_i(+1)\ell_j(\eta)\frac{\mathrm{d}s}{\mathrm{d}\eta}\mathrm{d}\eta \\ \approx\ & \sum_{q=0}^{k} H(u_h^K(+1,\xi_q), u_h^{K_e}(\cdot,\cdot), n_{K,e}(\xi_q))\ell_i(+1)\ell_j(\xi_q)\frac{\mathrm{d}s}{\mathrm{d}\eta}(+1,\xi_q)\omega_q \\ =\ & \sum_{q=0}^{k} H_K(+1,\xi_q)\ell_i(+1)\delta_{jq}\frac{\mathrm{d}s}{\mathrm{d}\eta}(+1,\xi_q)\omega_q \\ =\ & H_K(+1,\xi_j)\ell_i(+1)\frac{\mathrm{d}s}{\mathrm{d}\eta}(+1,\xi_j)\omega_j\end{aligned}$$

The solution from neighbouring cell $u_h^{K_e}$ has to obtained accounting for the orientation of the local coordinate system in that cell. We have introduced the notation $H_K(+1,\xi_j)$ to

denote the numerical flux which is directed out of the element K and is obtained from a Riemann solver. To compute the numerical flux, we require the state the two states at the face. In case of GLL nodes, these states are directly available since GLL nodes are located at the boundaries of cells. In case GL nodes, we have evaluate the solution at the face by summing over the basis functions. For example at $(+1, \xi_j)$

$$u_h(+1,\xi_j) = \sum_r \sum_s u_{rs}^K \ell_r(+1)\ell_s(\xi_j) = \sum_r \sum_s u_{rs}^K \ell_r(+1)\delta_{sj} = \sum_{r=0}^{k} u_{rj}^K \ell_r(+1)$$

and we only need to compute a 1-D sum[1] which can be implemented as a dot product. Need figure

Putting it together By combining the three approximation derived above, we obtain the semi-discrete nodal DG scheme as

$$\begin{aligned}\frac{\mathrm{d}u_{ij}^K}{\mathrm{d}t} \det J_K(\xi_i,\xi_j)\omega_i\omega_j - \omega_j \sum_{p=0}^{k} \tilde{f}_{pj}\ell_i'(\xi_p)\omega_p + \omega_i \sum_{q=0}^{k} \tilde{g}_{iq}\ell_j'(\xi_q)\omega_q \\ + H_K(-1,\xi_j)\ell_i(-1)\frac{\mathrm{d}s}{\mathrm{d}\eta}(-1,\xi_j)\omega_j \\ + H_K(+1,\xi_j)\ell_i(+1)\frac{\mathrm{d}s}{\mathrm{d}\eta}(+1,\xi_j)\omega_j \\ + H_K(\xi_i,-1)\ell_j(-1)\frac{\mathrm{d}s}{\mathrm{d}\xi}(\xi_i,-1)\omega_i \\ + H_K(\xi_i,+1)\ell_j(+1)\frac{\mathrm{d}s}{\mathrm{d}\xi}(\xi_i,+1)\omega_i = 0\end{aligned}$$

The boundary terms in the above equation have been written in terms of the local coordinate system in element K. The sums can be evaluated as dot products or as matrix-vector products. Define the stiffness matrix

$$S_{ij} = \ell_i'(\xi_j)\omega_j, \qquad 0 \le i,j \le k$$

then the sums can be computed via matrix-vector products as follows

$$\sum_{p=0}^{k} \tilde{f}_{pj}\ell_i'(\xi_p)\omega_p = [S\tilde{f}_{:,j}]_i, \qquad \sum_{q=0}^{k} \tilde{g}_{iq}\ell_j'(\xi_q)\omega_q = [S\tilde{g}_{i,:}]_j$$

[1] This is true even in 3-D which saves a lot of zero calculations !!!

Chapter 17

DG on Cartesian grids

In these chapter we give a comprehensive description of discontinuous Galerkin method for 2-d Euler equations on Cartesian grids. TVB limiter and positivity limiter are also discussed.

17.1 Introduction

Consider the system of hyperbolic conservation laws

$$\frac{\partial \boldsymbol{u}}{\partial t} + \sum_{\alpha=1}^{d} \frac{\partial \boldsymbol{f}_\alpha}{\partial x_\alpha} = 0$$

where $\boldsymbol{u} \in \mathcal{U}_{\text{ad}} \subset \mathbb{R}^m$ are the conserved variables and $\boldsymbol{f}_\alpha : \mathcal{U}_{\text{ad}} \to \mathbb{R}^m, \quad \alpha = 1, \ldots, d$ are the Cartesian components of the flux. For the Euler equations in 2-d we have $m = 4$ and the physically admissible set of states is given by

$$\mathcal{U}_{\text{ad}} = \left\{ \boldsymbol{u} \in \mathbb{R}^4 : w_1 > 0, \quad w_4 - \frac{w_2^2 + w_3^2}{2w_1} > 0 \right\}$$

where

$$\boldsymbol{u} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \qquad E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2)$$

and the flux is given by

$$\boldsymbol{f}_1 = \begin{bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ (E + p)u \end{bmatrix}, \qquad \boldsymbol{f}_2 = \begin{bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ (E + p)v \end{bmatrix}$$

17.2 Basis functions and solution representation

Let $\mathcal{T}_h$ be a triangulation of the domain Ω into disjoint rectangular cells. Let $\mathbb{P}_k$ denote the space of polynomials on $[-1, +1] \times [-1, +1]$ of degree at most k. We will form a basis

for this space using scaled Legendre polynomials $\tilde{P}_n$ as follows (see appendix (17.11))

$$\{\phi_j(\xi,\eta)\}_{j=1}^{N(k)} = \begin{cases} \tilde{P}_0(\xi)\tilde{P}_0(\eta), & \tilde{P}_1(\xi)\tilde{P}_0(\eta), \quad \ldots, & \tilde{P}_{k-1}(\xi)\tilde{P}_0(\eta), & \tilde{P}_k(\xi)\tilde{P}_0(\eta) \\ \tilde{P}_0(\xi)\tilde{P}_1(\eta), & \tilde{P}_1(\xi)\tilde{P}_1(\eta), \quad \ldots, & \tilde{P}_{k-1}(\xi)\tilde{P}_1(\eta) & \\ \vdots & & & \\ \tilde{P}_0(\xi)\tilde{P}_{k-1}(\eta), & \tilde{P}_1(\xi)\tilde{P}_{k-1}(\eta) & & \\ \tilde{P}_0(\xi)\tilde{P}_k(\eta) & & & \end{cases}$$

where

$$N(k) = \frac{1}{2}(k+1)(k+2)$$

Consider cell $K \in \mathcal{T}_h$ whose size is $\Delta x_K, \Delta y_K$ and center (x_K, y_K); the i'th component of the solution $\boldsymbol{u}$ is represented inside K as

$$\vec{x} = (x,y) \in K: \qquad w_i(x,y,t) = \sum_{j=1}^{N(k)} w_{i,j}^K(t)\phi_j^K(x,y)$$

where

$$\phi_j^K(x,y) = \phi_j(\xi,\eta), \qquad \xi = \frac{x - x_K}{\frac{1}{2}\Delta x_K}, \quad \eta = \frac{y - y_K}{\frac{1}{2}\Delta y_K}$$

The basis functions ϕ_j^K are also orthogonal

$$\int_K \phi_j^K \phi_l^K \mathrm{d}\vec{x} = |K|\delta_{jl} = \Delta x_K \Delta y_K \delta_{jl}$$

where $\mathrm{d}\vec{x} = \mathrm{d}x\mathrm{d}y$ is the 2-d measure. We will sometimes also write the solution in terms of the scaled variables

$$(x,y) \in K: \qquad w_i(\xi,\eta,t) = \sum_{j=1}^{N(k)} w_{i,j}^K(t)\phi_j(\xi,\eta)$$

Note that since $\phi_1(\xi,\eta) = \tilde{P}_0(\xi)\tilde{P}_0(\eta) \equiv 1$ then $w_{i,1}^K$ is the cell average value of w_i on cell K

$$w_{i,1}^K = \frac{1}{|K|}\int_K w_i \mathrm{d}\vec{x}$$

17.3 DG formulation

To derive the DG scheme on cell K we multiply the i'th conservation law by the basis function ϕ_l^K

$$\int_K \left(\frac{\partial w_i}{\partial t} + \sum_\alpha \frac{\partial f_{\alpha,i}}{\partial x_\alpha} \right) \phi_l^K \mathrm{d}\vec{x} = 0$$

Integrate by parts on the flux divergence term

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_K w_i \phi_l^K \mathrm{d}\vec{x} - \sum_\alpha \int_K f_{\alpha,i} \frac{\partial \phi_l^K}{\partial x_\alpha} \mathrm{d}\vec{x} + \int_{\partial K} \sum_\alpha f_{\alpha,i} n_\alpha \phi_l^K \mathrm{d}s = 0$$

