Application of a High Order Finite Volume Scheme on Unstructured Grids to Fluid Dynamics and Aerothermochemistry

F. Haider, N. Bertier, B. Courbet, F. Vuillot, J.P. Croisille

Application of a High Order Finite Volume Scheme on Unstructured Grids to Fluid Dynamics and Aerothermochemistry

Florian Haider* Nicolas Bertier* Bernard Courbet* François Vuillot*
Jean-Pierre Croisille†

December 16, 2014

Abstract

This paper presents a high order finite volume scheme built on a new k-exact reconstruction algorithm for general unstructured grids. The algorithm is capable of computing accurate polynomial approximations using data from adjacent cells only, overcoming a major obstacle to extend classical finite volume schemes beyond 2^{nd} order spatial accuracy. Moreover, it can easily be integrated in a cell or vertex centered finite volume method that uses the cell averages as the only unknown per grid cell and physical quantity. It is therefore particularly suited to upgrade existing 2^{nd} order finite volume solvers to higher accuracy without huge efforts in software development. Three numerical test cases demonstrate the viability of the scheme in practical applications. This work was announced in [13, 14].

Keywords: Finite Volume Method - High Order Accuracy - k-exact Reconstruction

1 Introduction

In this paper we investigate a class of high order finite volume approximation for fluid dynamics on general grids. Our method follows along the lines of the MUSCL method [22, 9]. First attempts to extend the MUSCL approach to unstructured grids go back 25 years and more, see [5] and the references therein. These pioneering efforts have led to 2^{nd} order accurate finite volume schemes with highly successful applications to the Euler and Navier Stokes equations. The possibility to handle complex geometries and the easy modeling of boundary conditions are among the benefits of these schemes. Two specific aspects are:

*Onera - The French Aerospace Lab, F-92322 Châtillon, France
†Inst. Elie Cartan de Lorraine, Univ. Lorraine, UMR CNRS 7502, Ile du Saulcy, 57045 Metz, France

1. In each cell, 2^{nd} order interpolation (piecewise linear) requires only data from direct cell neighbors. This is often referred to as interpolation on *compact stencils*.

2. The numerical solutions are completely determined by one single degree of freedom per physical quantity and grid cell. This unknown is usually represented by the cell average.

However, it is now widely accepted that a 2^{nd} order accurate scheme is insufficient for a growing number of applications. Unfortunately, when trying to extend a 2^{nd} order scheme to 3^{rd} order and beyond, a well-known difficulty appears: high order spatial approximation in cell $\mathcal{T}$ requires data from cells far beyond cells adjacent to $\mathcal{T}$. On a general grid with indirect data access, this fact severely impacts computational efficiency and performance scaling.

A seminal idea to increase spatial accuracy while keeping a compact stencil is to supplement the cell average with new degrees of freedom. This approach has triggered the development of a number of successful high order discretization methods for unstructured grids such as the Discontinuous Galerkin [8, 7], the Spectral Finite Volume [17, 16] and the Residual Distribution scheme [1, 20]. All these schemes have many desirable properties yet their implementation in an existing general purpose finite volume code for practical applications might be a difficult task. This is particularly true in the case of legacy softwares, such as CEDRE [1] , which handles many complex models. To name a few, this includes modeling of multi-species and multi-phase flow, combustion, turbulence, and the geometric representation of moving and overlapping grids. Supplementing cell averages with additional degrees of freedom obviously requires an important restructuring of the code. Therefore keeping the cell average as the sole unknown per cell and physical quantity remains an attractive option.

This raises again the question how to efficiently calculate piecewise high order reconstruction based on data located in adjacent cells only. This question is central to the k-exact approach for finite volumes. The main topic of this paper is an original approximation algorithm [2] that is able to carry out this difficult task. It has the crucial advantage that its implementation in an existing finite volume scheme modifies only the interpolation step. The other ingredients need just minimal upgrades to guarantee the global accuracy of the scheme, such as high order quadratures for the fluxes and high order time stepping methods. In particular, the application of this high order spatial discretization to a conservation law results in the same number of semi-discrete equations as for a 2^{nd} order finite volume scheme. This simplifies the use of implicit time stepping methods and the integration of complex physical models in the numerical scheme.

[1] The present study was carried out within the project CEDRE. CEDRE is the reference software for aerothermochemistry at ONERA, http://www.onera.fr/cedre.

[2] This algorithm was first described in a private communication by Pierre Brenner, AIRBUS Defence and Space SAS, 51-61 Route de Verneuil 78130 Les Mureaux, France.

Our second purpose is to show numerical evidence that a drastic higher order accuracy is effectively reached in complex situations with the new scheme. Specifically we numerically analyze the grid convergence of our scheme on two test cases having significance for applications involving sound or vortex propagation. In the first test-case, the numerical propagation of an acoustic wave in a three-dimensional channel is shown. The second test compares the numerical propagation of an inviscid isothermal vortex with the exact one. The domain is a periodic box and the challenge for this nonlinear case is to preserve the shape of the vortex after a large number of periods. Finally we illustrate the capability of the scheme to handle real world applications with a laminar flame test case. For this case, additional treatments such as slope limiting and positivity preserving are used. These two specific questions are not studied in detail here and will be the topic of a forthcoming paper.

The outline is as follows. Sec. 2 gives a brief summary of the physical modeling background in which the scheme is implemented so far. In Sec. 3 we recall the basic setting of finite volume methods based on k-exact reconstruction. Sec. 4 presents the new compact k-exact reconstruction algorithm. Sec. 5 presents the numerical test cases and finally conclusion and several perspectives are drawn in Sec. 6.

2 Modeling in Aerothermochemistry

Although the scheme presented here is suitable for general conservation laws, its intended use is mainly modeling in aerothermochemistry, the study of fluids taking into account the effect of motion, heat fluxes and chemical reactions. Here we restrict ourselves to fluid mixtures in *mechanical and thermal equilibrium*, so that a single valued pressure and temperature field is defined everywhere in the flow. The equations that are considered are the compressible Euler (or Navier-Stokes) equations for multi-species flows with a general equation of state. These equations are closely connected with turbulence modeling, chemical reactions, etc. This section gives a brief summary.

2.1 Conservation Laws for Multi-species Flow

The general form of a convection-diffusion-reaction system to be solved on a domain $\Omega \subset \mathbb{R}^3$ for a time interval $[t_0, t_1]$ is

$$\partial_t \boldsymbol{q} + \operatorname{div}(\boldsymbol{f} + \boldsymbol{\varphi}) = \boldsymbol{\varsigma}. \tag{2.1}$$

The unknown $\boldsymbol{q}$ in (2.1) is a function $\Omega \times [t_0, t_1] \ni (\boldsymbol{x}, t) \mapsto \boldsymbol{q}(\boldsymbol{x}, t) \in \mathbb{R}^m$. In the present setting, $\boldsymbol{q}$ comprises four types of variables:

- $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_{n_{\mathrm{sp}}}]^T$ are the densities of n_{sp} species and the total density is $\rho = \rho_1 + \rho_2 + \cdots + \rho_{n_{\mathrm{sp}}}$.
- $\rho \boldsymbol{v} = \rho\,[v_1, v_2, v_3]^T$ is the momentum vector.

- ρE is the total energy per volume unit. Let ε_{int} denote the internal energy of the mixture, then $E = \varepsilon_{\text{int}} + \frac{1}{2}|\boldsymbol{v}|^2$.
- $\rho \boldsymbol{z}$ with $\boldsymbol{z} = [z_1, z_2, ...z_{n_{\text{sc}}}]^T$ represents n_{sc} convected scalar quantities. The z_i are either passive scalars or used in the modeling of turbulence, combustion, etc.

The total number of unknowns is $m = n_{\text{sp}} + n_{\text{sc}} + 4$ and the vector $\boldsymbol{q}$ is expressed as

$$\boldsymbol{q} = [\boldsymbol{\rho}, \rho\boldsymbol{v}, \rho E, \boldsymbol{z}]^T \ . \tag{2.2}$$

Let p and T denote the pressure and the temperature, respectively, and define the vector of *mass fractions* $y_k = \frac{\rho_k}{\rho}$ by

$$\boldsymbol{y} \triangleq [y_1, \ldots, y_{n_{\text{sp}}}]^T \ . \tag{2.3}$$

By definition $\boldsymbol{\rho} = \rho\boldsymbol{y}$ and $\sum_{k=1}^{n_{\text{sp}}} y_k = 1$. The quantities $\boldsymbol{f}$, $\boldsymbol{\varphi}$ and $\boldsymbol{\varsigma}$ in (2.1) represent the convective fluxes, the viscous fluxes and the source terms, respectively. The convective fluxes $\boldsymbol{f}$ are the fluxes of the Euler equations with components $\boldsymbol{f} = \left[\boldsymbol{f}_\rho, \boldsymbol{f}_v, \boldsymbol{f}_E, \boldsymbol{f}_z\right]$ defined by

$$\begin{cases} \quad \boldsymbol{f}_{\rho,i} = v_i \left[\rho_1, \ldots, \rho_{n_{\text{sp}}}\right]^T \\ (\boldsymbol{f}_v)_{i,j} = \rho v_i v_j + p\delta_{ij} \ , \ 1 \le j \le 3 \\ \quad f_{E,i} = (\rho E + p)\, v_i \\ \quad \boldsymbol{f}_{z,i} = \rho v_i \left[z_1, \ldots, z_{n_{\text{sc}}}\right]^T . \end{cases} , 1 \le i \le 3 \, . \tag{2.4}$$

2.2 Specific Physical Modeling

Equations of State The system (2.1) with the fluxes (2.4) cannot be solved without additional closure conditions. These include at least a *caloric equation of state*

$$\varepsilon_{\text{int}} = \varepsilon\,(p, \boldsymbol{\rho}) \ . \tag{2.5}$$

The case $\boldsymbol{\varphi} \neq 0$ or $\boldsymbol{\sigma} \neq 0$ usually requires the specification of a *thermal equation of state*

$$T = T\,(p, \boldsymbol{\rho}) \tag{2.6}$$

or a *pressure law*

$$p = p\,(\boldsymbol{\rho}, T) \ . \tag{2.7}$$

These equations of state establish a relation between the quantities $\boldsymbol{q}$ and the so-called *physical variables*, which are defined by

$$\boldsymbol{u} = [\boldsymbol{y}, p, T, \boldsymbol{v}, \boldsymbol{z}]^T \ . \tag{2.8}$$

Note that (2.8) represents a vector with $m + 1 = n_{\text{sp}} + n_{\text{sc}} + 5$ components (instead of m components in (2.2) due to the constraint on the mass fractions in (2.3). For certain applications, it can be more convenient to use $\boldsymbol{u}$ than $\boldsymbol{q}$.[3]

[3] We assume in this paper that $\boldsymbol{q} \mapsto \boldsymbol{u}(\boldsymbol{q})$is one-to-one.

Diffusive Fluxes Depending on the modeling, the diffusive fluxes $\boldsymbol{\varphi}$ include the following terms:

- The tensor of viscous strains $\boldsymbol{\varphi}_v$ is given by the *Stokes law*

$$\boldsymbol{\varphi}_v = 2\mu_v(\boldsymbol{u})\, S(\boldsymbol{u}) + \left(\kappa_v(\boldsymbol{u}) - \frac{2}{3}\mu_v(\boldsymbol{u})\right)(\operatorname{div}\boldsymbol{v})\,\boldsymbol{I}, \tag{2.9}$$

  where the strain rate tensor is given by $S(\boldsymbol{u}) = \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{v} + \boldsymbol{\nabla}\boldsymbol{v}^T\right)$.

- Standard modeling of heat conduction is *given by Fourier's law*

$$\boldsymbol{\varphi}_E = -\lambda_H(\boldsymbol{u})\,\boldsymbol{\nabla}T\,. \tag{2.10}$$

- A classical modeling of species diffusion is *given by Fick's law* with a diagonal tensor $\boldsymbol{D}$

$$\boldsymbol{\varphi}_{\rho,i} = -\boldsymbol{D}(\boldsymbol{u})\,\boldsymbol{\nabla}y_i\,. \tag{2.11}$$

Turbulence Modeling Turbulence modeling is based on filtering of the conservative variables $\boldsymbol{q}$ at a specified scale. The unresolved scales of the turbulent flow are classically represented by additional terms in the diffusive fluxes $\boldsymbol{\varphi}$. In particular RANS or LES models, the coefficients in (2.9), (2.10) and (2.11) are deduced from scalars such as the turbulent kinetic energy k and its rate of dissipation ϵ. These scalars are collected as components of the variable $\boldsymbol{z} = [k, \epsilon, ...]$ and they are therefore part of the physical variable $\boldsymbol{u}$. Therefore the notational formalism remains the same as in the case without turbulence model. In particular LES [4] modeling is taken in account as follows:

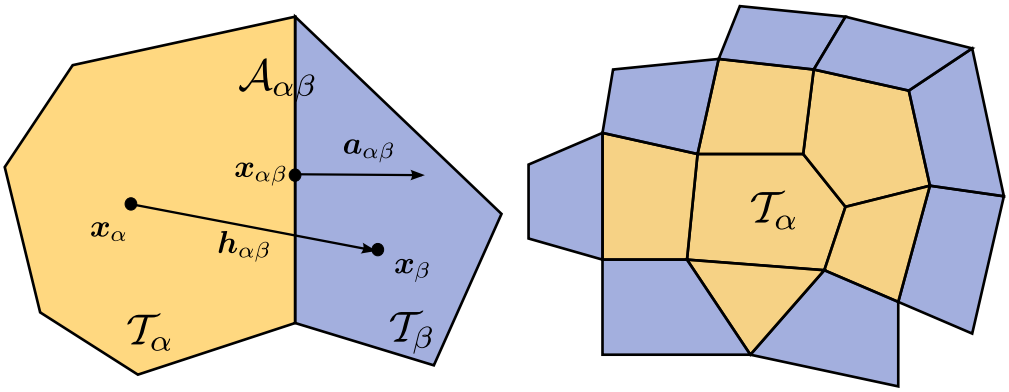
- In laminar calculations, (2.9), (2.10) and (2.11) include only the physical transport coefficients. This hypothesis is coined as the MILES [5] approach. In this case numerical dissipation is assumed to handle unresolved turbulent scales.

