Notes on the Discontinuous Galerkin methods for the numerical simulation of hyperbolic equations 1 General Context 1.1 Bibliography

Adam Larat

Notes on the Discontinuous Galerkin methods for the numerical simulation of hyperbolic equations

Adam Larat

February 10, 2016

1 General Context

1.1 Bibliography

The roots of Discontinuous Galerkin (DG) methods is usually attributed to Reed and Hills in a paper published in 1973 on the numerical approximation of the neutron transport equation [18]. In fact, the adventure really started with a rather thoroughfull series of five papers by Cockburn and Shu in the late 80's [7, 5, 9, 6, 8]. Then, the fame of the method, which could be seen as a compromise between Finite Elements (the center of the method being a weak formulation) and Finite Volumes (the basis functions are defined cell-wise, the cells being the elements of the primal mesh) increased and slowly investigated successfully all the domains of Partial Differential Equations numerical integration. In particular, one can cite the ground papers for the common treatment of convection-diffusion equations [4, 3] or the treatment of pure elliptic equations [2, 17]. For more information on the history of Discontinuous Galerkin method, please refer to section 1.1 of [15].

Today, DG methods are widely used in all kind of manners and have applications in almost all fields of applied mathematics. (TODO: cite applications and structured/unstructured meshes, steady/unsteady, etc...). The methods is now mature enough to deserve entire text books, among which I cite a reference book on Nodal DG Methods by Henthaven and Warburton [15] with the ground basis of DG integration, numerical analysis of its linear behavior and generalization to multiple dimensions.

Lately, since 2010, thanks to a ground work of Zhang and Shu [26, 27, 25, 28, 29], Discontinuous Galerkin methods are eventually able to combine high order accuracy and certain preservation of convex constraints, such as the positivity of a given quantity, for example. These new steps forward are very promising since it brings us very close to the "*Ultimate Conservative Scheme*", [23, 1].

1.2 Hyperbolic conservation laws

This section is just a quick introduction to hyperbolic conservation laws in order to settle the notations.

Let $d \in \mathbb{N}^*$ be the number of spatial dimensions and $\Omega \subset \mathbb{R}^d$ be the domain of study. $\partial\Omega$ denotes the frontiers of the domain. To greatly simplify the following notes, we will ignore boundary conditions.

Let $m \in \mathbb{N}^*$ be the number of conserved variables and $\mathbf{W}$ the vector of these variables. Then, $\mathbf{W}$ is ruled by a *Conservation Law* if there exists a *Flux Function*

$$\begin{cases} \vec{\boldsymbol{\mathcal{F}}} : & \boldsymbol{\mathcal{S}} \subset \mathbb{R}^m & \longrightarrow & (\mathbb{R}^m)^d \\ & \mathbf{W} & \longmapsto & \vec{\boldsymbol{\mathcal{F}}}(\mathbf{W}) \end{cases} \tag{1}$$

such that $\mathbf{W}$ verifies the PDE:

$$\frac{\partial \mathbf{W}}{\partial t} + \vec{\nabla} \cdot \vec{\boldsymbol{\mathcal{F}}}(\mathbf{W}) = 0. \tag{2}$$

Here, the set $\boldsymbol{\mathcal{S}}$ denotes the possible physical constraints on the conserved variables (positivity of the density, realizability of the transported moments, etc...).

This conservation law is said to be *hyperbolic*, if the Jacobian

$$\mathbf{J}(\mathbf{W}) = \frac{\partial \vec{\boldsymbol{\mathcal{F}}}}{\partial \mathbf{W}} \tag{3}$$

is diagonalizable with real eigenvalues. In this case, we denote α the maximal absolute eigenvalue of $\mathbf{J}$:

$$\alpha(\mathbf{W}) = \max\left\{|\lambda|,\ \exists \mathbf{V} \in \mathbb{R}^m \text{ such that } \mathbf{J}(\mathbf{W}).\mathbf{V} = \lambda \mathbf{V}\right\}. \tag{4}$$

2 Runge-Kutta discontinuous Galerkin methods

2.1 Degrees of freedom

Let $\Omega \subset \mathbb{R}^d$ be the *domain of study* and

$$\bigcup_{k=1}^{N} \mathrm{T}_k = \Omega \quad \text{such that} \quad \forall i,j \in [\![1,N]\!], \quad \dim(\mathrm{T}_i \cap \mathrm{T}_j) < d, \tag{5}$$

a *tesselation* of the domain.

In each element T_k, we define a funcitonal basis $\left(\psi_l^k\right)_{l=1,\dots,\mathrm{P}_k}$ and associate T_k with the finite dimensional vector space:

$$V_h^k = \left\{ \varphi : \Omega \longrightarrow \mathbb{R}, \quad \varphi = \left(\sum_{l=1}^{\mathrm{P}_k} a_l \psi_l^k \right) \chi_k \right\}, \tag{6}$$

where χ_k denote the characteristic function of T_k, taking value 1 in T_k and 0 elsewhere.

Therefore, a numerical solution on $\mathcal{M}_h = (\mathrm{T}_k)_{k=1,\dots,N}$ has exactly $\mathrm{P} = \sum_{k=1}^{N} \mathrm{P}_k$ *Degrees of Freedom (DoF).*

2.2 Spatial weak formulation

The continuous hyperbolic conservation law (2) is approximated on the high order functional vectorial space V_h defined in (6). It reads:

Weak Formulation

Find $\mathbf{W}_h(t,\boldsymbol{x}) = \sum_{k=1}^{N} \chi_k(\boldsymbol{x}) \sum_{j=1}^{\mathrm{P}_k} \mathbf{W}_k^j(t)\psi_k^j(\boldsymbol{x})$ a piecewise high order polynomial solution, such that:

$$\forall k' \in [\![1,N]\!], \forall i \in [\![1,\mathrm{P}_{k'}]\!], \quad \int_\Omega \left(\partial_t \mathbf{W}_h(t,\boldsymbol{x}) + \overrightarrow{\nabla} \cdot \vec{\boldsymbol{\mathcal{F}}}\left(\mathbf{W}_h(t,\boldsymbol{x})\right)\right) \psi_{k'}^i(\boldsymbol{x}) d\boldsymbol{x} = 0. \tag{7}$$