Since the solution is discontinuous across ∂K, we introduce a numerical flux function $\hat{\boldsymbol{f}}(\boldsymbol{u}^-, \boldsymbol{u}^+, n)$ for the interface flux leading to the semi-discrete DG scheme

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_K w_i \phi_l^K \mathrm{d}\vec{x} - \sum_\alpha \int_K f_{\alpha,i} \frac{\partial \phi_l^K}{\partial x_\alpha} \mathrm{d}\vec{x} + \int_{\partial K} \hat{f}_i \phi_l^K \mathrm{d}s = 0 \tag{17.1}$$

In order to derive equations more suitable for numerical implementation, we substitute the solution representation in terms of basis functions to get

$$\sum_{j=1}^{N(k)} \frac{\mathrm{d}w_{i,j}^K}{\mathrm{d}t} \int_K \phi_j^K \phi_l^K \mathrm{d}\vec{x} - \sum_\alpha \int_K f_{\alpha,i} \frac{\partial \phi_l^K}{\partial x_\alpha} \mathrm{d}\vec{x} + \int_{\partial K} \hat{f}_i \phi_l^K \mathrm{d}s = 0$$

Finally, using the orthogonality of the basis functions

$$|K| \frac{\mathrm{d}w_{i,l}^K}{\mathrm{d}t} - \sum_\alpha \int_K f_{\alpha,i} \frac{\partial \phi_l^K}{\partial x_\alpha} \mathrm{d}\vec{x} + \int_{\partial K} \hat{f}_i \phi_l^K \mathrm{d}s = 0$$

To approximate the integrals we transform the cell and edge to the reference domains of $[-1,+1]^2$ and $[-1,+1]$ respectively. For example,

$$\begin{aligned} \int_K f_{1,i} \frac{\partial \phi_l^K}{\partial x_1} \mathrm{d}\vec{x} &= \int_{[-1,+1]^2} f_{1,i} \left(\frac{2}{\Delta x_K} \frac{\partial \phi_l}{\partial \xi} \right) \left(\frac{1}{4} \Delta x_K \Delta y_K \mathrm{d}\vec{\xi} \right) \\ &= \frac{1}{2} \Delta y_K \int_{[-1,+1]^2} f_{1,i} \frac{\partial \phi_l}{\partial \xi} \mathrm{d}\vec{\xi} \end{aligned}$$

The cell integrals are approximated by tensor product of Gauss quadrature rule with $k+1$ points[1] while the edge integral is approximated with a Gauss rule of $k+1$ points (see Fig. 17.1), leading to

$$\begin{aligned} |K| \frac{\mathrm{d}w_{i,l}^K}{\mathrm{d}t} &- \frac{1}{2} \Delta y_K \sum_{q=1}^{M_k} f_{1,i}(\boldsymbol{u}(\vec{\xi}_q, t)) \frac{\partial \phi_l}{\partial \xi}(\vec{\xi}_q) \tilde{\omega}_q^{k+1} \\ &- \frac{1}{2} \Delta x_K \sum_{q=1}^{M_k} f_{2,i}(\boldsymbol{u}(\vec{\xi}_q, t)) \frac{\partial \phi_l}{\partial \eta}(\vec{\xi}_q) \tilde{\omega}_q^{k+1} \\ &+ \frac{1}{2} \sum_{e \in \partial K} |e| \sum_{q=1}^{k+1} \hat{f}_i(\vec{\xi}_q^e, t) \phi_l(\vec{\xi}_q^e) \omega_q^{k+1} = 0, \quad i = 1, \ldots, 4, \quad l = 1, \ldots, N(k) \end{aligned}$$

Define the vector of degrees of freedom associated to cell K by

$$w^K = [w_{1,1}^K, \ldots, w_{1,N(k)}^K, \ldots, w_{4,1}^K, \ldots, w_{4,N(k)}^K]^\top$$

Then we have an ODE of the form

$$\frac{\mathrm{d}w^K}{\mathrm{d}t} + R_K(w) = 0$$

which will be solved by a Runge-Kutta method.

[1] i.e., we have a total of $M_k = (k+1)^2$ quadrature nodes denoted by $\vec{\xi}_q = \vec{\xi}_q^{k+1} = (\xi_r^{k+1}, \xi_s^{k+1}), q = 1, \ldots, M_k$, see appendix (17.12)

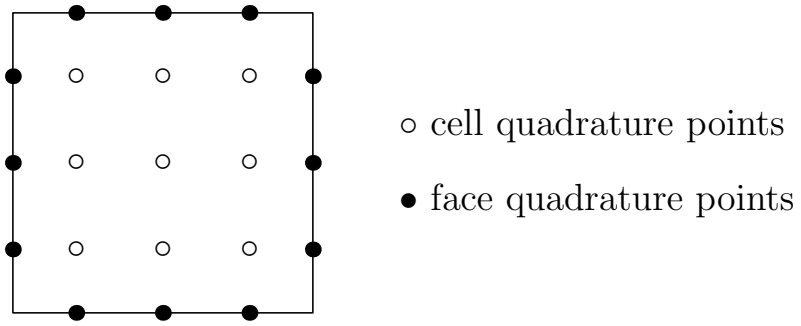


Figure 17.1: Gauss quadrature points for degree $k = 2$

Remarks

1. For $l = 1$, we have $\phi_1 \equiv 1$ and we obtain the equation for the cell average value

$$|K|\frac{\mathrm{d}w_{i,1}^K}{\mathrm{d}t} + \frac{1}{2}\sum_{e\in\partial K}|e|\sum_{q=1}^{k+1}\hat{f}_i(\vec{\xi}_q^e, t)\phi_1(\vec{\xi}_q^e)\omega_q^{k+1} = 0$$

2. If the cell face lies on the boundary of the domain, then the numerical flux $\hat{\boldsymbol{f}}$ must account for the boundary condition.

3. There are a large number of numerical flux functions that can be used. For the Lax-Friedrich's flux, see appendix (17.13).

4. To perform the quadature, we need to evaluate the solution $\boldsymbol{u}$ at the quadrature points $\vec{\xi}_q = (\xi_r, \eta_s)$, see appendix (17.12)

$$w_i(\vec{\xi}_q, t) = \sum_{j=1}^{N(k)} w_{i,j}(t)\phi_j(\vec{\xi}_q)$$

   The set of numbers

$$\phi_j(\vec{\xi}_q), \quad j = 1, \ldots, N(k), \quad q = 1, \ldots, M_k$$

   do not depend on the element K; hence they can be computed and stored in a common location that can be used by all elements.

5. Each basis function ϕ_l is of the form

$$\phi_l(\vec{\xi}) = P_m(\xi)P_n(\eta) \quad \text{for some } m, n \in \{0, 1, \ldots, k\}$$

   and hence

$$\frac{\partial}{\partial\xi}\phi_l(\vec{\xi}) = P_m'(\xi)P_n(\eta), \qquad \frac{\partial}{\partial\eta}\phi_l(\vec{\xi}) = P_m(\xi)P_n'(\eta)$$

   The set of numbers

$$\frac{\partial}{\partial\xi}\phi_l(\vec{\xi}_q), \quad \frac{\partial}{\partial\eta}\phi_l(\vec{\xi}_q), \quad l = 1, \ldots, N(k), \quad q = 1, \ldots, M_k$$

   do not depend on the element and can be computed and stored once.

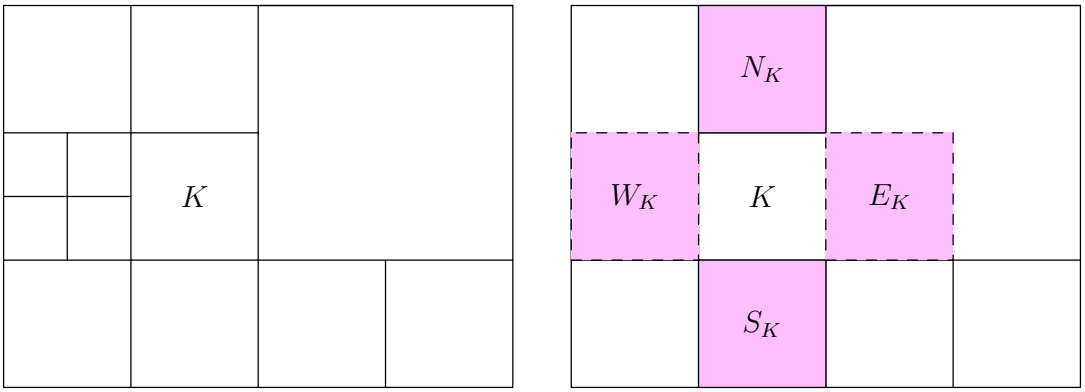


Figure 17.2: Definition of neighbours for TVD limiter

17.4 Limiting

High order schemes can generate oscillations in the solutions which are physically incorrect. Limiting tries to eliminate such oscillations. Let $\boldsymbol{u}$ be the solution obtained after applying the RK scheme. The i'th component of solution inside cell K is of the form

$$(x,y) \in K : \qquad w_i = w_{i,1}^K + \underbrace{w_{i,2}^K \phi_2^K + w_{i,3}^K \phi_3^K}_{\text{linear terms}} + \text{higher order terms}$$