- The diffusion coefficients in (2.9), (2.10) and (2.11) can also include a specific algebraic model. For instance in the Smagorinsky model , the laminar viscosity μ is replaced by $\mu+\mu_{sgs}$, with $\mu_{sgs} = \rho\,(C_s h)^2\,|\boldsymbol{S}(\boldsymbol{u})|$, where h is the local mesh size and $C_s \sim 0.17$ [21].

Source Terms The source term $\boldsymbol{\varsigma}$ in (2.1) represents body forces and chemical phenomena:

- Body forces appear as a source term in the momentum equation. The corresponding work is a source term in the energy equation.

[4] Large Eddy Simulation
[5] Monotonically Integrated Large Eddy Simulation



(a) Two cells $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$ sharing a face $\mathcal{A}_{\alpha\beta}$.

(b) Neighborhoods of a cell $\mathcal{T}_\alpha$: cells in $\mathbb{V}_\alpha^{(1)}$ in light color and cells in $\mathbb{V}_\alpha^{(2)} \setminus \mathbb{V}_\alpha^{(1)}$ in dark color.

Figure 3.1: The geometry of two-dimensional polyhedral grids.

- Chemical reactions induce source terms in the density equation for each species. Whenever R reactions indexed by $r = 1, \cdots, R_{\text{ch}}$ are involved, the source term for the density ρ_j is expressed as a sum in the form:

$$\varsigma_j = \mathcal{M}_j \sum_{r=1}^{R_{\text{ch}}} \nu_{j,r} W_r. \tag{2.12}$$

  In (2.12), $\mathcal{M}_j$ is the molar mass of species j and $\nu_{j,r}$ is the algebraic stoichiometric coefficient of species j in reaction r. For an Arrhenius reaction, W_r is a function of T and $\frac{\rho_j}{\mathcal{M}_j}$.

3 High Order Finite Volume Schemes

In this section we briefly review the setting of a finite volume scheme based on k-exact reconstruction on general unstructured grids.

3.1 Geometric Notation for Unstructured Meshes

A general unstructured grid of the domain $\Omega \subset \mathbb{R}^d$ consists of N general polyhedra $\mathcal{T}_\alpha$ with an arbitrary number of faces $\Omega = \bigcup_{\alpha=1}^N \mathcal{T}_\alpha$. The following notation is depicted in Fig. 3.1a. Two cells $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$ are called *adjacent* if they share a face denoted by $\mathcal{A}_{\alpha\beta} = \mathcal{T}_\alpha \cap \mathcal{T}_\beta$. Note that no connectivity restriction is imposed on the mesh. The volume of a cell $\mathcal{T}_\alpha$ and the area of a face $\mathcal{A}_{\alpha\beta}$ are denoted by $|\mathcal{T}_\alpha|$ and $|\mathcal{A}_{\alpha\beta}|$, respectively. The barycenters of $\mathcal{T}_\alpha$

and $\mathcal{A}_{\alpha\beta}$ are $\boldsymbol{x}_\alpha$ and $\boldsymbol{x}_{\alpha\beta}$, respectively. It is useful to define the vector

$$\boldsymbol{h}_{\alpha\beta} \triangleq \boldsymbol{x}_\beta - \boldsymbol{x}_\alpha \,. \tag{3.1}$$

Calculating a numerical flux in a finite volume scheme require a precise definition of the geometry of the interface $\mathcal{A}_{\alpha\beta}$, which is in general not planar. In our case, $\mathcal{A}_{\alpha\beta}$ is defined as the disjoint union of the triangles formed by two consecutive points of the contour $\partial\mathcal{A}_{\alpha\beta}$ and a point $O_{\alpha\beta}$ of the face. A convenient choice for $O_{\alpha\beta}$ is the barycenter $\boldsymbol{x}_{\alpha\beta}$ of $\mathcal{A}_{\alpha\beta}$. The surface vector $\boldsymbol{a}_{\alpha\beta}$ is deduced from the knowledge of $\mathcal{A}_{\alpha\beta}$. The normal unit vector of the face $\mathcal{A}_{\alpha\beta}$, with orientation from $\mathcal{T}_\alpha$ to $\mathcal{T}_\beta$, is denoted by $\boldsymbol{\nu}_{\alpha\beta}$. It is deduced from the surface vector $\boldsymbol{a}_{\alpha\beta}$ by $\boldsymbol{\nu}_{\alpha\beta} = \boldsymbol{a}_{\alpha\beta}/|\mathcal{A}_{\alpha\beta}|$.

The following definitions of neighborhoods of a cell $\mathcal{T}_\alpha$ are used in the design of the algorithm hereafter, see Fig. 3.1b:

- The 1^{st} *neighborhood* of a cell $\mathcal{T}_\alpha$ is the set

$$\mathbb{V}_\alpha^{(1)} \triangleq \{\beta \,|\mathcal{T}_\beta \text{ is adjacent to } \mathcal{T}_\alpha\} \cup \{\alpha\} \,. \tag{3.2}$$

- The n^{th} *neighborhood* is defined recursively by

$$\mathbb{V}_\alpha^{(n)} \triangleq \bigcup_{\gamma\in\mathbb{V}_\alpha^{(n-1)}} \mathbb{V}_\gamma^{(1)} \,. \tag{3.3}$$

Each neighborhood $\mathbb{V}_\alpha = \cup_{k=1}^m \mathcal{T}_{\beta_k}$ of a cell $\mathcal{T}_\alpha$ is associated with a matrix H_α defined by

$$H_\alpha^T = \operatorname{col}(\boldsymbol{h}_{\alpha\beta_1}, \boldsymbol{h}_{\alpha\beta_2}, \ldots, \boldsymbol{h}_{\alpha\beta_m}) \in \mathbb{M}_{3,m}(\mathbb{R}) \,. \tag{3.4}$$

The matrix (3.4) is useful for the definition of 2^{nd} order interpolation in Sec. 4 and 7. For each cell $\mathcal{T}_\alpha$, the tensor $\boldsymbol{x}_\alpha^{(k)}$ is the moment defined by its components

$$x^{(k)}_{\alpha;i_1,\ldots,i_k} \triangleq \frac{1}{|\mathcal{T}_\alpha|}\int_{\mathcal{T}_\alpha} (x_{i_1} - x_{\alpha;i_1})\cdots(x_{i_k} - x_{\alpha;i_k})\, dx \,. \tag{3.5}$$

For each pair of cells $\mathcal{T}_\alpha$ and $\mathcal{T}_\beta$, the tensor $\boldsymbol{z}_{\alpha\beta}^{(k)}$ is defined by its components

$$z^{(k)}_{\alpha\beta;i_1,\ldots,i_k} \triangleq \frac{1}{|\mathcal{T}_\beta|}\int_{\mathcal{T}_\beta} (x_{i_1} - x_{\alpha;i_1})\cdots(x_{i_k} - x_{\alpha;i_k})\, dx \,. \tag{3.6}$$

Furthermore the k^{th} power $\boldsymbol{x}^k$ of a vector $\boldsymbol{x} \in \mathbb{R}^3$ is defined as the tensor of order k with components:

$$\left(\boldsymbol{x}^k\right)_{i_1,i_2,\ldots,i_k} \triangleq x_{i_1}x_{i_2}\cdots x_{i_k} \;,\; 1 \le i_1, i_2, \ldots, i_k \le 3 \,. \tag{3.7}$$

For tensors $\boldsymbol{a}$ and $\boldsymbol{b}$ of rank k with components $a_{i_1,\ldots,i_k}$ and $b_{i_1,\ldots,i_k}$, respectively, we use the dot product $\bullet$ defined by

$$\boldsymbol{a} \bullet \boldsymbol{b} = \sum_{i_1,\ldots,i_k=1}^{d} a_{i_1,\ldots,i_k} b_{i_1,\ldots,i_k} \,. \tag{3.8}$$

3.2 Basic design of Finite Volume Schemes

In this section we review the setup of the semi-discrete finite volume method with the notation of Sec. 3.1. Integration of the conservation law (2.1) over a given non moving cell $\mathcal{T}_\alpha$ gives

$$\frac{d}{dt}\left(\frac{1}{|\mathcal{T}_\alpha|}\int_{\mathcal{T}_\alpha}\boldsymbol{q}\,d\boldsymbol{x}\right)=-\frac{1}{|\mathcal{T}_\alpha|}\sum_{\beta\in\mathbb{V}_\alpha^{(1)}}\int_{\mathcal{A}_{\alpha\beta}}\boldsymbol{n}_{\alpha\beta}\cdot(\boldsymbol{f}+\boldsymbol{\varphi})\;d\boldsymbol{s}+\frac{1}{|\mathcal{T}_\alpha|}\int_{\mathcal{T}_\alpha}\boldsymbol{\varsigma}\,d\boldsymbol{x}\,. \tag{3.9}$$

Note that identity (3.9) can be extended to the *Arbitrary Eulerian Lagrangian* (ALE) context of moving and deformable meshes.

The identity (3.9) infers a semi-discrete time-dependent system that approximates (2.1) as follows. First, the cell averages on the left-hand side of (3.9) are defined by

$$\overline{\boldsymbol{q}}_\alpha(t)\triangleq\frac{1}{|\mathcal{T}_\alpha|}\int_{\mathcal{T}_\alpha}\boldsymbol{q}\,(\boldsymbol{x},t)\;d\boldsymbol{x} \tag{3.10}$$

The $\overline{\boldsymbol{q}}_\alpha(t)$ are collected into a vector $\overline{\mathfrak{q}}\,(t)=\left[\overline{\boldsymbol{q}}_1\,(t),\ldots,\overline{\boldsymbol{q}}_N\,(t)\right]^T$ that becomes the principal unknown. [6].

Consider now the quantities $\boldsymbol{f}$, $\boldsymbol{\varphi}$ and $\boldsymbol{\sigma}$ in the right hand side of (3.9). They depend on the continuous unknown $\boldsymbol{q}\,(\boldsymbol{x},t)$ and its derivatives. They have to be approximated with quantities depending on the vector $\overline{\mathfrak{q}}\,(t)$. For the applications presented here, the treatment of the convective fluxes $\boldsymbol{f}$ is the crucial point and in the rest of Sec. 3, we set $\boldsymbol{\varphi}=0$ and $\boldsymbol{\sigma}=0$ for the sake of simplicity. In the simplest setting, the normal convective flux in (3.9) is approximated using a so called *numerical flux function* :

$$f_{\alpha\beta}^{\text{num}}\left(\overline{\boldsymbol{q}}_\alpha\,(t),\overline{\boldsymbol{q}}_\beta\,(t)\right)\approx\boldsymbol{n}_{\alpha\beta}\cdot\boldsymbol{f}\,(\boldsymbol{q}\,(\boldsymbol{x},t))\;. \tag{3.11}$$

The numerical flux function in (3.11) is required to be *consistent*, *conservative*, *monotone* and *Lipschitz continuous* [3]. Insertion of (3.11) in (3.9) gives the system

$$\frac{d\overline{\boldsymbol{q}}_\alpha\,(t)}{dt}=-\frac{1}{|\mathcal{T}_\alpha|}\sum_{\beta\in\mathbb{V}_\alpha^1}\int_{\mathcal{A}_{\alpha\beta}}f_{\alpha\beta}^{\text{num}}\left(\overline{\boldsymbol{q}}_\alpha\,(t),\overline{\boldsymbol{q}}_\beta\,(t)\right)\;d\boldsymbol{s}\,. \tag{3.12}$$

The dynamical system (3.12) represents an approximation to (3.9), which is at most first order accurate [3]. It is widely accepted that this poor spatial accuracy makes it unsuitable for most practical applications. The next section describes a classical remedy to this problem.

