Bibliographic Report
Development of low-order shock-capturing scheme for the discontinuous Galerkin method

Sana Amri[*]

Supervisor: Prof. Thierry Magin

* Aeronautic and Aerospace Department

Introduction

At a given altitude, when an object moves through the air at a speed which is way more faster than the speed of sound some critical phenomena may occur near the shock wave. The static pressure, the temperature and the gas density increase brutally. The wind tunnel experiment showed that the shape of the body [1] have a considerable impact on the flowfield behaviour. In the case of a blunt body , the supersonic wind tunnel experiment [2] showed that a strong detached shock wave appears in front of the blunted nose with a very high gas temperature between the nose and the shock wave. Beyond the observations made, ones need to have more information of the flowfield (such as the heat transfer, the pressure and the velocity distribution over the body) in order to have a better understanding of the flowfield. As pointed out by Anderson in [3], no general theoritical analysis of supersonic (and hence hypersonic) flow exists up to now. That is why, the field of computational fluid dynamics (CFD), that can be seen as a bridge between theoritical mathematical analysis and experimental method, became fondamental in fluid mechanics. The main purpose of CFD is to build extremely precise simulations of complex flowfield to have a better understanding of the observations made in the facilities. Notice that these simulations require powerful computer resources (hence costs) that pushes the CFD field to think numerical methods not only in terms of stability and consistency but also in terms of computational cost. The following bibliographic study will be oriented to the numerical methods and shock capturing techniques implemented in the Argo software on which the research project is based.
Argo is a multi-physics CFD platform initially developed by Hilleweart [4]. The code is now developed at Cenaero, a research centre in Belgium. It is based on a discontinuous Galerkin method (DGM), to be presented in Section 2.1, and can handle both structured and unstructured meshes[5]. Furthermore, it is designed to run on parallel architectures, using a hybrid of Open Multi-Processing (Open MP) and Message Passing Interface (MPI) libraries.

1 Physical Phenomena

1.1 Hypersonic flow challenges

> "Mach number is like an aborigine counting: one two, three, four, many. Once you reach many, the flow is hypersonic" (1969, Rice University)

The statement of H.K. Beckmann [6] regarding hypersonic flows is often quoted because it highlights the lacking of knowledge of these types of flowfield in the scientific community.

In his book [3], Anderson defines the hypersonic flow as a regime where some of the listed phenomena become important proportionally with the increase of the Mach number.

- Thin shock layer: When a hypersonic flow invests a body, the distance between the generated shock wave and the body is smaller compared to supersonic conditions.
- Entropy layer: Across a shock wave, the entropy tends to increase brutally at . This phenomenon is a source of analytical problems when we wish to perform a standard boundary layer calculation.
- Viscous interaction: The viscosity coefficient increases with temperature which leads to thicker boundary layer (viscous dissipation).
- High-temperature: The severe post-shock conditions can create a very high heat that can cause dissociation and even ionization within the gas. The nose region of a blunt body is a high-temperature flow behind the strong shock wave.
- Low-density flow: At high altitude (above 92km), the assumption of a continuous medium to apply the Navier- Stokes equations becomes tenuous therefore the Boltzmann equations should be considered.

1.2 Shock Capturing and Shock Fitting

As previously mentioned, one of the challenges of hypersonic flows is to localise precisely the thin shock region.

The state of the art reports two main methods [7–10] to tackle this problem by CFD.

The shock capturing method consist in letting appear naturally within the computational space the shock region without any specific treatment of the shocks themselves. The domain is computed as a uniform flowfield and the zone concerned by drastic discontinuous changes in the primitive variables (pressure, density, velocity and temperature). The region of the shock can be enhanced by using artificial viscosity and limiting in order to thickened the shock region and reduce the spurious oscillations (Gibbs phenomenon).

The shock fitting method treats the shock region independantly from the flowfield by treating the shock as an internal boundary condition. The governing flow equations (Navier-Stokes) are used to calculate all the flowfield except the shock region that uses the exact Rankine-Hugoniot relation.