The quantities $w_{i,2}^K$, $w_{i,3}^K$ are proportional to the derivatives along x and y directions respectively. Let E_K, W_K, N_K, S_K denote the cells to the east, west, north and south of cell K, see Fig. 17.2. We first compute the limited derivatives using minmod function[2]

$$\tilde{w}_{i,2}^K = \frac{1}{\sqrt{3}} \text{minmod}\left(\sqrt{3} w_{i,2}^K, \beta(w_{i,1}^K - w_{i,1}^{W_K}), \beta(w_{i,1}^{E_K} - w_{i,1}^K)\right)$$

$$\tilde{w}_{i,3}^K = \frac{1}{\sqrt{3}} \text{minmod}\left(\sqrt{3} w_{i,3}^K, \beta(w_{i,1}^K - w_{i,1}^{S_K}), \beta(w_{i,1}^{N_K} - w_{i,1}^K)\right)$$

where $\beta \in [1/2, 1]$ and

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

If $\tilde{w}_{i,2}^K = w_{i,2}^K$ and $\tilde{w}_{i,3}^K = w_{i,3}^K$ then the limited solution is same as the original solution

$$\Lambda\Pi_h(w_i) = w_i$$

else the limited solution is an affine solution of the form

$$\Lambda\Pi_h(w_i) = w_{i,1}^K + \tilde{w}_{i,2}^K \phi_2^K + \tilde{w}_{i,3}^K \phi_3^K$$

Note that if the limiter modifies the solution, then it preserves the cell average value. What we have described above is sometimes refered to as the *component-wise limiter* since we limit each component w_i, $i = 1, \ldots, 4$ individually.

Remark 17.1. *The parameter β controls the amount of limiting; $\beta = \frac{1}{2}$ leads to a TVD scheme for a scalar problem but is more dissipative. A value of $\beta = 1$ leads to a less restrictive limiter which may give more accurate solutions.*

[2]The factors involving $\sqrt{3}$ are necessary because we use scaled Legendre polynomials, see appendix (17.11).

Characteristic limiter

Define

$$\bar{\boldsymbol{u}}^K = \begin{bmatrix} w_{1,1}^K \\ w_{2,1}^K \\ w_{3,1}^K \\ w_{4,1}^K \end{bmatrix}, \qquad \boldsymbol{u}_2^K = \begin{bmatrix} w_{1,2}^K \\ w_{2,2}^K \\ w_{3,2}^K \\ w_{4,2}^K \end{bmatrix}, \qquad \boldsymbol{u}_3^K = \begin{bmatrix} w_{1,3}^K \\ w_{2,3}^K \\ w_{3,3}^K \\ w_{4,3}^K \end{bmatrix}$$

Compute the right eigenvector matrices $\mathcal{R}_x^K, \mathcal{R}_y^K$ and left eigenvector matrices $\mathcal{L}_x^K, \mathcal{L}_y^K$ of the corresponding flux Jacobian based on the cell average value $\bar{\boldsymbol{u}}_K$, see appendix (17.14). Define

$$\boldsymbol{c}_2^K = \mathcal{L}_x^K \boldsymbol{u}_2^K, \qquad \boldsymbol{c}_3^K = \mathcal{L}_y^K \boldsymbol{u}_3^K$$

Compute limited quantities by applying minmod function component-wise

$$\tilde{\boldsymbol{c}}_2^K = \frac{1}{\sqrt{3}} \text{minmod}\left(\sqrt{3}\boldsymbol{c}_2^K, \beta\mathcal{L}_x^K(\bar{\boldsymbol{u}}^K - \bar{\boldsymbol{u}}^{W_K}), \beta\mathcal{L}_x^K(\bar{\boldsymbol{u}}^{E_K} - \bar{\boldsymbol{u}}^K)\right)$$

$$\tilde{\boldsymbol{c}}_3^K = \frac{1}{\sqrt{3}} \text{minmod}\left(\sqrt{3}\boldsymbol{c}_3^K, \beta\mathcal{L}_y^K(\bar{\boldsymbol{u}}^K - \bar{\boldsymbol{u}}^{S_K}), \beta\mathcal{L}_y^K(\bar{\boldsymbol{u}}^{N_K} - \bar{\boldsymbol{u}}^K)\right)$$

If $\tilde{\boldsymbol{c}}_2^K = \boldsymbol{c}_2^K$ and $\tilde{\boldsymbol{c}}_3^K = \boldsymbol{c}_3^K$ then

$$\Lambda\Pi_h(\boldsymbol{u}) = \boldsymbol{u}$$

else

$$\tilde{\boldsymbol{u}}_2^K = \mathcal{R}_x^K \tilde{\boldsymbol{c}}_2^K, \qquad \tilde{\boldsymbol{u}}_3^K = \mathcal{R}_y^K \tilde{\boldsymbol{c}}_3^K, \qquad \Lambda\Pi_h(\boldsymbol{u}) = \bar{\boldsymbol{u}}^K + \tilde{\boldsymbol{u}}_2^K \phi_2 + \tilde{\boldsymbol{u}}_3^K \phi_3$$

TVB version

The minmod limiter is too strict and leads to clipping of smooth extrema which are identified as shocks by the limiter. In order to avoid such effects, we can use a less strict limiter which involves replacing the minmod function with the following function

$$\text{minmodB}\,(a,b,c) = \begin{cases} a & \text{if } |a| < Mh^2 \\ \text{minmod}\,(a,b,c) & \text{otherwise} \end{cases}$$

The parameter M is related to the second derivative of the solution at smooth extrema. This is usually not known and a proper choice of the parameter M has to be done for each problem.

17.5 Positivity limiter

We assume that the first order finite volume scheme is positivity preserving under the time step restriction

$$\Delta t_K^n \leq \left(\frac{|u_K^n| + c_K^n}{\Delta x_K} + \frac{|v_K^n| + c_K^n}{\Delta y_K} \right)^{-1}$$

where (u_K, v_K) is the velocity and c_K is the sound speed corresponding to the cell average value $\bar{\boldsymbol{u}}^K$. For example, this property holds true if we use the Lax-Friedrichs flux. The positivity preserving DG scheme with degree k solution polynomials is built as follows.

Choose the smallest integer m such that $2m-3 \geq k$. Let $\hat{\xi}_1^m < \ldots < \hat{\xi}_m^m \in [-1,+1]$ be the Gauss-Lobatto-Legendre (GLL) points[3]. Note that $\hat{\xi}_1^m = -1$ and $\hat{\xi}_m^m = +1$. Consider the two sets of points

$$S_x = \{(\hat{\xi}_r^m, \xi_s^{k+1}) : 1 \leq r \leq m, \quad 1 \leq s \leq k+1\}$$

[3]These are the roots of $(1-\xi^2)P'_{m-1}(\xi)$

k	$1/(2k+1)$	m	$\frac{1}{2}\hat{w}_1^m$
1	1/3	2	1/2
2	1/5	3	1/6
3	1/7	3	1/6
4	1/9	4	1/12

$$S_y = \{(\xi_r^{k+1}, \hat{\xi}_s^m) : 1 \le r \le k+1, \quad 1 \le s \le m\}$$

Note that S_x, S_y are tensor products of Gauss points in one direction and Lobatto points in another direction. The next two steps are applied in each cell K. Choose a small number $\epsilon = 10^{-13}$, usually of the order of machine precision.

Step 1: Let $\rho_h(\xi,\eta)$ be the density inside K. Compute the minimum value of density at the points $S_x \cup S_y$

$$\rho_{min}^K = \min_{(\xi,\eta)\in S_x\cup S_y} \rho_h(\xi,\eta)$$

and define

$$\theta_1^K = \min\left\{ \left| \frac{\bar{\rho}^K - \epsilon}{\bar{\rho}^K - \rho_{min}^K} \right|, 1 \right\}$$

Let $\{w_{1,j}^K : 1 \le j \le N(k)\}$ be the dofs corresponding to density. If $\theta_1^K < 1$ the density is negative ($< \epsilon$) in cell K. In this case we modify the density without changing its average value

$$w_{1,j}^K \leftarrow \theta_1^K w_{1,j}^K, \qquad j = 2, \ldots, N(k)$$

At this stage the density is positive ($\ge \epsilon$) at all points in $S_x \cup S_y$.

