A Dynamic Multiscale Viscosity Algorithm for Shock Capturing in Runge Kutta Discontinuous Galerkin Methods

by

Jean-Baptiste Brachet

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aerospace Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2005 [June 2005]

Author .....
Department of Aeronautics and Astronautics
May 20, 2005

Certified by .....
Jaime Peraire
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by .....
Jaime Peraire
Chairman, Department Committee on Graduate Students

AERO

A Dynamic Multiscale Viscosity Algorithm for Shock Capturing in Runge Kutta Discontinuous Galerkin Methods

by

Jean-Baptiste Brachet

Submitted to the Department of Aeronautics and Astronautics
on May 20, 2005, in partial fulfillment of the
requirements for the degree of
Master of Science in Aerospace Engineering

Abstract

In order to improve the performance of higher-order Discontinuous Galerkin finite element solvers, a shock capturing procedure has been developed for hyperbolic equations. The Dynamic Multiscale Viscosity method, originally presented by Oberai and Wanderer [8, 9] in a Fourier Galerkin context, is adapted to the Discontinuous Galerkin discretization. The notions of diffusive model term, artificial viscosities, and the Germano identity are introduced. A general technique for the evaluation of the multiscale model term's parameters is then presented. This technique is used to perform efficient shock capturing on an one-dimensional stationary Burgers' equation with 1-parameter and 2-parameter model terms. Corresponding numerical results are shown.

Thesis Supervisor: Jaime Peraire
Title: Professor of Aeronautics and Astronautics

Acknowledgments

I would like to express my gratitude to the many people who supported me during these two years at MIT.

First of all, I would like to thank all the ACDL staff: my advisor, Professor Peraire, for his help and availability during this research; the Project X team, especially Professor Darmofal and Bob Haimes, for having shared their insights with me; and Jean for her support and patience.

My stay at MIT would not have been such a wonderful experience without the numerous friends I met here. I therefore would like to thank my friends from the lab: Garrett, Krzysztof, Todd, Mike, Dan and Sudeep. Thanks you for your cheerfulness ! Thanks also to the french colonies at ICAT and GTL, to Jeff, Rick, Emmanuel, Theo, to the Thomas and to Pierre, who didn't unfortunately swim across the Charles as promised. Thanks to Jorge for having helped me to practice my English as well as my French and German accents.

Finally, I would like to thank my wife, Pascale, for her love and support during these two years.

Contents

List of Figures

List of Tables

e

Chapter 1

Introduction

Since their introduction during the 1970s, Computational Fluid Dynamics tools have become increasingly used in the industry, resulting in enhanced designs, shorter development costs and increased safety levels. This success calls for CFD tools that are both faster and more accurate, as they are expected to deal with more and more complex problems. Although grid refinement related techniques are a possible mean to increase the accuracy of CFD codes, they do not seem to provide a satisfactory solution for many applications when used on their own. Compared to them, higher-order methods, presented by Cockburn and Shu (see for example [1]), seem much more attractive and could ultimately change the current paradigm in CFD by introducing new codes that would combine high accuracy and reliability in a cost efficient manner.

However, many challenges still need to be addressed before a CFD code taking full advantage of high-order methods can be put into operations. These challenges are various and range from generating high-order grids to the visual presentation of information. Many of these challenges are of course related to solving techniques and, among those, obtaining efficient shock capturing capabilities is one of the most critical. .

One possible approach to shock capturing is to introduce artificial viscosity in the equation. Provided the correct amount of viscosity is put into the system, the shock can be properly and accurately solved. Recently, Assad Oberai and John Wanderer

have presented a promising method for computing this artificial viscosity from the system itself [9]. This method, called the Dynamic Multiscale Viscosity Method, is based on a variational formulation of the Germano identity, which is widely used in the Large Eddy Simulation of turbulent flows, and has been implemented and tested in a spectral context, using Fourier Galerkin discretizations.

The purpose of this thesis is to investigate the potential of this Dynamic Multiscale Viscosity Method (dMSV) in a Discontinuous Galerkin (DG) context. Although a number of finite differences or finite elements methods can achieve high order accuracy, the finite element Discontinuous Galerkin method has several advantages. In this method, the coupling between elements is achieved through numerical fluxes evaluated at the boundaries of each element, making the implementation of boundary conditions a straightforward task, whereas other higher-order schemes often require large stencils that complicate the implementation of these conditions. Additional benefits are obtained through the use of DG, such as an easy implementation of parallel computing techniques.

Although several methods for capturing shocks, such as Weighted Essentially Non-Oscillatory (WENO) schemes [12], have been developped, none of them has yet set a standard. Many of these methods are based on the detection of shocks and on the reconstruction of lower-order, oscillations-free, solutions on the corresponding cells. The design of an efficient and reliable shock detection algorithm is often a challenge. Similarly, the reconstruction of the solution over the troubled cells requires the use of a stencil, which may lead to difficult implementations of some kinds of boundary conditions. As we shall see it later, the dMSV approach has neither of these issues: it does not require any shock detection nor does it need any reconstruction for the solution.

In this context, a dMSV-based shock-capturing algorithm for stationary shocks has been developped. This algorithm has been tested with two different model terms on a fifth-order solution, the stationary shock being generated by Burgers' equation. The results of these tests are presented in the last chapter of this thesis.

Chapter 2

Time and space discretizations

This chapter is devoted to the presentation of the equations studied in this thesis. Their mathematical formulations will be described first. We will then focus on how to discretize them, both in space and time. For the space discretization, we use a Discontinuous Galerkin method while a Runge-Kutta scheme is used to perform the time marching.

2.1 Mathematical formulations

2.1.1 Purely convective equations

The first type of equation that will be discussed in this thesis is the nonlinear, purely convective equation. The general form of such an equation can be written:

$$\frac{\partial u}{\partial t} + \nabla \cdot f(u) = 0 \tag{2.1}$$

where:

- **u**, the state, is a scalar.
- f the flux, is a function of u.

When combined with a set of boundary conditions, such equations result in mathematical problems which, in general, cannot be solved analytically.

2.1.2 Convection-diffusion equations

The development of a multi-scale viscosity algorithm requires a viscous model term to be added to the equation. We will therefore encounter convection-diffusion equations, which can be written:

$$\frac{\partial u}{\partial t} + \nabla \cdot f(u) = \nu \Delta u \tag{2.2}$$

where the viscosity ν is a positive real number and the Laplacian operator Δ can operate in a one-dimensional or in a multi-dimensional space. Like purely convective equations, convection-diffusion equations cannot, in general, be solved analytically.

2.1.3 Burgers' equations

In this thesis, we will concentrate on the one-dimensional scalar Burger's equation, which is written:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(\frac{u^2}{2}) = 0 \tag{2.3}$$

This shock-generating equation will be used as a model to test our multi-scale viscosity algorithm. As mentioned before, the multi-scale viscosity algorithm requires the addition of a viscous model term in the equation. For example, in the case of a 1-viscosity model term, we will have to deal with the viscous Burger's equation, which is a convection-diffusion equation:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(\frac{u^2}{2}) = \nu \frac{\partial^2 u}{\partial x^2} \tag{2.4}$$

2.1.4 Weak formulations

The weak formulations of the two preceding types of equations are the basis of many finite elements formulations, such as the Discontinuous Galerkin formulation. Let us call Ω a convex domain, $\partial\Omega$ its boundary, and $\mathcal{U}(\Omega)$ the Sobolev space of acceptable solutions to the problem.

In the general case of a purely convective problem, we have:

$$u \in \mathcal{U}(\Omega), \qquad \frac{\partial u}{\partial t} + \nabla \cdot f(u) = 0$$

$$\Rightarrow \qquad \forall v \in \mathcal{U}(\Omega), \qquad (\frac{\partial u}{\partial t} + \nabla \cdot f(u))\, v = 0$$

$$\Rightarrow \qquad \forall v \in \mathcal{U}(\Omega), \qquad \int_\Omega \frac{\partial u}{\partial t}\, v\, dx + \int_\Omega (\nabla \cdot f(u))\, v\, dx = 0$$

Through integration by parts, we get:

$$\forall v \in \mathcal{U}(\Omega), \quad \int_\Omega (\frac{\partial u}{\partial t}\, v)\, dx + \int_{\partial\Omega} (f(u) \cdot \mathbf{n})\, v\, ds - \int_\Omega f(u) \cdot (\nabla v)\, dx = 0 \qquad (2.5)$$

which is the weak form of the problem, $\mathbf{n}$ being a unit normal vector pointing toward the outside of Ω. A similar formulation can be derived for a convection-diffusion equation.

When the exact solutions to these problems present discontinuities, such as a shock, weak formulations can lead to non unique solutions. In such cases, the physically relevant physical solutions are the solutions satisfying the following entropy inequality:

$$\frac{\partial U(u)}{\partial t} + \frac{\partial F(u)}{\partial x} \leq 0 \qquad (2.6)$$

for any convex function U and consistent entropy flux F satisfying $\nabla F = \nabla U \cdot \nabla f$. In the case of the Burger's equation, we can choose $U = u^2$.

2.2 Discontinuous Galerkin method

The Discontinuous Galerkin method, which was first introduced in 1973 by Reed and Hill in the context of neutron transport, is an efficient way to discretize convection

dominated or convection-diffusion problems. This section will provide the reader with a short introduction to the method and its application to these two types of problems. For a more detailed description of the method, the reader may consult [1].

Like any other numerical method, the Discontinuous Galerkin formulation aims at approximating the exact mathematical solution of a problem (which is defined in a space of infinite dimension), by a function that possesses a finite number of degrees of freedom. In our case, we will approximate the Sobolev space of acceptable solutions to a given problem, $\mathcal{U}(\Omega)$ by the finite dimensional space $\mathcal{U}_h^p(\Omega)$. The space $\mathcal{U}_h^p(\Omega)$ is built by discretizing the physical space Ω into a grid $(\mathcal{T}_h)_h$ and by assuming that, each element of the original space $\mathcal{U}(\Omega)$ can be approximated by a linear combination of basis functions $(\phi_i)_{i=0...p}$, so that we have the inclusion $\mathcal{U}_h^p(\Omega) \subset \mathcal{U}(\Omega)$.

2.2.1 Application to purely convective problems

The discontinuous Galerkin formulation of a purely convective problem is obtained by applying its weak form (equation 2.5) to elements of $\mathcal{U}_h^p$. By doing so, we get:

$$Find \quad u^h \in \mathcal{U}_h^p(\Omega), \quad \forall v^h \in \mathcal{U}_h^p(\Omega)$$

$$\int_{(\mathcal{T}_h)} (\frac{\partial u^h}{\partial t}\, v^h)\, dx + \int_{\partial(\mathcal{T}_h)} (f(u^h)\cdot \mathbf{n})\, v^h\, ds - \int_{(\mathcal{T}_h)} f(u^h)\cdot(\nabla v^h)\, dx = 0$$

It is then possible to choose v^h equal to a basis function $\phi_i \in \mathcal{U}(\Omega)_h^p$, which leads to:

$$\int_{(\mathcal{T}_h)} (\frac{\partial u^h}{\partial t}\, \phi_i)\, dx + \int_{\partial(\mathcal{T}_h)} (f(u^h)\cdot \mathbf{n})\, \phi_i\, ds - \int_{(\mathcal{T}_h)} f(u^h)\cdot(\nabla \phi_i)\, dx = 0 \qquad (2.7)$$

Since u^h belongs to $\mathcal{U}(\Omega)_h^p$ we can expand it on the basis $(\phi_i)_{i=0..p}$ such that $u^h(\mathbf{x},t) = \sum_{i=0}^{p} u_i^h(t)\phi_i(\mathbf{x})$. By substituting this expression into (2.7), we get:

$$\sum_{j=0}^{p} \int_{(\mathcal{T}_h)} (\frac{du_j^h}{dt}\phi_j\, \phi_i)\, dx + \int_{\partial(\mathcal{T}_h)} (f(u^h)\cdot \mathbf{n})\, \phi_i\, ds - \int_{(\mathcal{T}_h)} f(u^h)\cdot(\nabla \phi_i)\, dx = 0$$

which can be rewritten:

$$\sum_{j=0}^{p} \frac{du_j^h}{dt} \int_{(\mathcal{T}_h)} (\phi_j \, \phi_i) \, dx + \int_{\partial(\mathcal{T}_h)} (f(u^h) \cdot \mathbf{n}) \, \phi_i \, ds - \int_{(\mathcal{T}_h)} f(u^h) \cdot (\nabla \phi_i) \, dx = 0 \quad (2.8)$$

The first term of this expression involves the mass matrix, which will be called $\mathcal{M}$. It is defined by:

$$\forall (i,j) \in [0,...,p]^2 \,, \quad \mathcal{M}_{i,j} = \int_{(\mathcal{T}_h)} \phi_i \phi_j \, dx$$

The remaining part of (2.8) is called the j^{th}-residual associated to $\mathbf{u}^h$:

$$\forall j \in [0,...,p] \,, \quad \mathbf{R}_j(u^h) = \int_{\partial(\mathcal{T}_h)} (f(u^h) \cdot \mathbf{n}) \, \phi_j \, ds - \int_{(\mathcal{T}_h)} f(u^h) \cdot (\nabla \phi_j) \, dx$$

If we call $\mathbf{U}^h$ the vector $[u_0^h, ..., u_p^h]$, the problem can be rewritten:

$$\mathcal{M} \frac{d\mathbf{U}^h}{dt} + \mathbf{R}(\mathbf{U}^h) = 0 \quad (2.9)$$

The specificity of the Discontinuous Galerkin method comes from the choice of $(\phi_i)_{i=0,...,p}$. Equation (2.9) shows that the stencil of the method will be determined by the mass matrix $\mathcal{M}$ and the choice of the basis functions. Understandably, the size of the stencil plays an important role in the determination of the robustness (i.e. ability to handle complex boundary conditions) and the computationnal cost of the method. Typically, the smaller the stencil is, the easier the method can handle complex boundary conditions and the faster it will be.

The Discontinuous Galerkin method aims at taking the smallest stencil possible by taking the smallest possible support for the basis functions. Each basis function will have a support limited to one cell $\mathcal{T}_\lambda$ of the grid $(\mathcal{T}_h)_h$. A typical example of such a basis is a set of local polynomial basis functions $(\phi_i^h)_{i,h}$ with $i \in [0,...,p]$ and

$h \in [0, ..., card((\mathcal{T}_h))]$. As a consequence, with this kind of basis, the mass matrix is block-diagonal, hence the small stencil which makes the Discontinuous Galerkin methods well-adapted to high-order solvers.

Because of the basis used, the solutions found through the Discontinuous Galerkin method are discontinuous from one cell to another. In order to take into account the dependency of the solution on a given cell to its restriction to the surrounding cells, a numerical flux $\mathcal{F}$ is introduced in the expression of the residuals. For example, the the i^{th} residual on element $\mathcal{T}_\lambda$ is written:

$$\mathbf{R}_i^\lambda(u^h) = \int_{\partial\mathcal{T}_\lambda \setminus \partial\Omega} \mathcal{F}((f(u_+^h), f(u_-^h), \mathbf{n})\, \phi_i^\lambda \, ds + \int_{\partial\mathcal{T}_\lambda \cap \partial\Omega} \mathcal{F}_{bc}((f(u_+^h), f(u_-^h), \mathbf{n})\, \phi_i^\lambda \, ds$$

$$- \int_{\mathcal{T}_\lambda} f(u^h) \cdot (\nabla \phi_i^\lambda)\, dx$$

where the symbols $()^+$ and $()^-$ refer to the respectively outside and inside values of $\mathbf{u}^h$ on a given edge. The numerical flux $\mathcal{F}$ is used for interior edges whereas $\mathcal{F}_{bc}$ corresponds to edges that are boundaries of the physical domain Ω. These two functions do not need to be the same.