Both of the methods have their own limits. For the shock capturing method, the calculation of all the flow field without taking into account the sharp discontinuous change of the primitive variables may crash the simulation. That is why, the conservation form of the governing equation that resolve the unknowns in terms of flux are commonly used.
Even if the shock fitting technique allows to resolve shocks accurately, many issues related to the mesh make this technique difficult to use in practise. One of them beeing the difficulty to obtain a perfect alignment of the mesh with the discontinuity.

2 Numerical treatment of the Gouverning equations

Let's recall the hypervectorial form of the Navier Stokes (conservative form).

$$\partial_t \boldsymbol{U} + \nabla.\boldsymbol{F}_c(\boldsymbol{U}) = \nabla.\boldsymbol{F}_v(\boldsymbol{U}, \nabla\boldsymbol{U}) + S(\boldsymbol{U}, \nabla\boldsymbol{U}) \quad (1)$$

$$\boldsymbol{U} = [\rho, \rho u_i, \rho E]^T, \quad \boldsymbol{F}_c(\boldsymbol{U}) = \begin{bmatrix} \rho u_i \\ \rho u_i u_j + \delta_{ij} p \\ \rho u_i H \end{bmatrix},$$

$$\boldsymbol{F}(\boldsymbol{U}, \nabla\boldsymbol{U}) = \begin{bmatrix} 0 \\ \tau_{ij} \\ u_j \tau_{ij} + k_T \frac{\partial T}{\partial x_i} \end{bmatrix},$$

with $\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \delta_{ij} \lambda \frac{\partial u_k}{\partial x_k}$

2.1 Space discretisation

2.1.1 The Discontinuous Galerkin Method

The discontinuous Galerkin method (DGM) [5, 11–16] of order p approximate each control volume of the domain Ω by a polynomial of order inferior or equal to p. This method join the high mesh adaptability of the finite volume method (FVM) with the polynomial approximation of each cells of the finite element method (FEM).
For the DGM, the polynomial functions are functions of $L^2(\Omega)$ ie they are defined in Ω and their respective 2^{nd}power are Lebesgue integrable.
Notice that they are not continuous in all the domain as in the FEM. Instead, the solution in each cell K_i (which form a partition of $\Omega = \cup_{i \in N} K_i$) is approximate by a combination of p+1 polynomials $\{\phi_i^{(0)}, \phi_i^{(1)}, \ldots, \phi_i^{(p)}\}$ defined as follow.
For $k \in$⟦0,p⟧,

$$\phi_i^{(k)}(x, y, z) = \begin{cases} P_k(t, x, y, z) & in\ K_i \\ \\ 0 & otherwise \end{cases}$$

where P_k is a generic notation that denotes a polynomial of order k.

These functions define a functional space (vectorial space as a close space of $L^2(\Omega)$) of the entire domain. The restricted solution to K_i is therefore a linear combinaison of these functions:
$u_{|K_i}(t, x, y, z) = \sum_{k=0}^{p} a_i^{(k)}(t) \phi_i^{(k)}(x, y, z)$

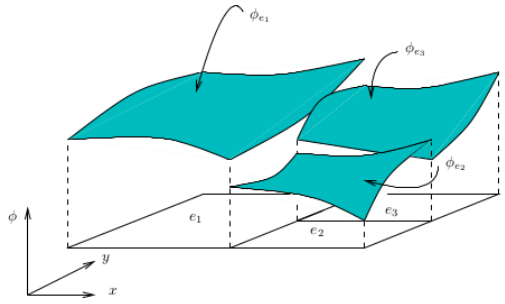

Figure 1 Scheme from [6]

The conservativity of the flux [**6**] allows (as the FVM) to regularize the values at each interface of the mesh. In theory all types of elements are allowed, as long as their union covers the entire domain without overlapping. In Argo the element types are restricted to triangles and quadrilaterals in two dimensions, and tetrahedra, prisms, hexahedra and pyramids in three dimensions. Moreover conformal meshes are used, meaning that any boundary face of the element can only connect to a single other element.