Step 2: Let $p_h(\xi,\eta) = p(\boldsymbol{u}_h(\xi,\eta))$ be the pressure. At each point $(\xi,\eta) \in S_x \cup S_y$ compute

$$\tau(\xi,\eta) = \begin{cases} 1 & \text{if } p_h(\xi,\eta) \ge \epsilon \\ \tau_* & \text{such that } p((1-\tau_*)\bar{\boldsymbol{u}}^K + \tau_* \boldsymbol{u}_h(\xi,\eta)) = \epsilon \end{cases}$$

Then define

$$\theta_2^K = \min_{(\xi,\eta)\in S_x\cup S_y} \tau(\xi,\eta)$$

If $\theta_2 < 1$, then the pressure is negative so that we modify the entire solution without changing the average value

$$w_{i,j}^K \leftarrow \theta_2^K w_{i,j}^K, \qquad 1 \le i \le 4, \qquad 2 \le j \le N(k)$$

Time step: Let $\hat{w}_1^m, \ldots, \hat{w}_m^m$ be the GLL quadrature weights that satisfy

$$\sum_{q=1}^m \hat{w}_q^m = 2 \qquad \text{since our interval is } [-1,+1]$$

The the DG update preserves positivity of density and pressure in the cell average values if

$$\Delta t_K^n \le \frac{1}{2}(\min_{1\le q\le m} \hat{w}_q^m) \left(\frac{|u_K^n| + c_K^n}{\Delta x_K} + \frac{|v_K^n| + c_K^n}{\Delta y_K} \right)^{-1}$$

Remark 17.2. *(1) The computation of τ_* in step 2 requires solution of a quadratic equation which must be performed carefully so that we obtain one root which is strictly in the interval $[0,1]$. Mathematically this root exists if $p(\bar{\boldsymbol{u}}^K) \ge \epsilon$ but roundoff errors can make things difficult. (2) The positivity limiter is local to each cell and does not require any information from neighbouring cells.*

17.6 Time integration

After discretizing in space using the DG scheme, we end with a system of coupled ODEs of the the form

$$\frac{\mathrm{d}w}{\mathrm{d}t} + R(w) = 0$$

We will solve them using the 3'rd order SSP Runge-Kutta scheme which is made of three stages:

$$\begin{aligned} w^{(0)} &= w^n \\ w^{(1)} &= w^{(0)} - \Delta t^n R(w^{(0)}) \\ w^{(2)} &= \frac{3}{4}w^n + \frac{1}{4}\left[w^{(1)} - \Delta t^n R(w^{(1)})\right] \\ w^{(3)} &= \frac{1}{3}w^n + \frac{2}{3}\left[w^{(2)} - \Delta t^n R(w^{(2)})\right] \\ w^{n+1} &= w^{(3)} \end{aligned}$$

The time step is computed based on a CFL condition. For each cell K, compute the local time step

$$\Delta t_K^n = \frac{1}{2k+1}\left(\frac{|u_K^n| + c_K^n}{\Delta x_K} + \frac{|v_K^n| + c_K^n}{\Delta y_K}\right)^{-1}$$

where (u_K, v_K) is the velocity and c_K is the sound speed corresponding to the cell average value $\bar{\boldsymbol{u}}^K$. If we use the positivity limiter also, then the local time step is given by

$$\Delta t_K^n = \min\left\{\frac{1}{2k+1}, \frac{\hat{w}_1^m}{2}\right\}\left(\frac{|u_K^n| + c_K^n}{\Delta x_K} + \frac{|v_K^n| + c_K^n}{\Delta y_K}\right)^{-1}$$

The global time step is

$$\Delta t^n = \mathrm{CFL} \cdot \min_{K \in \mathcal{T}_h} \Delta t_K^n, \qquad 0 < \mathrm{CFL} \leq 1$$

where a CFL number slightly less than unity may be used.

17.7 Setting the initial condition

Given the initial condition for the i'th component of the solution

$$w_i = g_i \qquad \text{at} \qquad t = 0$$

we approximate the initial condition on cell K by an L^2-projection

$$\min_{\{w_{i,j}^K\}_j} \int_K (w_i - g_i)^2 \mathrm{d}\vec{x}$$

which leads to

$$w_{i,l}^K = \frac{1}{|K|}\int_K g_i \phi_l^K \mathrm{d}\vec{x} = \frac{1}{4}\int_{[-1,+1]^2} g_i \phi_l \mathrm{d}\vec{\xi}$$

The integral is approximated by a tensor product Gauss quadrature of $M_k = (k+1)^2$ points

$$w_{i,l}^K = \frac{1}{4}\sum_{q=1}^{M_k} g_i(\vec{x}_q)\phi_l(\vec{\xi}_q)\tilde{\omega}_q^{k+1}, \qquad l = 1, 2, \ldots, N(k)$$

where the real location of quadrature nodes $\vec{\xi}_q = (\xi_r, \eta_s)$ is given by

$$\vec{x}_q = \left(\frac{1}{2}\xi_r^{k+1}\Delta x_K + x_K, \frac{1}{2}\xi_s^{k+1}\Delta y_K + y_K\right)$$

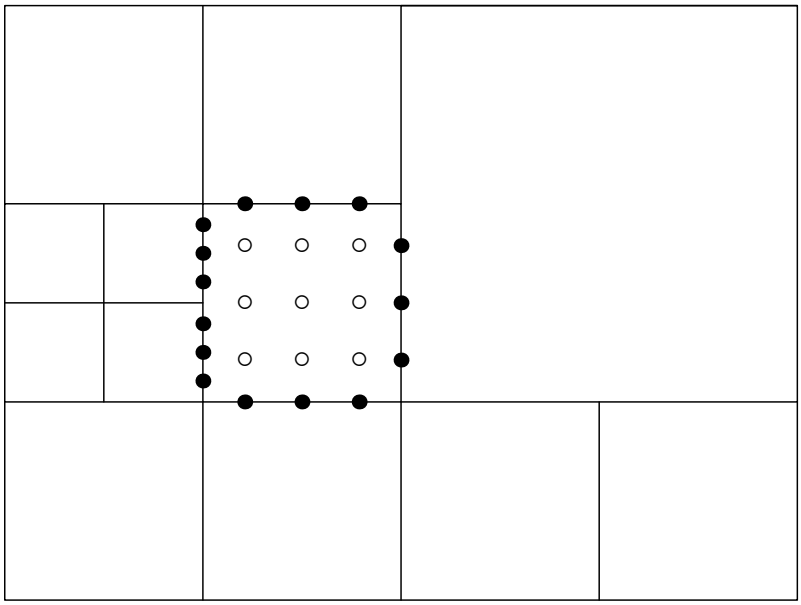


Figure 17.3: Example of Gauss quadrature points for degree $k = 2$ on adapted grids

17.8 Summary of algorithm

1. Compute projection matrices, quadrature data, etc.
2. Set initial condition by L^2-projection, $t = 0$
3. Apply TVD limiter (if needed)
4. Apply positivity limiter (if needed)
5. While $t < T$
   (a) Compute time step
   (b) For $r = 1, 2, 3$
      i. Compute rhs
      ii. Update solution to next RK stage
      iii. Apply TVD limiter
      iv. Apply positivity limiter
   (c) $t = t + \Delta t$

17.9 Adaptive grids

The DG scheme can be easily applied on adapted grids. It is usual practice to keep the level difference between any two neighbouring cells to be at most one. When two neighbouring cells are at different level of refinement, then the face quadrature is performed on the smaller faces, see Fig. 17.3.

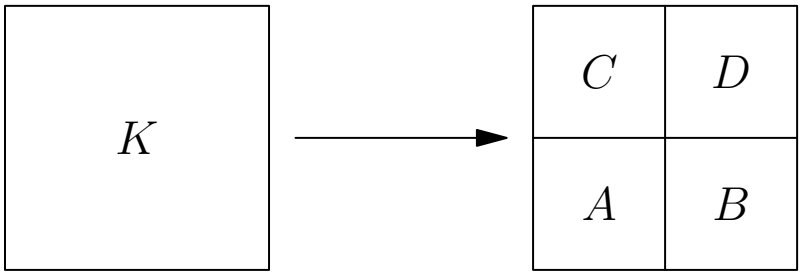


Figure 17.4: Refinement of cell K into four cells

Refinement

When a cell K is divided into four smaller cells, say A, B, C, D, see Fig. 17.4, then the solution from the parent cell K has to be projected onto the child cells. We have the i'th component of the solution

$$\vec{x} \in K: \qquad w_i^K = \sum_j w_{i,j}^K \phi_j^K$$

and we want to compute

$$\vec{x} \in A: \qquad w_i^A = \sum_j w_{i,j}^A \phi_j^A$$

We determine $\{w_{i,j}^A, j = 1, \ldots, N(k)\}$ by performing an L^2-projection, i.e.,

$$\min_{\{w_{i,j}^A\}_j} \int_A \left(w_i^K - w_i^A\right)^2 \mathrm{d}\vec{x}$$

which leads to

$$w_{i,l}^A = \frac{1}{|A|} \int_A w_i^K \phi_l^A \mathrm{d}\vec{x} = \frac{1}{|A|} \sum_{j=1}^{N(k)} w_{i,j}^K \int_A \phi_j^K \phi_l^A \mathrm{d}\vec{x}, \qquad l = 1, \ldots, N(k)$$

This can be written as a matrix vector product

$$w_i^A = P_1 w_i^K, \qquad (P_1)_{jl} = \frac{1}{|A|} \int_A \phi_j^K \phi_l^A \mathrm{d}\vec{x}$$

By transforming the integral on A to the reference cell $[-1, +1] \times [-1 + 1]$, we can rewrite P_1 as

$$(P_1)_{jl} = \frac{1}{4} \int_{-1}^{+1} \int_{-1}^{+1} \phi_j \left((\xi + 1)/2, (\eta + 1)/2\right) \phi_l(\xi, \eta) \mathrm{d}\xi \mathrm{d}\eta$$

which is independent of the element K, A; the above integrals can be computed exactly using a Gauss quadrature of $(k + 1)^2$ points. Similarly, the computation of w_i^B, w_i^C, w_i^D can be expressed in terms of matrices P_2, P_3, P_4 respectively, which are independent of the elements.