Ideally, the numerical fluxes should satisfy two important properties:

- They must be consistent with the continuous flux.
- They must respect the cell entropy inequality.

The first property simply states that, in the case of a continuous function u^h across an edge, $u_+^h = u_-^h$, then the numerical flux $\mathcal{F}$ must satisfy:

$$\mathcal{F}(u_+^h, u_-^h, \mathbf{n}) = f(u^h) \cdot \mathbf{n}$$

With this condition on $\mathcal{F}$, if (2.9) converges, it will converge to a weak solution of the original convective problem. However, this condition is not sufficient to insure that this weak solution corresponds to the physical solution of the problem. Fortunately, Jiang [6] have proved that the cell entropy inequality holds for a class of

high-order Discontinuous Galerkin methods. This property confers a great advantage to Discontinuous Galerkin methods over other finite elements methods: with a properly chosen flux, one can be sure that if the Discontinuous Galerkin converges, then it will converge to the entropy-satisfying weak solution, i.e. the relevant physical solution to the problem. It can be shown that an E-flux (as defined by Osher [10]) satisfies the cell entropy inequality (see for example Serrano [13, p. 20]). In the context of this thesis, we used the Godunov flux, which is a E-flux for the Burger's one-dimensional equation.

If the numerical flux $\mathcal{F}$ satisfies the two properties mentioned above, then $\mathbf{u}^h \in \mathcal{U}_p^h$ will be an approximation to the physical solution of the original problem if it satisfies the following Discontinuous Galerkin formulation:

$$\mathcal{M}\frac{d\mathbf{U}^h}{dt} + \mathbf{R}(\mathbf{U}^h) = 0 \tag{2.10}$$

2.2.2 Application to convection-diffusion problems

The Discontinuous Galerkin method was first adapted to convection-diffusion problems by Bassi and Rebay in 1997 [3] then followed by Cockburn and Shu in 1998 [2] who introduced the Local Discontinuous Galerkin formulation, which is the formulation that has been used for all the results presented in this thesis. The underlying idea of the Local Discontinuous Galerkin method is to rewrite the convection-diffusion problem as a first order system and then apply the Discontinuous Galerkin space discretization we saw in the preceding section.
Let us illustrate the method by considering (2.2):

$$\frac{\partial u}{\partial t} + \nabla \cdot f(u) = \nu \Delta u$$

The first step is to rewrite the equation as a first-order system by introducing a new unknown vector $\mathbf{q}$:

$$\frac{\partial u}{\partial t} + (\nabla \cdot f(u)) = \nu \nabla \cdot \mathbf{q}$$

$$\mathbf{q} = \nabla u$$

which leads to the following weak forms:

$$\forall v \in \mathcal{U}_h^p(\Omega), \quad \int_\Omega \frac{\partial u^h}{\partial t}\, v\, dx + \int_\Omega (\nabla \cdot f(u^h) - \nu \nabla \cdot \mathbf{q}^h)\, v dx = 0$$

$$\forall (w,i) \in \mathcal{U}_h^p(\Omega) \times [0,...,p], \quad \int_\Omega q_i^h \cdot w\, dx - \int_\Omega (\nabla u^h) \cdot w\, dx = 0$$

if the vector $\mathbf{q}^h$ is written $\left[q_0^h, ..., q_p^h\right]$, and to the following Discontinuous Galerkin formulations:

$$\sum_{i=0}^{p} \frac{du_i^h}{dt} \int_\Omega \phi_i^h \phi_j^h\, dx + \int_{\partial\Omega} (\widehat{f(u^h)} \cdot \mathbf{n} - \nu \widehat{\mathbf{q}^h \cdot \mathbf{n}})\, \phi_j^h\, ds - \int_\Omega (f(u^h) - \nu \mathbf{q}^h) \cdot \nabla \phi_j^h\, dx = 0$$

$$\sum_{i=0}^{p} q_i^h \int_\Omega \phi_i^h \phi_j^h\, dx = \int_{\partial\Omega} \widehat{u_i^h}\, \phi_j^h\, ds - \int_\Omega u_i\, \nabla \phi_j^h\, dx$$

for all $j \in [0,...,p]$.

The numerical flux $\widehat{f(\mathbf{u}^h)}$ can be chosen to be the same as in the purely convective case whereas the fluxes $\widehat{\mathbf{q}^h}$ and $\widehat{u^h}$ have to be carefully chosen so that the scheme satisfies a discrete energy inequality and it remains stable. For further information about how these two fluxes can be chosen in a multi-dimensional context, the reader may refer to the review article by Cockburn and Shu [1, p.238].

2.2.3 One-dimensional implementations

Grid and basis

In this thesis, we will limit ourselves to one-dimensional equations and to the corresponding spaces $\mathcal{U}(\Omega)$ and $\mathcal{U}_h^p(\Omega)$. More precisely, the physical space $\Omega = [-1,1]$ will be divided into segments $\mathcal{T}_h = \left[x_{h+\frac{1}{2}}, x_{h+\frac{1}{2}}\right]$ and the local basis will consist of Legendre polynomial functions: $\forall (h,i) \in (\mathcal{T}_h) \times [0,...,p], \quad \phi_i^h = L_i^h$.

For the sake of simplicity we chose to normalize the local Legendre basis, so that two elements L_i^h and L_j^h verify the relation:

$$\int_{\mathcal{T}_{h'}} L_i^h L_j^h \, dx = \delta_{ij}\delta_{hh'} \tag{2.11}$$

With this choice of a local basis, the mass matrix $\mathcal{M}$ will be equal to the identity matrix.

Numerical fluxes

For both the purely convective problem and the convection-diffusion problem, the numerical flux $\widehat{f(u^h)}$ will be taken equal to the Godunov flux, which is an E-flux:

$$\widehat{f(u^h)} = f^{G}(\widehat{u_+^h, u_-^h}) = \begin{cases} \min_{u_+^h \leq u \leq u_-^h} f(u), & if\ u_+^h \leq u_-^h \\ \max_{u_-^h \leq u \leq u_+^h} f(u), & otherwise \end{cases}$$

In the convection-diffusion problem, the two remaining numerical fluxes $\widehat{u^h}$ and $\widehat{q^h}$ are computed by taking alternatively the left and right limits of u^h and q^h. This expression was proposed by Cockburn and Shu in [2]. It guarantees an optimal rate of convergence for the solution.

$$\widehat{u^h} = u_-^h$$

$$\widehat{q^h} = q_+^h$$

One-dimensional Burgers' equation

As mentioned previously, we will make an extensive use of the one-dimensional viscous version of Burger's equation in this thesis. With the conventions we set so far, the Local Discontinuous Galerkin discretization of this problem is:

$$\forall (h, h', i, j) \in (\mathcal{T}_h)^2 \times [0, ..., p]^2 \,,$$

$$\frac{du_i^h(t)}{dt}\int_{\mathcal{T}_h} L_i^h(x)L_i^h(x)\,dx + \left[(\widehat{\frac{(u^h)^2}{2}}^G - \nu\widehat{q^h})L_i^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h}(\frac{(u^h)^2}{2} - \nu q^h)(L_i^h)_x\,dx = 0$$

$$q_j^{h'}\int_{\mathcal{T}_{h'}} L_j^{h'}L_j^{h'}\,dx = \left[\widehat{u^{h'}}L_j^{h'}\right]_{h'-\frac{1}{2}}^{h'+\frac{1}{2}} - \int_{\mathcal{T}_{h'}} u^{h'}(L_j^{h'})_x\,dx$$

where the expressions for the fluxes $\widehat{\frac{(u^h)^2}{2}}^G$, $\widehat{q^h}$ and $\widehat{u^h}$ were given previously. Numerical integrations were peformed using the Gauss-Legendre quadrature rules.

2.3 Runge-Kutta schemes

Attention must be paid to the fact that the scheme has to be high-order accurate in order not to penalize the overall accuracy of the code. Runge-Kutta schemes have this property of being high-order accurate. In addition, being explicit schemes, they are relatively inexpensive, use a reasonable amount of memory and are easy to implement.

We decided to go for a fourth-order Runge-Kutta scheme, which allows for larger time steps, hence reducing the time needed to perform the calculations. With the notations introduced in (2.9) with $\mathbf{L}(\mathbf{U}^h) = -\mathcal{M}^{-1}\cdot\mathbf{R}(\mathbf{U}^h)$, the scheme we chose to implement can be written:

$$\begin{aligned}
\mathbf{K}_1 &= \Delta t\cdot\mathbf{L}(\mathbf{U}_n^h)\\
\mathbf{K}_2 &= \Delta t\cdot\mathbf{L}(\mathbf{U}_n^h + \tfrac{\mathbf{K}_1}{2})\\
\mathbf{K}_3 &= \Delta t\cdot\mathbf{L}(\mathbf{U}_n^h + \tfrac{\mathbf{K}_2}{2})\\
\mathbf{K}_4 &= \Delta t\cdot\mathbf{L}(\mathbf{U}_n^h + \mathbf{K}_3)\\
\mathbf{U}_{n+1}^h &= \mathbf{U}_n^h + \tfrac{\mathbf{K}_1}{6} + \tfrac{\mathbf{K}_2}{3} + \tfrac{\mathbf{K}_3}{3} + \tfrac{\mathbf{K}_4}{6}
\end{aligned}$$

In this scheme, the moments of the solution over a cell and its neighbours (or the corresponding boundary conditions, if the given cell is on the boundary of the domain) are required to compute the residual $\mathbf{R}(\mathbf{U}^h)$ and to perform the time marching. Since this matrix does not change with time, it can be computed once and stored. In the

case of a DG or LDG spatial discretization, the mass matrix is block diagonal. As a result, the computation of its inverse can be done a element-wise.

Chapter 3

Dynamic Multiscale Viscosity

High-order polynomials used in DG and LDG discretizations can provide high-order accurate representations of smooth solutions. However, whenever shocks appear, numerical oscillations usually start to show up in the DG and LDG solutions. These overshoots first begin in the vicinity of the discontinuities but eventually propagate and damage the accuracy of the overall solution. The proposed Dynamic Multiscale Viscoscity (dMSV) approach is a way to mitigate this issue and to keep the high-order accuracy of DG solvers.

The purpose of this chapter is threefold.

- Present our motivations to use the dMSV algorithm and the notations we will be using in the chapter.
- Describe the dMSV method in itself. This includes presenting the variational Germano identity, the multiscale equations and the evaluation procedures.
- Provide a comprehensive report on how the method has been implemented in a RKDG context and explain what are the challenges that arose.

3.1 Motivation and notation

In this section, we will first give a brief overview of an alternative limiting technique and compare it to the dMSV methodology. Then we will introduce the notation that is used in the dMSV method.

3.1.1 Existing shock-capturing techniques

A popular and promising class of shock-limiting and capturing techniques are the WENO and HWENO schemes. The WENO methodology was first introduced by Qiu (see for example [11]). It is based on first identifying which moments of the solutions are affected by the shock and then on reconstructing these moments using a weighted sum of reconstruction polynomials. This method has shown promising results and is perfectly able to produce oscillation-free solutions which are still high-order accurate in smooth regions. Yet, it has a major drawback: the reconstruction of the solution requires large stencils which are a problem both because they may complicate the implementation of some kinds of boundary conditions and because these large stencils prevent the code from benefiting from the advantages linked to DG discretizations. HWENO schemes were then developed [12]. Contrary to WENO schemes that use large stencils and only the 0^{th}-order moments of the solution for the reconstruction, HWENO methods, which take into account the higher-order moments of the solution to do the reconstruction, require smaller stencils. Nevertheless, even these smaller stencils may be burdensome in some cases.

3.1.2 Advantages of dMSV schemes

As we will see it later, dMSV-based schemes do not perform any shock detection or any reconstruction of the solution. The shock is captured by introducing one or more artificial viscosities in the original equation. These viscosities are computed through a set of consistency conditions. dMSV schemes do no present any of stencil-related problems we mentioned earlier. This is the main reason that pushed for the development and adaptation of these schemes to DG discretizations.

3.1.3 Model term

Let us consider the weak form of specified a problem, either purely convective or convective-diffusive. This problem can be expressed, following the notations introduced in the preceding chapter:

$$u^h \in \mathcal{U}_p^h \,/\, \forall v^h \in \mathcal{U}_p^h, \qquad B(v^h, u^h) = 0 \tag{3.1}$$

where B is a semi-linear form (linear in v^h): $\mathcal{U}_p^h \times \mathcal{U}_p^h \longrightarrow \mathbb{R}$.

We know that the solution of (3.1) breaks down in the presence of discontinuities. In order to perform the shock-capturing, we introduce a model term M. This model term depends on the solution u^h, the test function v^h but also on the space s and on a set of parameters $(\nu_k)_k$, the artificial viscosities. It is important to realize that this space s can be a physical space, such as a grid with a specified mesh size, or it can be a functional space, such as a polynomial basis of specified dimension. To take into account these dependencies, the functional M: $\mathcal{U}_p^h \times \mathcal{U}_p^h \times L \times \mathbb{R} \longrightarrow \mathbb{R}$ will be denoted $M(v^h, \mathbf{u}^h, s, (\nu_k)_k)$.
From now on, we will therefore refer to the following problem:

$$u^h \in \mathcal{U}_p^h \,/\, \forall v^h \in \mathcal{U}_p^h, \qquad B(v^h, u^h) + M(v^h, u^h, s, (\nu_k)_k) = 0 \tag{3.2}$$

The solutions of (3.1) and (3.2) are of course different, but the model term should be carefully designed so that it vanishes away from shocks..

3.2 Description of the dMSV methodology

As we saw it in the preceding section, the underlying idea of dMSV scheme is to introduce a model term in the equation. This model term is responsible for the capturing of the shock. To do so, it relies on a set of parameters $(\nu_k)_k$. In the

case of a diffusive model term, which is the case studied in this thesis, this set of parameters can be interpretated as artificial viscosities. The challenge is then to compute adequate viscosities from the numerical solution itself. If the viscosities are too large, then the accuracy of the solution will decrease dramatically and the shock capturing will be poor. On the other hand, if they happen to be too small, then the lack of dissipation will lead to an oscillatory solution. In the dMSV methodology, the evaluation of the model term's parameters is performed through the Germano identity. For more information about the method itself and the evaluation of the parameters, one may consult [8].

3.2.1 The variational Germano identity

The Germano identity has first been developed in the context of filtered Navier Stokes equations in 1991 (see the original paper from Germano et al. [4]), as an efficient tool for computing eddy viscosity in Large Eddy Simulation of turbulent flows. The variational expression of Germano's equation appears in ([8]) and can be generalized to the case of an abstract system, such as (3.2).