Since the supports T_k and $\mathrm{T}_{k'}$ are disjointed, $k' = k$ always and the problem comes to

$$\forall k, i, \quad \int_{\mathrm{T}_k} \sum_j d_t \mathbf{W}_k^j(t)\psi_k^j(\boldsymbol{x})\psi_k^i(\boldsymbol{x}) d\boldsymbol{x} + \int_{\mathrm{T}_k} \overrightarrow{\nabla} \cdot \vec{\boldsymbol{\mathcal{F}}}\left(\mathbf{W}_h(t,\boldsymbol{x})\right) \psi_k^i(\boldsymbol{x}) d\boldsymbol{x} = 0.$$

By integration by parts, one easily gets the following three fold formulation:

$$\underbrace{\sum_j \left(\int_{\mathrm{T}_k} \psi_k^j \psi_k^i d\boldsymbol{x}\right) d_t \mathbf{W}_k^j(t)}_{①} - \underbrace{\int_{\mathrm{T}_k} \vec{\boldsymbol{\mathcal{F}}}\left(\mathbf{W}_h(t,\boldsymbol{x})\right) \cdot \nabla \psi_k^i(\boldsymbol{x}) d\boldsymbol{x}}_{②} + \underbrace{\int_{\partial \mathrm{T}_k} \psi_k^i(\boldsymbol{s}) \vec{\boldsymbol{\mathcal{F}}}\left(\mathbf{W}_h(t,\boldsymbol{s})\right) \cdot \vec{\mathbf{n}}\, d\boldsymbol{s}}_{③} = 0 \tag{8}$$

Remark 1

One usually considers (but we don't have to)

$$\vec{\boldsymbol{\mathcal{F}}}\left(\boldsymbol{W}_h(t,\boldsymbol{x})\right) = \sum_j \vec{\boldsymbol{\mathcal{F}}}\left(\boldsymbol{W}_k^j(t)\right) \psi_k^j(\boldsymbol{x}). \tag{9}$$

If not, ② is integrated by means of an adequate volumic quadrature.

Remark 2

Along the boundaries ∂T_k of T_k, $\vec{\boldsymbol{\mathcal{F}}}\left(\boldsymbol{W}_h\right)$ does not have a mathematical sens other than the flux occuring locally between the two states on both sides. Then the flux appearing in ③ is replaced by the numerical flux:

$$\vec{\boldsymbol{\mathcal{F}}}\left(\boldsymbol{W}_h(t,\boldsymbol{s})\right) \cdot \vec{\mathbf{n}} = \boldsymbol{F}^*\left(\boldsymbol{W}_{ext}(t,\boldsymbol{s}), \boldsymbol{W}_{int}(t,\boldsymbol{s}); \vec{\mathbf{n}}\right). \tag{10}$$

In the case the flux function is also projected on the basis as in (9), the global formu-

lation simply comes down to:

$$\mathcal{M}_{ij} d_t \mathbf{W}_j - \vec{\mathcal{K}}^t_{ij} \cdot \vec{\boldsymbol{\mathcal{F}}}(\mathbf{W}_j) + \sum_{\varepsilon \in \partial \mathrm{T}_k} \underbrace{\int_\varepsilon \psi^i_k(\boldsymbol{s}) \mathbf{F}^* \left(\mathbf{W}_{\mathrm{ext}}(t, \boldsymbol{s}), \mathbf{W}_{\mathrm{int}}(t, \boldsymbol{s}); \vec{\mathbf{n}} \right) d\boldsymbol{s}}_{④} = 0 \qquad (11)$$

Remark 3

- *In general, last integral ④ is estimated by mean of an adequate quadrature of dimension $d-1$. This implies a numerous amount of numerical flux computations.*
- *Luckily, since we transport and update all the derivatives of the solution within the mesh, we don't need to be high-order accurate on all these numerical fluxes. A first order flux on all the degrees of freedom will provide a high order solution anyway. Therefore, the linear Rusanov flux (28) is very well suited for this purpose, since its evaluation is quite costless and it has good monotonicity properties (see section 2.5.1).*

2.3 Basis functions

DG methods are split into two main families: *modal* and *nodal* DG. These prefixes refer to the choice on the basis functions ψ^k_l. In general, the modal basis function will be chosen so that the corresponding mass matrix

$$\mathcal{M}^k_{ij} = \int_{\mathrm{T}_k} \psi^k_i \psi^k_j \qquad (12)$$

is everywhere diagonal, what greatly simplifies the updates. On the other hand, any access to the value of the solution at a given point (for example for the evaluation of a numerical flux somewhere) will require a combination of all the degrees of freedom in the element:

$$\mathbf{W}_h(x_k) = \sum_{l=1}^{\mathrm{P}_k} a_l \psi^k_l(x_k). \qquad (13)$$

On the contrary, nodal basis function are attached to certain points of the considered element. In general, the points are chosen in a way it simplifies the volumic and edge terms evaluations. In particular, on the edges of the elements degrees of freedom should be attached to some known quadrature points. Then, the numerical flux integral evaluation has all the necessary data for the numerous numerical fluxes evaluation directly at hand.

For dimensions stricly higher than 1 and polynomial order at least quadratic, there is a conjecture that there is no orthogonal nodal basis[1]. This might even be proven, alas I don't know yet about the demonstration.

[1] Personal conversation with V. Perrier

2.3.1 Modal basis

- **Legendre polynomials:** this is the orthogonal interlocked polynomial basis for measure one. It is simply obtained by a Gram-Schmidt orthogonalization process starting from any polynomial basis of increasing order.

$$\text{On } [-1,1]: \quad 1,\ x,\ x^2 - 1/3,\ \dots \tag{14}$$

- Other orthogonal basis: Jacobi, Tchebychev, trigonometric, etc.