Notice that, as showed in my report [17] , in the case of the scalar advection equation [18] $\partial_t u(t, x) + v \partial_x u(t, x) = 0$, the variational formulation will give a system of $p \times N^2$ integral form equations to solve (that can be reduced to $p \times N$ [17]).
The solution u_h will be the sum of the approximate

solutions $u_{|K_i}$ at each cell obtained by the system.
To not overwhelm the reader, let's consider the variational formulation in the case of a scalar version of the system (1).

The weak formulation of the DGM of order $p = 0$ reads (after summing the N equations of the system):

$$\sum_{i=1}^{N}\int_{K_i}\phi_i^{(0)}\partial_t u_{|K_i}dV+\sum_{i=1}^{N}\int_{K_i}\phi_i^{(0)}\partial_x F_c(u_{|K_i})dV = \sum_{i=1}^{N}\int_{K_i}\phi_i^{(0)}\partial_x F_v(u_{|K_i},\partial_x u_{|K_i})dV \tag{2}$$

2.1.2 Inviscid Flux Discretisation

The inviscid term F is discretised by integrating by parts and approximating F with a numerical flux $\hat{F}$ on each boundary ∂K_i, which results to:

$$\sum_{i=1}^{N}\int_{K_i}\phi_i^{(0)}\frac{\partial F(u_{|K_i})}{\partial x}dV = -\sum_{i=1}^{N}\int_{K_i}F(u_{|K_i})\frac{\partial\phi_i^{(0)}}{\partial x}dV + \sum_{i=1}^{N}\int_{\partial K_i}\hat{F}(u_{|K_i}{}^{+},u_{|K_i}{}^{-},\boldsymbol{n})[\phi_i^{(0)}]dS$$

with $[\bullet]$ being the jump operator such that,

$$[v] = v^- n^- + v^+ n^+$$

with the convention that + is for quantities approaching the interface in the direction of its normal and is associated with direction opposite to the normal.
In the case considered, the first term on the right-hand side disappears and the method becomes effectively a first-order FVM. The spatial discretisation of the inviscid flux in DGM can thus be seen as a high-order adaptation of FVM. Inviscid fluxes $\hat{\mathrm{F}}$ are determined by solving a Riemann problem at each element interface [CFCM course]. Two main categories of approximate Riemann solvers exist [7, 19, 20]:

- **Flux Difference Splitting** (FDS). The flux difference is split in two parts, one associated to the downstream travelling wave and the other with the upstream travelling one, as follows:
  $\boldsymbol{F}(\boldsymbol{U}_R)-\boldsymbol{F}(\boldsymbol{U}_L) = (\Delta\boldsymbol{F})^+(\boldsymbol{U}_L,\boldsymbol{U}_R)+(\Delta\boldsymbol{F})^-(\boldsymbol{U}_L,\boldsymbol{U}_R)$
- **Flux Vector Splitting** (FVS). The flux vector is split in a sum of two vectors, again each associated with the downstream or upstream travelling wave:
  $\boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{F}^+ + \boldsymbol{F}^-$

Hybrid flux splittings have appeared with the intention to combine the benefits of both categories. Specifically, the goal was to take the robustness of the FVS, the ability of selecting only entropy-satisfying approximate solutions of FVS, and the ability of capturing stationary contact discontinuities exactly of FDS. The most popular family of hybrid schemes is the Advection Upstream Splitting Method (AUSM), for which more information can be found in [10].

2.1.3 Viscous Flux Discretisation

Various techniques exist for the DGM to discretise the viscous fluxes F_v [21]. Two well-known methods are the interior penalty (IP) method and the scheme of Bassi and Rebay BR2 [2]. Only the IP method was implemented in Argo until recently. The effect of the IP scheme will be tested for the research project.