[6] Strictly speaking $\overline{\boldsymbol{q}}_\alpha(t)$ denotes the exact average over cell $\mathcal{T}_\alpha$, whereas the components of $\overline{\mathfrak{q}}\,(t)$ are approximate values to be calculated.

3.3 Increasing Accuracy by k-exact Reconstruction

This section recalls how the accuracy of the approximation (3.12) can be enhanced by the well known method of $k-$exact reconstructions. For simplicity we will consider all functions at fixed time t_0 and we drop the dependency on t. The main idea for restoring accuracy in (3.12) is to replace the crude approximations $\overline{\boldsymbol{q}}_\alpha$ and $\overline{\boldsymbol{q}_\beta}$ in the right-hand-side of (3.12) with more accurate values at the face $\mathcal{A}_{\alpha\beta}$. A simple way to do this is to use the cell average data $\overline{\mathfrak{q}} = [\overline{\boldsymbol{q}}_1, \ldots, \overline{\boldsymbol{q}}_N]^T$ to reconstruct in each cell $\mathcal{T}_\alpha$ a polynomial approximant $\boldsymbol{w}_\alpha$ of degree k. The reconstruction map $\overline{\mathfrak{q}} \mapsto \boldsymbol{w}_\alpha$ must be linear and *local*, i.e. $\boldsymbol{w}_\alpha$ should only depend on data in some close neighborhood of $\mathcal{T}_\alpha$. This particular neighborhood is called the *reconstruction stencil* of $\mathcal{T}_\alpha$. Two essential constraints which must be satisfied by the reconstruction procedure are:

1. *conservation*: the average of the polynomial $\boldsymbol{w}_\alpha$ over the cell $\mathcal{T}_\alpha$ should be the exact cell average $\overline{\boldsymbol{q}}_\alpha$.

2. *k-exactness:* whenever the cell averages on a neighborhood of the cell $\mathcal{T}_\alpha$ are those of a polynomial p of degree k then the reconstruction should recover p, i.e. $\boldsymbol{w}_\alpha = p$.

In the following we use the notation $\boldsymbol{w}_\alpha\left[\overline{\mathfrak{q}}\right](\boldsymbol{x})$ to mention that the coefficients of the polynomial $\boldsymbol{w}_\alpha\left[\overline{\mathfrak{q}}\right](\boldsymbol{x})$ depend on the data $\overline{\mathfrak{q}}$ and that $\boldsymbol{w}_\alpha\left[\overline{\mathfrak{q}}\right](\boldsymbol{x})$ represents a function of $\boldsymbol{x} \in \mathcal{T}_\alpha$. The first order approximation (3.11) is now replaced by

$$f_{\alpha\beta}^{\text{num}}\left(\boldsymbol{w}_\alpha\left[\overline{\mathfrak{q}}\left(t\right)\right](\boldsymbol{x}), \boldsymbol{w}_\beta\left[\overline{\mathfrak{q}}\left(t\right)\right](\boldsymbol{x})\right) \approx \boldsymbol{n}_{\alpha\beta} \cdot \boldsymbol{f}\left(\boldsymbol{q}\left(\boldsymbol{x}, t\right)\right) . \tag{3.13}$$

It can be shown that the two properties above imply the consistency identity

$$\sum_{\beta \in \mathbb{V}_\alpha^1} \int_{\mathcal{A}_{\alpha\beta}} f_{\alpha\beta}^{\text{num}}\left(\boldsymbol{w}_\alpha\left[\overline{\mathfrak{q}}\right](\boldsymbol{x}), \boldsymbol{w}_\beta\left[\overline{\mathfrak{q}}\right](\boldsymbol{x})\right) \, ds\left(\boldsymbol{x}\right)$$

$$= \sum_{\beta \in \mathbb{V}_\alpha^1} \int_{\mathcal{A}_{\alpha\beta}} \left(\boldsymbol{n}_{\alpha\beta} \cdot \boldsymbol{f}\left(\boldsymbol{q}\left(\boldsymbol{x}\right)\right)\right) \, ds\left(\boldsymbol{x}\right) + O(h^{k+1}) \tag{3.14}$$

Suitable quadratures rules to calculate the integral terms over the interface $\mathcal{A}_{\alpha\beta}$ in the left-hand side of (3.14) are required. These formulas are defined by appropriate weight/node couples $(\omega_q, \boldsymbol{x}_{\alpha\beta;q})$. These rules must be at least of order k to ensure that the $O(h^{k+1})$ accuracy in (3.14) is preserved.

The notion of k-exact reconstruction on unstructured grids already appears in [4] where it is implemented using a standard least square procedure. Its main advantage is its simplicity. But an important drawback is a lack of compactness. Specifically in order to get k-exact reconstruction for $k = 3$ on a three-dimensional grid, the reconstruction stencil of each cell must contain at least 20 cells. In practice however, due to stability constraints, the reconstruction stencil has to be much larger on, e.g., tetrahedral grids. We refer the reader to [15] where a stability analysis is carried out in the case of piecewise linear reconstruction

($k = 1$ in the present notation). As a consequence, due to memory access, practically implementing a k-exact reconstruction on a parallel computer can become severely inefficient on a general grid. The next section describes a novel algorithm to overcome this problem.

4 k-Exact Compact Reconstruction on General Grids

As observed in the preceding section, the practical use of the notion of k-exactness requires a precise algorithmic approach to the lack of stencil compactness of the reconstruction. In this section we present a new reconstruction method to solve this problem.

4.1 Principle of k-Exact Reconstruction

We are specifically interested in 4th order accurate methods and so we need to consider reconstructions of order $k = 3$ (cubic reconstructions).

Consider a smooth scalar function $u : \mathbb{R}^3 \to \mathbb{R}$. The problem is to calculate in cell $\mathcal{T}_\alpha$ a polynomial approximant w_α[7] to u depending on the cell averages $\overline{\mathrm{u}} = [\overline{u}_1, \ldots, \overline{u}_N]^T$ only (see (3.10)). The approximant is denoted by $x \in \mathcal{T}_\alpha \mapsto w_\alpha[\overline{\mathrm{u}}](\boldsymbol{x})$. It can be expressed in terms of $\boldsymbol{\sigma}_\alpha[\overline{\mathrm{u}}]$, $\boldsymbol{\theta}_\alpha[\overline{\mathrm{u}}]$ and $\boldsymbol{\psi}_\alpha[\overline{\mathrm{u}}]$ which are the 1st, 2nd and 3rd derivative of $w_\alpha[\overline{\mathrm{u}}](\boldsymbol{x})$ at $\boldsymbol{x}_\alpha$, respectively, as

$$w_\alpha[\overline{\mathrm{u}}](\boldsymbol{x}) = w_\alpha[\overline{\mathrm{u}}](\boldsymbol{x}_\alpha) + \boldsymbol{\sigma}_\alpha[\overline{\mathrm{u}}] \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha) + \frac{1}{2}\boldsymbol{\theta}_\alpha[\overline{\mathrm{u}}] \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha)^2 + \frac{1}{6}\boldsymbol{\psi}_\alpha[\overline{\mathrm{u}}] \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha)^3 \ . \quad (4.1)$$

This representation of the approximate function $w_\alpha[\overline{\mathrm{u}}](\boldsymbol{x})$ has to be matched with the Taylor expansion of the function $u(x)$ at $\boldsymbol{x}_\alpha$, which is :

$$u(\boldsymbol{x}) = u(\boldsymbol{x}_\alpha) + D^{(1)}u\Big|_{\boldsymbol{x}_\alpha} \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha) + \frac{1}{2} D^{(2)}u\Big|_{\boldsymbol{x}_\alpha} \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha)^2 + \frac{1}{6} D^{(3)}u\Big|_{\boldsymbol{x}_\alpha} \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha)^3 + O(h^4) \quad (4.2)$$

First, according to the conservation condition in Sec. 3.3, the approximant (4.1) must have the same cell average over $\mathcal{T}_\alpha$ as u, i.e. $\overline{u}_\alpha$. For this to hold, the value of $w_\alpha[\overline{\mathrm{u}}](\boldsymbol{x}_\alpha)$ in (4.1) must be defined as (see (3.5) and (3.8)) :

$$w_\alpha[\overline{\mathrm{u}}](\boldsymbol{x}_\alpha) = \overline{u}_\alpha - \frac{1}{2}\boldsymbol{\theta}_\alpha \bullet \boldsymbol{x}_\alpha^{(2)} - \frac{1}{6}\boldsymbol{\psi}_\alpha \bullet \boldsymbol{x}_\alpha^{(3)} \ . \quad (4.3)$$

Second, according to the $k-$exactness condition, whenever the cell averages on a neighborhood of the cell $\mathcal{T}_\alpha$ are those of a polynomial p of degree $k = 3$, the reconstruction must recover the 1st, 2nd and 3rd derivatives of p:

$$\boldsymbol{\sigma}_\alpha = D^{(1)}p\Big|_{\boldsymbol{x}_\alpha} \ , \ \boldsymbol{\theta}_\alpha = D^{(2)}p\Big|_{\boldsymbol{x}_\alpha} \ , \ \boldsymbol{\psi}_\alpha = D^{(3)}p\Big|_{\boldsymbol{x}_\alpha} \ . \quad (4.4)$$

[7] The notation for w_α is not bold here, as u is assumed to be scalar.

Equations (4.4) and (4.3) together with the expansions (4.1) and (4.2) imply the following consistency identities to hold:

$$\left.\begin{aligned} w_\alpha\left[\overline{\mathrm{u}}\right](\boldsymbol{x}_\alpha) &= u(\boldsymbol{x}_\alpha) + O\left(h^4\right), \qquad \boldsymbol{\theta}_\alpha = D^{(2)}u\Big|_{\boldsymbol{x}_\alpha} + O\left(h^2\right) \\ \boldsymbol{\sigma}_\alpha &= D^{(1)}u\Big|_{\boldsymbol{x}_\alpha} + O\left(h^3\right), \quad \psi_\alpha = D^{(3)}u\Big|_{\boldsymbol{x}_\alpha} + O(h) \end{aligned}\right\} \tag{4.5}$$

Substituting (4.5) into (4.1) and again comparing (4.1) with (4.2) shows that $w_\alpha\left[\overline{\mathrm{u}}\right](\boldsymbol{x})$ satisfies the uniform consistency equation (recall that $\boldsymbol{x} - \boldsymbol{x}_\alpha = O\left(h^4\right)$):

$$\sup_{\boldsymbol{x}\in\mathcal{T}_\alpha} \left| w_\alpha\left[\overline{\mathrm{u}}\right](\boldsymbol{x}) - u(\boldsymbol{x}) \right| = O\left(h^4\right). \tag{4.6}$$

In summary the cubic reconstruction problem is equivalent to the computation of three approximate values $\boldsymbol{\sigma}_\alpha$, $\boldsymbol{\theta}_\alpha$ and ψ_α with the accuracy specified by (4.5).

4.2 Approximation Algorithm for the Derivatives

4.2.1 Introduction

In this section we present a general method to solve the approximation problem of Sec. 4.1. For the clarity of the presentation, we limit here ourselves to a slightly simpler problem than the cubic case (4.1): we seek a quadratic reconstruction of the form

$$w_\alpha\left[\overline{\mathrm{u}}\right](\boldsymbol{x}) = w_\alpha\left[\overline{\mathrm{u}}\right](\boldsymbol{x}_\alpha) + \boldsymbol{\sigma}_\alpha \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha) + \frac{1}{2}\boldsymbol{\theta}_\alpha \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha)^2. \tag{4.7}$$

The calculation principle for (4.7) readily translates to (4.1) at the cost of an additional step in Algorithm 4.5 hereafter. The accuracy requirement (4.5) now simply becomes

$$\left.\begin{aligned} w_\alpha\left[\overline{\mathrm{u}}\right](\boldsymbol{x}_\alpha) &= u(\boldsymbol{x}_\alpha) + O\left(h^3\right), \qquad \boldsymbol{\theta}_\alpha = D^{(2)}u\Big|_{\boldsymbol{x}_\alpha} + O(h) \\ \boldsymbol{\sigma}_\alpha &= D^{(1)}u\Big|_{\boldsymbol{x}_\alpha} + O\left(h^2\right), \end{aligned}\right\}. \tag{4.8}$$

The new reconstruction algorithm proceeds iteratively along the following steps:

1. Calculate a predicted value $\boldsymbol{\sigma}_{\boldsymbol{\alpha}}^{(\mathbf{1})}$ of the first derivative $D^{(1)}u\Big|_{\boldsymbol{x}_\alpha}$
2. Calculate a predicted value $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(\mathbf{1})}$ of the second derivative $D^{(2)}u\Big|_{\boldsymbol{x}_\alpha}$
3. Calculate a corrected value $\boldsymbol{\theta}_{\boldsymbol{\alpha}}^{(\mathbf{2})}$ of the second derivative $D^{(2)}u\Big|_{\boldsymbol{x}_\alpha}$
4. Calculate a corrected value $\boldsymbol{\sigma}_{\boldsymbol{\alpha}}^{(\mathbf{2})}$ of the first derivative $D^{(1)}u\Big|_{\boldsymbol{x}_\alpha}$
5. Evaluate the quadratic reconstruction by

$$w_\alpha\left[\overline{\mathrm{u}}\right](\boldsymbol{x}) = w_\alpha\left[\overline{\mathrm{u}}\right](\boldsymbol{x}_\alpha) + \boldsymbol{\sigma}_\alpha^{(2)} \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha) + \frac{1}{2}\boldsymbol{\theta}^{(\mathbf{2})}{}_\alpha \bullet (\boldsymbol{x} - \boldsymbol{x}_\alpha)^2 \tag{4.9}$$

4.2.2 Approximation Algorithm for the Derivatives

In this section we show how to implement the five steps mentioned in the preceding section. It takes the form of an algorithm which calculates the approximate derivatives $\boldsymbol{\sigma}_\alpha^{(2)}$ and $\boldsymbol{\theta}_\alpha^{(2)}$ in equation (4.9). The principle remains the same whenever the additional term $\boldsymbol{\psi}_\alpha$ is present (see (4.1)).