2.3.2 Nodal basis

Let us restrict to 1D for the moment. Givent a set $(x_i)_{i=1,\dots,\mathrm{P}_k}$ of points in T_k, the associated *Lagrangian* basis is the only basis of polynomials of order $\mathrm{P}_k - 1$ such that $\mathrm{L}_j(x_i) = \delta_{ij}$. Then it is easy and classic to write

$$\mathrm{L}_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}. \tag{15}$$

In general, the set of nodes is:

- a regular distribution within the cell:

$$x_j = x_{i-1/2} + j\frac{\Delta x}{\mathrm{P}_k - 1}, \quad j = 0, \dots, \mathrm{P}_k - 1, \tag{16}$$

- a set of smart quadrature points, like Gauss-Lobatto quadrature points since they contain the bounds of the interval.

In more dimensions, the generalization of the former discussion is more complex. Only the regular distribution can be algorithmically extended. In particular, the generalization of the Gauss-Lobatto Lagrange polynomials to 2 and 3 D is not obvious, see discution in [15], chapter 6.

2.4 Time integration

Now, thanks to the spatial semi-integration, the numerical scheme (11) comes down to an Ordinary Differential Equation. It can be integrated by any numerical procedure for ODEs. Here we focus on Runge-Kutta methods, since they offer a hierarchy of increasing order, which is suitable for the time integration of high order spatial weak formulations.

2.4.1 Runge-Kutta methods

Let

$$y'(t) = f(t, y(t)) \tag{17}$$

be an arbitrary ordinary differential equation. An s-stages explicit Runge-Kutta method applied to this equation can be written in the form:

$$\left\{\begin{array}{rcl} y_{n+1} & = & y_n + \Delta t \sum_{i=1}^{s} b_i k_i, \\ k_i & = & f\left(t_n + c_i \Delta t, y_n + \Delta t \sum_{j<i} a_{ij} k_j\right). \end{array}\right. \tag{18}$$

This writing can be summed up in the so-called *Butcher Tableau*:

$$\begin{array}{c|cccc} 0 & & 0 & & \\ c_2 & a_{21} & & & \\ \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & \cdots & a_{s,s-1} & \\ \hline & b_1 & \cdots & \cdots & b_s \end{array} \tag{19}$$

These methods are consistent with equation (17) when

$$c_i = \sum_j a_{ij}. \tag{20}$$

Then, under the additional constraint that $\sum b_j = 1$, the explicit numerical procedure is stable if the time step is smaller than a bound which depends on f Lipschitz constant.

In certain cases, especially when dealing with stiff multiscale problems, this stability constraint may be too harsh. One can overcome it by going implicit. The intermediate updates now become

$$k_i = f\left(t_n + c_i \Delta t, y_n + \Delta t \sum_j a_{ij} k_j\right), \tag{21}$$

and the Butcher tableau now looks like

$$\begin{array}{c|ccc} c_1 & & & \\ \vdots & & A & \\ c_s & & & \\ \hline & b_1 & \cdots & b_s \end{array} \tag{22}$$

where A is a full matrix. The price to pay for the increased range of stability is a interdependance between all the intermediate steps, which is not the case in the explicit version, thanks to the triangular shape of A. The solution at time $t + \Delta t$ is usually the solution of a big non-linear system.

The accuracy study of Runge-Kutta methods is more complex. All I want to say here is that it is known that starting from $s \geqslant 5$, there is no more RK method of order s.

2.4.2 Strong Stability Preserving integrators

In a fundamental paper published in 1988 [19], Chi-Wang Shu selects among all the RK methods a family of integrators he first calls "*Total Variation Diminishing time discretizations*", that will later be renamed as *Strong Stability Preserving* time discretizations in a review paper [14].

These methods can be seen as those which can be written as a convex combination of Explicit Euler (EE) time integrations:

$$y_{n+1} = \sum_{i=0}^{s-1} a_i y_{(i)} + \Delta t b_i f\left(t + i\Delta t, y_{(i)}\right). \tag{23}$$

Since the stability constraint of the EE scheme is 1.0, these methods are stable under the fact that the time step Δt is smaller than the Lipschitz constant of f time

$$\mathrm{CFL}_{\max} = \max_i \frac{a_i}{b_i}. \tag{24}$$

Next, Shu and Osher [20, 21] proved that up to fourth order, there exist an optimal SSP-RK integrator, meaning being both

- of the order of the number of stages,
- with the largest possible stability constraint:

$$\mathrm{CFL}_{\max} = 1.0.$$

Example 1 (*Heun's Method*)

This is the optimal second order SSP-RK integration :

$$\left\{ \begin{array}{rcl} y_{n+\frac{1}{2}} & = & y_n + \Delta t f\left(t, y_n\right), \\ y_{n+1} & = & \frac{1}{2}\left[y_n + \left(y_{n+\frac{1}{2}} + \Delta t f\left(t + \Delta t, y_{n+\frac{1}{2}}\right)\right)\right]. \end{array} \right. \tag{25}$$

2.5 Convex state preserving DG methods

2.5.1 Convex state preserving numerical flux

Let $\boldsymbol{\mathcal{S}} \subset \mathbb{R}^m$, convex, be the set of physical states.