Generalization of the variational Germano identity

Following Oberai [8], we write that the function $u^h \in \mathcal{U}_p^h$ is the solution of our modified system (3.2), if:

$$\forall v^h \in \mathcal{U}_p^h, \qquad M(v^h, u^h, s, (\nu_k)_k) = -B(v^h, u^h) \tag{3.3}$$

Now, we take a subspace of $\mathcal{U}_p^h$. This subspace can be a polynomial subspace of $\mathcal{U}_p^h$, in that case, it will be denoted $\mathcal{U}_{p_1}^h$ with $p_1 < p$ to insure that $\mathcal{U}_{p_1}^h \subset \mathcal{U}_p^h$. But it could also be a physical subspace of $\mathcal{U}_p^h$. In that case, it would be noted $\mathcal{U}_p^{h_1}$ with the grids $(\mathcal{T}_{h_1})$ and $(\mathcal{T}_h)$ verifying the inclusion $(\mathcal{T}_{h_1}) \subset (\mathcal{T}_h)$.

Let us consider a projection $\mathbb{P}^1$ from $\mathcal{U}_p^h$ to this subspace $\mathcal{U}_{p_1}^h$ (for example), and call $\mathbf{u}_1^h$ the projection of the solution $\mathbf{u}^h$ over this subspace. A problem similar to

(3.2) can be written over the subspace $\mathcal{U}_{p_1}^h$, that is:

$$\exists w_1^h \in \mathcal{U}_{p_1}^h \;/\; \forall v_1^h \in \mathcal{U}_{p_1}^h, \qquad B(v_1^h, w_1^h) + M(v_1^h, w_1^h, s_1, (\nu_k)_k) = 0 \tag{3.4}$$

We then make the assumption that the solution of (3.4), w_1^h is equal to u_1^h, which we consider to be the optimal solution of u^h in $\mathcal{U}_{p_1}^h$. This assumption implies that the model term M leads to solutions that can be related one to another through a projection operator. This property of the model term will enable us to find relationships between the solutions of the same problem seen at different scales.
Since $\mathcal{U}_{p_1}^h \subset \mathcal{U}_p^h$, (3.3) implies that:

$$\forall v_1^h \in \mathcal{U}_{p_1}^h, \qquad M(v_1^h, u^h, s, (\nu_k)_k) = -B(v_1^h, u^h) \tag{3.5}$$

Under our assumption, (3.4) leads to:

$$\forall v_1^h \in \mathcal{U}_{p_1}^h, \qquad M(v_1^h, u_1^h, s_1, (\nu_k)_k) = -B(v_1^h, u_1^h) \tag{3.6}$$

Substracting (3.6) and (3.5) gives the variational Germano identity that relates scales s and s_1:

$$\forall v_1^h \in \mathcal{U}_{p_1}^h, \quad M(v_1^h, u_1^h, s_1, (\nu_k)_k) - M(v_1^h, u^h, s, (\nu_k)_k) = B(v_1^h, u^h) - B(v_1^h, u_1^h) \tag{3.7}$$

By applying either the dissipation or least-squares evaluation procedure, which will be presented later in the chapter, these equations provide us with a consistency condition on the model term. Since this condition only depends on the numerical solutions u_1^h and u^h, it can be used to compute dynamically the unknown parameter(s) of the model term.

Comparing more than two scales

In the case of a model term M that contains more than one artificial viscosity, the preceding equations will, through the use of one of the evaluation procedure that will be explained later, give only one consistency condition by relating two scales via the variational Germano identity. It is not enough to close the system. The solution is then to generalize (3.7). Let consider a set of several scales $(s_j)_{j=1...N}$, and the associated subspaces $(\mathcal{U}^h_{p_j})_{j=1...N}$. These subspaces must verify $\mathcal{U}^h_{p_N} \subset \mathcal{U}^h_{p_{N-1}} \subset \cdots \subset \mathcal{U}^h_{p_1} \subset \mathcal{U}^h_p$. Under the assumption that the model term is such that, for every scale, the optimal solution of the problem at this scale is equal to the projection of u^h over the corresponding subspace, we can write the following set of Germano identities:

$\forall j \in [1, ..., N], \forall \mathbf{v}^h_j \in \mathcal{U}^h_{p_j}$,

$$M(v^h_j, u^h_j, s_j, (\nu_k)_k) - M(v^h_j, u^h, s, (\nu_k)_k) = B(v^h_j, u^h) - B(v^h_j, u^h_j) \tag{3.8}$$

With this generalized expression of the variational Germano identity, developed by Oberai [8], a number of consistency conditions can be derived, which, in turn, can be used to determine the values of the artificial viscosities that appear in the model term. As a result, the number of parameters in the model terms is only limited by the dimension of $\mathcal{U}^h_p$ since it is related to the number of subspaces available to write the consistency conditions.

Choice of subspaces

As mentioned earlier, the different scales can be either spatial scales or spectral scales. In the case of spatial scales, the subspaces $\mathcal{U}^{h_j}_p$ are simply build upon grids that are coarser and coarser. The projection operators $\mathbb{P}^j$ are then interpolation operators. Spectral spaces are constructed in a different manner: the subspaces have not the same polynomial (or spectral) basis while the grid remains the same. In this case, $\mathbb{P}^j$ are projections on given polynomials or spectral subspaces. A choice must be made between the two types of subspaces. For high-order schemes, the choice of spectral subspaces seems much more appropriate since it does not require incorporating inter-

polation schemes in the code, these schemes being both computationally expensive and likely to introduce annoying stencils. As a result, we chose to use spectral scales in this thesis.

3.2.2 Evaluation of the model parameters

In this section, we will show how the generalized variational Germano identity is utilized to evaluate the artificial viscosities $(\nu_k)_k$. For more detailed information, the reader may refer to [8].

The consistency conditions (3.8) claim that the model term is such that on each scale s^j, the optimal solution u_j^h is equal to the projection of u^h (solution of the problem at scale s) over the subspace $\mathcal{U}_{p_j}^h$, that is $u_j^h = \mathbb{P}^j u^h$. The consistency conditions (3.8) can therefore be rewritten:

$$M(v_j^h, \mathbb{P}^j u^h, s_j, (\nu_k)_k) - M(v_j^h, u^h, s, (\nu_k)_k) = B(v_j^h, u^h) - B(v_j^h, \mathbb{P}^j u^h)$$

for $j \in [1, ..., N]$.
Let us assume that the model term contains q parameters. Q+1 scales are needed to be able to write q Germano identities and obtain q consistency conditions. Each of these expressions must be satisfied for each element $\mathbf{v}_j^h$ of $\mathcal{U}_{p_j}^h$, $j \in [1, ..., q]$. Consequently, each consistency condition generates $\dim(\mathcal{U}_{p_j}^h)$ equations. The system is therefore overdefined and specific procedures are needed to evaluate $(\nu_k)_{k=1,...,q}$. We will present here two different procedures to perform the evaluation.

Dissipation method

The first procedure involves taking $\mathbb{P}^j u^h$ as a test function in consistency condition j. That is, each consistency condition becomes:

$$M(\mathbb{P}^j u^h, u_j^h, s_j, (\nu_k)_k) - M(\mathbb{P}^j u^h, u^h, s, (\nu_k)_k) = B(\mathbb{P}^j u^h, u^h) - B(\mathbb{P}^j u^h, u_j^h) \quad (3.9)$$

for $j \in [1, ..., q]$.

In turbulence modeling, this equation has an energetic interpretation, the left-hand side representing the difference in the dissipation of kinetic energy produced by LES models acting at different scales. For further information about the method, one may refer to the original work by Germano [4].

Least-squares method

Another possible approach to evaluate $(\nu_k)_{k=1,...q}$ is the least-squares method, originally developped by Goshal et al.[5] and Lily [7].

Let us consider the j^{th} consistency condition, the corresponding subspace $\mathcal{U}^h_{p_j}$ and $(\phi^h_l)_{l=0,...,p_j}$ a basis of the subspace. For all $j \in [1, ..., q]$, the j^{th} consistency condition will be evaluated for each basis function ϕ^h_l of $\mathcal{U}^h_{p_j}$, resulting in $\sum_{j=1}^{q} \dim(\mathcal{U}^h_{p_j})$ equations:

$$M(\phi^h_l, u^h_j, s_j, (\nu_k)_k) - M(\phi^h_l, u^h, s, (\nu_k)_k) = B(\phi^h_l, u^h) - B(\phi^h_l, u^h_j)$$

with $l \in [0, ..., p_j]$, $j \in [1, ..., q]$. In order to generate q equations the preceding equations will be written in residual form:

$$M(\phi^h_l, u^h_j, s_j, (\nu_k)_k) - M(\phi^h_l, u^h, s, (\nu_k)_k) - B(\phi^h_l, u^h) + B(\phi^h_l, u^h_j) = 0$$

and the quantity $W((\nu_k)_k)$ will be formed:

$$W((\nu_k)_k) = \sum_{j=1}^{q} \sum_{l=0}^{p_j} (M(\phi^h_l, u^h_j, s_j, (\nu_k)_k) - M(\phi^h_l, u^h, s, (\nu_k)_k) - B(\phi^h_l, u^h) + B(\phi^h_l, u^h_j))^2$$

We now seek the values of $(\nu_k)_k$ that minimizes this quantity, that is we compute

$(\nu)_{k=1,...,q}$ such that:

$$\forall k \in [1, ..., q], \quad \frac{\partial W}{\partial \nu_k} = 0 \tag{3.10}$$

Which generates the correct number of equations and therefore enables the computation of the model term's parameters.

3.3 Discontinuous Galerking implementation

As mentioned earlier, the dMSV methodology has been implemented by John Wanderer and Assad Oberai [9] in a Fourier Galerkin context. However, we think that Runge-Kutta Discontinuous Galerkin formulations offer several advantages for high-order schemes, hence our effort to adapt dMSV shock capturing techniques to Discontinuous Galerkin formulations. Implementing these techniques raise several challenging issues that will be presented in the following subsection.

3.3.1 Challenges for a DG implementation

Diffusive terms

The model term $M(v^h, u^h, s, (\nu_k)_k)$ is of diffusive nature. Although the implementation of such a term in a Fourier Galerkin context is straightforward, its implementation in a DG context is more complicated and one needs to pay particular attention to which numerical fluxes are used in order to have a consistent scheme. In this thesis, we chose to use Cockburn and Shu's LDG method [2] to discretize our model terms. In the simple case of a one-viscosity model term, $M(v^h, u^h, s, \overline{\nu}) = -\overline{\nu} \int_{(\mathcal{T}_h)} \Delta u^h \, v^h \, dx$, applied to the one-dimensional Burger's equation, the discretization is:

$$\forall (h, h', i, j) \in (\mathcal{T}_h)^2 \times [0, ..., p]^2 ,$$

$$\frac{du_i^h(t)}{dt} \int_{\mathcal{T}_h} L_i^h(x) L_i^h(x) \, dx + \left[\left(\frac{\widehat{(u^h)^2}}{2}^G - \overline{\nu} \widehat{q^h} \right) L_i^h \right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} \left(\frac{(u^h)^2}{2} - \overline{\nu} q^h \right) (L_i^h)_x \, dx = 0$$

$$q_j^{h'} \int_{\mathcal{T}_{h'}} L_j^{h'} L_j^{h'} \, dx = \left[\widehat{u^{h'}} L_j^{h'} \right]_{h'-\frac{1}{2}}^{h'+\frac{1}{2}} - \int_{\mathcal{T}_{h'}} u^{h'} (L_j^{h'})_x \, dx$$

with the fluxes: $\begin{array}{l} \widehat{u^h} = u_-^h \\ \widehat{q^h} = q_+^h \end{array}$ and $\widehat{\frac{(u^h)^2}{2}}^G = \begin{cases} \min_{u_+^h \le u \le u_-^h} \frac{\widehat{(u^h)^2}}{2}, & if \ u_+^h \le u_-^h \\ \max_{u_-^h \le u \le u_+^h} \frac{\widehat{(u^h)^2}}{2}, & otherwise \end{cases}$

Spectral decomposition of local Legendre polynomials

In a Fourier Galerkin context, the model term is chosen such that, near the shock, the high frequencies of the solution, which are responsible for the oscillations, are more smoothed than the lower ones. The Fourier basis allows a differentiation between the high frequencies of the solutions and the lower ones, hence enabling an efficient treatment of the oscillations near the shock. In our Discontinuous Galerkin discretization, where *local* Legendre basis are used, this distinction can no longer be made, and one needs to be particularly careful because of the fact that each element in a DG context had a bounded support. Each basis function over this element is therefore not periodic and, as a consequence, there is no direct analogy between the spectrum of a solution discretized on a DG cell and the same solution expressed in terms of a finite Fourier series over the same domain.

Loss of properties

The spectral formulation brings a certain number of properties that are lost when switching to Legendre basis, making the implementation of the dMSV more complicated. In particular, we can mention the following loss of properties when going from Fourier to Legendre:

- *non-orthogonality of Legendre polynomials' derivatives and non commutativity of the projection and space-differentiation operators*

  In a Fourier context, the derivative of the basis functions remain orthogonal one with another and the projection and space-differentiation operators commute; this is not the case anymore when Legendre basis are used. As we will see it in the section dedicated to the model term, if the model term is such that it in-

cludes more than one parameter, these properties have important consequences.

- *non-invariance in translation.*

  The Fourier basis is invariant under translations; Legendre series are not. Consequently, whereas the transition from stationary to moving shocks are straightforward in a Fourier context, the same operation in a Legendre context is a real challenge and has yet to be solved.

Gibbs phenomenon

The implementation of the dMSV methodology, which relies essentially on the Germano identity, involves projections of the solution on various scales. In Fourier Galerkin and Discontinuous Galerkin contexts, these projections generate the so-called Gibbs phenomenon when the solution is discontinuous, introducing spurious high-frequency oscillations. In some cases, these oscillations may be large enough to affect the evaluation of the model term's parameters, and lead to undervalued (hence inefficient) viscosities.

One possible way to mitigate the Gibbs phenomenon is to take a projection operator that is more suited to the discontinuities of the solution than the plain L^2 norm projection operator. The following plot shows two different projections of the "step" function $f(x)$:

$$f(x) = \begin{cases} 1, & if\ x \leq 0 \\ -1, & if x \geq 0 \end{cases}$$

on the Legendre basis $(L_i)_{i=0,\ldots,25}$.