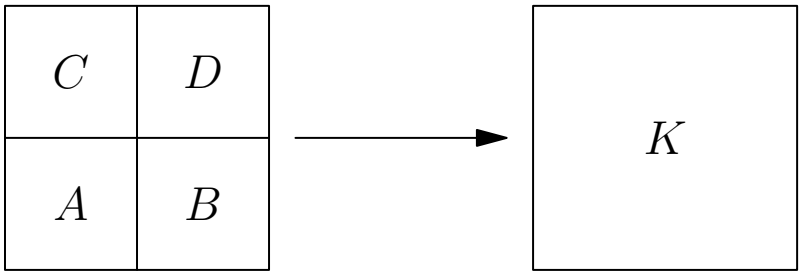


Figure 17.5: Coarsening of cells A, B, C, D into cell K

Coarsening

Consider the case when cells A, B, C, D are merged into one cell K, see Fig. 17.5. We determine the solution on K by performing an L^2-projection of the solutions on A, B, C, D, i.e., we solve the following minimization problem,

$$\min_{\{w_{i,j}^K\}_j} \int_A \left(w_i^K - w_i^A\right)^2 \mathrm{d}\vec{x} + \int_B \left(w_i^K - w_i^B\right)^2 \mathrm{d}\vec{x} + \int_C \left(w_i^K - w_i^C\right)^2 \mathrm{d}\vec{x} + \int_D \left(w_i^K - w_i^D\right)^2 \mathrm{d}\vec{x}$$

which leads to

$$w_{i,l}^K = \frac{1}{|K|}\left(\int_A w_i^A \phi_l^K \mathrm{d}\vec{x} + \int_B w_i^B \phi_l^K \mathrm{d}\vec{x} + \int_C w_i^C \phi_l^K \mathrm{d}\vec{x} + \int_D w_i^D \phi_l^K \mathrm{d}\vec{x}\right), \qquad l = 1, \ldots, N(k)$$

The above equations can be expressed as a matrix-vector product

$$w_i^K = \frac{1}{16}\left(P_1^\top w_i^A + P_2^\top w_i^B + P_3^\top w_i^C + P_4^\top w_i^D\right)$$

where the matrices P_1, P_2, P_3, P_4 are same as those described in the previous section. Moreover we have used the property that $|A| = |B| = |C| = |D| = \frac{1}{4}|K|$.

17.10 Angular momentum conservation

Define the angular momentum

$$L = x\rho v - y\rho u = xw_3 - yw_2$$

Because the momentum flux tensor

$$\begin{bmatrix} f_{1,2} & f_{2,2} \\ f_{1,3} & f_{2,3} \end{bmatrix} = \begin{bmatrix} p + \rho u^2 & \rho uv \\ \rho uv & p + \rho v^2 \end{bmatrix}$$

is symmetric, we obtain the conservation law for angular momentum

$$\frac{\partial L}{\partial t} + \frac{\partial}{\partial x}(xf_{1,3} - yf_{1,2}) + \frac{\partial}{\partial y}(xf_{2,3} - yf_{2,2}) = 0$$

Integrating this over element K, we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_K L \mathrm{d}\vec{x} + \int_{\partial K} [x(f_{1,3}n_1 + f_{2,3}n_2) - y(f_{1,2}n_1 + f_{2,2}n_2)] \mathrm{d}s = 0 \tag{17.2}$$

The DG scheme is linear in the test functions. Provided the degree $k \geq 1$ we can take the test function to be $\phi = y$ in the x-momentum equation and $\phi = x$ in the y-momentum equation.

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_K w_2 y \mathrm{d}\vec{x} - \int_K f_{2,2}\mathrm{d}\vec{x} + \int_{\partial K} \hat{f}_2 y \mathrm{d}s = 0$$
$$\frac{\mathrm{d}}{\mathrm{d}t}\int_K w_3 x \mathrm{d}\vec{x} - \int_K f_{1,3}\mathrm{d}\vec{x} + \int_{\partial K} \hat{f}_3 x \mathrm{d}s = 0$$

Subtracting the above two equations and noting that $f_{1,3} = f_{2,2} = \rho uv$ we get the equation for angular momentum under the DG scheme to be

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_K L \mathrm{d}\vec{x} + \int_{\partial K} (\hat{f}_3 x - \hat{f}_2 y)\mathrm{d}s = 0$$

This is consistent with the exact equation (17.2) and thus the DG scheme conserves angular momentum. However the limiting process can modify angular momentum. If the limiter is inactive in smooth regions then the angular momentum will be conserved.

Experimental section

Let us compute the angular momentum relative to cell center (x_K, y_K)

$$\begin{aligned} L_K &= \int_K ((x - x_K)w_3 - (y - y_K)w_2)\mathrm{d}\vec{x} \\ &= \frac{1}{2}\Delta x_K \int_K \xi w_3 \mathrm{d}\vec{x} - \frac{1}{2}\Delta y_K \int_K \eta w_2 \mathrm{d}\vec{x} \\ &= \frac{1}{2\sqrt{3}}\Delta x_K |K| w_{3,2} - \frac{1}{2\sqrt{3}}\Delta y_K |K| w_{2,3} \end{aligned}$$

If the TVB limiter does not change the solution inside K then the angular momentum is conserved in the cell K. But if the TVB limiter modifies the solution, then there is no guarantee that L_K is preserved. We will modify the limited values $\tilde{w}_{2,3}$, $\tilde{w}_{3,2}$ to $\hat{w}_{2,3}$, $\hat{w}_{3,2}$ by solving the following problem

$$\begin{aligned} &\min_{\hat{w}_{2,3}, \hat{w}_{3,2}} (\hat{w}_{2,3} - \tilde{w}_{2,3})^2 + (\hat{w}_{3,2} - \tilde{w}_{3,2})^2 \\ &\text{subject to} \quad \Delta x_K \hat{w}_{3,2} - \Delta y_K \hat{w}_{2,3} = l_K = \Delta x_K w_{3,2} - \Delta y_K w_{2,3} \end{aligned}$$

To solve this explicity, eliminate $\hat{w}_{3,2}$ to get

$$\min_{\hat{w}_{2,3}} (\hat{w}_{2,3} - \tilde{w}_{2,3})^2 + \left(\frac{l_K + \Delta y_K \hat{w}_{2,3}}{\Delta x_K} - \tilde{w}_{3,2} \right)^2$$

whose solution is

$$\hat{w}_{2,3} = \frac{\Delta x_K^2 \tilde{w}_{2,3} - (l_K - \Delta x_K \tilde{w}_{3,2})\Delta y_K}{\Delta x_K^2 + \Delta y_K^2}$$

17.11 Legendre polynomials

Legendre polynomials are obtained as the solution of Legendre's differential equation

$$\frac{\mathrm{d}}{\mathrm{d}\xi}\left[(1-\xi^2)\frac{\mathrm{d}}{\mathrm{d}\xi}P_n(\xi)\right] + n(n+1)P_n(\xi) = 0, \qquad n = 0, 1, 2, \ldots$$