Example 2 (*Convex Constraints*)

- ***Euler Equations:*** *Density and pressure have to stay positive. If the Equation Of State (EOS) follows the Bethe-Weil conditions, these conditions imply the convexity of the physical states*[2].
- ***Moments of a Repartition Function:*** *if the transported conserved quantities are the moments of a positive repartition function, then these moments need to stay moments of a positive distribution. In many cases, this implies convex constraints on the set of moments, [10].*

CSP Numerical Flux

A numerical flux $\mathbf{F}^*\left(\mathbf{W}^+, \mathbf{W}^-; \vec{\mathbf{n}}\right)$ is said to be **Convex State Preserving (CSP)**, when, for any states

$$\mathbf{W}_{i-1}^n, \mathbf{W}_i^n, \mathbf{W}_{i+1}^n \in \boldsymbol{\mathcal{S}},$$

the Explicit Euler update belongs to $\boldsymbol{\mathcal{S}}$,

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \nu\left[\mathbf{F}^*\left(\mathbf{W}_{i+1}^n, \mathbf{W}_i^n; \vec{\mathbf{n}}\right) - \mathbf{F}^*\left(\mathbf{W}_i^n, \mathbf{W}_{i-1}^n; \vec{\mathbf{n}}\right)\right] \in \boldsymbol{\mathcal{S}}, \tag{26}$$

under a CFL constraint:

$$\nu \leqslant C. \tag{27}$$

Example 3

- ***Rusanov Flux:***

$$\boldsymbol{F}^*\left(\boldsymbol{W}^+, \boldsymbol{W}^-; \vec{\mathbf{n}}\right) = \frac{\vec{\boldsymbol{\mathcal{F}}}(\boldsymbol{W}^+) + \vec{\boldsymbol{\mathcal{F}}}(\boldsymbol{W}^-)}{2} \cdot \vec{\mathbf{n}} - \alpha\left(\boldsymbol{W}^+ - \boldsymbol{W}^-\right). \tag{28}$$

  α being defined by (4).

- ***Godunov Flux:*** *since the Godunov flux is a convex combination of the physical states coming from the exact resolution of the Riemann problem at the interface, the update is physical and* (26) *comes true.*

- ***HLLx Solvers:*** *the same reasoning applies to HLL solvers, since the updated states are convex combinations of physical states, even though the resolution of the Riemann problem is approximated.*

Corrolary 4 (*SSP Methods*)

This naturally extends to SSP integrators, since they are convex combination of Explicit Euler updates. Only the CFL condition must be multiplied by the additional constraint (24)*. Hence the particular role of optimal SSP-RK integrators.*

2.5.2 Application to RK-DG methods

Recall the general update of all the degrees of freedom of an element T_k:

$$\mathcal{M}_{ij} d_t \mathbf{W}_j^k - \int_{\mathrm{T}_k} \vec{\boldsymbol{\mathcal{F}}}\left(\mathbf{W}(t, \boldsymbol{x})\right) \cdot \nabla \psi_i^k \, d\boldsymbol{x} + \int_{\partial \mathrm{T}_k} \psi_k^i(\boldsymbol{s}) \mathbf{F}^*\left(\mathbf{W}_{\mathrm{ext}}(t, \boldsymbol{s}), \mathbf{W}_{\mathrm{int}}(t, \boldsymbol{s}); \vec{\mathbf{n}}\right) \, d\boldsymbol{s} = 0. \tag{11}$$

Once summed up on all the DoFs, we obtain the equation ruling the *mean value* $\overline{\mathbf{W}_k}$:

$$|\mathrm{T}_k| \frac{d\overline{\mathbf{W}_k}}{dt} + \int_{\partial \mathrm{T}_k} \mathbf{F}^* \cdot \vec{\mathbf{n}} \, d\boldsymbol{s} = 0. \tag{29}$$

If the ODE (11) is integrated by Explicit Euler, the update simply gives:

$$\overline{\mathbf{W}_k}^{n+1} = \overline{\mathbf{W}_k}^{n} - \frac{\Delta t}{|\mathrm{T}_k|}\int_{\partial \mathrm{T}_k} \mathbf{F}^* \cdot \vec{\mathbf{n}}\, d\boldsymbol{s} = 0. \tag{30}$$

Note that by Corrolary 4, this discussion naturally extends to SSP integrators.

Now, let's switch to 1D and assume that the polynomial order is $\mathrm{P}_k - 1$, so that there exists a number Q of Gauss-Lobatto quadrature points, $(2Q-3) \geqslant (\mathrm{P}_k - 1)$, such that the following equality is exact:

$$\overline{\mathbf{W}_k} = \sum_{q=1}^{Q} \omega_q \mathbf{W}_k(x_q). \tag{31}$$

Then, discrete update (30) can be rewritten into

$$\begin{aligned}
\overline{\mathbf{W}_k}^{n+1} &= \sum_{q=1}^{Q} \omega_q \mathbf{W}_k^n(x_q) - \frac{\Delta t}{\Delta x_k}\Big\{ \mathbf{F}^*\left(\mathbf{W}_{k+1}^-, \mathbf{W}_k^+\right) - \mathbf{F}^*\left(\mathbf{W}_k^-, \mathbf{W}_{k-1}^+\right) \Big\}, \qquad (32)\\
&= \sum_{q=1}^{Q} \omega_q \left[\mathbf{W}_k^n(x_q) - \frac{\Delta t}{\omega_q \Delta x_k}\left\{ \mathbf{F}^*\Big(\mathbf{W}_k^n(x_{q+1}), \mathbf{W}_k^n(x_q)\Big) - \mathbf{F}^*\Big(\mathbf{W}_k^n(x_q), \mathbf{W}_k^n(x_{q-1})\Big) \right\} \right],
\end{aligned}$$

with the convention that $\mathbf{W}_k^n(x_{Q+1}) = \mathbf{W}_{k+1}^-$ and $\mathbf{W}_k^n(x_0) = \mathbf{W}_{k-1}^+$. In this form, we see that the DG update in the mean writes as a convex combination of abstract Euler Explicit updates at the Gauss-Lobatto quadrature points. So that

- if the numerical flux is CSP (26),
- if the solution at time t_n belongs to $\boldsymbol{\mathcal{S}}$ at all the Gauss-Lobatto quadrature points,

$$\forall q \in [\![0, Q+1]\!], \quad \mathbf{W}_k^n(x_q) \in \boldsymbol{\mathcal{S}}, \tag{33}$$

the updated mean value $\overline{\mathbf{W}_k}^{n+1}$ will be in $\boldsymbol{\mathcal{S}}$, under a CFL constraint

$$\nu \leqslant \omega_1 C, \tag{34}$$

since it is known that the smallest weights of the Gauss-Lobatto quadrature are always at the borders of the interval: $\omega_1 = \omega_Q = \min \omega_q$.