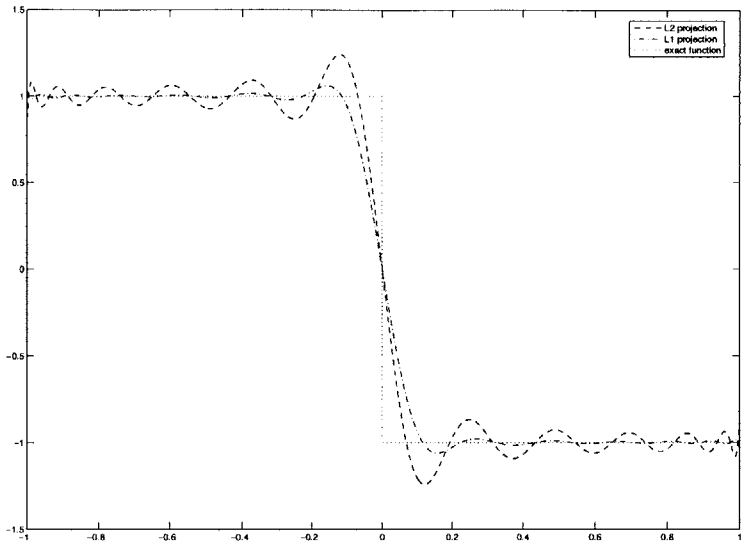

Figure 3-1: Projections of the step function on $(L_i)_{i=0,\ldots,25}$

This simple example shows how the L^1 projection can be used to reduce the amplitude of the Gibbs oscillations. As we will see it later, we will make extensive use of this projection in one of the test cases we ran. The methodology for the computation of L^1 projections will be detailed further in the chapter.

3.3.2 1-viscosity and 2-viscosity model terms

This section will describe the model terms we chose to implement in our shock-capturing algorithm and give more general insights on the way multi-parameter model terms are to be constructed.

1-viscosity model term

The first model term we implemented is the simple 1-viscosity model term. This term is of diffusive nature and its weak form can be written:

$$M(v, u, s, \overline{\nu}) = -\overline{\nu} \int_{\Omega} \Delta u \cdot v \, dx$$

The corresponding one-dimensional Local Discontinuous Galerkin representation is:

$$\begin{aligned} M(v^h, u^h, s, \overline{\nu}) &= -\overline{\nu} \left[\widehat{q^h} v^h \right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} + \overline{\nu} \int_{\mathcal{T}_h} q^h (v^h)_x \, dx \\ q_j^{h'} \int_{\mathcal{T}_{h'}} L_j^{h'} L_j^{h'} \, dx &= \left[\widehat{u^{h'}} L_j^{h'} \right]_{h'-\frac{1}{2}}^{h'+\frac{1}{2}} - \int_{\mathcal{T}_{h'}} u^{h'} (L_j^{h'})_x \, dx \end{aligned} \tag{3.11}$$

with the fluxes: $\begin{cases} \widehat{u^h} = u_-^h \\ \widehat{q^h} = q_+^h \end{cases}$.

This model term has essentially been implemented for the validation of the method. As we will see later, its performance is not as high as that of more elaborated model terms and its main advantage relies on the relative simplicity of its LDG formulation.

We will now introduce a useful way of representing on which moments of the solution u^h and of the test function v^h the artificial viscosity is acting. Let us consider two axes representing the Legendre moments of the solution u^h (x-axis) and the test function v^h (y-axis). Given a fifth-order space $\mathcal{U}_5^h$ and the third-order subspace $\mathcal{U}_3^h$ (we need two scales to close the system), in our 1-viscosity model term, the parameter $\overline{\nu}$ is acting on the following squares:

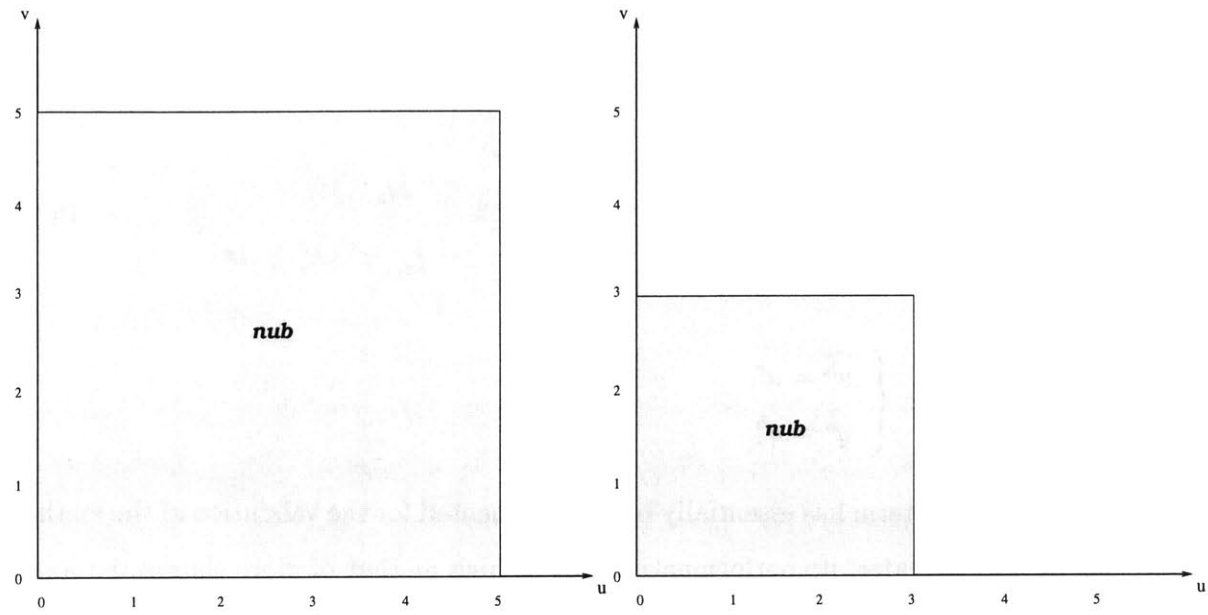

Figure 3-2: domains affected by the artificial viscosity $\overline{\nu}$ at scales 5 (right) and 3 (left)

In this case, the artificial viscosity $\overline{\nu}$ acts on every moment of the solution or the weighting function so its representation is straightforward. For more complicated model terms however, these plots give very valuable insights on whether or not these model terms fit within the dMSV framework.

2-viscosity model term

The goal of multiple-parameter model terms, such as our proposed 2-viscosity model term, is to introduce a separation of the modes of the solution, to group them and, to assign a different viscosity to each group. As we will see it in the next chapter, this increased complexity in the dMSV formulation is more than offset by the achieved shock-capturing performance.

If $\mathbb{P}^{\lfloor \frac{p}{2} \rfloor}$ denotes the projection over the subspace $\mathcal{U}_h^{\lfloor \frac{p}{2} \rfloor}$ and $\mathbb{I}$ the identity function then the weak form of our proposed 2-viscosity model term is:

$$M(v,u,s,\overline{\nu},\overline{\overline{\nu}}) = -\overline{\nu}\int_\Omega \Delta u \cdot v\, dx - \overline{\overline{\nu}}\int_\Omega \Delta(\mathbb{I}-\mathbb{P}^{\lfloor \frac{p}{2} \rfloor})u \cdot (\mathbb{I}-\mathbb{P}^{\lfloor \frac{p}{2} \rfloor})v\, dx \qquad (3.12)$$

The underlying idea is to distinguish between the modes that correspond to high-

order Legendre polynomials ($L_i, i \geq \lfloor \frac{p}{2} \rfloor$) and the ones that correspond to lower-order Legendre polynomials ($L_i, i \leq \lfloor \frac{p}{2} \rfloor$). Each of these two groups of moments has a different behavior near discontinuities. As a result, we are expecting better shock-capturing capabilities by assigning a specific artificial viscosity to each of them.
The one-dimensional Local Discontinuous Galerkin formulation for this 2-viscosity model term is as follow:

$$M(v^h, u^h, s, \overline{\nu}, \overline{\overline{\nu}}) = -\overline{\nu}(\left[\widehat{\overline{q}} \cdot v\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} \overline{q} \cdot (v)_x \, dx)$$
$$-\overline{\overline{\nu}}(\left[\widehat{\overline{\overline{q}}} \cdot (\mathbb{I} - \mathbb{P}^{\lfloor \frac{p}{2} \rfloor})\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} \overline{\overline{q}} \cdot \left[(\mathbb{I} - \mathbb{P}^{\lfloor \frac{p}{2} \rfloor})v\right]_x \, dx)$$

The auxiliary quantities $\overline{q}$ and $\overline{\overline{q}}$ being:

$$\overline{q}_j^{h'} \int_{\mathcal{T}_{h'}} L_j^{h'} L_j^{h'} \, dx = \left[\widehat{u^{h'}} L_j^{h'}\right]_{h'-\frac{1}{2}}^{h'+\frac{1}{2}} - \int_{\mathcal{T}_{h'}} u^{h'} (L_j^{h'})_x \, dx$$
$$\overline{\overline{q}}_{j'}^{h''} \int_{\mathcal{T}_{h''}} L_{j'}^{h''} L_{j'}^{h''} \, dx = \left[\widehat{(\mathbb{I} - \mathbb{P}^{\lfloor \frac{p}{2} \rfloor})} u^{h''} L_{j'}^{h''}\right]_{h''-\frac{1}{2}}^{h''+\frac{1}{2}} - \int_{\mathcal{T}_{h''}} (\mathbb{I} - \mathbb{P}^{\lfloor \frac{p}{2} \rfloor}) u^{h''} (L_{j'}^{h''})_x \, dx$$

and the fluxes: $\begin{cases} \widehat{u^{h'}} = u_-^{h'} \\ \widehat{(\mathbb{I} - \mathbb{P}^{\lfloor \frac{p}{2} \rfloor})} u^{h''} = (\mathbb{I} - \mathbb{P}^{\lfloor \frac{p}{2} \rfloor}) u_-^{h''} \\ \widehat{\overline{q}^h} = \overline{q}_+^h \\ \widehat{\overline{\overline{q}}^h} = \overline{\overline{q}}_+^h \end{cases}$.

With a 2-viscosity model term, three different scales are needed to form enough consistency conditions to close the system. Given a fifth-order space $\mathcal{U}_5^h$, the following two subspaces: $\mathcal{U}_3^h$ and $\mathcal{U}_1^h$ can be used as scales in the generalized Germano indentity (3.8) to determine $\overline{\nu}$ and $\overline{\overline{\nu}}$. In which case, the domains affected by each parameter at each scale are:

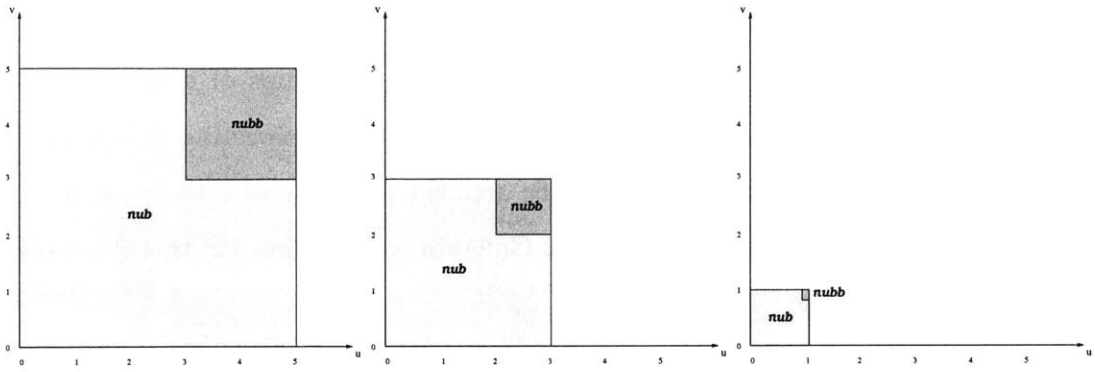

Figure 3-3: domains affected by the artificial viscosities $\overline{\nu}$ and $\overline{\overline{\nu}}$ at scales 5 (right), 3 (middle), and 1 (left)

Two remarks can then be made about this model term:

- Our proposed 2-viscosity model term is a generalization of the 2-viscosity model term proposed by Oberai and Wanderer in [9] to a context in which the space-differentiation and the projection operators do not commute and in which the derivatives of the basis functions do not have any orthogonality relationship.

- It illustrates the fact that, when dealing with multi-parameter model terms, the subspaces that will be used to derive the consistency conditions cannot be arbitrary. In our example, the modes are equally divided into two groups and this division is to remain the same for every scale on which the model term is to be applied. For example, choosing $\mathcal{U}_4^h$ would have led to an unbalanced partition of the moments between the two groups and must therefore be avoided since it may lead to an incorrect evaluation of the parameters $\overline{\nu}$ and $\overline{\overline{\nu}}$.
  The choice of a multi-parameter model term is therefore constrained by the structure of the model term itself, which has to be consistent at every scale on which the term will be applied. This consistency requirement reduces the number of scales that are actually available to write the Germano identities, as we showed it in the preceding paragraph. The reduced number of scales available reduces in turn the number of possible parameters in the model term.

3.3.3 Projections

Writing the consistency conditions for the model term's parameters involves projections on Legendre basis of various dimensions. Several projection operators can be used, but when Gibbs oscillations begin to develop, the choice of a specific projection operator has a direct impact on the performance of the algorithm. The purpose of this section is to present two projection operators that have been used in our dMSV routine.

L^2 projection

The L^2 projection is straightforward with our choice of a normalized Legendre basis on each element of the DG discretization. The L^2 projection operator $\mathbb{P}^m$ is defined by:

$$\mathbb{P}^m : \quad \mathcal{U}_p^h \longrightarrow \mathcal{U}_m^h, \quad p \geq m+1$$

$$\mathbb{P}^m(L_i^h) = \begin{cases} 0 \quad if \quad i \geq m+1 \\ L_i^h \quad if \quad i \leq m \end{cases}$$

Although very simple to implement and inexpensive, the results given by this L^2 projection are subject to the Gibbs phenomenon. In certain cases, another projection operator, such as the L^1 projection operator, which be detailed in the next section, will be needed to achieve a correct evaluation of the parameters.

L^1 projection

Let us consider a known function f defined over $[-1, 1]$ and $P_n = \sum_{i=0}^{n} \alpha_i L_i$ its L^1 projection over the space of polynomials $\mathbb{R}_n[X]$. Then the L^1 projection is defined by the α_i, which are such that the quantity (which represents the L^1 norm of $f - P_n$) is minimized:

$$\{\alpha_0, \alpha_1, \cdots, \alpha_n\} = \arg\min_{\alpha} \int_{-1}^{+1} |f(x) - P_m(x)| \, dx \tag{3.13}$$

This equation is then turned into a Linear Programming problem and solved.