The **Interior Penalty** method is based on the boundary penalty approach from Nitsche [7]. Values at the boundary are imposed in a weak manner with the use of a penalty term that minimises the average interpolation error between the value of the solution and the value of the boundary condition. This leads to a lower global error due to smoother variation of the solution.

$$\begin{aligned}\sum_{i=1}^{N}\int_{K_i}\phi_i^{(0)}\partial_x F_v(u_{|K_i},\partial_x u_{|K_i})dV &= \sum_{i=1}^{N}\int_{K_i}\phi_i^{(0)}\partial_x\left(D\partial_x u_{|K_i}\right)dV\\ &= \sum_{i=1}^{N}\int_{K_i}\left(D\partial_x u_{|K_i}\right)\partial_x\phi_i^{(0)}dV\\ -\sum_{i=1}^{N}\int_{\partial K_i}\left\langle D(u_{|K_i})\partial_x u_{|K_i}\right\rangle\left[\phi_i^{(0)}\right]dS\\ -\theta\sum_{i=1}^{N}\int_{\partial K_i}\left\langle D(u_{|K_i})\partial_x\phi_i^{(0)}\right\rangle\left[u_{|K\,i}\right]dS\\ +\alpha\sum_{i=1}^{N}\int_{\partial K_i}\left[\phi_i^{(0)}\right]\left[u_{|K_i}\right]dS\end{aligned}$$

where $\langle\bullet\rangle = \frac{1}{2}\left(\bullet^- + \bullet^+\right)$

On the right hand side, the first two parameters are the result of the discretisation by parts. The third and forth term on the right hand side are the penalty and symmetric terms, respectively, and ensure the method is stable. The penalty parameter θ needs to be chosen within a bounded interval, for which the limits can be found in the work of Hilleweart. The parameter θ has three options {1,-1,0}, see [7].

A key drawback of the Interior Penalty method is that it is inconsistent at p = 0. Indeed, all the derivatives on the right-hand side become zero, cancelling all terms except the penalty one.

A popular method which is consistent at p = 0 is the **second scheme of Bassi and Rebay** (BR2) which is a modification of the first scheme of Bassi Rebay (BR1). In the weak formulation (2), the spatial discretisation of

the viscous term of the Navier–Stokes equations is constructed by resorting to a mixed finite element formulation. The first-order derivatives of the conservative variables appearing in the vector $\boldsymbol{F}_v$ in (1) lead to second-order derivatives when ones evaluate the divergence of the viscous fluxes. However, second-order derivatives cannot be accommodated directly in a weak variational formulation using a discontinuous function space (Hilbert space). The rewritting of the gradient of the conservative variables $\nabla \boldsymbol{U} = \boldsymbol{S}(\boldsymbol{U})$ leads to the following coupled system for the unknowns $\boldsymbol{S}$ and $\boldsymbol{U}$,

$$\begin{aligned} \boldsymbol{S} - \nabla \boldsymbol{U} &= 0 \\ \partial_t \boldsymbol{U} + \nabla . \boldsymbol{F}_c(\boldsymbol{U}) + \nabla . \boldsymbol{F}_v(\boldsymbol{U}, \boldsymbol{S}) &= 0 \end{aligned} \tag{3}$$

System (3) can be approximated by means of a discontinuous finite element formulation in a way similar to that employed for the inviscid part of the equations. The use of an explicit time-stepping scheme greatly simplifies the mixed finite element formulation, since it allows a decoupled solution of system (3). At each time step n, in fact, we first compute a discontinuous approximation of $\boldsymbol{S}^n$ by solving the first equation of the system and then use $\boldsymbol{U}^n$ and $\boldsymbol{S}^n$ to evaluate the inviscid and viscous fluxes of the second equation which is then advanced in time.