Starting from that, Zhang and Shu [26] provided a limitation procedure which was proven to conserve accuracy. Given that the updated mean value $\overline{\mathbf{W}_k}^{n+1}$ belongs to $\boldsymbol{\mathcal{S}}$ and that $\boldsymbol{\mathcal{S}}$ is a convex set, we now look at the values of the update solution at the Gauss-Lobatto quadrature points. If any of these values $\mathbf{W}_k^{n+1}(x_q)$ is outside $\boldsymbol{\mathcal{S}}$, there exists a unique value $\theta_q \in]0,1[$, such that $\theta_q \mathbf{W}_k^{n+1}(x_q) + (1-\theta_q)\overline{\mathbf{W}_k}^{n+1}$ is back on the frontier of $\boldsymbol{\mathcal{S}}$. By setting

$$\theta = \min_q \theta_q, \tag{35}$$

and

$$\widetilde{\mathbf{W}}_k^{n+1} = \theta \mathbf{W}_k^{n+1} + (1-\theta)\overline{\mathbf{W}_k}^{n+1}, \tag{36}$$

one obtains a new solution which is as accurate as $\mathbf{W}_k^{n+1}$ but limited in a way that (32) will propagate the convex constraint preservation further.

Remark 5

- *In a paper published on Arxiv in 2012, [16], Johnson and Rossmanith proved that such a limitation procedure on Gauss-Lobatto quadrature points is optimal in 1D.*
- *We have restrict the discussion to 1D because it is much simpler to explain in this context. However, the procedure generalizes to any dimension [27, 29, 16] but in a rather cumbersome manner. The interested reader is encouraged to read these articles and citation within.*

2.6 Limits of such procedure

Looking for a Strong Stability Preserving strategy or not, generalization of Runge-Kutta integration to higher orders is rather complex, since an increasing number of intermediate stages is needed to obtain an additional order in time. Moreover, all these intermediate stages generally need to be stored for the final t^{n+1} update. This may be limiting in term of memory usage.

On the top of that, the global time step of the method is chosen a priori and kept during the whole multi-step Runge-Kutta process. Even if a security coefficient is applied globally on the time step, the local physics may evolve rapidly within the time step and the stability constraint on the time progress may be a posteriori violated. This risk increasing with the number of substeps.

These reasons explain why people start to turn to space-time formulation. A fully space-time formulation of DG methods is conceivable but finally looks like a huge implicit formulation: all the degrees of freedom of the mesh are coupled, especially with non-linear equations.

On the other hand, arbitrary high order one-step explicit space-time have appeared under the name of ADER (for **A**rbitrary high-order schemes using **DER**ivatives). Even though these formulations are intrinsically space-time, the volume and flux terms (2) and (3), in (8), are in fact somehow extrapolated from the available information at time t^n and the method can be considered as one-step and explicit. This is the topic of the next section.

3 ADER-DG integration

ADER-DG methods always start by a space-time integration of a weak formulation in space of equation (2).

$$\int_{t^n}^{t^{n+1}} \left[\int_{\mathrm{T}_k} \left(\frac{\partial \mathbf{W}}{\partial t} + \overrightarrow{\nabla} \cdot \vec{\boldsymbol{\mathcal{F}}}(\mathbf{W}) \right) \cdot \psi_i^k(\mathbf{X})\, d\boldsymbol{x} \right] dt = 0, \tag{37}$$

$$\Leftrightarrow$$

$$\mathcal{M}_{ij} \left(\mathbf{W}_j^{k,n+1} - \mathbf{W}_j^{k,n} \right) - \underbrace{\int_{t^n}^{t^{n+1}} \int_{\mathrm{T}_k} \vec{\boldsymbol{\mathcal{F}}}(\mathbf{W}) \cdot \nabla \psi_i^k \, d\boldsymbol{x} dt}_{\textcircled{5}} + \underbrace{\int_{t^n}^{t^{n+1}} \int_{\partial \mathrm{T}_k} \psi_i^k \mathbf{F}^*(t, \vec{\mathbf{n}})\, d\boldsymbol{s}\, dt}_{\textcircled{6}} = 0.$$

In the two next sections, the goal is to set up a method to obtain the necessary data needed to integrate the two integrals (5) and (6) in last equation. In a first procedure, the time evolution is gotten from an arbitrary high order Taylor expansion in time of the conserved variables. Thanks to the *Cauchy-Kowaleski* procedure (also known as *Lax-Wendroff* procedure), the time derivatives of $\mathbf{W}$ are functions of the spatial derivatives of the fluxes, which are known at time t^n. A *Generalized Riemann Problem* (GRP) is solved at each interface and the problem can be moved to next time step.

In a second procedure, a predictor step is ran in the form of a *local space-time DG scheme*. This allows to get a local information on the evolution of the data. The predicted solution is then used to compute the space-time flux and volume terms of equation (37).

3.1 Cauchy-Kowaleski ADER procedure

3.1.1 Generalized Riemann Problem

Most of this paragraph takes its roots in chapter 19 and 20 of Toro's book [22].