If we divide the interval $[-1, 1]$ into M equal subinterval, the integral (3.13) can be approximated by the following expression, with $h = \frac{2}{M}$, $F_m = hF(x_m)$, $L_m^i = hL^i(x_m)$ and $x_m = 2\frac{m-1}{M-1} - 1$:

$$\sum_{m=1}^{M} |F(x_m) - P_n(x_m)|h = \sum_{m=1}^{M} |F_m - \sum_{i=0}^{n} \alpha_i L_m^i|$$

whose associated Linear Programming problem is:
Find $\alpha_0, ..., \alpha_n, \tau_1, ..., \tau_M$ such that:

$$\min \sum_{m=0}^{M} \tau_m$$

under the constraints:

$$\begin{array}{rcl}
F_1 - \sum_{i=0}^{n} \alpha_i L_1^i & \leq & \tau_1 \\
F_2 - \sum_{i=0}^{n} \alpha_i L_2^i & \leq & \tau_2 \\
 & \cdots & \\
F_M - \sum_{i=0}^{n} \alpha_i L_M^i & \leq & \tau_M \\
-\tau_1 & \leq & F_1 - \sum_{i=0}^{n} \alpha_i L_1^i \\
-\tau_2 & \leq & F_2 - \sum_{i=0}^{n} \alpha_i L_2^i \\
 & \cdots & \\
-\tau_M & \leq & F_M - \sum_{i=0}^{n} \alpha_i L_M^i
\end{array}$$

which, in turn, can be written more conveniently in the following matrix form:

$$y = [\alpha_0, ..., \alpha_n, \tau_1, ..., \tau_M]^T$$
$$\min \overbrace{[0, ..., 0}^{n+1}, \overbrace{1, ..., 1]}^{M} \times y$$

under the constraints:

$$\begin{bmatrix} -L_1^0 & -L_1^1 & \cdots & -L_1^n & -1 & 0 & \cdots & 0 \\ -L_2^0 & -L_2^1 & \cdots & -L_2^n & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -L_M^0 & -L_M^1 & \cdots & -L_M^n & 0 & 0 & \cdots & -1 \\ L_1^0 & L_1^1 & \cdots & L_1^n & -1 & 0 & \cdots & 0 \\ L_2^0 & L_2^1 & \cdots & L_2^n & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ L_M^0 & L_M^1 & \cdots & L_M^n & 0 & 0 & \cdots & -1 \end{bmatrix} \cdot \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \\ \tau_1 \\ \tau_2 \\ \vdots \\ \tau_M \end{bmatrix} + \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_M \\ -F_1 \\ -F_2 \\ \vdots \\ -F_M \end{bmatrix} \leq 0$$

3.3.4 Algorithms structure

Regardless of which model term is used, our dMSV algorithms all have a common structure, which is presented in the following figure:

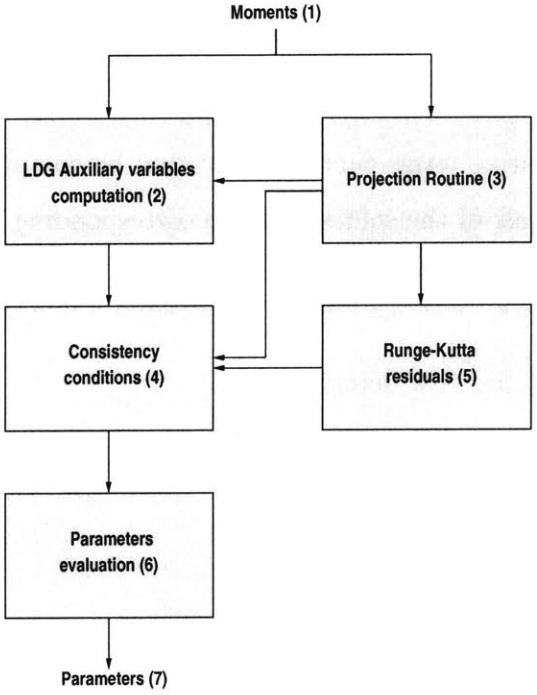

Figure 3-4: dMSV algorithms structure

where:

1. The moments of the solution over the whole domain are given as inputs to the algorithm. Our implementation used a Runge-Kutta time-marching scheme; the tests showed that it is usually unnecessary to run the algorithm at each stage of the Runge-Kutta scheme.

2. The LDG auxiliary variables (i.e. the variables q) are computed. The number of auxiliary variables needed are directly linked to the structure of the model term.

3. The moments have to be projected on different scales during the parameter evaluation process, and it usually does make sense to compute these projections at the beginning of the algorithm since they will be used in (2), (4) and (5). When straightforward projections are used, such as the L^2 projection, the gain is not significant, but it becomes important when the computation of the projection is expensive, for example when L^1 projections are used.

4. The various quantities involved in the consistency conditions are evaluated, leaving the model term parameters as unknowns.

5. The time-derivative terms that are evaluated by computing the Discontinuous Galerkin residuals of the solution at the corresponding scales.

6. The linear system derived from the consistency conditions is solved.

7. The parameters are now determined.

Chapter 4

Results

In this chapter, the numerical results that have been obtained using the previously described Dynamic Multiscale Viscosity methodology are presented. Two model terms have been tested, the first one involving only one parameter and the second one involving two artificial viscosities. Regardless of which model term is used, we achieved optimal accuracy in the case of smooth solutions. The shock-capturing capability of each model term will also be discussed.

4.1 1-viscosity model term

The procedure presented in the section uses the simple 1-viscosity model term (3.11). The Runge-Kutta discontinuous Galerkin scheme is built upon fifth-order elements and a fourth-order time marching algorithm.
For the derivation of the consistency conditions and the evaluation of the artificial viscosity, the L^1 projection operator presented in the preceding chapter, the scales $\mathbb{R}_3^h[X]$ $\mathbb{R}_5^h[X]$ and the dissipation method have been used.

4.1.1 Smooth solution

The first test aimed at determining whether the accuracy of the Runge-Kutta Discontinuous Galekin scheme was affected by the presence of a model term, in the case of a

smooth solution. To do so, we considered the one-dimensional linear transport equation, to which we added the 1-viscosity model term (3.11) presented in the preceding chapter.

$$\int_\Omega \frac{\partial u^h}{\partial t} v^h \, dx + \int_\Omega \frac{\partial u^h}{\partial x} v^h \, dx + M(v^h, u^h, s, \overline{\nu}) = 0$$

with the initial condition: $u(x,0) = \frac{1}{2} + \sin(\pi x)$ and periodic boundary conditions.

Iterated up to $t = \frac{1.5}{\pi}$, the numerical solution is then compared to the analytical one. The L^2 norms of the errors are shown in the table below for various mesh sizes:

number of elements	20	30	40
$\|\|e\|\|_{L^2}$	5.366236E-8	4.418526E-9	9.367500E-10

Table 4.1: L^2 error for the transport equation with a smooth initial condition and a 1-viscosity model term

These results are used to compute the rate of convergence of the scheme. The rate of convergence of the procedure is presented on the figure below, the x-axis of the plot corresponding to the logarithm of the number of elements in the grid and the y-axis to the logarithm or the error:

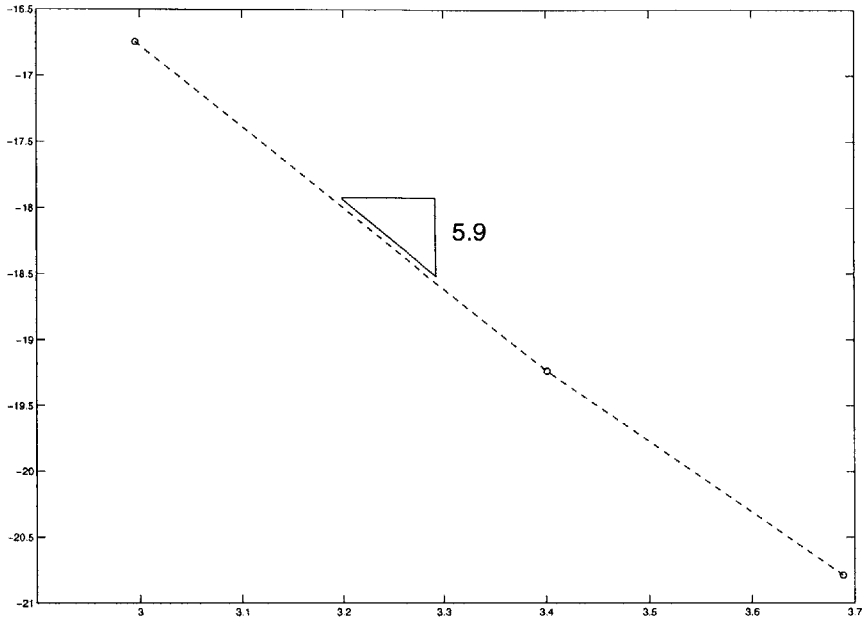

Figure 4-1: 1-viscosity dMSV - convergence rate for a smooth solution

We observe that the rate of convergence of the algorithm is close to the optimal one which, in our case, is $O(h^{5+\frac{1}{2}})$. As a result, when the solution is smooth, the code achieves optimal accuracy.

4.1.2 Burgers' equation

The 1-viscosity model term procedure has been tested on the viscous Burgers' equation, on a 19-element grid, with the initial condition $u(x,0) = -\sin(\frac{\pi}{2}x)$ and the boundary conditions $u^h(-1) = 1$, $u^h(1) = -1$ and $\overline{q}(1) = 0$. The strength of the shock generated by the equation is represented by the Peclet number $Pe = \frac{1.5 \cdot \pi}{\nu}$, where ν is the physical viscosity.

The results for various Peclet numbers are presented below:

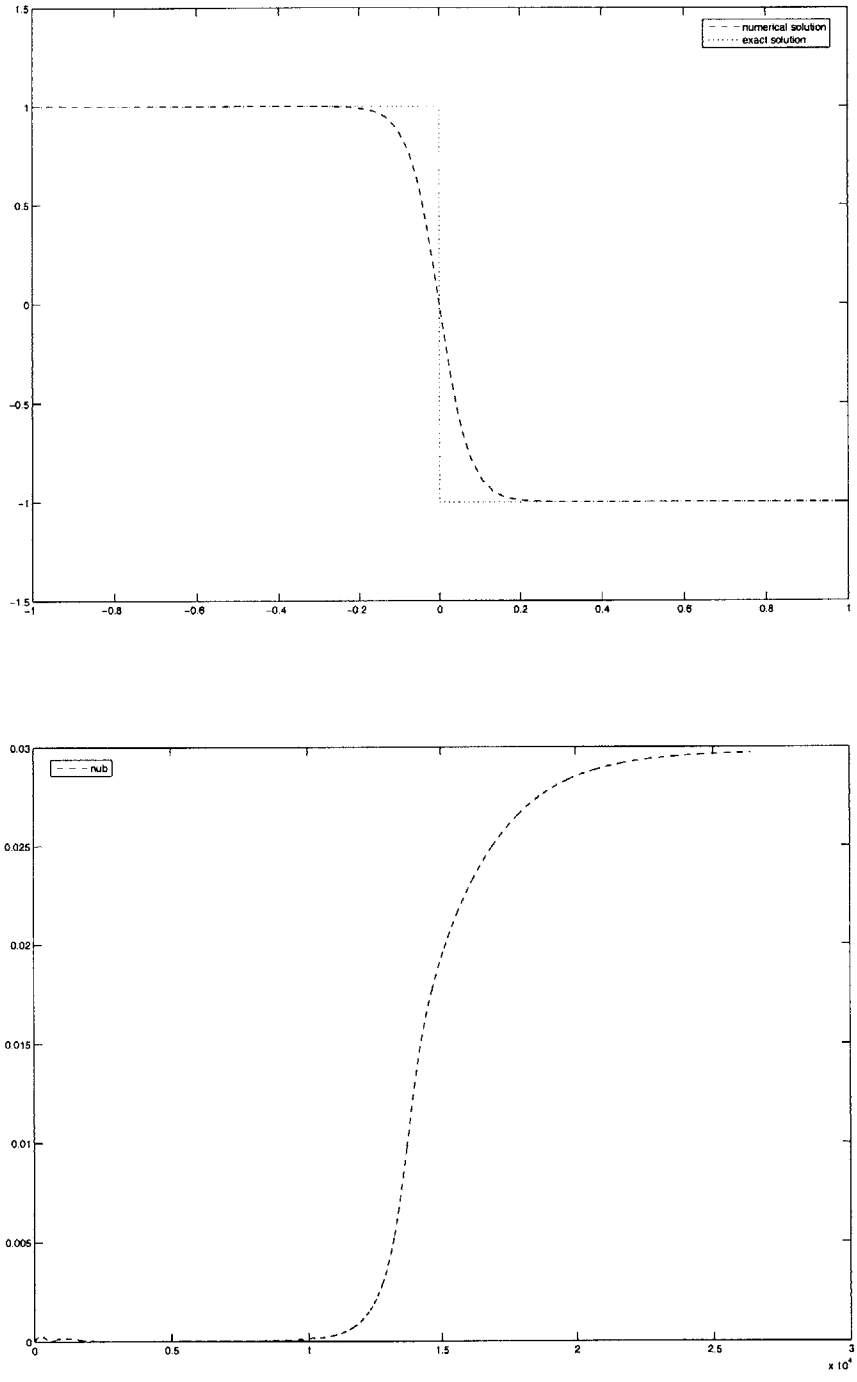

Figure 4-2: iterated solution (top) and artificial viscosity evolution (bottom) over the time period $\left[0, \frac{3.5}{\pi}\right]$ for Pe=100

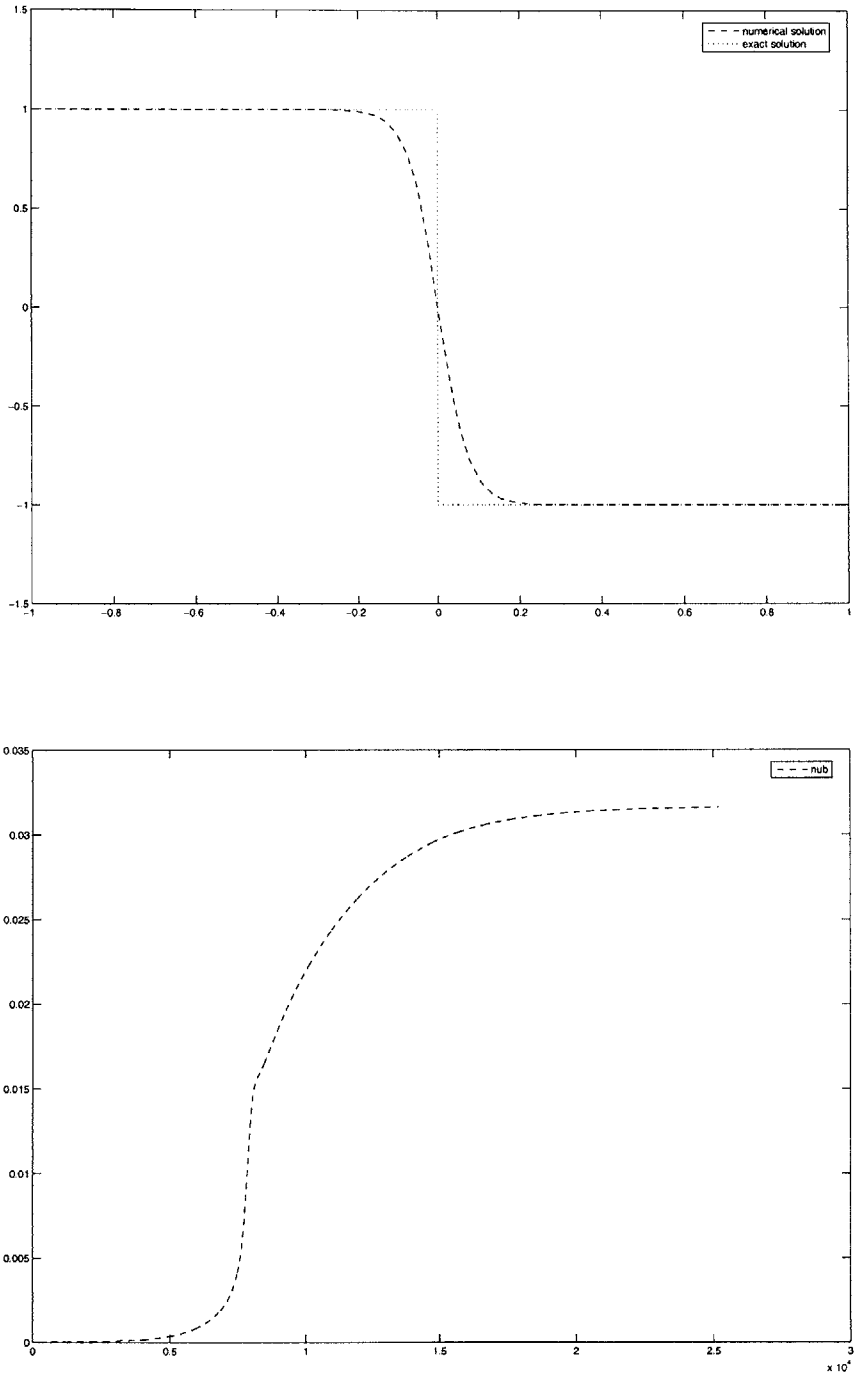

Figure 4-3: iterated solution (top) and artificial viscosity evolution over the time period $\left[\frac{1.5}{\pi}, \frac{4.5}{\pi}\right]$ (bottom) for Pe=+inf

These figures validate the dMSV methodology as a way of capturing shocks in a Discontinuous Galerkin context. Regardless of the Peclet number, the shock was properly captured in every case we tested. However, two factors negatively impacted the cost of the algorithm:

- The amount of artificial viscosity produced by the dMSV routine is such that it requires smaller CFL numbers (i.e. more time iterations) in order to keep the algorithm stable.