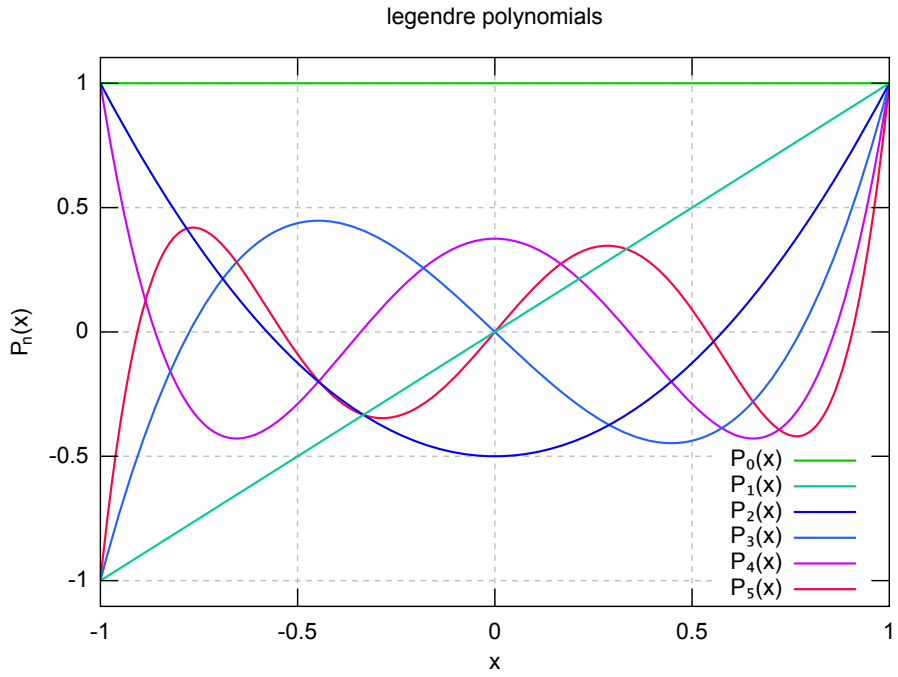


Figure 17.6: A few legendre polynomials

A few of them are listed below, also see Fig. 17.6

$$\begin{aligned} P_0(\xi) &= 1 & P_1(\xi) &= \xi \\ P_2(\xi) &= \tfrac{1}{2}(3\xi^2 - 1) & P_3(\xi) &= \tfrac{1}{2}(5\xi^3 - 3\xi) \\ P_4(\xi) &= \tfrac{1}{8}(35\xi^4 - 30\xi^2 + 3) & P_5(\xi) &= \tfrac{1}{8}(63\xi^5 - 70\xi^3 + 15\xi) \end{aligned}$$

A good way to compute them is by the following recursion relation

$$(n+1)P_{n+1}(\xi) = (2n+1)\xi P_n(\xi) - nP_{n-1}(\xi), \qquad n = 1, 2, \ldots$$

We notice that P_n is a polynomial of degree n. The Legendre polynomials have the orthogonality property

$$\int_{-1}^{+1} P_j(\xi)P_k(\xi)\mathrm{d}\xi = \begin{cases} 0 & \text{if } j \neq k \\ \frac{2}{2j+1} & \text{if } j = k \end{cases}$$

Another useful property is

$$P_n(1) = 1, \qquad P_n(-1) = (-1)^j, \qquad n = 0, 1, 2, \ldots$$

The derivatives of P_n can be computed from

$$P_0'(\xi) = 0, \qquad \frac{\xi^2 - 1}{n} P_n'(\xi) = nP_n(\xi) - P_{n-1}(\xi), \qquad \xi \in (-1, +1)$$

Let us also define the scaled functions

$$\tilde{P}_n(\xi) = \sqrt{2n+1}P_n(\xi), \qquad n = 0, 1, 2, \ldots$$

for which we have

$$\int_{-1}^{+1} \tilde{P}_j(\xi)\tilde{P}_k(\xi)\mathrm{d}\xi = \begin{cases} 0 & \text{if } j \neq k \\ 2 & \text{if } j = k \end{cases}$$

17.12 Gauss quadrature

To integrate a function $f : [-1, +1] \to \mathbb{R}$, the Gauss rule of N points is

$$\int_{-1}^{+1} f(x)\mathrm{d}x \approx \sum_{r=1}^{N} f(\xi_r^N)\omega_r^N$$

where $\{\xi_r^N\}_r \subset (-1, +1)$ are the Gauss quadrature nodes and $\{\omega_r^N\}_r$ are the corresponding weights. The N point Gauss quadrature is exact for polynomials of degree $2N - 1$.

To integrate a 2-d function $f : [-1, +1]^2 \to \mathbb{R}$ we use the tensor product of the N Gauss points

$$\int_{-1}^{+1}\int_{-1}^{+1} f(x, y)\mathrm{d}x\mathrm{d}y \approx \sum_{r=1}^{N}\sum_{s=1}^{N} f(\xi_r^N, \xi_s^N)\omega_r^N\omega_s^N$$

From a programming point of view, one may want to arrange the Gauss points in a 1-d array indexed by a single integer q; there is a unique mapping from $q \to (r, s)$. Then we can write the quadrature rule as

$$\int_{-1}^{+1}\int_{-1}^{+1} f(x, y)\mathrm{d}x\mathrm{d}y \approx \sum_{q=1}^{N^2} f(\vec{\xi}_q^N)\tilde{\omega}_q^N$$

where

$$\vec{\xi}_q^N = (\xi_r^N, \xi_s^N), \qquad \tilde{\omega}_q^N = \omega_r^N\omega_s^N$$

17.13 Lax-Friedrich's flux

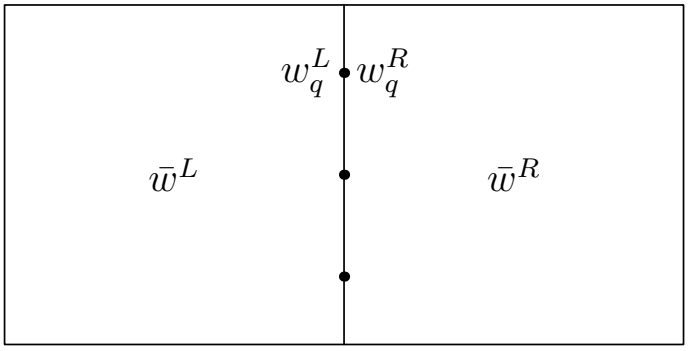


Figure 17.7: Computation of numerical flux

Consider the face between cells L and R as shown in Fig. 17.7. The quantities $\boldsymbol{u}_q^L$, $\boldsymbol{u}_q^R$ are the two trace values at the q'th quadrature point and $\bar{\boldsymbol{u}}^L$, $\bar{\boldsymbol{u}}^R$ are the cell average values. The Lax-Friedrich's flux at the q'th quadrature point on this face is given by

$$\hat{\boldsymbol{f}}_q = \frac{1}{2}[\boldsymbol{f}_1(\boldsymbol{u}_q^L) + \boldsymbol{f}_1(\boldsymbol{u}_q^R)] - \frac{1}{2}\lambda_{LR}(\boldsymbol{u}_q^R - \boldsymbol{u}_q^L)$$

where

$$\lambda_{LR} = \max\{\lambda_1(\bar{\boldsymbol{u}}^L), \lambda_1(\bar{\boldsymbol{u}}^R)\}, \qquad \lambda_1(\boldsymbol{u}) = |u(\boldsymbol{u})| + c(\boldsymbol{u})$$

17.14 Left and right eigenvectors of Euler flux Jacobian

The right eigenvector matrices of the flux Jacobians $\frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{u}}$, $\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{u}}$ are respectively

$$\mathcal{R}_x = \begin{bmatrix} 1 & 0 & 1 & 1 \\ u & 0 & u+c & u-c \\ v & -1 & v & v \\ k & -v & h+cu & h-cu \end{bmatrix}, \qquad \mathcal{R}_y = \begin{bmatrix} 1 & 0 & 1 & 1 \\ u & 1 & u & u \\ v & 0 & v+c & v-c \\ k & u & h+cv & h-cv \end{bmatrix}$$

The left eigenvector matrices are

$$\mathcal{L}_x = \begin{bmatrix} 1-\frac{\phi}{c^2} & \gamma_1 \frac{u}{c^2} & \gamma_1 \frac{v}{c^2} & -\frac{\gamma_1}{c^2} \\ v & 0 & -1 & 0 \\ \beta(\phi - cu) & \beta(c - \gamma_1 u) & -\beta\gamma_1 v & \beta\gamma_1 \\ \beta(\phi + cu) & -\beta(c + \gamma_1 u) & -\beta\gamma_1 v & \beta\gamma_1 \end{bmatrix}$$

$$\mathcal{L}_y = \begin{bmatrix} 1-\frac{\phi}{c^2} & \gamma_1 \frac{u}{c^2} & \gamma_1 \frac{v}{c^2} & -\frac{\gamma_1}{c^2} \\ -u & 1 & 0 & 0 \\ \beta(\phi - cv) & -\beta\gamma_1 u & \beta(c - \gamma_1 v) & \beta\gamma_1 \\ \beta(\phi + cv) & -\beta\gamma_1 u & -\beta(c + \gamma_1 v) & \beta\gamma_1 \end{bmatrix}$$

where

$$\gamma_1 = \gamma - 1, \quad k = \frac{1}{2}(u^2 + v^2), \quad \phi = \gamma_1 k, \quad \beta = \frac{1}{2c^2}, \quad h = \frac{c^2}{\gamma - 1} + k$$