In a general context, the numerical flux needed to compute last integral (6) of equation (37) comes from the (possibly approximate) resolution of a Riemann problem:

$$\begin{cases} \partial_t \mathbf{W} + \partial_{\vec{\mathbf{n}}} \vec{\boldsymbol{\mathcal{F}}}(\mathbf{W}) = 0, \quad t > 0,\ \vec{\boldsymbol{x}} \cdot \vec{\mathbf{n}} \in \mathbb{R}, \\ \\ \mathbf{W}(t = 0, \vec{\boldsymbol{x}} \cdot \vec{\mathbf{n}} < 0) = \mathbf{W}^-, \\ \mathbf{W}(t = 0, \vec{\boldsymbol{x}} \cdot \vec{\mathbf{n}} > 0) = \mathbf{W}^+. \end{cases} \tag{38}$$

By autosimilarity, this solution depends only on one variable $\xi = \vec{\boldsymbol{x}} \cdot \vec{\mathbf{n}}/t$ and the numerical flux usually writes

$$\mathbf{F}^*(\mathbf{W}^+, \mathbf{W}^-; \vec{\mathbf{n}}) = \vec{\boldsymbol{\mathcal{F}}}(\mathbf{W}^*(\xi = 0)) \cdot \vec{\mathbf{n}}, \tag{39}$$

where $\mathbf{W}^*(\xi = 0)$ is the values taken by the (possibly approximate) solution of the Riemann Problem along the ordinate axis.

At higher order in space, input values $\mathbf{W}^+$ and $\mathbf{W}^-$ are fed with the limit values at the interface of the polynomial in each cell: $\mathbf{W}_{k+1}^n(x_{i+\frac{1}{2}}^+)$ and $\mathbf{W}_k^n(x_{i+\frac{1}{2}}^-)$. Therefore a first order error in time comes from the fact that the Riemann solvers see a constant extrapolation of $\mathbf{W}^+$ and $\mathbf{W}^-$ in the neighboring cells.

This stimulated people to look at *Generalized Riemann Problems* (GRP) where the numerical flux would now be:

$$\mathbf{F}^*\left(t;\mathbf{W}^+(x>0),\mathbf{W}^-(x<0);\vec{\mathbf{n}}\right)=\vec{\boldsymbol{\mathcal{F}}}\left(\mathbf{W}^*(t,x=0)\right)\cdot\vec{\mathbf{n}},\quad t\in]0,\Delta t[. \tag{40}$$

The accuracy in time would then be achieved by a Taylor expansion of $\mathbf{W}$ in time along the ordinate axis

$$\mathbf{W}^*(t,x=0)=\mathbf{W}^*(0^+,x=0)+\sum_{k=1}^{N}\frac{t^k}{k!}\frac{\partial^k\mathbf{W}^*}{\partial t^k}(0^+,x=0)+\mathcal{O}\left(t^{N+1}\right), \tag{41}$$

and to look at an accurate enough way to solve all the unknowns (meaning the successive time derivatives at $t=0^+$). It came out that the problem can be split into N sub-Riemann problems, one being the full space consistent Riemann problem (38), called the *space high-order Riemann problem*, the $(N-1)$ others being rather simple.

In fact, since $\mathbf{W}$ is the solution of (2), its k^{th}-order time derivatives can be written as a function of its spatial derivatives up to order k:

$$\partial_t^{(k)}\mathbf{W}(t,x)=\mathscr{G}\left(\mathbf{W}(t,x),\partial_x^{(1)}\mathbf{W}(t,x),\ldots,\partial_x^{(k)}\mathbf{W}(t,x)\right). \tag{42}$$

Now, one can derivate problem (38) k times in space to get a Riemann problem on $\partial_x^{(k)}\mathbf{W}(t,x)$. In fact, since only the solution at $(t=0^+,x=0)$ is needed, the associated Riemann problem can be simplified into

$$\begin{cases}\partial_t\left(\partial_x^{(k)}\mathbf{W}(t,x)\right)+\mathbf{J}\Big(\mathbf{W}^*(t=0^+,x=0)\Big)\partial_x\left(\partial_x^{(k)}\mathbf{W}(t,x)\right)=0,\\ \\ \partial_x^{(k)}\mathbf{W}(t=0,x)=\begin{cases}\partial_x^{(k)}\mathbf{W}(t=0,x=0^-)\ \text{ if }\ x<0,\\ \partial_x^{(k)}\mathbf{W}(t=0,x=0^+)\ \text{ if }\ x>0,\end{cases}\end{cases} \tag{43}$$

where $\mathbf{W}^*(t=0^+,x=0)$ is the solution of the space high-order Riemann problem and $\mathbf{J}$ is the Jacobian of the flux at this state.

3.1.2 Integration within DG formulation

Once this is done, the Taylor expansion (41) can be filled up to desired order N thanks to the Lax-Wendroff expansions (42), and the numerical flux is gained as the exact integration on $t\in[0,\Delta t]$ of

$$\vec{\boldsymbol{\mathcal{F}}}\Big(\mathbf{W}^*(t,x=0)\Big).$$

Next, the volume integral (5) needs to be evaluated. This is generally done by a numerical quadrature at a sufficient order of accuracy:

$$\int_{t^n}^{t^{n+1}}\int_{\mathrm{T}_k}\vec{\boldsymbol{\mathcal{F}}}\left(\mathbf{W}(t,\boldsymbol{x})\right)\cdot\nabla\psi_i^k\,d\boldsymbol{x}dt=\sum_{t_s=1}^{N_t}\sum_{\boldsymbol{x}_q=1}^{N_x}\omega_s\omega_q\Delta t\Delta x\vec{\boldsymbol{\mathcal{F}}}\left(\mathbf{W}(t_s,\boldsymbol{x}_q)\right)\cdot\nabla\psi_i^k(t_s,\boldsymbol{x}_q). \tag{44}$$

But the solution is supposed to be regular within the cell (since it is locally approximated by a polynomial), so that the Taylor and Lax-Wendroff expansions (41) and (42) still

apply at any spatial quadrature point $\boldsymbol{x}_q$. The same procedure can then be applied at every spatial quadrature point and the whole method is evaluated everywhere and each degree of freedom is updated thanks to (37).