- To mitigate the effects of the Gibbs phenomenon, a L^1 projection was used in the evaluation routine (trials with a L^2 projection showed oscillations for Peclet numbers greater than 25). As we explained it in chapter 3, each L^1 projection requires the solving of linear programming problem, which greatly increases the computational cost of each time iteration.

Even though, the 1-viscosity model term dMSV algorithm performs the shock capturing correctly, it is computationally costly and this cost prevents it from being used for anything else than the mere validation of the algorithm. For operational purposes, the 2-viscosity model term algorithm, although more complicated to implement, must be preferred.

4.2 2-viscosity model term

In this section, we describe the implementation of the proposed 2-viscosity model term (3.12). The space discretization is a fifth-order Discontinuous Galerkin and the time iteration is performed by the fourth-order Runge-Kutta scheme presented at the end of chapter 2.
The parameter evaluation routine uses L^2 projections and the scales $\mathbb{R}_5^h[X]$, $\mathbb{R}_3^h[X]$ and $\mathbb{R}_1^h[X]$ for the derivation of the consistency conditions. As for the 1-parameter case, the evaluation of the parameters is achieved through a dissipation method.

4.2.1 smooth solution

In order to assess the impact of the addition of a model term on the accuracy of the Runge-Kutta Discontinuous Galerkin method, an accuracy study was conducted on the one-dimensional linear transport equation, to which we added our 2-viscosity model term:

$$\int_\Omega \frac{\partial u^h}{\partial t} v^h \, dx + \int_\Omega \frac{\partial u^h}{\partial x} v^h \, dx + M(v^h, u^h, s, \overline{\nu}, \overline{\overline{\nu}}) = 0$$

with the initial condition: $u(x,0) = \frac{1}{2} + \sin(\pi x)$ and periodic boundary conditions. The solution is interated up until $t = \frac{1.5}{\pi}$ and the L^2 norm of the error between the computed and the numerical solutions is then estimated. The results are presented in the table below:

number of elements	20	30	40
$\|\|e\|\|_{L^2}$	5.296765E-8	4.458097E-9	9.419499E-10

Table 4.2: L^2 error for the transport equation with a smooth initial condition and a 2-viscosity model term

The rate of convergence of the algorithm is presented on the figure below, the x-axis of the plot corresponding to the logarithm of the number of elements in the grid and the y-axis to the logarithm or the error:

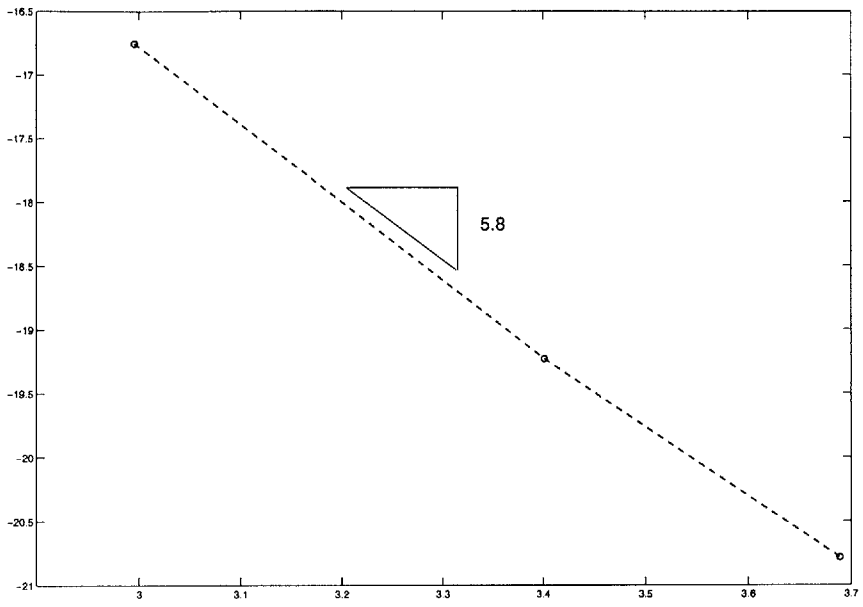

Figure 4-4: 2-viscosity dMSV - convergence rate for a smooth solution

The rate of convergence of the scheme is close to the optimal one ($O(h^{5+\frac{1}{2}})$). The accuracy of our RKDG scheme is therefore not penalized by the presence of the shock-capturing model term in the case of a smooth solution.

4.2.2 Burgers' equation

The scheme was then tested with a shock-generating equation. Similarly to the 1-viscosity model term case, we used the viscous Burgers' equation with the initial condition $u(x,0) = -\sin(\frac{\pi}{2}x)$ and the boundary conditions $u^h(-1) = 1$, $u^h(1) = -1$, $\overline{q}(1) = 0$ and $\overline{\overline{q}}(1) = 0$. The strength of the shock generated by Burgers' equation is still asssessed by a cell Peclet number, defined by $Pe = \frac{1.5 \cdot h}{\nu}$ where ν is the physical viscosity.

The results of these tests are shown on the following plots for various Peclet numbers together with the associated evolution of the artificial viscosities with respect to time (for every of these tests, the number of elements in the grid is constant and equal to 19):

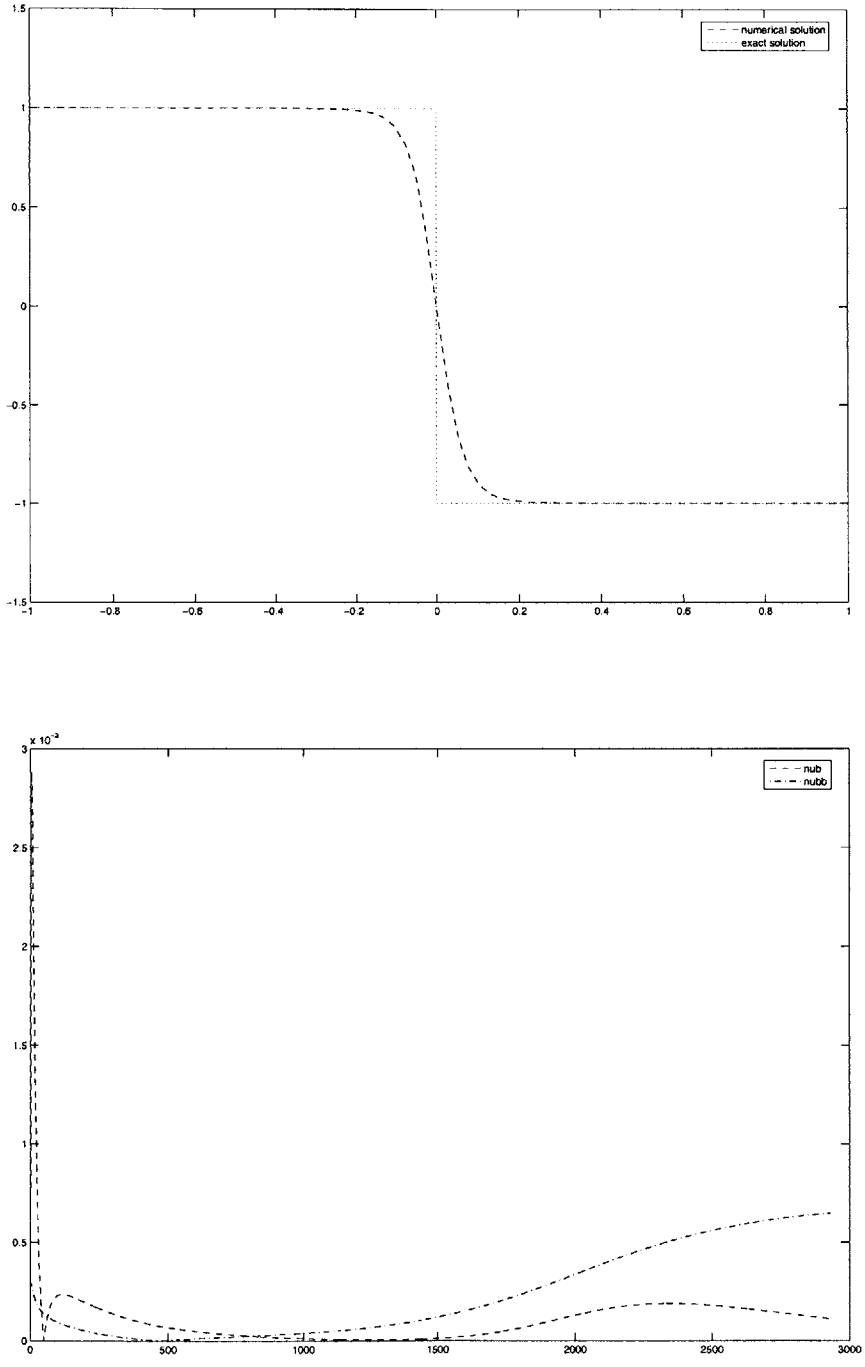

Figure 4-5: iterated solution (top) and artificial viscosities evolution over the time period $\left[0, \frac{3.5}{\pi}\right]$ (bottom) for Pe=5

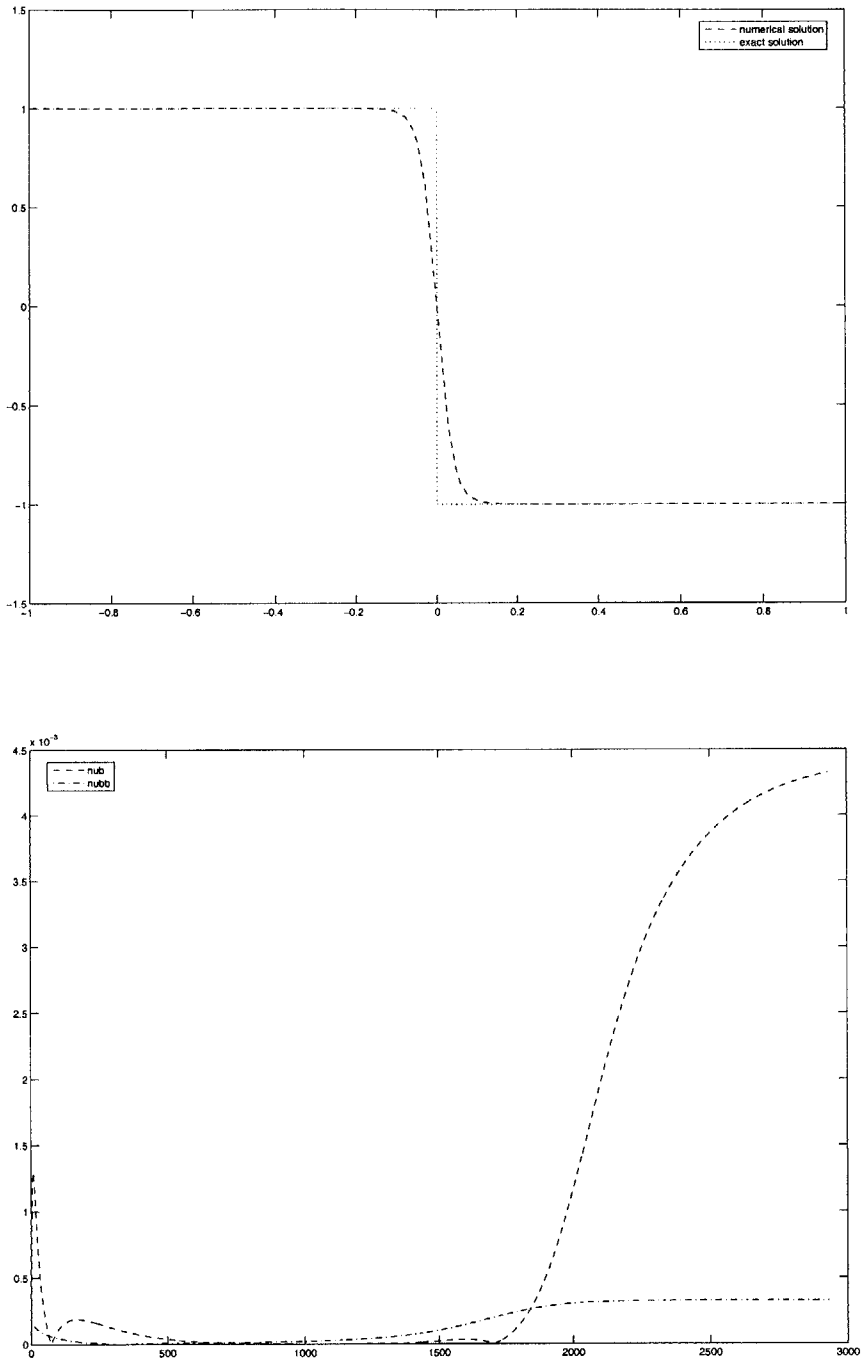

Figure 4-6: iterated solution (top) and artificial viscosities evolution over the time period $\left[0, \frac{3.5}{\pi}\right]$ (bottom) for Pe=10