As a preliminary observation, consider the piecewise linear reconstruction of a typical second order MUSCL method. Such a reconstruction is labeled $1-$exact reconstruction in the present framework. In this case, only the 1^{st} derivative $\boldsymbol{\sigma}_\alpha$ is required and, as mentioned in Sec. 4.2 , this 1^{st} derivative is easily calculated using data from the 1^{st} neighborhood $\mathbb{V}_\alpha^{(1)}$ only. For this reason, due to easy data access, A MUSCL 2^{nd} order scheme can be efficiently implemented on parallel computers. However, when considering a k exact reconstruction with $k \geq 2$ this advantage disappears: stencils required for k-exactness become prohibitively large regarding computing efficiency. We now show how to solve this stencil problem. Specifically we show how to calculate a 2-exact reconstruction of the 1^{st} derivative based on data on the 1^{st} neighborhood $\mathbb{V}_\alpha^{(1)}$ only. This is the key to calculate approximate derivatives $\boldsymbol{\sigma}_\alpha$ and $\boldsymbol{\theta}_\alpha$ satisfying consistency conditions (4.8).

As a starting point suppose given in each cell $\mathcal{T}_\beta$ a 1-exact reconstruction $\boldsymbol{\sigma}_\beta$ of the 1^{st} derivative. This calculation is typically performed by least squares on the neighborhood $\mathbb{V}_\alpha^{(1)}$, (see Sec.7). The 1^{st} derivative $\boldsymbol{\sigma}_\alpha$ is expressed as

$$\boldsymbol{\sigma}_\alpha = \sum_{\beta \in \mathbb{V}_\alpha^{(1)}} \boldsymbol{\sigma}_{\alpha\beta} \overline{u}_\beta. \tag{4.10}$$

with coefficients $\boldsymbol{\sigma}_{\alpha\beta}$ given in (7.4). Our main result is the following :

Proposition 4.1. *Suppose that the function u is a polynomial of degree 2 on the 2^{nd} neighborhood $\mathbb{V}_\alpha^{(2)}$ of the cell $\mathcal{T}_\alpha$. Then the vector of the second order derivative $\partial_i\partial_j u|_{x_\alpha}$ satisfies the linear identity*

$$\partial_i\partial_j u\big|_{\boldsymbol{x}_\alpha} + \frac{1}{4} \sum_{\beta \in \mathbb{V}_\alpha^{(1)}} \sum_{\gamma \in \mathbb{V}_\beta^{(1)}} (\sigma_{\alpha\beta,i}\sigma_{\beta\gamma,j} + \sigma_{\alpha\beta,j}\sigma_{\beta\gamma,i}) \sum_{k,l=1}^{3} z^{(2)}_{\beta\gamma,kl}\, \partial_k\partial_l u\big|_{\boldsymbol{x}_\alpha} = \frac{1}{2} \sum_{\beta \in \mathbb{V}_\alpha^{(1)}} (\sigma_{\alpha\beta,i}\sigma_{\beta,j}\,[\overline{u}] + \sigma_{\alpha\beta,j}\sigma_{\beta,i}\,[\overline{u}]) \tag{4.11}$$

Let us comment this result:

- The vector with components $D^{(2)}u\Big|_{\boldsymbol{x}_\alpha} = \partial_i\partial_j u|_{x_\alpha}$ is a symmetric tensor with $1 \leq i \leq j \leq 3$. It is therefore identified with a vector in $\mathbb{R}^6$. This vector is the solution of the linear system (4.11).

- The right hand side of (4.11) is obtained by applying twice the operator (4.10) to $\overline{\mathfrak{u}} = [\overline{u}_1, \ldots, \overline{u}_N]^T$. This can be formally seen as a discrete counterpart of the differential identity $\partial^2 u = (\partial \circ \partial) u$.

Proposition 4.1 obviously suggests a way to calculate a consistent second derivative $\boldsymbol{\theta}_\alpha$. We omit the proof of the following

Corollary 4.2. *Consider any smooth function u with average values $\overline{\mathfrak{u}} = [\overline{u}_1, \ldots, \overline{u}_N]^T$. in the cells $\mathcal{T}_\alpha$. Then the linear system*

$$\boldsymbol{\theta}_{\alpha,ij} + \frac{1}{4} \sum_{\beta \in \mathbb{V}_\alpha^{(1)}} \sum_{\gamma \in \mathbb{V}_\beta^{(1)}} (\sigma_{\alpha\beta,i}\sigma_{\beta\gamma,j} + \sigma_{\alpha\beta,j}\sigma_{\beta\gamma,i}) \sum_{k,l=1}^{3} z^{(2)}_{\beta\gamma,kl} \boldsymbol{\theta}_{\alpha,kl} = \frac{1}{2} \sum_{\beta \in \mathbb{V}_\alpha^{(1)}} (\sigma_{\alpha\beta,i}\sigma_{\beta,j}[\overline{u}] + \sigma_{\alpha\beta,j}\sigma_{\beta,i}[\overline{u}]) \quad (4.12)$$

obtained by substituting $\boldsymbol{\theta}_\alpha$ to $D^{(2)}u\Big|_{x_\alpha}$ in (4.11) is a first order approximation of the 2^{nd} derivative of u : $\boldsymbol{\theta}_\alpha = D^{(2)}u\Big|_{\boldsymbol{x}_\alpha} + O(h)$.

Let us now outline the proof of Prop. 4.1

Proof. Suppose that u is a second order polynomial. For all cells $\mathcal{T}_\gamma$ and $\mathcal{T}_\beta$, the cell averages $\overline{u}_\gamma$ of u can be expressed in terms of values in cell $\mathcal{T}_\beta$, as

$$\overline{u}_\gamma = \overline{u}_\beta + \sum_{i=1}^{3} h_{\beta\gamma,i}\, \partial_i u|_{\boldsymbol{x}_\beta} + \frac{1}{2} \sum_{k,l=1}^{3} z^{(2)}_{\beta\gamma,kl}\, \partial_k \partial_l u|_{\boldsymbol{x}_\beta}, \forall \gamma \quad (4.13)$$

In addition, u satisfies for all α and for all $\beta \in \mathbb{V}_\alpha^{(2)}$

$$\partial_i \partial_j u|_{\boldsymbol{x}_\beta} = \partial_i \partial_j u|_{\boldsymbol{x}_\alpha} \text{ and } \partial_j u|_{\boldsymbol{x}_\beta} = \partial_j u|_{\boldsymbol{x}_\alpha} + \sum_{k=1}^{3} h_{\alpha\beta,k} \cdot \partial_k \partial_j u|_{\boldsymbol{x}_\alpha}, 1 \le i,j,k \le 3. \quad (4.14)$$

Unfortunately, the reconstructions $\boldsymbol{\sigma}_\beta$ applied to (4.13) do not give in general a consistent approximation to the 1^{st} derivative $\partial_j u|_{\boldsymbol{x}_\beta}$. This is due to the fact that $\boldsymbol{\sigma}_\beta$ is only 1-exact and not 2-exact. However, (4.14) and (7.2) allow to express the *reconstruction error* of $\boldsymbol{\sigma}_\beta$ on each cell $\mathcal{T}_\beta$ as

$$\sigma_{\beta,j}[\overline{u}] = \sum_{\gamma \in \mathbb{V}_\beta^{(1)}} \boldsymbol{\sigma}_{\beta\gamma,j} \overline{u}_\gamma = \partial_j u|_{\boldsymbol{x}_\beta} + \underbrace{\frac{1}{2} \sum_{\gamma \in \mathbb{V}_\beta^{(1)}} \sigma_{\beta\gamma,j} \sum_{k,l=1}^{3} z^{(2)}_{\beta\gamma,kl}\, \partial_k \partial_l u|_{\boldsymbol{x}_\beta}}_{\text{Reconstruction Error}}, \ \beta \in \mathbb{V}_\alpha^{(1)}. \quad (4.15)$$

Identities (4.14) are now used to express the derivatives of u at $\boldsymbol{x}_\beta$ in (4.15) in terms of the derivatives at $\boldsymbol{x}_\beta$ as:

$$\sigma_{\beta,j}\left[\overline{u}\right] = \partial_j u|_{\boldsymbol{x}_\alpha} + \sum_{k=1}^{3} h_{\alpha\beta,k}\cdot \partial_k\partial_j u|_{\boldsymbol{x}_\alpha} + \underbrace{\frac{1}{2}\sum_{\gamma\in\mathbb{V}_\beta^{(1)}} \sigma_{\beta\gamma,j}\sum_{k,l=1}^{3} z^{(2)}_{\beta\gamma,kl}\,\partial_k\partial_l u|_{\boldsymbol{x}_\alpha}}_{\text{Reconstruction Error}}\ ,\ \beta\in\mathbb{V}_\alpha^{(1)}\,. \tag{4.16}$$

Observe that the system of equations (4.16) contains only the derivatives of u at $\boldsymbol{x}_\alpha$. Consider now applying the operator $\boldsymbol{\sigma}_\alpha$ to (4.16) . This is operated by multiplying (4.16) with $\sigma_{\alpha\beta,i}$ and summing over $\beta\in\mathbb{V}_\alpha^{(1)}$. Taking the symmetric part in i and j of the result and using the consistency properties (7.2) of $\boldsymbol{\sigma}_\alpha$ yields

$$\sum_{\beta\in\mathbb{V}_\alpha^{(1)}} \sigma_{\alpha\beta,i}\,\partial_j u|_{\boldsymbol{x}_\alpha} = 0 \ \text{ and } \ \sum_{k=1}^{3}\sum_{\beta\in\mathbb{V}_\alpha^{(1)}} \sigma_{\alpha\beta,i}h_{\alpha\beta,k}\cdot\partial_k\partial_j u|_{\boldsymbol{x}_\alpha} = \partial_i\partial_j u|_{\boldsymbol{x}_\alpha}\,. \tag{4.17}$$

Finally identity (4.11) is deduced after some algebraic calculation. ■

An easy consequence is that the solution $\boldsymbol{\theta}_\alpha$ of (4.12) is a first order approximation to second derivative $D^{(2)}u\Big|_{\boldsymbol{x}_\alpha}$. Equation (4.12) is typical to our approach: it expresses how the discrete operator in the right-hand side of (4.12) has to be modified in order to restore consistency.

Remark 4.3. It is assumed that the matrix in the left-hand-side of (4.12) is non singular. In the case of an uniform grid, the sums $\sum_{\gamma\in\mathbb{V}_\beta^{(1)}}\sigma_{\beta\gamma,j}z^{(2)}_{\beta\gamma,kl}$ are identical for all cells, so that the constraints for 1-exactness (7.2) show that the matrix becomes the identity matrix. On a general grid we assume that this matrix is a perturbation of the identity matrix. This is actually an assumption on the smoothness of the grid. Numerical results over a large series of meshes have shown so far that it is non singular, as long as the 1-exact 1$^{\text{st}}$ derivatives satisfy the stability conditions discussed in [15].