For a general algorithm to obtain the successive time derivative as in (42), in particular in the context of the Euler flux, see [13] and references therein.

3.2 Local space-time predictor ADER procedure

As we can see in the previous section, the Taylor expansion in time of the solution may lead to extremely complex moving from known spatial derivatives to time derivatives through Lax-Wendroff recursive expansion. Especially in the context of non-linear fluxes, or even worse, when the flux is an unknown black box...

Anyway, another much more straightforward method is under development. As far as I know, this method really starts in [12] in the finite volume context and has recent applications in the DG context [24, 13] (clearly not a throughful bibliography).

The idea here is that, in order to compute the time dependant numerical flux (6) in (37), only the outgoing information from each cell is needed. So, on a local space-time mesh, the solution can be evolved within each cell independantly. A predictor solution $\mathrm{Q}_k^n(t,x)$ is obtained and is next used to compute integrals (5) and (6).

Now, we restrict our study to the space-time slab $\mathrm{T}_k^n = \mathrm{T}_k \times [t^n, t^{n+1}]$ and suppose that two functional basis $(\psi_s^t)_{s=1,\dots,N_t}$ and $\left(\psi_q^x\right)_{q=1,\dots,N_x}$ are respectively defined on $[t^n, t^{n+1}]$ and T_k, see sections 2.1 and 2.3, such that Q_k^n expands as

$$\mathrm{Q}_k^n(t,\boldsymbol{x}) = \sum_{s=1}^{N_t}\sum_{q=1}^{N_x} \mathrm{Q}_k^{s,q}\psi_s^t(t)\psi_q^x(\boldsymbol{x}). \tag{45}$$

Then Q_k^n is supposed to locally verify the space-time problem

$$\begin{cases} \partial_t \mathrm{Q}_k^n + \overrightarrow{\nabla}\cdot\vec{\boldsymbol{\mathcal{F}}}\left(\mathrm{Q}_k^n\right) = 0, & \text{in } \mathrm{T}_k\times]t^n, t^{n+1}], \\ \mathrm{Q}_k^n(t=0,\boldsymbol{x}) = \mathbf{W}_k^n(\boldsymbol{x}), & \text{in } \mathrm{T}_k, \\ \mathrm{Q}_k^n(t,\boldsymbol{x}) = \mathrm{Q}_k^n(t,\boldsymbol{x}), & \text{on } \partial\mathrm{T}_k. \end{cases} \tag{46}$$

The last boundary condition is a little bit mysterious and corresponds to a certain personal point-of-view...

Next, a local space-time DG formulation is led with an integration by part only in the time direction:

$$\begin{aligned} &-\int_{\mathrm{T}_k^n} \mathrm{Q}_k^n(t,\boldsymbol{x})\partial_t\psi_s^t(t)\psi_q^x(\boldsymbol{x})\,d\boldsymbol{x}dt \\ &+\int_{\mathrm{T}_k} \left(\mathrm{Q}_k^n(t^{n+1},\boldsymbol{x})\psi_s^t(t^{n+1})\psi_q^x(\boldsymbol{x}) - \mathrm{Q}_k^n(t^n,\boldsymbol{x})\psi_s^t(t^n)\psi_q^x(\boldsymbol{x})\right) d\boldsymbol{x} \\ &+\int_{\mathrm{T}_k^n} \overrightarrow{\nabla}\cdot\vec{\boldsymbol{\mathcal{F}}}\left(\mathrm{Q}_k^n\right)\psi_s^t(t)\psi_q^x(\boldsymbol{x})\,d\boldsymbol{x}dt = 0, \quad \forall s\in[\![1,N_t]\!], q\in[\![1,N_x]\!]. \end{aligned} \tag{47}$$

If Q_k^n and $\vec{\boldsymbol{\mathcal{F}}}(\mathrm{Q}_k^n)$ are supposed to be spanned by the $\psi_s^t\psi_q^x$ space-time basis, and the vector of local unknown $\mathrm{Q}_k^{s,q}$ is supposed to be ordered by time layers:

$$\mathbb{Q} = \left(\mathrm{Q}_k^{s,q}\right)_{s=1,\dots,N_t,\ q=1,\dots,N_x} = \left(\mathrm{Q}_k^{1,1},\dots,\mathrm{Q}_k^{1,N_x},\dots,\mathrm{Q}_k^{N_t,1},\dots,\mathrm{Q}_k^{N_t,N_x}\right)^T, \tag{48}$$

where superscript $(.)^T$ stands for transposition, then equation (47) can be rewritten into

$$-\left[\left(\mathcal{K}^t\right)^T \otimes \mathcal{M}^x\right]\mathbb{Q} + \left[\mathbb{1}^t_{n+1} \otimes \mathcal{M}^x\right]\mathbb{Q} - \left[\mathbb{1}^t_n \otimes \mathcal{M}^x\right]\mathbb{Q} + \left[\mathcal{M}^t \otimes \vec{\mathcal{K}}^x\right]\vec{\boldsymbol{\mathcal{F}}}(\mathbb{Q}) = 0,$$

$$\Leftrightarrow$$

$$\left[\left((\mathcal{K}^t)^T - \mathbb{1}^t_{n+1} + \mathbb{1}^t_n\right) \otimes \mathcal{M}^x\right]\mathbb{Q} = \left[\mathcal{M}^t \otimes \vec{\mathcal{K}}^x\right]\vec{\boldsymbol{\mathcal{F}}}(\mathbb{Q}), \tag{49}$$

with the following matrices:

$$\mathcal{M}^x_{i,j} = \int_{\mathrm{T}_k} \psi^x_i(\boldsymbol{x})\psi^x_j(\boldsymbol{x})\, d\boldsymbol{x}, \tag{50}$$

$$\mathcal{M}^t_{i,j} = \int_{t^n}^{t^{n+1}} \psi^t_i(t)\psi^t_j(t)\, dt, \tag{51}$$

$$\mathcal{K}^t_{i,j} = \int_{t^n}^{t^{n+1}} \psi^t_i(t)\partial_t\psi^t_j(t)\, dt, \tag{52}$$

$$\left(\vec{\mathcal{K}}^x_{i,j}\right)_l = \int_{\mathrm{T}_k} \psi^x_i(\boldsymbol{x})\partial_{x_l}\psi^x_j(\boldsymbol{x})\, d\boldsymbol{x} \tag{53}$$

$$(\mathbb{1}^t_\xi)_{i,j} = \int_{t^n}^{t^{n+1}} \psi^t_i(t^\xi)\psi^t_j(t)\, dt. \tag{54}$$

We have got a local non-linear system which needs to be solved. Fortunately, according to [11], the process would be always contractant and the solution can be reached within a few iterations per cells.