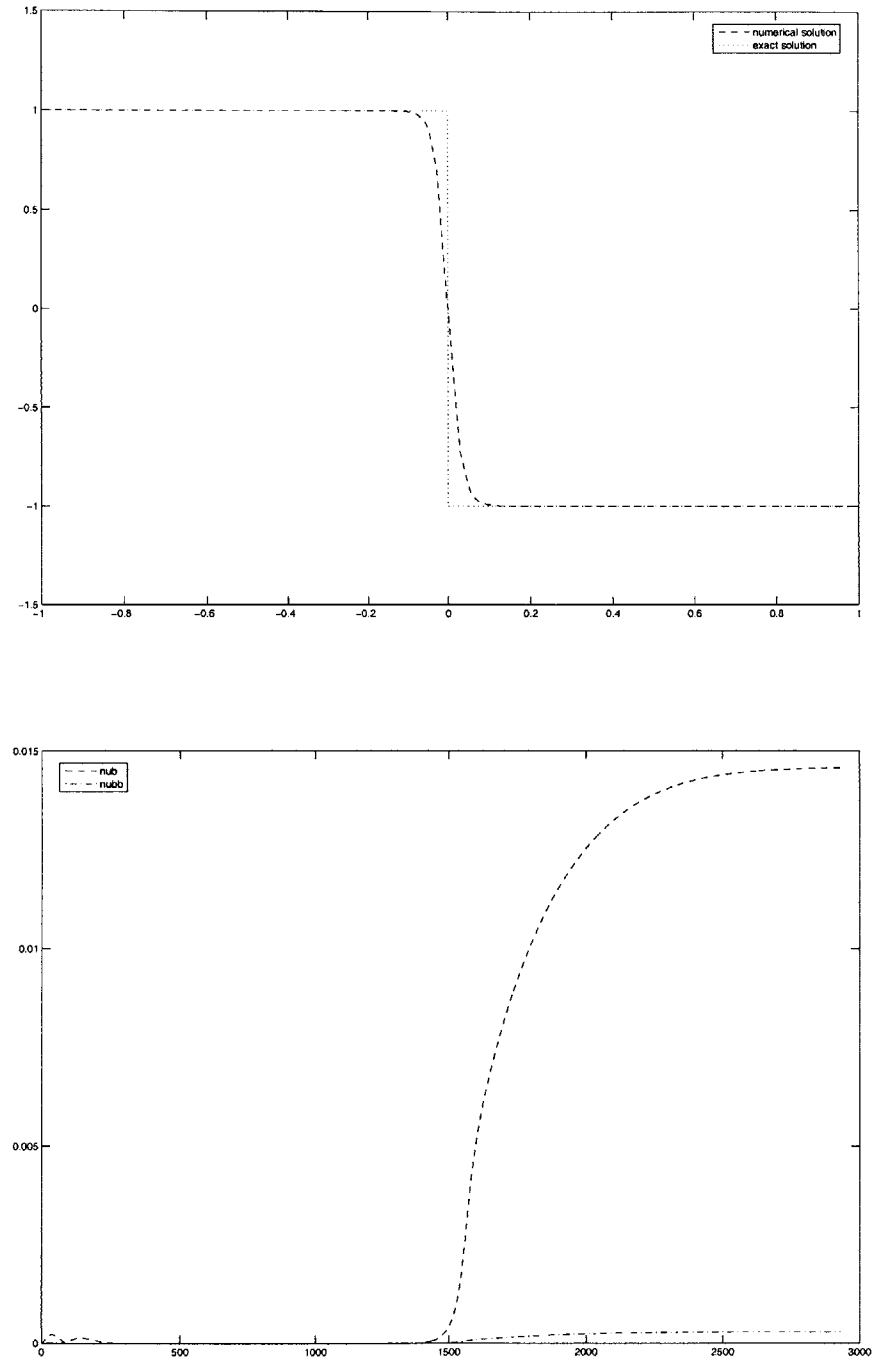

Figure 4-7: iterated solution (top) and artificial viscosities evolution over the time period $\left[0, \frac{3.5}{\pi}\right]$ (bottom) for Pe=100

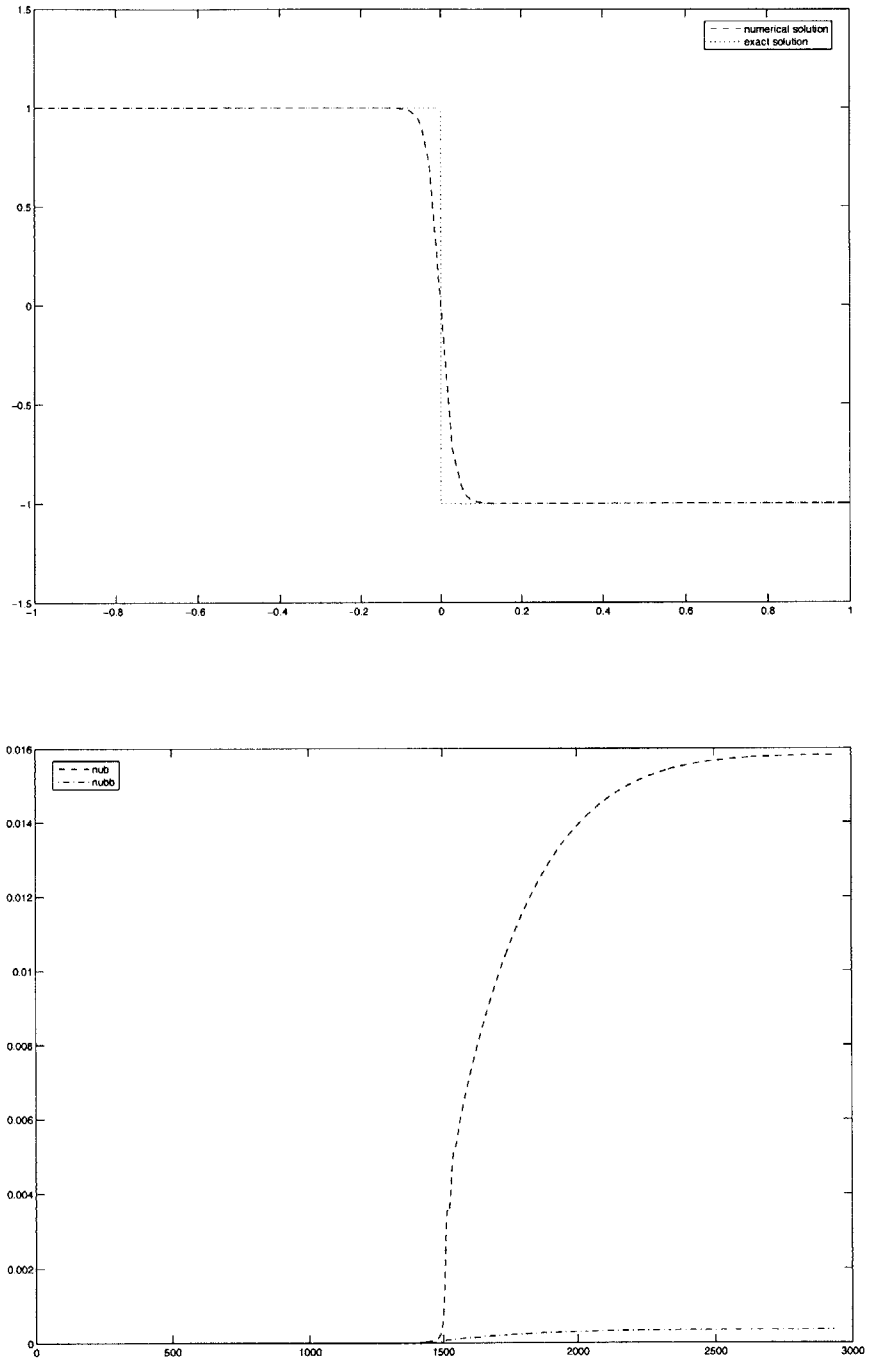

Figure 4-8: iterated solution (top) and artificial viscosities evolution over the time period $\left[0, \frac{3.5}{\pi}\right]$ (bottom) for Pe=10^6

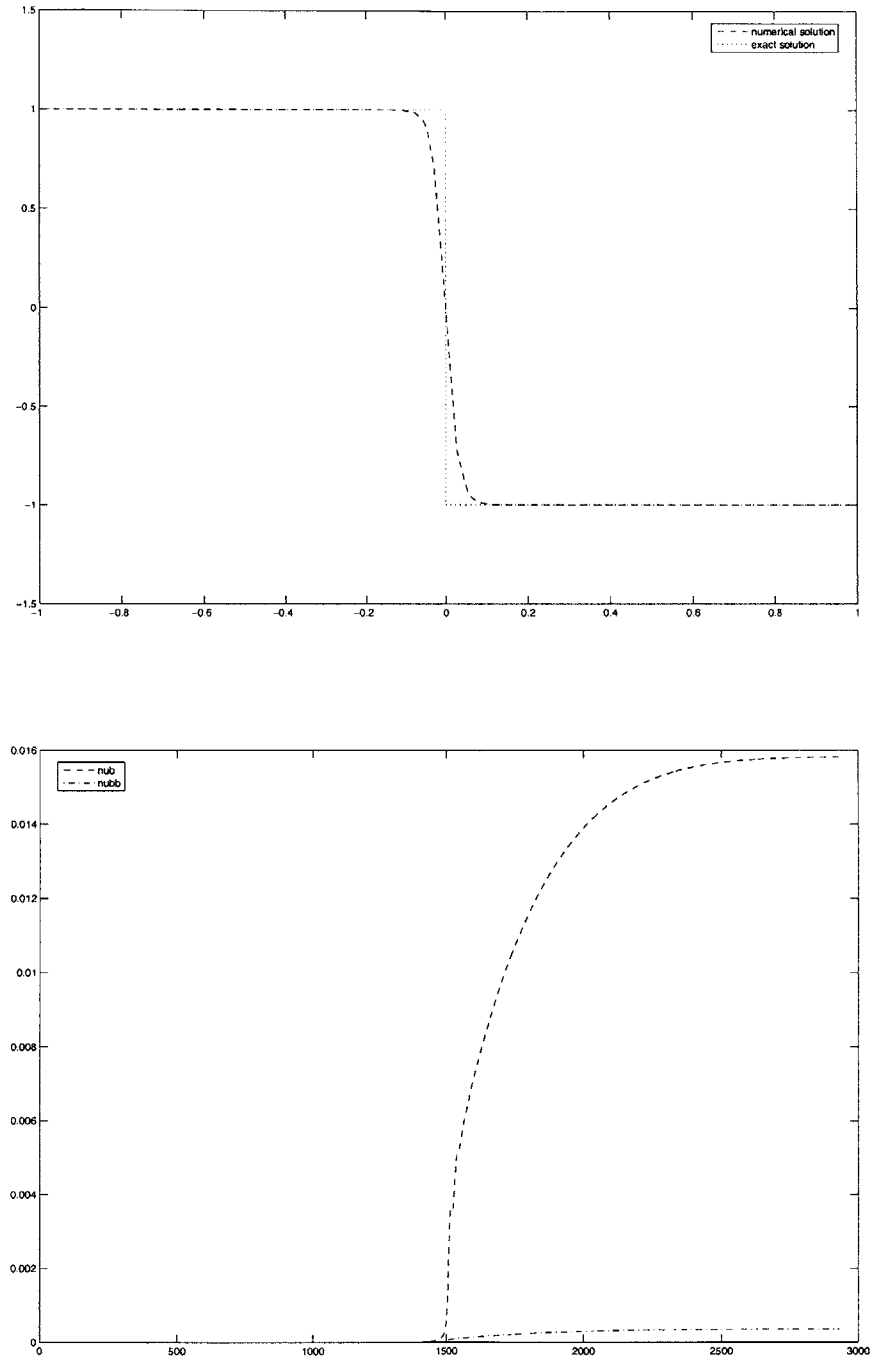

Figure 4-9: iterated solution (top) and artificial viscosities evolution over the time period $\left[0, \frac{3.5}{\pi}\right]$ (bottom) for Pe=+ inf

The following obsevations can be made about these results:

- Compared to the 1-viscosity model term case, the shock is better captured. As a result, we conclude that increasing the number of parameters of the model term to better adapt the smoothing to the Legendre-spectrum of the computed solution yields more accurate results. In addition to that, the amount of artificial viscosity needed for the shock capturing is less than what was required by the 1-viscosity model term, thus allowing for the use of larger CFL numbers.

- The viscosity plots show that the viscosity $\overline{\overline{\nu}}$ related to the high Legendre modes is usually smaller than $\overline{\nu}$, which is computed from the whole Legendre spectrum of the solution. This result may look paradoxical and contradictory to the results obtained by John Wanderer and Assad Oberai in [9] who found that, in a Fourier Galerkin context, $\overline{\overline{\nu}}$ was larger than $\overline{\nu}$ - result which makes intuitive sense-. However, one has to keep in mind that there is no intuitive correlation, in our situation, between the Fourier frequencies and the amplitudes associated to each Legendre polynomial in the Legendre decomposition of the solution. The fact that, in our case, $\overline{\overline{\nu}}$ is greater than $\overline{\nu}$ is explained by the fact that high frequency Fourier oscillations does not necessarily correspond to high frequency Legendre oscillations.

- Unlike the 1-viscosity model term case which required the use of an L^1 projection operator, the 2-viscosity model term case works fine with the L^2 projection, hence decreasing the computational cost of the algorithm.

4.3 Conclusion

A higher-order one-dimensional Burgers' equation Runge-Kutta Discontinuous Galerkin solver was developped. Its accuracy was successfully tested and the advantage of using higher-order polynomials over grid refinement was proven. In order to enhance the shock capturing capabilities of the algorithm, a procedure was developed and implemented.

This scheme is based on the Dynamic Multiscale Viscosity (dMSV) methodology presented by Oberai in [9] and [8]. This methodology was adapted to the Runge-Kutta Discontinuous Galerkin context. This method is general and does not depend on the space dimension nor does it depend on the order of the interpolants. As a result, provided the order of the interpolants are compatible with the required scales for the evaluation of the model term's parameters, the method is applicable to any order and any spatial dimension.

Using this method, a shock capturing routine for the Burgers' equation in a fifth-order Runge Kutta Discontinuous Galerkin context was developped. This routine is able to handle 1-parameter and 2-parameter model terms and uses a Local Discontinuous Galerkin formulation presented by Shu in [2] to discretize the diffusive parts of the model terms. With either the 1-parameter model term or the 2-parameter one, the routine maintains high accuracy in the regions where the solution is smooth and presents good shock capturing capabilities for stationary shocks. The result is a new shock capturing alternative for Runge Kutta Discontinuous Galerkin solvers.

To enhance these results, future research directions may include:

- An investigation of new, computationally cheaper, projection operators, or the implementation of new projection methods, especially in the case of the 1-viscosity model term, whose L^1 projection operator is very expensive.

- The moving shock case, which may require more elaborated projection operators or eveluation methods.

- Alternative model terms. Going from one parameter to two parameters in the model term greatly improves the shock capturing capabilities of the code. So increasing the number of parameters in the model term is definitely a promising way to improved performances.

Appendix A

1-viscosity model term consistency condition

In this appendix, the consistency condition for the evaluation of the parameter of the model term (3.11) for the viscous Burgers' equation in a DG context is presented.

Weak formulation

The weak formulation of the viscous Burgers' equation, using a LDG scheme, is:

$$\sum_{h\in(\mathcal{T}_h)} \int_{\mathcal{T}_h} \frac{\partial u^h}{\partial t} v^h dx + \left[\mathcal{F}(\frac{(u^h)^2}{2} - \nu q^h)\, v^h \right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} (\frac{(u^h)^2}{2} - \nu q^h) \frac{dv^h}{dx} dx + M(v^h, u^h, s, \overline{\nu}) = 0$$

with:

$$\frac{\Delta^h}{2} \cdot q_i^h = \left[\mathcal{F}(u^h) L_i^h \right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} u^h \frac{dL_i^h}{dx}\, dx$$

$(L_i^h)_{i,h}$ being the local Legendre basis we introduced in chapter 2 and Δ^h the length of an element.