Remark 4.4. The calculation of the approximant (4.1) up to 3$^{\text{rd}}$ order requires also a 2-exact 1$^{\text{st}}$ derivative denoted by $\boldsymbol{\sigma}^{(2)}_\alpha$ and satisfying $\sigma^{(2)}_{\alpha,j}\left[\overline{u}\right] = \partial_j u|_{\boldsymbol{x}_\alpha}$. It can be obtained by setting $\beta=\alpha$ and $\partial_k\partial_l u|_{\boldsymbol{x}_\alpha} = \theta_{\alpha,kl}$ in (4.15)

$$\sigma^{(2)}_{\alpha,j}\left[\overline{u}\right] = \sigma_{\alpha,j}\left[\overline{u}\right] - \frac{1}{2}\sum_{\beta\in\mathbb{V}_\alpha^{(1)}}\sigma_{\alpha\beta,j}\sum_{k,l=1}^{3} z^{(2)}_{\alpha\beta,kl}\theta_{\alpha,kl} = \partial_j u|_{\boldsymbol{x}_\alpha}\,. \tag{4.18}$$

The key equation (4.12) for $\boldsymbol{\theta}_\alpha$ is the foundation of the following reconstruction algorithm, which applies to all smooth functions u with average data $\overline{\mathrm{u}} = [\overline{u}_1,\ldots,\overline{u}_N]^T$

Algorithm 4.5. *Perform the following steps*

1. *Calculate in each cell* $\mathcal{T}_\beta$ *an approximate* 1*-exact* 1^{st} *derivative* $\boldsymbol{\sigma}_\beta$ *satisfying* $\boldsymbol{\sigma}_\beta = D^{(1)}u\Big|_{\boldsymbol{x}_\beta} + O(h)$*. This is typically the* 1^{st} *derivative calculated by least squares, (see Sec. 7).*

2. *Calculate in each cell the* 1*-exact* 1^{st} *derivative of the values* $\boldsymbol{\sigma}_\beta$ *. This gives the right hand side of (4.12).*

3. *Solve the* 6×6 *linear system (4.12) to obtain* $\boldsymbol{\theta}_\alpha$*, a consistent approximation of the* 2^{nd} *derivative satisfying* $\boldsymbol{\theta}_\alpha = D^{(2)}u\Big|_{\boldsymbol{x}_\alpha} + O(h)$*.*

4. *Using (4.18), calculate a* 2*-exact* 1^{st} derivative satisfying $\boldsymbol{\sigma}^{(2)}_\alpha = D^{(1)}u\Big|_{\boldsymbol{x}_\alpha} + O(h^2)$*.*

5. *Calculate the quadratic approximation (4.9) to* u*. This is a third order approximation of the form (4.7) .*

Remark 4.6. To extend Algorithm 4.5 to the approximation of the 3^{rd} derivative $\boldsymbol{\psi}_\alpha = D^{(3)}u\Big|_{\boldsymbol{x}_\alpha} + O(h)$, one needs to consider the reconstruction error of the 2-exact 2^{nd} derivative $\boldsymbol{\theta}_\alpha$ on a 3^{rd} degree polynomial u to obtain a linear equation for $\boldsymbol{\psi}_\alpha$ analogous to (4.11).

In conclusion, the Algorithm 4.5 combines both *high order accuracy* – the reconstruction is k-exact – and *locality* – only the first neighborhood is used.

5 Numerical Test Cases

In this section we show three numerical test cases showing the interest of the previously introduced cubic reconstruction. As mentioned in Sec. 1, the main focus of the two first test cases is on the accuracy evaluation of the fourth order approach, compared to lower order variants of the same scheme. The third case shows the applicability of the present approach to a complex test case in aerothermochemistry, involving multispecies and chemical reactions.

5.1 Acoustic Wave in a 3D Channel

5.1.1 Context in aeroacoustics

Aeroacoustics computations are usually performed by associating two kinds of codes:

- an unsteady CFD computation code for LES (Large Eddy Simulation) . This code calculates acoustic sources generated by the turbulent flow and performs the numerical propagation in the acoustic near field region,

- an acoustic code to reconstruct the far field from unsteady CFD results recorded at CFD-acoustic interfaces (e.g. [19]).

It is thus very important to assert the efficiency of the CFD code for accurately convecting acoustic waves in the near field up to the CFD-acoustic interface without excessive damping or phase errors. In practical situations, complex geometries are often to be considered and unstructured meshes are desirable when not mandatory. In general, when 2^{nd} order CFD schemes are used, fine meshes are required in the near field region to compensate the limited frequency bandwidth of these schemes. The works presented in [10, 18] show recent numerical results obtained with a 2^{nd} order scheme for aeroacoustics problems. It is believed that higher order accurate schemes will demonstrate higher bandwidth thus allowing less refined meshes for a fixed precision. A popular measure of the bandwidth of a scheme is simply expressed as the minimum number of grid points (or cells) per wavelength (denoted by *ppw*). This is required to properly propagate a given wave pattern over a significant length [6]. When using a 2^{nd} order schemes, a *ppw* of the order of 20-25 is often mentioned while with recent high order schemes, the *ppw* parameter is likely of the order 4-6 for *ppw.*

5.1.2 Acoustic wave in a channel: parametric data

To evaluate high order schemes for aeroacoustic computations, we have choosed a simple benchmark case for acoustic wave propagation in a uniform flow. The advantage of this simple configuration is that linear acoustic theory easily provides analytical solutions. Our test case is as follows. The geometry consists of a square duct with axis Ox and uniform cross section of 0.2×0.2m. The length is L=10 m. The duct is filled with a uniform flow of air at Vx =1 m.s^{-1} at ambient conditions: 101 325 Pa and 300 K. Under these conditions the sound velocity is $c = 347$ m.s^{-1}. Boundary conditions are set to:

- *left end (x=0)*: subsonic inlet conditions set at Vx =1 m.s^{-1}, T =300 K.
- *right end (x=L)*: subsonic outlet conditions set at P=101 325 Pa.
- *lateral surfaces* : symmetry conditions .

Since our code is compatible with a general grid, we selected a uniform mesh of isotropic tetrahedra of size a =20 mm. Only at the right hand side four layers of prisms of uniform height of 16 mm were added. This part of the grid was intended to allow for a planar acoustic wave generation as described below. The mesh consists of 288876 cells and 600053 faces, see fig. 5.1.

The simulation is initiated by a sinusoidal perturbation at the right end of the duct: a time harmonic, isothermal (300 K), distributed mass source enters the first layer of prismatic cells on the right of the computational domain. It is expressed as:

$$\dot{m} = A \sin(2\pi f t) \tag{5.1}$$

In (5.1) the amplitude A is set so as to generate a low amplitude acoustic wave (smaller than 20 Pa of physical amplitude) such that non linear effects can be ignored. This source

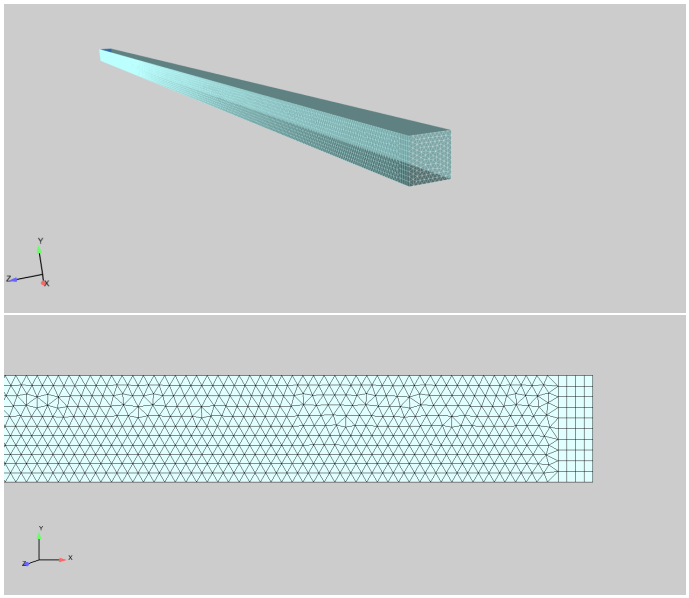


Figure 5.1: Duct overview and grid details at the right end

generates a plane acoustic wave propagating upstream, from the right to the left with a phase velocity $V_\varphi = c - V_x 346\ 1\,\mathrm{m.s^{-1}}$. The corresponding wave length is thus:

$$\lambda = \frac{V_\varphi}{f} \tag{5.2}$$

The physical elapsed time of each run is 20 ms.This corresponds to a propagation length of 6.9 m in the counter flow direction ($x < 0$).The inlet boundary is therefore not impacted, thus alleviating any acoustic reflection issues at the left end boundary. A total of 7578 time steps are performed at $CFL = 0.4$. By using a series of values for the source frequency f, we estimate the resolution of the acoustic wave for a given grid, as represented by the parameter ppw (number of grid points per wave length) defined by

$$ppw = \frac{\lambda}{a} \tag{5.3}$$

Runs were performed for a series of 4 values of the frequency f, which corresponds to a ppw number varying from 6 (coarse mesh) to 59 (extra fine mesh). Table 5.1 summarizes the results.

When 2^{nd} order CFD schemes are used in aeroacoustic, a value of 20-25 ppw is typically used. This corresponds to a specific grid step size, according to the expected frequency range of the acoustic sources. This value was selected for the runs of the R2 series. Then for each run conditions, computations were performed for schemes of various orders, as

Run	R1	R2	R3	R4
f (Hz)	300	700	1500	3000
λ (m)	1.1533	0.4943	0.2307	0.1153
ppw = λ/a	59	25	12	6

Table 5.1: Values of tested wave frequency

described in Table 5.2, and the wave quality was observed and compared at different numerical transducers located along the duct axis.

Run	R1	R2	R3	R4
O2 - RK2	x	x	x	x
O3 - RK3		x	x	x
O4 - RK4		x	x	x
O4 - RK4, CFL = 0.8			x	

Table 5.2: Computation matrix

5.1.3 Amplitude and phase error analysis

The analysis of the numerical results was performed using three numerical transducers located on the duct axis at 7 m, 8 m and 9 m, respectively, see Fig. 5.2 top. For example we show on Fig. 5.2 (bottom) the pressure time history for a specific run.

The quality of the schemes was estimated from the wave amplitude damping along the propagation direction. The phase error was measured with respect to an ideal linear acoustic wave propagation solution of the form

$$P(x,t_i) = \Re\left[P_0 e^{i(\omega t - kr + \varphi_0)}\right]$$

with $\omega = 2\pi f \in \mathbb{R}$ and $k = k^r + ik^i \in \mathbb{C}$.
The calculated time series is assumed to have the form

$$F\left(x,t_i\right) = A_0\left(x\right)\sin\left(2\pi f\left(x\right)t_i + \varphi\left(x\right)\right)e^{-\alpha(x)t_i}\,. \tag{5.4}$$

where the four parameter functions $A_0(x)$, $f(x)$, $\varphi(x)$ and $\alpha(x)$ have to be estimated. Identifying these four functions is obtained by a fitting approximation technique using exponential (in time) sums. This kind of fitting is thoroughly presented e.g. in [23]. Here we assume that our numerical data have the form of series

$$F(x,t_i) = \sum_{n=1}^{N} a_n e^{i\sigma_n t_i} \text{ with } (a_n,\sigma_n) \in \mathbb{Z}^2 \tag{5.5}$$

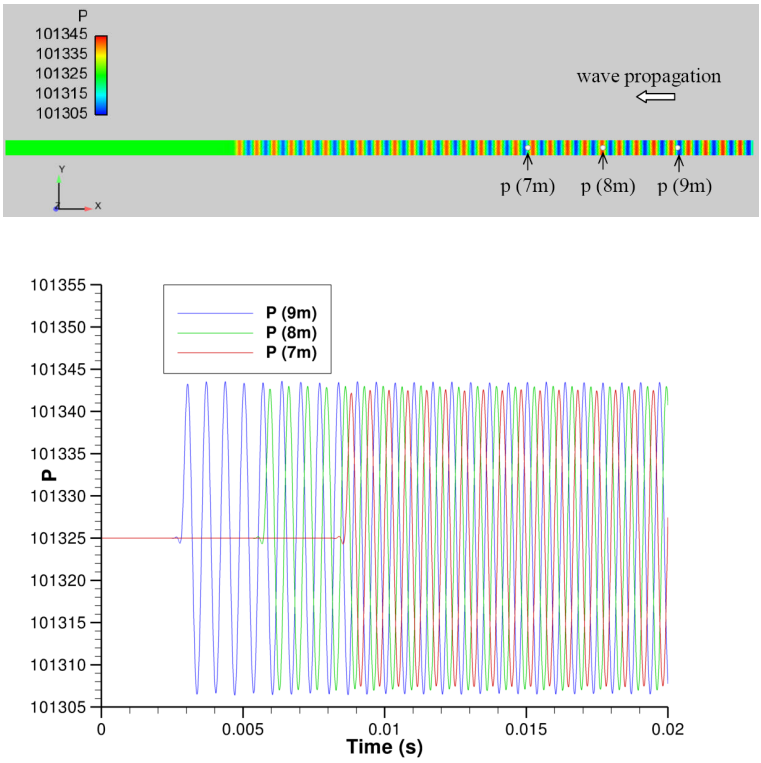


Figure 5.2: Illustration of wave propagation and pressure time histories at numerical transducers case R3/O4-RK4