Once this is done, the predictor Q^n_k is simply used in (5) and (6) to update equation (37):

$$\forall \mathrm{T}_k \in \mathcal{M}_h, \forall i \in \mathrm{T}_k, \tag{55}$$

$$\mathcal{M}_{ij}\left(\mathbf{W}^{k,n+1}_j - \mathbf{W}^{k,n}_j\right) - \int_{\mathrm{T}^n_k} \vec{\boldsymbol{\mathcal{F}}}\left(\mathrm{Q}^n_k\right)\cdot\nabla\psi^k_i\, d\boldsymbol{x}dt + \int_{t^n}^{t^{n+1}}\int_{\partial\mathrm{T}_k} \psi^k_i \mathbf{F}^*(t;\mathrm{Q}^{\mathrm{ext}},\mathrm{Q}^{\mathrm{int}};\vec{\mathbf{n}})\, d\boldsymbol{s}\, dt = 0.$$

References

[1] R. Abgrall. Toward the ultimate conservative scheme: Following the quest. *J. Comput. Phys*, 167(2):277–315, 2001.

[2] Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, May 2001.

[3] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations. *Journal of Computational Physics*, 131(2):267 – 279, 1997.

[4] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2d euler equations. *Journal of Computational Physics*, 138(2):251 – 285, 1997.

[5] B. Cockburn and Chi-Wang Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Mathematics of Computation*, 52(186):411–435, 1989.

[6] B. Cockburn and Chi-Wang Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws iv: the multidimensional case. *Mathematics of Computation*, 54:545–581, 1990.

[7] B. Cockburn and Chi-Wang Shu. The runge-kutta local projection p^1-discontinuous-galerkin finite element method for scalar conservation laws. *M2AN*, 25(3):337–361, 1991.

[8] B. Cockburn and Chi-Wang Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V - multidimensional systems. *Journal of Computational Physics*, 141(2):199–124, 1998.

[9] Bernardo Cockburn, San-Yih Lin, and Chi-Wang Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws iii: One-dimensional systems. *Journal of Computational Physics*, 84(1):90 – 113, 1989.

[10] H. Dette and W. J. Studden. *The theory of canonical moments with applications in statistics, probability, and analysis*. John Wiley & Sons Inc., New York, 1997.

[11] Michael Dumbser, Dinshaw S. Balsara, Eleuterio F. Toro, and Claus-Dieter Munz. A unified framework for the construction of one-step finite volume and discontinuous galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227(18):8209 – 8253, 2008.

[12] Michael Dumbser, Cedric Enaux, and Eleuterio F. Toro. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *Journal of Computational Physics*, 227(8):3971 – 4001, 2008.

[13] Michael Dumbser and Claus-Dieter Munz. Building blocks for arbitrary high order discontinuous galerkin schemes. *Journal of Scientific Computing*, 27(1):215–230, 2005.

[14] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, 2001.

[15] Jan S. Hestaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*. Springer-Verlag, 2008.

[16] Evan Alexander Johnson and James A. Rossmanith. Outflow positivity limiting for hyperbolic conservation laws. part i: Framework and recipe. submitted Dec 19, 2012, `http://arxiv.org/abs/1212.4695v1`.

[17] Christoph Ortner and Endre Süli. Discontinuous galerkin finite element approximation of nonlinear second-order elliptic and hyperbolic systems. *SIAM Journal on Numerical Analysis*, 45(4):1370–1397, 2007.

[18] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. In *National topical meeting on mathematical models and computational techniques for analysis of nuclear systems*, Oct 1973.

[19] Chi-Wang Shu. Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.*, 9(6):1073–1084, November 1988.

[20] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439 – 471, 1988.

[21] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, {II}. *Journal of Computational Physics*, 83(1):32 – 78, 1989.

[22] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, New York, 1999.

[23] B. van Leer. Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method. *J. Comput. Phys.*, 32:101–136, 1979.

[24] Olindo Zanotti, Francesco Fambri, Michael Dumbser, and Arturo Hidalgo. Space–time adaptive ader discontinuous galerkin finite element schemes with a posteriori sub-cell finite volume limiting. *Computers & Fluids*, 118:204 – 224, 2015.

[25] X. Zhang. *Maximum-Principle-Satisfying and Positivity-Preserving High Order Schemes for Conservation Laws*. PhD thesis, Brown University, 2011.

[26] Xiangxiong Zhang and Chi-Wang Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.*, 229(9):3091–3120, May 2010.

[27] Xiangxiong Zhang and Chi-Wang Shu. On positivity-preserving high order discontinuous galerkin schemes for compressible euler equations on rectangular meshes. *J. Comput. Phys.*, 229(23):8918–8934, November 2010.

[28] Xiangxiong Zhang and Chi-Wang Shu. Positivity-preserving high order discontinuous galerkin schemes for compressible euler equations with source terms. *J. Comput. Phys.*, 230(4):1238–1248, February 2011.

[29] Xiangxiong Zhang, Yinhua Xia, and Chi-Wang Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous galerkin schemes for conservation laws on triangular meshes. *J. Sci. Comput.*, 50(1):29–62, January 2012.