Model term

Forcing an explicit dependance of the parameters on the order of the solution, the model term can be written, for u^h of order n:

$$M(v^h, u^h, s, \overline{\nu}) = -\sum_{h\in(\mathcal{T}_h)} \frac{\overline{\nu}}{n} \int_{\mathcal{T}_h} \Delta u^h \, v^h \, dx$$

or, using the LDG methodology:

$$M(v^h, u^h, s, \overline{\nu}) = -\frac{\overline{\nu}}{n} \left[\mathcal{F}(q^h) v^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} + \frac{\overline{\nu}}{n} \int_{\mathcal{T}_h} q^h (v^h)_x \, dx$$

with:

$$\frac{\Delta^h}{2} \cdot q_i^h = \left[\mathcal{F}(u^h) L_i^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} u^h (L_i^h)_x \, dx$$

Consistency condition

For a fifth-order solution, using the scales $\mathbb{R}_5[X]$, $\mathbb{R}_3[X]$ and the dissipation method, the consistency condition can be written:

$$-A\,\overline{\nu} = B$$

$$A = \sum_{h\in(\mathcal{T}_h)} \left[\mathcal{F}\left(\frac{\widetilde{q^h}}{3} - \frac{\overline{q^h}}{5}\right) \mathbb{P}^3 u^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} \left(\frac{\widetilde{q^h}}{3} - \frac{\overline{q^h}}{5}\right) (\mathbb{P}^3 u^h)_x \, dx$$

$$B = \sum_{h\in(\mathcal{T}_h)} \int_{\mathcal{T}_h} \frac{\partial}{\partial t}(u^h - \mathbb{P}^3 u^h)(\mathbb{P}^3 u^h)\, dx + \left[\mathcal{F}\left(\frac{(u^h)^2}{2} - \nu\overline{q^h} - \frac{(\mathbb{P}^3 u^h)^2}{2} + \nu\widetilde{\overline{q^h}}\right) \mathbb{P}^3 u^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}$$

$$- \int_{\mathcal{T}_h} \left(\frac{(u^h)^2}{2} - \nu\overline{q^h} - \frac{(\mathbb{P}^3 u^h)^2}{2} + \nu\widetilde{\overline{q^h}}\right) (\mathbb{P}^3 u^h)_x \, dx$$

where $\mathcal{F}(x)$ means that an appropriate flux of the expression x must be used, instead of just x.

Appendix B

2-viscosity model term consistency conditions

In this appendix, the consistency conditions for the evaluation of the parameters of the model term (3.12) for the viscous Burgers' equation in a DG context are presented.

Weak formulation

The weak formulation of the viscous Burgers' equation, using a LDG scheme, is:

$$\sum_{h\in(\mathcal{T}_h)} \int_{\mathcal{T}_h} \frac{\partial u^h}{\partial t} v^h dx + \left[\mathcal{F}(\frac{(u^h)^2}{2} - \nu q^h)\, v^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} (\frac{(u^h)^2}{2} - \nu q^h) \frac{dv^h}{dx} dx + M(v^h, u^h, s, \overline{\nu}, \overline{\overline{\nu}}) = 0$$

with:

$$\frac{\Delta^h}{2} \cdot q_i^h = \left[\mathcal{F}(u^h) L_i^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} u^h \frac{dL_i^h}{dx}\, dx$$

$(L_i^h)_{i,h}$ being the local Legendre basis we introduced in chapter 2 and Δ^h the length of an element.

Model term

Forcing an explicit dependance of the parameters on the order of the solution, the model term can be written, for u^h of order n:

$$M(v^h, u^h, s, \overline{\nu}, \overline{\overline{\nu}}) = -\sum_{h\in(\mathcal{T}_h)} \frac{\overline{\nu}}{n}\int_{\mathcal{T}_h} \Delta u^h\, v^h\, dx + \frac{\overline{\overline{\nu}}}{n}\int_{\mathcal{T}_h} (\Delta(\mathbb{I}-\mathbb{P}^{\lfloor\frac{n}{2}\rfloor})u^h)((\mathbb{I}-\mathbb{P}^{\lfloor\frac{n}{2}\rfloor})v^h)\, dx$$

or, using the LDG methodology:

$$M(v^h, u^h, s, \overline{\nu}, \overline{\overline{\nu}}) = \sum_{h\in(\mathcal{T}_h)} -\frac{\overline{\nu}}{n}\Big(\Big[\mathcal{F}(\overline{q^h})v^h\Big]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathbb{T}_h} \overline{q^h}\frac{dv^h}{dx}\, dx\Big)$$

$$-\frac{\overline{\overline{\nu}}}{n}\Big(\Big[\mathcal{F}(\overline{\overline{q^h}})(\mathbb{I}-\mathbb{P}^{\lfloor\frac{n}{2}\rfloor})v^h\Big]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_(} \overline{\overline{q^h}}\frac{d}{dx}((\mathbb{I}-\mathbb{P}^{\lfloor\frac{n}{2}\rfloor})v^h)\, dx\Big)$$

and:

$$\frac{\Delta^h}{2}\cdot\overline{q_i^h} = \big[\mathcal{F}(u^h)L_i^h\big]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} u^h\frac{d}{dx}(L_i^h)\, dx$$

$$\frac{\Delta^h}{2}\cdot\overline{\overline{q_i^h}} = \big[\mathcal{F}((\mathbb{I}-\mathbb{P}^{\lfloor\frac{n}{2}\rfloor})u^h)L_i^h\big]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} ((\mathbb{I}-\mathbb{P}^{\lfloor\frac{n}{2}\rfloor})u^h)\frac{d}{dx}(L_i^h)\, dx$$

Consistency conditions

For a fifth-order solution, using the scales $\mathbb{R}_5[X]$, $\mathbb{R}_3[X]$, $\mathbb{R}_1[X]$ and the dissipation method, the consistency conditions can be written:

$$-\begin{bmatrix} A & B \\ D & E \end{bmatrix}\begin{bmatrix} \overline{\nu} \\ \overline{\overline{\nu}} \end{bmatrix} = \begin{bmatrix} C \\ F \end{bmatrix}$$

with:

$$A = \sum_{h\in(\mathcal{T}_h)} \left[\mathcal{F}\Big(\frac{\widetilde{\overline{q^h}}}{3} - \frac{\overline{q^h}}{5}\Big)\,\mathbb{P}^3u^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h}\Big(\frac{\widetilde{\overline{q^h}}}{3} - \frac{\overline{q^h}}{5}\Big)\,(\mathbb{P}^3u^h)_x\, dx$$

$$B = \sum_{h\in(\mathcal{T}_h)} \frac{1}{3}\left[\mathcal{F}(\widetilde{\overline{\overline{q^h}}})(\mathbb{I}-\mathbb{P}^1)(\mathbb{P}^3u^h)\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \frac{1}{5}\left[\mathcal{F}(\overline{\overline{q^h}})(\mathbb{I}-\mathbb{P}^2)(\mathbb{P}^3u^h)\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}$$

$$-\frac{1}{3}\int_{\mathcal{T}_h}\widetilde{\overline{q^h}}\left[(\mathbb{I}-\mathbb{P}^1)(\mathbb{P}^3u^h)\right]_x\,dx+\frac{1}{5}\int_{\mathcal{T}_h}\overline{\overline{q^h}}\left[(\mathbb{I}-\mathbb{P}^2)(\mathbb{P}^3u^h)\right]_x\,dx$$

$$C=\sum_{h\in(\mathcal{T}_h)}\int_{\mathcal{T}_h}\frac{\partial}{\partial t}(u^h-\mathbb{P}^3u^h)(\mathbb{P}^3u^h)\,dx+\left[\mathcal{F}(\frac{(u^h)^2}{2}-\nu\overline{q^h}-\frac{(\mathbb{P}^3u^h)^2}{2}+\nu\widetilde{\overline{q^h}})\,\mathbb{P}^3u^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}$$

$$-\int_{\mathcal{T}_h}(\frac{(u^h)^2}{2}-\nu\overline{q^h}-\frac{(\mathbb{P}^3u^h)^2}{2}+\nu\widetilde{\overline{q^h}})\,(\mathbb{P}^3u^h)_x\,dx$$

$$D=\sum_{h\in(\mathcal{T}_h)}\left[\mathcal{F}(\frac{\widehat{\overline{q^h}}}{1}-\frac{\overline{q^h}}{5})\,\mathbb{P}^1u^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}-\int_{\mathcal{T}_h}(\frac{\widehat{\overline{q^h}}}{1}-\frac{\overline{q^h}}{5})\,(\mathbb{P}^1u^h)_x\,dx$$

$$E=\sum_{h\in(\mathcal{T}_h)}\frac{1}{1}\left[\mathcal{F}(\widehat{\overline{q^h}})(\mathbb{I}-\mathbb{P}^0)(\mathbb{P}^1u^h)\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}-\frac{1}{5}\left[\mathcal{F}(\overline{\overline{q^h}})(\mathbb{I}-\mathbb{P}^2)(\mathbb{P}^1u^h)\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}$$

$$-\frac{1}{1}\int_{\mathcal{T}_h}\widehat{\overline{\overline{q^h}}}\left[(\mathbb{I}-\mathbb{P}^0)(\mathbb{P}^1u^h)\right]_x\,dx+\frac{1}{5}\int_{\mathcal{T}_h}\overline{\overline{q^h}}\left[(\mathbb{I}-\mathbb{P}^2)(\mathbb{P}^1u^h)\right]_x\,dx$$

$$F=\sum_{h\in(\mathcal{T}_h)}\int_{\mathcal{T}_h}\frac{\partial}{\partial t}(u^h-\mathbb{P}^1u^h)(\mathbb{P}^1u^h)\,dx+\left[\mathcal{F}(\frac{(u^h)^2}{2}-\nu\overline{q^h}-\frac{(\mathbb{P}^1u^h)^2}{2}+\nu\widehat{\overline{q^h}})\,\mathbb{P}^1u^h\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}$$

$$-\int_{\mathcal{T}_h}(\frac{(u^h)^2}{2}-\nu\overline{q^h}-\frac{(\mathbb{P}^1u^h)^2}{2}+\nu\widehat{\overline{q^h}})\,(\mathbb{P}^1u^h)_x\,dx$$

where the auxiliary LDG variables are $\forall(h,i)\in(\mathcal{T}_h)\times[0,...,n]$:

$$\frac{\Delta^h}{2}\cdot\overline{q_i^h}=\left[\mathcal{F}(u^h)L_h^i\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}-\int_{\mathcal{T}_h}u^h\frac{d}{dx}(L_h^i)\,dx$$

$$\frac{\Delta^h}{2}\cdot\widehat{\overline{q_i^h}}=\left[\mathcal{F}(\mathbb{P}^1u^h)L_h^i\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}-\int_{\mathcal{T}_h}(\mathbb{P}^1u^h)\frac{d}{dx}(L_h^i)\,dx$$

$$\frac{\Delta^h}{2}\cdot\widetilde{\overline{q_i^h}}=\left[\mathcal{F}(\mathbb{P}^3u^h)L_h^i\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}-\int_{\mathcal{T}_h}(\mathbb{P}^3u^h)\frac{d}{dx}(L_h^i)\,dx$$

$$\frac{\Delta^h}{2}\cdot\overline{\overline{q_i^h}}=\left[\mathcal{F}((\mathbb{I}-\mathbb{P}^2)u^h)L_h^i\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}-\int_{\mathcal{T}_h}((\mathbb{I}-\mathbb{P}^2)u^h)\frac{d}{dx}(L_h^i)\,dx$$

$$\frac{\Delta^h}{2}\cdot\widehat{\overline{\overline{q_i^h}}}=\left[\mathcal{F}((\mathbb{I}-\mathbb{P}^2)\mathbb{P}^1u^h)L_h^i\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}}-\int_{\mathcal{T}_h}((\mathbb{I}-\mathbb{P}^2)\mathbb{P}^1u^h)\frac{d}{dx}(L_h^i)\,dx$$

$$\frac{\Delta^h}{2} \cdot \widetilde{\widetilde{q_i^h}} = \left[\mathcal{F}((\mathbb{I} - \mathbb{P}^2)\mathbb{P}^3 u^h) L_h^i\right]_{h-\frac{1}{2}}^{h+\frac{1}{2}} - \int_{\mathcal{T}_h} ((\mathbb{I} - \mathbb{P}^2)\mathbb{P}^3 u^h) \frac{d}{dx}(L_h^i)\, dx$$

where $\mathcal{F}(x)$ means that an appropriate flux of the expression x must be used, instead of just x.

Bibliography

[1] Cockburn B and C.-W. Shu. Runge-kutta discontinuous galerkin methods for convection-dominated problems. *Journal of Scientific Computing, Vol. 16, No. 3*, 2001.

[2] B. Cockburn and C.-W. Shu. The local discontinuous galerkin method for time dependant convection-diffusion systems. *SIAM Journal of Numerical Analysis, Vol. 35, pp.2440-2463*, 1998.

[3] Bassi F. and S. Rebay. High-order accurate discontinuous finite element method for the numerical solution of the compressible navier-stokes equations. *Journal of Computational Physics, Vol. 138, pp. 251-285*, 1997.

[4] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of fluids, 3(7), pp.1760-1765*, 1991.

[5] S. Goshal, T. S. Lund, P. Moin, and K. Akselvoll. A dynamic localization model for large-eddy simulation of turbulent flows. *Journal of Fluids Mechanics, vol. 186, pp. 229-255*, 1995.

[6] G. Jiang and C.-W. Shu. On a cell entropy inequality for discontinuous galerkin methods. *Methematics of Computation, Vol. 62, Number 162, pp. 531-538*, 1994.

[7] D. K. Lily. A proposed modification of the germano subgrid-scale closure method. *Physics of fluids A, 4(3), pp. 633-635*, 1992.

[8] Assad A. Oberai. Evaluating parameters in a numerical method using the variational germano identity. *Report No. 03-004, Department of Aerospace and Mechanical Engineering, Boston University*, 2003.

[9] Assad A. Oberai and John Wanderer. A dynamic multiscale viscosity method for the spectral approximation of conservation laws. *Report. Department of Aerospace and Mechanical Engineering, Boston University*, 2004.

[10] S. Osher. Riemann solvers, the entropy condition, and difference approximation. *SIAM J. Numer. Anal. 21, pp. 217-235*, 1984.

[11] J. Qiu and C.-W. Shu. Runge-kutta discontinuous galerkin method using weno limiters. *submitted to SIAM Journal on Scientific Computing.*

[12] J. Qiu and C.-W. Shu. Hermite weno schemes and their application as limiters for runge-kutta discontinuous galerkin method: one-dimensional case. *Journal of Computational Physics 193, pp.115-135*, 2003.

[13] M. Serrano. A discontinuous galerkin hweno scheme for hyperbolic equations. *Massachusetts Institute of Technology , Department of Aerospace Engineering, thesis*, 2004.