Let's reconsider the previous scalar case. The coupled system reads:

$$\begin{aligned} s_{|K_i} - \partial_x u_{|K_i} &= 0 \\ \partial_t u_{|K_i} + \partial_x F_c(u_{|K_i}) + \partial_x F_v(u_{|K_i}, \partial_x u_{|K_i}) &= 0 \end{aligned} \tag{4}$$

The weak formulation of the first equation of (4) is:

$$\sum_{i=1}^{N} \int_{K_i} \phi_i^{(0)} s_{|K_i} dV - \sum_{i=1}^{N} \int_{\partial K_i} \phi_i^{(0)} u_{|K_i} n dS + \sum_{i=1}^{N} \int_{K_i} \left(\partial_x \phi_i^{(0)} \right) u_{|K_i} dV = 0$$

where, due to the discontinuous function approximation at internal interfaces, the unknowns $u_{|K_i}$ appearing in the boundary integral is not uniquely defined. In analogy with the procedure described for the inviscid part of the equations, it is therefore necessary to introduce a numerical flux function $\hat{F}_1$.

$$\hat{F}_1(u^-, u^+, n) = \langle u \rangle .n$$

The computed auxiliary variables $s_{|K_i}$ are then used in the weak form of the second equation of system (4),

$$\begin{aligned} 0 = & \sum_{i=1}^{N} \int_{K_i} \phi_i^{(0)} \partial_t u_{|K_i} dV + \sum_{i=1}^{N} \int_{\partial K_i} \phi_i^{(0)} \partial_x F_c(u_{|K_i}) . n dS \\ & - \sum_{i=1}^{N} \int_{K_i} \left(\partial_x \phi_i^{(0)} \right) F_c(u_{|K_i}) dV \\ & + \sum_{i=1}^{N} \int_{\partial K_i} \phi_i^{(0)} F_v(u_{|K_i}, s_{|K_i}) . n dS \\ & - \sum_{i=1}^{N} \int_{K_i} \left(\partial_x \phi_i^{(0)} \right) F_v(u_{|K_i}, s_{|K_i}) dV \end{aligned}$$

in which, once again, the boundary integral contains flux terms which are not uniquely defined. It is therefore necessary to replace the term $\partial_x F_v(u_{|K_i}, s_{|K_i}).n$ with the numerical flux function $\hat{F}_2$

$$\hat{F}_2 \left((u^-, s^-), (s^+, u^+), n \right) = \left\langle F_v(u_{|K_i}, s_{|K_i}) \right\rangle n$$

The treatment of the boundary integrals when K_i is part of $\partial\Omega$ is detailed in [].

This is the original method of Bassi and Rebay (BR1) introduced in [] for which, however, several problems have been observed: In contrast to most other DGM where an element communicates with its direct neighboring elements only, the stencil of the BR1 discretisation is considerably larger as it includes also neighbors of neighbors (high computational cost). Furthermore, this discretisation is unstable (only weakly stable).

By slightly modify, the BR1 scheme (introducing a lifting operator and adding a stabilization parameter), the BR2 discretisation solve these issue. Indeed, the BR2 scheme has several advantages over the BR1 scheme. The stencil of BR2 scheme includes only first neighbors instead of additional second neighbors as does the BR1 scheme. Furthermore, as shown in [] the BR2 discretisation is stable.

However, due to the use of lifting operators the BR2 discretisation is significantly more complicated and more computing time expensive than the IP discretisation.

2.2 Time Discretisation

2.2.1 Time Discretisation Technique

The explicit Runge-Kutta scheme was integrated in the DGM by Cockburn, Shu, and collaborators, leading to the popular Rung-Kutta Discontinuous Galerkin (RKDG) method for which the development is treated in [7]. The method has been extensively used and analysed with limiters and reconstruction methods.

The main drawback of explicit methods is the constrain on the time-step which can be significant in certain problems. This issue can be overcome by the use of implicit schemes. The implicit time marching scheme used

for this project is the second order multistep Backward Differentiation Formula (BDF2), which works as follows. After the space discretisation, the time step is computed with a simple BDF1, given by:

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

where,

$$R(\boldsymbol{U}) = -\nabla.\boldsymbol{F}(\boldsymbol{U}) + \nabla.\boldsymbol{F}_v(\boldsymbol{U}, \nabla\boldsymbol{U}) + S(\boldsymbol{U}, \nabla\boldsymbol{U})$$

The scheme is given by:

$$\boldsymbol{U}^{n+1} = \frac{4}{3}\boldsymbol{U}^n - \boldsymbol{U}^{n-1} - \frac{2}{3}\Delta t R(\boldsymbol{U}^{n+1}) \tag{5}$$

2.2.2 Treatement of the Linear System

The scheme (5) is solved with a Newton-Raphson (NR) scheme, where you can find a detailed in [9].