As F represents a real monochromatic wave, so that we limit the sum to two terms ($N = 2$ in (5.5)). By identification the functions $A_0(x)$, $f(x)$, $\varphi(x)$ and $\alpha(x)$ are constant and they are expressed in terms the values $a_1, a_2, \sigma_1$and σ_2 as

$$\begin{aligned} A_0 &= \sqrt{4a_1a_2} & 2\pi f &= \frac{\sigma_1 - \sigma_2}{2} \\ \varphi &= \frac{1}{2i}\log\left(\frac{-a_1}{a_2}\right) & \alpha &= \frac{\sigma_1 + \sigma_2}{2i} \end{aligned} \tag{5.6}$$

An iterative least square procedure is then used to identify these four constants.
Turning back to our damping and dispersion evaluation, we use the now estimated functions $A_0(x)$ and $\varphi(x)$ as a basis of our measurement [8]. The numerical wave damping for one

[8]Note that the two other functions $f(x)$ and $\alpha(x)$ are used only to guarantee the accuracy of the exponential fitting.

single wavelength is given by

$$\begin{aligned} \frac{A_0(x_2)}{A_0(x_1)} &= e^{k^i(x_2-x_1)} = e^{\lambda k^i\left(\frac{x_2-x_1}{\lambda}\right)} \\ e^{\lambda k^i} &= e^{\left(\frac{\lambda}{x_2-x_1}\right)\log\left(\frac{A_0(x_2)}{A_0(x_1)}\right)} \end{aligned} \tag{5.7}$$

where $e^{\lambda k^i}$ represents the wave damping per wave length traveled. Similarly, the numerical dispersion can be represented by the phase difference between two transducer locations as

$$\begin{aligned} \Delta\varphi_{21} &= \varphi(x_2) - \varphi(x_1) \\ (\Delta\varphi_{21})_{\text{th}} &= -k^r(x_2 - x_1) \end{aligned} \tag{5.8}$$

where $k^r = \frac{2\pi f}{V_\varphi}$.
The relative error on the phase error can be finally expressed as

$$\frac{\partial V_\varphi}{V_\varphi} = \frac{\partial(\Delta\varphi_{21})}{\frac{2\pi(x_2-x_1)}{\lambda}}. \tag{5.9}$$

5.1.4 Numerical Results

The comparison analysis of Sec 5.1.3 was carried out with the results obtained from different runs summarized in Table 5.2. These results are reported in Tables 5.3, 5.4 and in Fig 5.3 , 5.4.

Run (ppw)	O2-RK2	O3-RK3	O4-RK4	O4-RK4(CFL=0.8)
R1 (59)	0.9995			
R2 (25)	0.9961	0.9965	0.9990	
R3 (12)	0.9843	0.9866	0.9942	0.9939
R4 (6)	0.9188	0.9276	0.9754	

Table 5.3: Summary of results for wave damping pwl

Run (ppw)	O2-RK2	O3-RK3	O4-RK4	O4-RK4(CFL=0.8)
R1 (59)	-0.0119%			
R2 (25)	-0.0695%	-0.0054%	0.0000%	
R3 (12)	-0.3046%	-0.0183%	-0.0256%	-0.0287%
R4 (6)	-1.0516%	-0.0466%	-0.0744%	

Table 5.4: Summary of results for relative phase velocity error

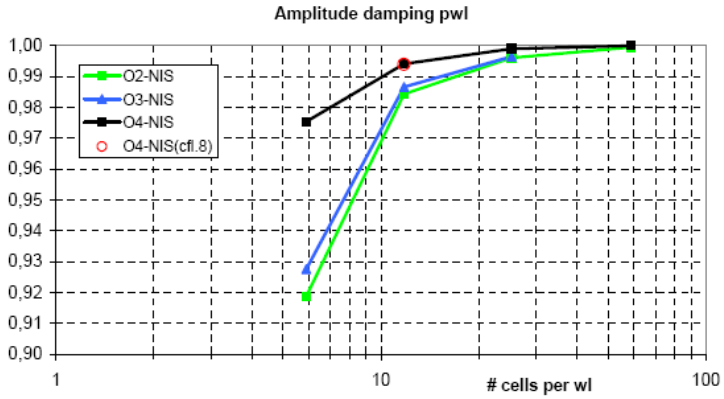


Figure 5.3: Wave damping per wavelength

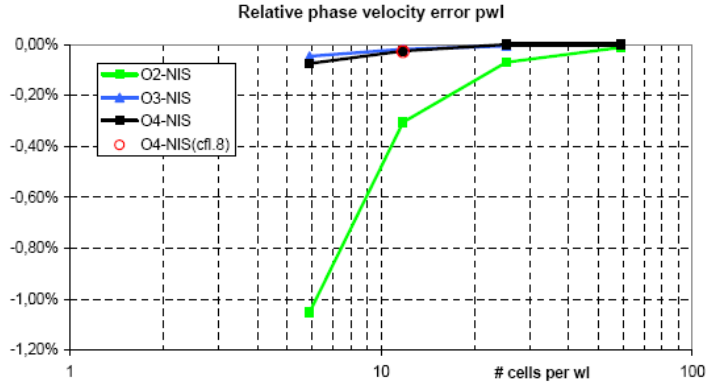


Figure 5.4: Relative phase velocity error

Tables 5.3 , 5.4, and Fig. 5.3 , 5.4 show that the higher order scheme provides a drastic improvement in the ability of the code to propagate acoustic waves. Even on a coarse grid (up to 6 points per wavelength) a very small damping and phase error was observed. Regarding a detailed comparison, Fig. 5.3 shows a similar damping between 2^{nd} order and the 3^{rd} order scheme whereas Fig. 5.4 shows that the 3^{rd} order and the 4^{th} order schemes have a similar phase error. This is easily interpreted in terms of the *modified equation* associated with the linear transport equation. A 2^{nd} and a 3^{rd} order scheme have a first dissipative error $h^3\partial_x^{(4)}u$, whereas a 3^{rd} and 4^{th} order schemes have a first dispersive error of the form $h^4\partial_x^{(5)}u$. Overall the 4^{th}scheme outperforms the two other schemes.

Table 5.5reports numerical values regarding computational cost. The computational complexity of the 3^{rd} and 4^{th} schemes was numerically measured against the 2^{nd} order scheme. Two values are reported. First the intrinsic CPU ratio, which is a numerical

a	O2-RK2	O3-RK3	O4-RK4	O4-RK4(CFL=0.8)
Intrinsic CPU ratio	1	4.7	12.0	12.0
Run CPU ratio	1	4.7	12.0	6.0

Table 5.5: Relative CPU cost

evaluation of the cost of one single time iteration. Second the CPU ratio for the full simulation. The latter also depends on the stability properties of each scheme. As an example, the RK4 (explicit) time stepping scheme allows a stability limit of $CFL = 0.8$ with the 4^{th} order scheme instead of $CFL = 0.4$with the second order scheme.The overall CPU ratio is therefore reduced to a factor , as reported in Table 5.5. The change from $CFL = 0.4$ to $CFL = 0.8$ did not bear any loss in terms of scheme accuracy, as shown in Tables 5.3 and 5.4 and in Fig. 5.3 and 5.4.

5.2 Periodic Advection of an Isothermal Two-dimensional Vortex

5.2.1 Inviscid vortices

This test case is based on the advection of a inviscid vortex in a periodic box. This is a popular non linear test case based on an analytical solution to the Euler equations, which is used by many authors to assess the accuracy of numerical schemes. As in the acoustic test case of Sec. 5.1 the behaviour of the scheme under investigation is mostly the numerical dissipation and the dispersion of the vortex for large physical time. This is known to be a severe challenge impacting many simulations related to turbulence.

The Euler equations for one single species and with no source term corresponds with the notation for conservation laws in Section (2.2) to the parameters:

$$n_{sp} = 1 \; n_{sc} = 1, \; \boldsymbol{\varphi} = 0, \; \varsigma = 0$$

The vector of physical variables is $\boldsymbol{u} = [p, T, \boldsymbol{v}]$ and the only parameter is the state equation $\rho = \rho(p, T)$. Let us briefly recall how the vortex is defined: as a first step a time-independent solution with radial symmetry is defined as follows. A polar reference frame is introduced in the x, y plane with basis vectors

$$\boldsymbol{e_r} = \cos\theta \boldsymbol{e}_x + \sin\theta \boldsymbol{e}_y, \boldsymbol{e}_\theta = -\sin\theta \boldsymbol{e}_x + \cos\theta \boldsymbol{e}_y$$

In this basis the velocity is $\boldsymbol{v} = v_r e_r + v_\theta \boldsymbol{e}_\theta$ and we seek time independent solutions $\boldsymbol{u} = \boldsymbol{u}_s(\boldsymbol{x})$ to (2.1) with no radial velocity $v_{r,s} = 0$. Rotational invariance around the Oz axis translates to $v_\theta = v_{\theta,s}(r), p = p_s(r), T = T_s(r)$. With these assumptions, the mass and energy equation are automatically satisfied, and the momentum equation reduces to the radial equilibrium, which is expressed as

$$v_{\theta,s}^2 = \frac{r}{\rho_s(p_s, T_s)} \frac{dp_s}{dr}. \tag{5.10}$$

Then the pressure $p_s(r)$ and the temperature $T_s(r)$ are free parameter functions and the radial velocity is deduced from Equation (5.10). Finally, Galilean invariance of (2.1) implies that for any constant velocity vector $\boldsymbol{v}_0$ the time dependent vortex

$$\boldsymbol{u}(x,t) = \boldsymbol{u}_s(\boldsymbol{x} - \boldsymbol{v}_0 t) \tag{5.11}$$

is also a solution of the Euler system.

5.2.2 Test Case: Inviscid Isothermal Vortex

Assume now a constant temperature and a Gaussian pressure field

$$T(r) = T_0, \;\; p(r) = p_0 \, (1 - a_0 \, exp(\frac{r^2}{\sigma^2}))$$

A constant temperature is a suitable choice as possible temperature oscillations are a suggestive marker of numerical defects. Note that instead of selecting a constant temperature, another popular choice is the constant entropy assumption [24] . The specific parameters of our test are as follows. The fluid is supposed to be a perfect gas of molar mass $\mathcal{M}$ =0.029 kg.mol^{-1},with constant heat capacities c_p, c_v with ratio $\gamma = c_p/c_v = 1.4$ If $\mathcal{R} = 8.314\,472$ Jmol^{-1}K^{-1},is the perfect gas constant, the density field is then given by

$$\rho(r) = \frac{p(r)}{\frac{\mathcal{R}}{\mathcal{M}} T(r)},$$

and the velocity field follows from (5.10) and (5.11).

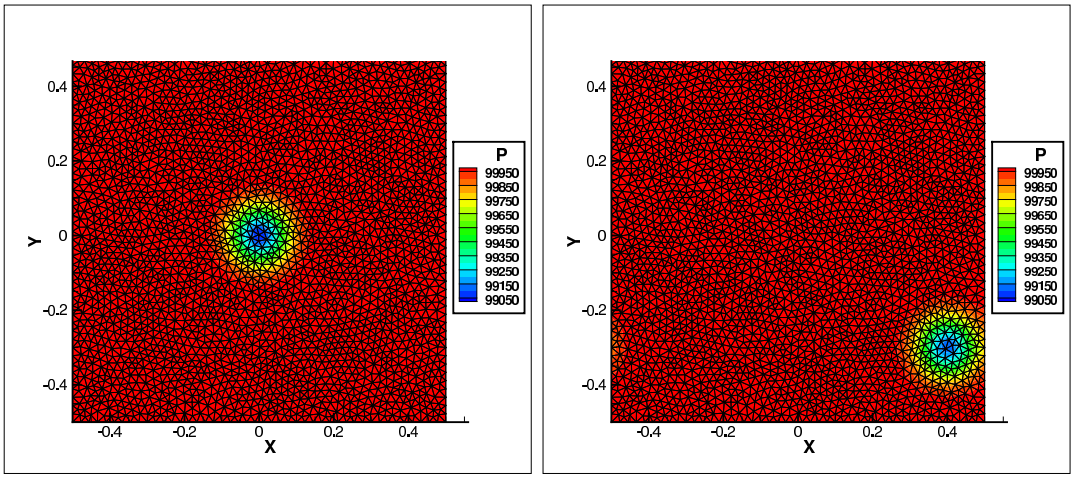
The domain Ω is a square of edge length $L = 1\,\mathrm{m}$. Periodic boundary conditions are applied on the boundary. The temperature is $T' = T_0'$ and the pressure function is selected as

$$p'(r) = p_0 \left(1 - a_0 \exp\left(\frac{r^2}{\sigma^2}\right)\right), \tag{5.12}$$

where $T_0' = 300\,\mathrm{K}$, $p_0 = 10^5\,\mathrm{Pa}$, $a_0 = 10^{-2}$. The value $\sigma = 0,075\,\mathrm{m}$ is selected so that the pressure is constant up to machine accuracy on $\partial\Omega$. The advection velocity is selected as $\tilde{\boldsymbol{v}} = (50.25)\,\mathrm{m.s}^{-1}$, and the physical final time is $t_{\mathrm{end}} = 0.2\,\mathrm{s}$, so that the vortex travels across Ω 10 times in the x-direction and 5 times in the y-direction.