These eigenvectors are orthonormal

$$\mathcal{R}_x \mathcal{L}_x = I = \mathcal{R}_y \mathcal{L}_y$$

Chapter 18

Nodal DG on triangles

The automatic generation of triangular grids in 2-D is a very well developed technology. Hence they form an important choice for numerical solution of PDEs. We can formulate DG scheme on triangles directly in physical space or in mapped space using modal or nodal basis. The first two approaches require quadrature which adds to the cost of DG schemes. On linear triangles, the use nodal basis can lead to a quadrature-free scheme which can be efficiently implemented by matrix-vector operations. The construction of nodal basis has been explained in Chapter (XXX). The solution in each triangle is a polynomial $\mathbb{P}_k$ of the form

$$u_h = \sum_{j=0}^{N(k)-1} u_j^K \ell_j(\xi, \eta)$$

where $N(k) = \frac{1}{2}(k+1)(k+2)$ is the number of dofs in each cell and (r, s) are coordinates in the reference cell

$$\hat{K} = \{(\xi, \eta) : \xi, \eta \geq -1, \xi + \eta \leq 0\}$$

We will consider a scalar conservation

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0$$

Multiplying this by $\ell_i(\xi, \eta)$ and integrating over one cell

$$\int_K \frac{\partial u_h}{\partial t} \ell_i \mathrm{d}x\mathrm{d}y - \int_K \left[f(u_h) \frac{\partial \ell_i}{\partial x} + g(u_h) \frac{\partial \ell_i}{\partial y} \right] \mathrm{d}x\mathrm{d}y + \int_{\partial K} H(u_h^-, u_h^+, n) \ell_i \mathrm{d}s = 0$$

Let us assume that we have a linear triangle which means that the Jacobian of the transformation $\vec{F}_K : \hat{K} \to K$ is a constant matrix. The first integral transforms as

$$\int_K \frac{\partial u_h}{\partial t} \ell_i \mathrm{d}x\mathrm{d}y = J_K \int_{\hat{K}} \frac{\partial u_h}{\partial t} \ell_i \mathrm{d}\xi\mathrm{d}\eta = J_K \sum_j \frac{\mathrm{d}u_j^K}{\mathrm{d}t} \int_{\hat{K}} \ell_i \ell_j \mathrm{d}\xi\mathrm{d}\eta$$

To compute the elements of the mass matrix, we utilize the representation of the Lagrange polynomial in terms of orthogonal polynomials

$$\begin{aligned}
\int_{\hat{K}} \ell_i \ell_j \mathrm{d}\xi \mathrm{d}\eta &= \sum_r \sum_s c_{ir} c_{js} \int_{\hat{K}} \hat{\varphi}_r \hat{\varphi}_s \mathrm{d}\xi \mathrm{d}\eta \\
&= \sum_r \sum_s c_{ir} c_{js} \delta_{rs} \\
&= \sum_r c_{ir} c_{jr} \\
&= \sum_r V_{ri}^{-1} V_{rj}^{-1} \\
&= \sum_r V_{ir}^{-\top} V_{rj}^{-1} \\
&= (V^{-\top} V^{-1})_{ij} \\
&= (V V^{\top})_{ij}^{-1}
\end{aligned}$$

Hence the mass matrix is

$$M_K = J_K (V V^{\top})^{-1}$$

To apply time integration, we require $M_K^{-1} = \frac{1}{J_K} V V^{\top}$ which is easy to compute and there is no need to compute a matrix inverse.

To compute the flux integral, we first interpolate the flux using the nodes where the solution is stored, i.e.

$$f(u_h) \approx f_h = \sum_j f(u_j^K) \ell_j(\xi, \eta), \qquad g(u_h) \approx g_h = \sum_j g(u_j^K) \ell_j(\xi, \eta)$$

$$\frac{\partial \ell_i}{\partial x} = \frac{\partial \ell_i}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \ell_i}{\partial \eta} \frac{\partial \eta}{\partial x}$$

Appendix A

Quadrature rules

A.1 Quadrature in 1-D

The DG scheme involves integrals which must be approximated by quadrature. Let

$$f : [-1, +1] \to \mathbb{R}$$

Choose n quadrature nodes $\{\xi_1, \xi_2, \ldots, \xi_n\} \subset [-1, +1]$.

$$\int_{-1}^{+1} f(\xi) \mathrm{d}\xi \approx \sum_{q=1}^{n} \omega_q f(\xi_q)$$

The accuracy of a quadrature rule is usually characterized in terms of the set of polynomials which it can exactly integrate.

- Gauss-Legendre quadrature
  - The nodes $\{\xi_q\}$ are the n roots of Legendre polynomial $P_n(\xi)$
  - n-point rule is exact for any $f \in \mathbb{P}_{2n-1}$
- Gauss-Lobatto-Legendre quadrature
  - The nodes include $\{-1, +1\}$ and the $n-2$ roots of $P'_{n-1}(\xi)$,
  - n-point rule is exact for any $f \in \mathbb{P}_{2n-3}$

Since any quadrature rule must be exact atleast for the constant function, the weights must sum to two, i.e.,

$$\sum_{q=1}^{n} \omega_q = 2$$

For a function defined on a general interval $f : [a, b] \to \mathbb{R}$, we first perform a change of variable

$$x(\xi) = \frac{1-\xi}{2} a + \frac{1+\xi}{2} b, \qquad \xi \in [-1, +1]$$

and then apply the quadrature rule

$$\begin{aligned}\int_a^b f(x)\mathrm{d}x &= \frac{1}{2}(b-a)\int_{-1}^{+1} f(x(\xi))\mathrm{d}\xi \\ &\approx \frac{1}{2}(b-a)\sum_{q=1}^{n}\omega_q f(x(\xi_q)) \\ &= \sum_{q=1}^{n}\tilde{\omega}_q f(x(\xi_q)), \qquad \tilde{\omega}_q = \frac{1}{2}(b-a)\omega_q\end{aligned}$$

In many finite element integrals, the function being integrated is already given in terms of the reference coordinate ξ and in this case

$$\int_a^b f_h(\xi)\mathrm{d}x \approx \sum_{q=1}^{n}\tilde{\omega}_q f_h(\xi_q)$$

A.2 Quadrature in 2-D

A two dimensional integral over the reference cell $\hat{K} = [-1,+1]\times[-1,+1]$ is computed using a tensor product of one dimensional quadrature points. Hence

$$\int_{\hat{K}} f(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta \approx \sum_{i=1}^{n}\sum_{j=1}^{n} f(\xi_i,\xi_j)\omega_i\omega_j$$

If we have a general domain K, e.g., an isoparametric element, we will have to mapping to the reference cell $\vec{F}_K : \hat{K} \to K$ so that $(x,y) = \vec{F}_K(\xi,\eta)$. Then

$$\begin{aligned}\int_K f(x,y)\mathrm{d}x\mathrm{d}y &= \int_{\hat{K}} f(x(\xi,\eta),y(\xi,\eta))\det J_K(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta \\ &\approx \sum_{i=1}^{n}\sum_{j=1}^{n} f(x(\xi_i,\xi_j),y(\xi_i,\xi_j))\det J_K(\xi_i,\xi_j)\omega_i\omega_j\end{aligned}$$

where J_K and $\det J_K$ are the Jacobian of the map and its determinant, and they are explained in Section (12.4). In many finite element integrals, the function being integrated is already known as a function of the reference coordinates (ξ,η) and in such a situation, we can approximate the integral as

$$\int_K f_h(\xi,\eta)\mathrm{d}x\mathrm{d}y \approx \sum_{i=1}^{n}\sum_{j=1}^{n} f_h(\xi_i,\xi_j)\det J_K(\xi_i,\xi_j)\omega_i\omega_j$$

Appendix B

Evaluation of Lagrange polynomials

Consider the reference cell to be $[-1, +1]$ and let us choose n distinct points in the reference cell

$$-1 \leq \xi_0 < \xi_1 < \ldots < \xi_{n-1} \leq +1$$

The Lagrange polynomials are

$$\ell_j(\xi) = \frac{(\xi - \xi_0)\ldots(\xi - \xi_{j-1})(\xi - \xi_{j+1})\ldots(\xi - \xi_{n-1})}{(\xi_j - \xi_0)\ldots(\xi_j - \xi_{j-1})(\xi_j - \xi_{j+1})\ldots(\xi_j - \xi_{n-1})}$$

Each ℓ_j is a polynomial of degree $n-1$ and has the delta property

$$\ell_j(\xi_i) = \delta_{ij}, \qquad 0 \leq i, j \leq n-1$$

The polynomial of degree $n-1$ given by

$$p(\xi) = \sum_{j=0}^{n-1} \ell_j(\xi) f_j$$

interpolates the function $f : [-1, +1] \to \mathbb{R}$ at the n points in the sense that

$$p(\xi_j) = f_j = f(\xi_j), \qquad 0 \leq j \leq n-1$$

Since a constant function must be exactly interpolated, the Lagrange polynomials form a *partition of unity*

$$\sum_{j=0}^{n-1} \ell_j(\xi) = 1, \qquad \forall \xi$$

If we want to evaluate $p(\xi)$ for any other value of ξ then the direct evaluation requires $O(n^2)$ floating point operations, which is very expensive.