2.3 Boundary Condition

The boundary conditions are weakly imposed, meaning that boundaries are treated as an interface between elements at the limit of the domain and ghost elements, on which the desired conditions are specified. In [Pierre] , the boundary condition implemented in Argo are detailed for the convective and diffusive terms.

3 Shock Capturing Methods

3.1 Limiters

The main idea behind limiters is to eliminate the spurious oscillations by lowering the order of accuracy near discontinuities. They do so by modifying the fluxes or the slopes of the solution, and are thus given the names of flux or slope limiters [19]. This tools would not be treated here because they will not be needed for this project.

3.2 Artificial viscosity

The artificial viscosity (AV) technique used in this work is largely based on the method proposed by Persson & Peraire and the improvement proposed by Barter & Darmofal in [4]. The detector by Persson & Peraire works on the principle that expansions of different order will be similar when the solution is smooth but different when it contains discontinuities. In each element K_i, a component of the solution of order p can be written as:

$$u_{|K_i}(t,x,y,z) = \sum_{k=0}^{N(p)} a_i^{(k)} \phi_i^{(k)}$$

where $N(p)$ corresponds to the expansion's number of terms. The solution can also be expanded in the same way but with terms up to order $p-1$ only:

$$\hat{u}_{|K_i}(t,x,y,z) = \sum_{k=0}^{N(p-1)} a_i^{(k)} \phi_i^{(k)}$$

The shock sensor is based on these two expansions. In regions where the solution is smooth, the decay of the expansion coefficients will be quick. However, discontinuities will lower the decay rate. Therefore, discontinuities can be detected by comparing the two expansions of different order. The sensor S e is given by:

$$S_{K_i} = \frac{(u-\hat{u}, u-\hat{u})_{L^2(\Omega)}}{(u,u)_{L^2(\Omega)}}$$

Clearly, a threshold is needed in order to define above which value of S_{Ki} is artificial viscosity (or another stabilisation technique) applied. In practice, the threshold value needs to be defined by trial and error and will be case-dependent.

Artificial viscosity is then added to the equations (1) in the form of a Laplacian term: $\nabla.(\epsilon\nabla\boldsymbol{U})$

The amount of artificial viscosity added ϵ is scaled by a user-defined magnitude k, the element size h, the polynomial order p and by the maximum wave speed λ max , such that: $\epsilon_0 = k\frac{h}{p}\lambda_{max}$.

This value is then inserted into Equation (2.60) to obtain a smoother variation depending on the sensor values.

$$\epsilon_i = \begin{cases} 0 & if\ s_i < s_0 - I \\ \frac{\epsilon_0}{2}\left(1 + sin\frac{\pi(s_i - s_0)}{2I}\right) & if\ s_0 - I \leq s_i \leq s_0 + I \\ \epsilon_0 & if\ s_i > s_0 + I \end{cases}$$

where $s_i = log_{10}(S_{K_i})$

- s_0 is the sensor threshold and scales with $\frac{1}{p^4}$
- k is the interval on which the artificial viscosity increases smoothly from zero to its maximum value

In Argo, at each interface, the maximum ϵ_i value is chosen to construct a linear polynomial interpolation for ϵ, which leads to a smooth AV field. Notice that, AV methods have parameters that need to be defined by the user. In this case, the user needs to specify three parameters before each simulation, namely the threshold s_0 , the interval I, and the magnitude k. Also this technique do not apply in the case of polynomial function of degree $p = 0$ in a cell K_i, because the comparison in this case with a lower degree polynomial is impossible.

Conclusion

In his master thesis, Marc Cruellas Bordes made different test cases and used the artificial viscosity to enhance the shock location by making the target zone broader. The impossibility to calculate in p=0 causing by the limitation of the IP method (The BR2 method was not yet implemented in Argo), made difficult the run of the code especially in the shock region.