Numerical simulations were performed using three kinds of grids: triangular, quadrangular and uniform Cartesian. In each case, the convergence analysis was observed with 14 grids ordered from the coarser to the finest. The time stepping scheme was the classical 4-stage explicit Runge Kutta method with time step $\Delta t = 81\,0^-\,6\,\mathrm{s}$ and maximum CFL number of 0.5. An identical time step Δt was used for all grids but on the finer unstructured grids the CFL constraint leads to the use of smaller time steps. Fig. 5.5 on the left displays the pressure contours at initial time.

The computation is done for k-exact reconstruction from $k = 1$ to $k = 3$. Note that for $k = 1$ the reconstruction is the standard least squares gradient for classical 2$^{\mathrm{nd}}$ finite volume schemes that does not make use of Alg. 4.5.



(a) Initial condition ($t = 0\,\mathrm{s}$) (b) 4^{th} order at $t = 0,0188\,\mathrm{s}$

Figure 5.5: Pressure contours of the vortex on the 3^{rd} triangular grid.

Mesh Type	Reconstruction		
	$k=1$	$k=2$	$k=3$
Triangles	1.4350	1.9734	4.4311
Quadrilaterals	1.1119	1.7450	4.5408
Cartesian	1.0453	1.6754	4.3642

(a) Convergence in $\mathcal{L}_2(\Omega)$

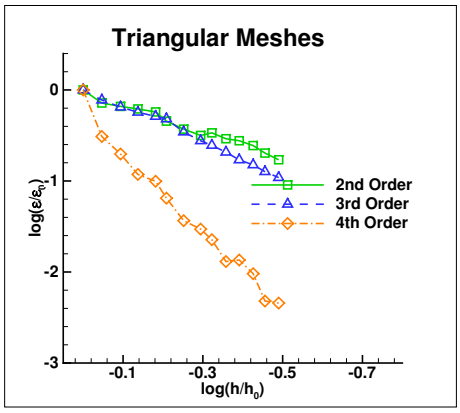
Mesh Type	Reconstruction		
	$k=1$	$k=2$	$k=3$
Triangles	1.3755	1.7675	4.3200
Quadrilaterals	1.0233	1.5637	4.3390
Cartesian	0.94045	1.4865	4.2684

(b) Convergence in $\mathcal{L}_\infty(\Omega)$

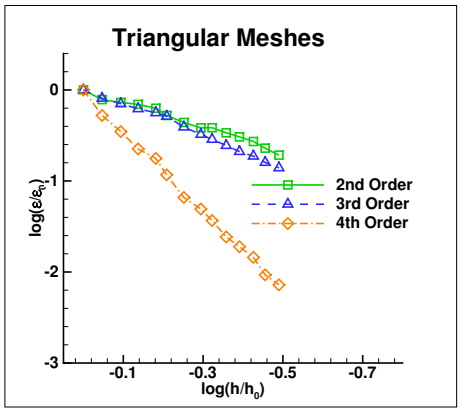
Table 5.6: Order of grid convergence estimated by a linear fit

5.2.3 Numerical results

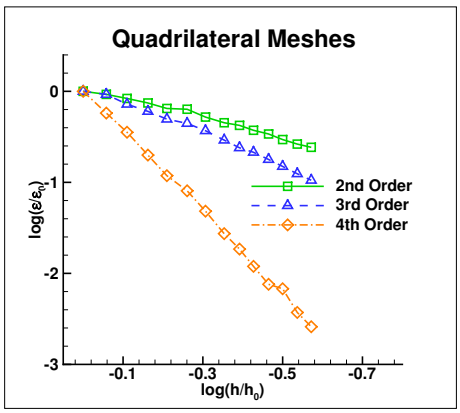
At time $t_{\text{end}} = 0.2\,\mathrm{s}$, the error ε on the pressure field was measured in the discrete $\mathcal{L}_2(\Omega)$ and $\mathcal{L}_\infty(\Omega)$ norms. Let ε_0 and h_0 denote the error and the approximate grid size for the coarsest mesh, respectively. Fig. 5.6 shows the plot of $\log(\varepsilon/\varepsilon_0)$ against $\log(h/h_0)$ for the different kinds of grid. Tab. 5.6 displays the slope of the linear fit of $\log(\varepsilon/\varepsilon_0)$ against $\log(h/h_0)$. The results show that the k-exact reconstruction by Alg. 4.5 actually achieves 4^{th} order grid convergence for $k = 3$ (cubic reconstruction). For $k = 1$ and $k = 2$, the order of convergence does not reach the expected values. This is explained by the fact that the grids used are too coarse for these lower order schemes. The grid size h is simply not in the range where the error of order $k+1$ decreases with h^{k+1}. This can be observed in Fig. 5.6 where the slopes for the 2^{nd} and 3^{rd} order schemes are not constant but tend to steepen as the grids become finer. Additional numerical results, not shown here, clearly support this



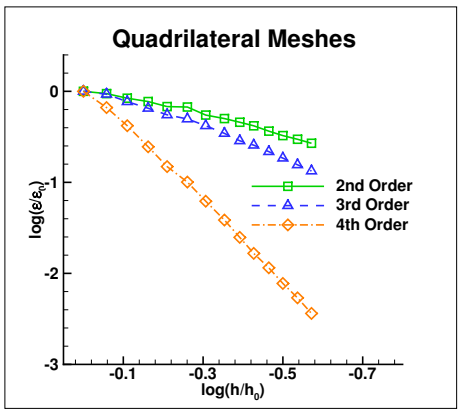
(a) Convergence in $\mathcal{L}_2(\Omega)$



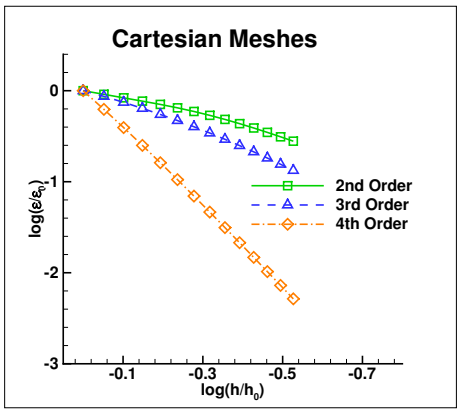
(b) Convergence in $\mathcal{L}_\infty(\Omega)$



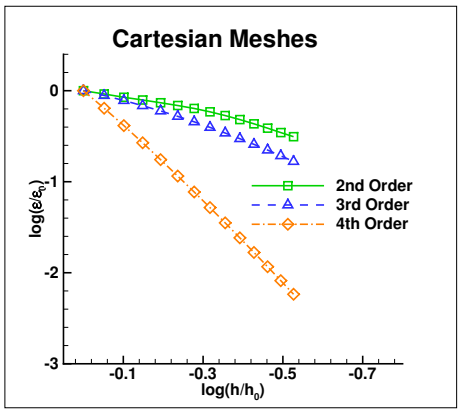
(c) Convergence in $\mathcal{L}_2(\Omega)$



(d) Convergence in $\mathcal{L}_\infty(\Omega)$



(e) Convergence in $\mathcal{L}_2(\Omega)$



(f) Convergence in $\mathcal{L}_\infty(\Omega)$

Figure 5.6: Grid convergence for the isothermal vortex.

claim. The 4^{th} order scheme, by contrast, presents a nearly uniform rate of convergence over all grids. It is worth to mention that the 4^{th}order accuracy can be observed even on very coarse grids. As can be observed on the fight of Fig. 5.5 , the distortion of the vortex is minimal on a quite coarse triangular grid.

5.3 Computation of a Laminar Flame with a Positivity Preserving Scheme

This section presents the computation of a reactive flow in a two-dimensional geometry. The goal is to show that the 4^{th} order finite-volume scheme is working in this configuration that is derived from the *Pagode test rig* of [9][11] and [2].

5.3.1 Two-dimensional Reactive Flow of a Fluid Mixture

The geometry representing the model of a Bunsen burner is displayed in Fig. 5.7. The mesh consists of about 150 000 triangular cells. A near stoechiometric methane/air mixture with an equivalence ratio of 1.05 is injected through the inlet boundary at the bottom. The chemical reaction produces a laminar flame that is anchored to the lips of the channel. Due to a constant mass flow rate, the flame has the shape of a triangle and is asymptotically time-independent.

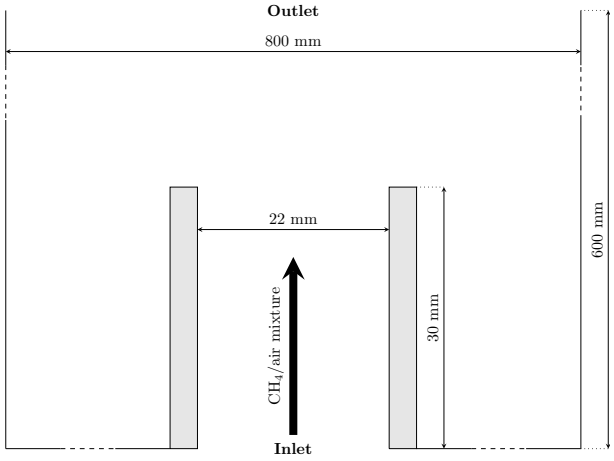


Figure 5.7: Computational domain and boundary conditions.

The propagation of the kind of premixed flame considered here is essentially determined by the diffusive fluxes $\boldsymbol{\varphi}$ in (2.1). However, in the implementation used for the computation, only the approximation of the convective fluxes $\boldsymbol{f}$ in (2.1) is 4^{th} order accurate whereas the diffusive fluxes $\boldsymbol{\varphi}$ and the source terms ς are modeled by 2^{nd} order approximations.

[9]

To obtain a test case that is more responsive to the accuracy of the convective fluxes, a periodically time-dependent inflow velocity is imposed as

$$v_{\text{inflow}}(t) = 0.97 + \sqrt{2} \times 0.19 \cos(2\pi \times 62.5t) \ .$$

The flame is laminar, so there is no need to take into account turbulence modeling or complex interactions between turbulence and chemical reactions. Molecular diffusion and heat conduction are modeled by Fick's laws and Fourier's law, respectively. In order to keep the computation time at a reasonable cost, a quasi-global kinetics scheme [12] involving six species and two reactions is used for the methane/air combustion:

$$CH_4 + \frac{3}{2}O_2 \ \rightarrow \ CO + 2H_2O \tag{5.13}$$

$$CO + \frac{1}{2}O_2 \ \leftrightarrow \ CO_2 \, . \tag{5.14}$$

5.3.2 Positivity Preserving of densities

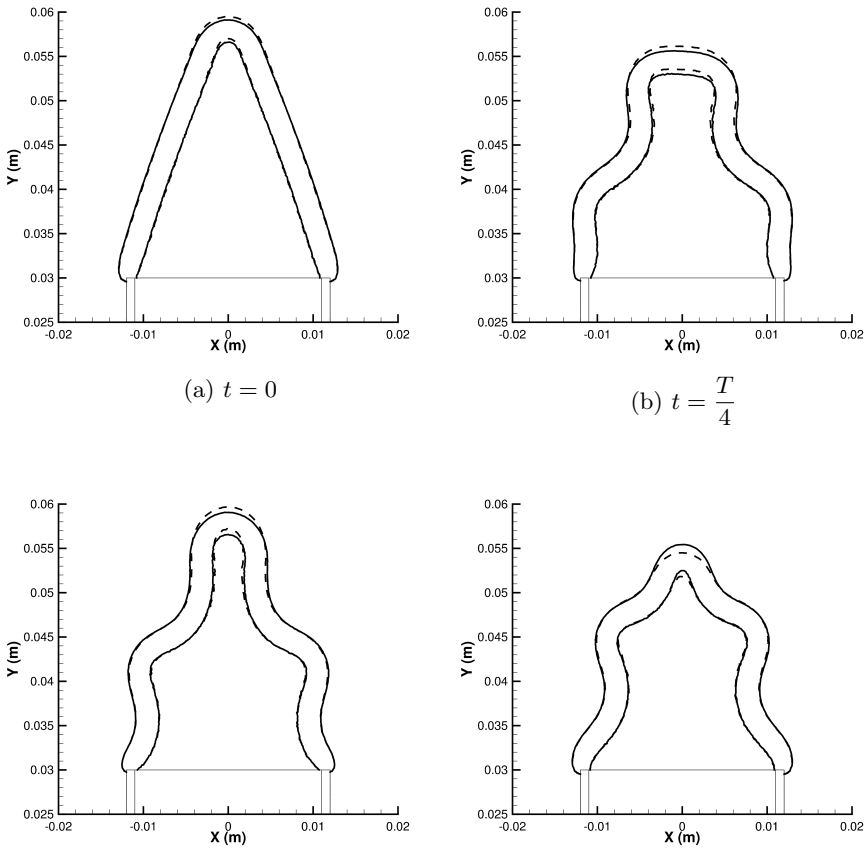
The faithful physical modeling of multi-species flow requires numerical schemes that keep the densities ρ_i and the mass fractions y_i inside their respective physical bounds: $0 \leq \rho_i$ and $0 \leq y_i \leq 1$. These bounds are compulsory for reactive flows because the source terms $\boldsymbol{\varsigma}$ in (2.1) commonly used to model the chemical reactions are only defined for non-negative y_i. A recently described mechanism [25, 26] to enforce global bounds on the numerical solution has been successfully adapted to the present finite volume scheme.