B.1 Barycentric Lagrange form

Define

$$\ell(\xi) = (\xi - \xi_0)(\xi - \xi_1)\ldots(\xi - \xi_{n-1})$$

Then it is easy to check that

$$\ell_j(\xi) = \frac{\ell(\xi)}{(\xi - \xi_j)\ell'(\xi_j)} = \ell(\xi)\frac{w_j}{\xi - \xi_j}, \qquad w_j = \frac{1}{\ell'(\xi_j)}$$

The Lagrange interpolation can be written as

$$p(\xi) = \ell_(\xi) \sum_{j=0}^{n-1} \frac{w_j}{\xi - \xi_j} f_j$$

Summing all the Lagrange polynomials yields us unity

$$1 = \sum_{j=0}^{n-1} \ell_j(\xi) = \ell(\xi) \sum_{j=0}^{n-1} \frac{w_j}{\xi - \xi_j}$$

so that

$$\ell(\xi) = \frac{1}{\sum_{j=0}^{n-1} \frac{w_j}{\xi - \xi_j}}$$

Finally we get the Barycentric Lagrange interpolation formula [2]

$$p(\xi) = \frac{\sum_{j=0}^{n-1} \frac{w_j}{\xi - \xi_j} f_j}{\sum_{j=0}^{n-1} \frac{w_j}{\xi - \xi_j}}$$

This looks like a rational polynomial but in reality it is an ordinary polynomial because of the particular values of the weights. The weights do not depend on the point ξ and can be precomputed and stored. Then the evaluation of $p(\xi)$ requires only $O(n)$ floating point operations. It is also proved that this form is stable in terms of round-off errors.

Here is an implementation in C. We first compute the weights.

```
// Given n distinct points in x[n], compute weights for barycentric
// formula.
void compute_weights(int n, double x[n], double w[n])
{
   for(int i=0; i<n; ++i) w[i] = 1.0;
   for(int i=1; i<n; ++i)
      for(int j=0; j<i; ++j)
      {
         t = x[i] - x[j];
         w[i] *=  t;
         w[j] *= -t;
      }
   for(i=0; i<n; ++i) w[i] /= 1.0;
}
```

Once the weights have been computed the evaluation can be accomplished by a function like this.

```
double evaluate(int n, double x[n], double w[n], double f[n],
                double y)
{
   double num = 0, den = 0;
   for(int i=0; i<n; ++i)
   {
      if(fabs(y - x[i]) < eps) return f[i];
      t = w[i]/(y - x[i]);
      num += t * f[i];
      den += t;
   }
   return num/den;
}
```

B.2 Computing derivatives

If we have constructed an interpolating polynomial, we can differentiate it to obtain approximations to derivatives.

$$p'(\xi) = \sum_{j=0}^{n-1} \ell_j'(\xi) f_j$$

Derivative at an arbitrary point The barycentric form can be written as

$$p(\xi) \sum_{j=0}^{n-1} \frac{w_j}{\xi - \xi_j} = \sum_{j=0}^{n-1} \frac{w_j f_j}{\xi - \xi_j}$$

and differentiating this on both sides, we get

$$p'(\xi) \sum_{j=0}^{n-1} \frac{w_j}{\xi - \xi_j} - p(\xi) \sum_{j=0}^{n-1} \frac{w_j}{(\xi - \xi_j)^2} = - \sum_{j=0}^{n-1} \frac{w_j f_j}{(\xi - \xi_j)^2}$$

Re-arranging this gives the derivative formula

$$p'(\xi) = \frac{\sum_{j=0}^{n-1} \frac{w_j}{\xi - \xi_j} \frac{p(\xi) - f_j}{\xi - \xi_j}}{\sum_{j=0}^{n-1} \frac{w_j}{\xi - \xi_j}}$$

Derivatives at the nodes Since

$$\ell_j(\xi) = \ell(\xi) \frac{w_j}{\xi - \xi_j} \quad \Longrightarrow \quad (\xi - \xi_j) \ell_j(\xi) = w_j \ell(\xi)$$

Differentiating both sides

$$\ell_j(\xi) + (\xi - \xi_j) \ell_j'(\xi) = w_j \ell'(\xi) \quad \Longrightarrow \quad \ell_j'(\xi) = \frac{w_j \ell'(\xi) - \ell_j(\xi)}{\xi - \xi_j}$$

At any node ξ_i, $i \neq j$

$$\ell_j'(\xi_i) = \frac{w_j \ell'(\xi_i) - \ell_j(\xi_i)}{\xi_i - \xi_j} = \frac{w_j \frac{1}{w_i} - \delta_{ij}}{\xi_i - \xi_j} = \frac{w_j}{w_i} \frac{1}{\xi_i - \xi_j}, \qquad i \neq j$$

Finally, by differentiating the partition of unity property

$$\sum_{i=0}^{n-1} \ell_i'(\xi) = 0$$

we get

$$\ell_j'(\xi_j) = - \sum_{i=0, i \neq j}^{n-1} \ell_i'(\xi_j)$$

We can arrange the derivatives into a differentiation matrix

$$D_{ij} = \ell_j'(\xi_i), \qquad 0 \leq i, j \leq n-1$$

Then the derivatives at all the nodes can be obtain as a matrix vector product

$$\begin{bmatrix} p'(\xi_0) \\ \vdots \\ p'(\xi_{n-1}) \end{bmatrix} = D \begin{bmatrix} f_0 \\ \vdots \\ f_{n-1} \end{bmatrix}$$

Bibliography

[1] M. Ainsworth, *Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods*, Journal of Computational Physics, 198 (2004), pp. 106–130.

[2] J.-P. Berrut and L. N. Trefethen, *Barycentric Lagrange Interpolation*, SIAM Review, 46 (2004), pp. 501–517.

[3] M. G. Blyth and C. Pozrikidis, *A Lobatto interpolation grid over the triangle*, IMA Journal of Applied Mathematics, 71 (2006), pp. 153–169.

[4] L. Bos, *Bounding the Lebesgue function for Lagrange interpolation in a simplex*, Journal of Approximation Theory, 38 (1983), pp. 43–59.

[5] Q. Chen and I. Babuška, *Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle*, Computer Methods in Applied Mechanics and Engineering, 128 (1995), pp. 405–417.

[6] B. Cockburn, S.-Y. Lin, and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems*, Journal of Computational Physics, 84 (1989), pp. 90–113.

[7] M. Dubiner, *Spectral methods on triangles and other domains*, Journal of Scientific Computing, 6 (1991), pp. 345–390.

[8] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods*, vol. 54 of Texts in Applied Mathematics, Springer New York, New York, NY, 2008.

[9] F. Q. Hu and H. L. Atkins, *Eigensolution Analysis of the Discontinuous Galerkin Method with Nonuniform Grids*, Journal of Computational Physics, 182 (2002), pp. 516–545.

[10] F. Q. Hu, M. Hussaini, and P. Rasetarinera, *An Analysis of the Discontinuous Galerkin Method for Wave Propagation Problems*, Journal of Computational Physics, 151 (1999), pp. 921–946.

[11] C. Hua, *An inverse transformation for quadrilateral isoparametric elements: Analysis and application*, Finite Elements in Analysis and Design, 7 (1990), pp. 159–166.

[12] T. Koornwinder, *Two-Variable Analogues of the Classical Orthogonal Polynomials*, in Theory and Application of Special Functions, Elsevier, 1975, pp. 435–495.

[13] J. F. B. M. Kraaijevanger, *Contractivity of Runge-Kutta methods*, BIT Numerical Mathematics, 31 (1991), pp. 482–528.

[14] E. J. Kubatko, J. J. Westerink, and C. Dawson, *Semi discrete discontinuous Galerkin methods and stage-exceeding-order, strong-stability-preserving Runge–Kutta time discretizations*, Journal of Computational Physics, 222 (2007), pp. 832–848.

[15] H. Luo, J. D. Baum, and R. Löhner, *A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids*, Journal of Computational Physics, 227 (2008), pp. 8875–8893.

[16] R. J. Spiteri and S. J. Ruuth, *A New Class of Optimal High-Order Strong-Stability-Preserving Time Discretization Methods*, SIAM Journal on Numerical Analysis, 40 (2002), pp. 469–491.

[17] M. A. Taylor, B. A. Wingate, and R. E. Vincent, *An Algorithm for Computing Fekete Points in the Triangle*, SIAM Journal on Numerical Analysis, 38 (2000), pp. 1707–1720.

[18] T. Warburton, *An explicit construction of interpolation nodes on the simplex*, Journal of Engineering Mathematics, 56 (2007), pp. 247–262.