The "DGM+BR2" discretisation allows to run the code in p=0 everywhere on the domain. The project will be the opportunity to test different cases and compare with the results obtained by Marc Cruellas Bordes by using the AV.

References

[1] Balakumar P. Kandil-O. A Kara K. "Effects of Nose Bluntness on Stability of Hypersonic Boundary Layers over Blunt Cone". In: *37th AIAA Fluid Dynamics Conference and Exhibit*. 2019.

[2] S. Rebay F. Bassi. "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier?Stokes Equations". In: *Journal of Computational Physics 131* (1997).

[3] John D. Anderson. *Hypersonic and high temperature gas dynamics*. Edited by American Institute of Aeronautics & Astronautics. 2006.

[4] Koen Hillewaert. "Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries". PhD thesis. Université Catholique de Louvain, Ecole Polytechnique de Louvain, Institut de Mécanique, Matériaux et Génie Civil, 2013.

[5] Pierre Schrooyen. "Argo introduction".

[6] John J. Bertin. *Hypersonic Aerothermodynamics*. Edited by American Institute of Aeronautics & Astronautics. AIAA Education Series, 1994.

[7] M. Cruellas Bordes. "An overview of shock capturing techniques for high-order discontinuous Galerkin methods Literature Study". TU Delft. 2019.

[8] G.Degrez E.Dick R.Grundmann J.D. Anderson Jr. *Computational Fluid Dynamics: An Introduction (Von Karman Institute Book)*. Edited by John F.Wendt. Springer; 3rd ed., 2008.

[9] M. Cruellas Bordes. "A study of an artificial viscosity technique for high-order discontinuous Galerkin methods". Master's thesis. Delft University of Technology, 2019.

[10] Jan Sokolowski Pavel Plotnikov. *Compressible Navier-Stokes Equations Theory and Shape Optimization*. Edited by Springer Basel. Monografie Matematyczne, 2014.

[11] Alexandre Ern Daniele Antonio Di Pietro. *Mathematical Aspects of Discontinuous Galerkin Methods*. Edited by Springer-Verlag Berlin and Heidelberg GmbH & Co. K. 2011.

[12] Tim Warburton Jan S. Hesthaven. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Edited by Springer-Verlag New York Inc. Texts in Applied Mathematics, 2007.

[13] Guido Kanschat. *Discontinuous Galerkin Methods for Viscous Incompressible Flow*. Edited by Deutscher Universitats-Verlag. 2007.

[14] Béatrice M. Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*. Edited by Cambridge University Press. 2008.

[15] Pierre Schrooyen. "Numerical Simulation of Aerothermal Flows through Ablative Thermal Protection System". PhD thesis. Université Catholique de Louvain, Ecole Polytechnique de Louvain, Institut de Mécanique, Matériaux et Génie Civil, 2015.

[16] Chi-Wang Shu Bernado Cockburn George E.Karniadakis. *Discontinuous Galerkin Method: Theory, Computation and Applications*. Edited by Springer-Verlag Berlin, Heidelberg GmbH, and Co. 2000.

[17] K. Leger S. Amri. "Méthode de Galerkine discontinue espace-temps appliquée à l'éqaution d'advection 1D". Master's thesis. Ecole d'ingénieur Sup Galilée, 2017.

[18] Lawrence C. Evans. *Partial Differential Equations*. Edited by American Mathematical Society. Volume 19. 2nd Revised edition, 2010.

[19] Randall J. Leveque. *Numerical Methods for Conservation Laws*. Edited by Birkhauser Verlag AG. Lectures in Mathematics. ETH Zurich, 1999.

[20] Miloslav Feistauer Vít Dolejsí. *Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow*. Edited by Springer. Volume Springer Series in Computational Mathematics Book 48. 14 septembre 2015.

[21] Stanford University; Matthias Ihme Stanford University Eric Ching Stanford University; Yu Lv. "Development of Discontinuous Galerkin method for Hypersonic Heating Prediction". In: *55th AIAA Aerospace Sciences Meeting*. 2017.