5.3.3 Results and Discussion

Results are shown on figure 5.8. It is worth noting that the 4^{th} order accurate scheme is sufficiently robust to handle this combustion problem with several species and stiff chemical reaction rates. However, there is no significant difference concerning the quality of the solution obtained. This can be explained by the fact that the mesh must be fine enough to obtain a good accuracy for the 2^{nd} order diffusive fluxes. In particular, it is difficult to coarsen the mesh while keeping the flame attached to the lips of the channel.

6 Conclusion

This paper presents a new k-exact reconstruction strategy for finite volume schemes with high order spatial accuracy. The central idea of Algorithm 4.5 is to iterate over adjacent cell neighbors which proceeds as follows. First the 1^{st} derivative is calculated from the cell averages in the adjacent cells. Second, the 2^{cd}derivative is calculated from the 1^{st} derivatives in adjacent cells. Then a 3^{rd} derivative is deduced from the 2^{nd} derivatives in the adjacent cells and so on. Each step involves the application of a 1-exact 1^{st} derivative



(a) $t = 0$

(b) $t = \dfrac{T}{4}$

(c) $t = \dfrac{T}{2}$

(d) $t = \dfrac{3T}{4}$

Figure 5.8: Heat release ($\mathrm{J.s^{-1}}$) iso-values (300 K and 2000 K) for $t = 0, T/4, T/2, T$. Solid lines: O4 scheme, dashed lines: O2 scheme.

to the m^{th} derivative calculated in the previous step, which results in a non-consistent approximation of the $(m+1)^{\text{th}}$ derivative. This non consistent approximation is transformed into a consistent one at the cost of a multiplication by an non singular local matrix in each cell [13, 14]. This generalizes the simple case of 1-exact 1^{st} derivative operators which are available in standard 2^{nd} order finite volume solvers. In this case they are often referred to as *gradient interpolation* or *gradient reconstruction*.

Convergence analysis on unstructured grids is presented in Sec. 5.1 and Sec. 5.2. The first case evaluates the influence of the spatial accuracy on the propagation of acoustic waves in a three-dimensional channel. This is of great importance for practical computations in aeroacoustics where an unsteady CFD computation must be able to transport acoustic waves in the near field without excessive damping. This analysis was performed on a sequence of tetrahedral grids: this is required for complex geometries modeling such as nozzle flows. The second test case is the advection of an isothermal inviscid vortex, an exact solution of the two-dimensional Euler equations. Whereas the first case is more or less a linear problem, the second case evaluates the capabilities of the scheme to capture genuinely non-linear solutions of the Euler equations. The two first test cases show that the 4^{th} order reconstruction achieves a significant reduction in numerical dissipation and a substantial improvement of the solution. Finally Sec 5.3 shows the capability of the scheme in a case where complex terms are involved.

To conclude let us mention that the present approach can be implemented in any finite volume scheme without huge efforts. Of course some additional upgrades are required to conform to the global accuracy of the scheme, such as high order quadratures rule (for the fluxes) or high order time stepping methods. An important advantage of this high order spatial discretization of a conservation law is that it involves the same number of semi-discrete equations as in 2^{nd} order scheme. This simplifies the use of implicit time stepping methods and complex physical models.

7 Appendix: Least Squares Slope

We briefly recall in this section how the slope $\boldsymbol{\sigma_\alpha}$ is calculated on the first neighborhood $\mathbb{V}_\alpha^{(1)}$ defined by (3.2). This calculation will serves as the initial step of the recursive algorithm presented in the next section. In the language of the preceding section, we are interested in a linear interpolant ($k=1$) based on some 1-exact first derivative called $\boldsymbol{\sigma}_\alpha$. By linearity, $\boldsymbol{\sigma}_\alpha$ can be written as

$$\boldsymbol{\sigma}_\alpha = \sum_{\beta \in \mathbb{V}_\alpha^{(1)}} \boldsymbol{\sigma}_{\alpha\beta} \overline{u}_\beta \tag{7.1}$$

and we need to determine the coefficients $\boldsymbol{\sigma}_{\alpha\beta}$. Suppose that u is a polynomial of degree 1 with first derivative $\boldsymbol{s} \in \mathbb{R}^3$. Then the cell averages of u satisfy $\overline{u}_\beta = \overline{u}_\alpha + \boldsymbol{h}_{\alpha\beta} \cdot \boldsymbol{s}$ for $\beta \in \mathbb{V}_\alpha^{(1)}$. For 1-exactness to hold, it is necessary and sufficient that the values $\boldsymbol{\sigma}_{\alpha\beta}$ satisfy

the identities

$$\left.\begin{aligned} \sigma_{\alpha\alpha,i} &= -\sum_{\beta\in\mathbb{V}_\alpha^{(1)}\setminus\{\alpha\}} \sigma_{\alpha\beta,i}\ ,\ 1\le i\le 3 \\ \delta_{ij} &= \sum_{j=1}^{3}\sum_{\beta\in\mathbb{V}_\alpha^{(1)}} \sigma_{\alpha\beta,i}h_{\alpha\beta,j}\ ,\ 1\le i,j\le 3 \end{aligned}\right\}. \tag{7.2}$$

The solution to the linear system (7.2) is in general not unique. Refer to [15] for a detailed discussion. Several choices for the values $\sigma_{\alpha\beta}$ satisfying (7.2) are available in practice. Here we retain the popular least squares solution where the values $\boldsymbol{\sigma}_{\alpha\beta}$ are taken as the solution of the problem

$$\min_{\boldsymbol{\sigma}_{\alpha\beta}\in\mathbb{R}^3,\beta\in\mathbb{V}_\alpha^{(1)}} \sum_{\beta\in\mathbb{V}_\alpha^{(1)}} \left(\overline{u}_\beta - \overline{u}_\alpha - \boldsymbol{\sigma}_{\alpha\beta}\cdot\boldsymbol{h}_{\alpha\beta}\right)^2. \tag{7.3}$$

In this case, the values $\boldsymbol{\sigma}_{\alpha\beta}$ of (7.3) are explicitly specified in terms of the matrix H_α (3.4) as

$$\boldsymbol{\sigma}_{\alpha\beta} = \left(H_\alpha^T H_\alpha\right)^{-1}\boldsymbol{h}_{\alpha\beta}\ ,\ \beta\neq\alpha \quad , \quad \boldsymbol{\sigma}_{\alpha\alpha} = -\sum_{\beta\in\mathbb{V}_\alpha^{(1)}\setminus\{\alpha\}} \boldsymbol{\sigma}_{\alpha\beta} \tag{7.4}$$

For simplicity we will use (7.4) throughout the rest of this paper.

Acknowledgement. The first author warmfully acknowledges P. Brenner [10] for the numerous discussions on high order finite volume approximations.

References

[1] R. Abgrall, A. Larat, and M. Ricchiuto. Construction of very high order residual distribution schemes for steady inviscid flow problems on hybrid unstructured meshes. *Jour. Comp. Phys.*, 230(11):4103 – 4136, 2011. Special issue High Order Methods for CFD Problems.

[2] P. Auzillon. *Modeling chemical flame structure and combustion dynamics in large eddy simulation*. Phd thesis, Ecole Centrale Paris, October 2011.

[3] T. Barth and M. Ohlberger. Finite volume methods: Foundation and analysis. In E. Stein, R. de Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*. John Wiley & Sons, 2004.

[4] T. J. Barth and P. O. Frederickson. Higher order solution of the Euler equation on unstructured grids using quadratic reconstruction. In *AIAA 90*, number AIAA-90-0013, pages 1–12, Reno Nevada, January 1990. AIAA.

[10] Airbus Defence and Space SAS, 51-61 Route de Verneuil 78130 Les Mureaux, France.

[5] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *AIAA 89*, number AIAA-89-0366, pages 1–12, Reno Nevada, January 1989. AIAA.

[6] C. Bogey and C. Bailly. A family of low dispersive and low dissipative explicit schemes for flow and noise computations. *Jour. Comp. Phys.*, 194(1):194 – 214, 2004.

[7] B. Cockburn. Devising Discontinuous Galerkin methods for non-linear hyperbolic conservation laws. *Jour. Comp. App. Math.*, 128(1-2):187 – 204, 2001.

[8] B. Cockburn and C.-W. Shu. Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16(3):173–261, 2002.

[9] P. Colella and P. R. Woodward. The piecewise parabolic method (ppm) for gas-dynamical simulations. *Jour. Comp. Phys.*, 54(1):174 – 201, 1984.

[10] F. de la Puente, L. Sanders, and F. Vuillot. On LAGOON nose landing gear CFD/CAA computation over unstructured mesh using a ZDES approach. In *Aviation,20th AIAA/CEAS Aeroacoustics Conference,20th AIAA/CEAS Aeroacoustics Conference, 16-20 June 2014, Atlanta, Ga, USA*.

[11] S. Ducruix, D. Durox, and S. Candel. Theoretical and experimental determinations of the transfer function of a laminar premixed flame. *Proc. of the Comb. Inst.*, 28(1):765 – 773, 2000.

[12] B. Franzelli, E. Riber, L.Y.M. Gicquel, and T. Poinsot. Large eddy simulation of combustion instabilities in a lean partially premixed swirled flame. *Combustion and Flame*, 159(2):621 – 637, 2012.

[13] F. Haider, P. Brenner, B. Courbet, and J-P. Croisille. Efficient implementation of high order reconstruction in finite volume methods. In Federica Corradi Dell'Acqua, Marina Reizakis, Raphaele Herbin, Jan Halama, and Jiří Fürst, editors, *Proc. Sixth Int. Symp. on Finite Volumes for Complex Applications VI (FVCA6)*, volume 1 of *Springer Proceedings in Mathematics*, pages 553 – 560. Springer, 2011.

[14] F. Haider, P. Brenner, B. Courbet, and J.-P. Croisille. Parallel implementation of k-exact finite volume reconstruction on unstructured grids. In Rémi Abgrall, Héloïse Beaugendre, Pietro Marco Congedo, Cécile Dobrzynski, Vincent Perrier, and Mario Ricchiuto, editors, *High Order Nonlinear Numerical Schemes for Evolutionary PDEs*, volume 99 of *Lect. Notes in Comp. Sci. Eng.*, pages 59–75. Springer International Publishing, 2014.

[15] F. Haider, J-P. Croisille, and B. Courbet. Stability analysis of the cell centered finite-volume MUSCL method on unstructured grids. *Numer. Math.*, 113:555 – 600, 2009.

[16] R. E. Harris and Z.J. Wang. High-order adaptive quadrature-free spectral volume method on unstructured grids. *Comp. & Fluids*, 38(10):2006 – 2025, 2009.

[17] R. E. Harris, Z.J. Wang, and Yen Liu. Efficient quadrature-free high-order spectral volume method on unstructured grids: Theory and 2D implementation. *Jour. of Comp. Phys.*, 227(3):1620 – 1642, 2008.

[18] M. Lorteau, F. Cléro, and F. Vuillot. Recent progress in les computation for aeroacoustics of turbulent hot jet. Comparison to experiments and near field analysis. Number AIAA-2014-3057, Atlanta, Ga, USA, June 16-20 2014. AIAA/CEAS Aeroacoustics Conference.

[19] G. Rahier, J. Prieur, F. Vuillot, N. Lupoglazoff, and A. Biancherin. Investigation of integral surface formulations for acoustic post-processing of unsteady aerodynamic jet simulations. *Aerospace Science and Technology*, 8(6):453 – 467, 2004.

[20] M. Ricchiuto and R. Abgrall. Explicit Runge–Kutta residual distribution schemes for time dependent problems: Second order case. *Jour. Comp. Phys.*, 229(16):5653 – 5691, 2010.

[21] P. Sagaut. *Large Eddy Simulation for Incompressible Flows*. Springer, 2nd ed. edition, 2002.

[22] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Jour. Comp. Phys.*, 32(1):101 – 136, 1979.

[23] W.J. Wiscombe and J.W Evans. Exponential-sum fitting of radiative transmission functions. *J. Comput. Phys.*, 24:416–444, 1977.

[24] H.C Yee, N.D Sandham, and M.J Djomehri. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *Jour. Comp. Phys.*, 150(1):199 – 238, 1999.

[25] X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *Jour. Comp. Phys.*, 229(9):3091 – 3120, 2010.

[26] X. Zhang, Y. Xia, and C.-W. Shu. Maximum-principle-satisfying and positivity-preserving high order discontinuous galerkin schemes for conservation laws on triangular meshes. *Jour. of Sci. Comp.*, 50(1):29–62, 2012.