



**HAL**  
open science

## A framework to perform high-order deconvolution for finite-volume method on simplicial meshes

Manuel Bernard, Ghislain Lartigue, Guillaume Balarac, Vincent Moureau,  
Guillaume Puigt

► **To cite this version:**

Manuel Bernard, Ghislain Lartigue, Guillaume Balarac, Vincent Moureau, Guillaume Puigt. A framework to perform high-order deconvolution for finite-volume method on simplicial meshes. International Journal for Numerical Methods in Fluids, 2020, 92 (11), pp.1551-1583. 10.1002/fld.4839 . hal-02558814v2

**HAL Id: hal-02558814**

**<https://hal.science/hal-02558814v2>**

Submitted on 7 Nov 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A framework to perform high-order deconvolution for Finite-Volume Method on simplicial meshes

Bernard M.<sup>1</sup>, Lartigue G. <sup>\*2</sup>, Balarac G.<sup>1,4</sup>, Moureau V.<sup>2</sup>, and Puigt G.<sup>3</sup>

<sup>1</sup>*Univ. Grenoble Alpes, CNRS, Grenoble INP, LEGI, 38000 Grenoble, France*

<sup>2</sup>*Normandie Univ., UNIROUEN, INSA Rouen, CNRS, CORIA, 76000 Rouen, France*

<sup>3</sup>*ONERA/DMPE, Université de Toulouse, F-31055 Toulouse, France*

<sup>4</sup>*Institut Universitaire de France (IUF)*

Article published in International Journal for Numerical Methods in Fluids (IJNMF).

First published: April 06, 2020

DOI: <https://doi.org/10.1002/flid.4839>

This is a postprint: final version which account for reviewers remarks.

Only template is different from published version.

---

\*Corresponding author (ghislain.lartigue@coria.fr).

## Abstract

In this paper, a new framework to design high-order approximations in the context of node-centered Finite Volumes on simplicial meshes is proposed. The major novelty of this method is that it relies on very simple and compact differential operators, which is a critical point to achieve good performances in the High-Performance Computing (HPC) context. This method is based on deconvolution between nodal and volume-average values which can be conducted to any order. The interest of the new method is illustrated through three different applications : mesh-to-mesh interpolation, levelset curvature computation and numerical scheme for convection. Higher-order can also be achieved within the present framework by introducing high-rank tensors. Although these tensors feature many symmetries, their manipulation can quickly become an overwhelming task. For this reason and without loss of generality, the present papers is limited to third-order expansion. This method, although tightly connected to the *k-exact schemes* theory, does not rely on successive corrections: the high-order property is obtained in a single operation, which makes them more attractive in terms of performances.

**Keywords**— Deconvolution, Node-Centered Finite Volume, Simplicial mesh, High-order approximation, Polynomial reconstruction, Compact stencil

# 1 Motivation and objectives

The Finite Volume Method (*FVM*) has become a major tool in CFD in the last decades, mostly because of its unique feature of naturally enforcing exact conservation at the discrete level. In this framework, the quantities of interest are the average of the transported variables over Control Volumes (*CV*). As a consequence, pointwise quantities are not directly available. Traditionally, the *direct* use of the cell-averaged quantities in the discretization leads to a first-order accurate scheme. But such a scheme is not accurate enough for most industrial and research applications. Following the procedure proposed by Van Leer [42], a second-order accuracy can be recovered if the flux integral over the control-volume surface is not directly expressed from the cell-centered data: a piece-wise linear extrapolation, eventually coupled with a slope limiter, is mandatory. Obtaining any other scheme accuracy is possible if the error between the exact solution and the numerical approximation at any point (or integrated over a CV) is of sufficient order. This numerical approximation can be estimated by combining and integrating several Taylor expansions.

This error estimation is illustrated hereafter by considering the transport equation of a scalar  $\phi$  at a constant velocity  $\mathbf{u}$  in conservative form:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}) = 0. \quad (1)$$

The finite-volume formulation of the problem is obtained by integrating this partial derivative equation (*PDE*) over a generic control-volume  $\Omega_I$  and by commuting the volume averaging and time differentiation operators (only time-independent meshes are considered here):

$$\frac{d\bar{\phi}^{\Omega_I}}{dt} + \frac{1}{V_I} \int_{\Omega_I} \nabla \cdot (\phi \mathbf{u}) dV = 0 \quad (2)$$

The measure of the control volume is defined as:

$$V_I = \int_{\Omega_I} dV, \quad (3)$$

which represents either a volume in 3D, an area in 2D or a length in 1D. The control-volume averaging operator is then:

$$\bar{\phi}^{\Omega_I} = \frac{1}{V_I} \int_{\Omega_I} \phi dV \quad (4)$$

The divergence of the convective flux in Eq. 2 may be rewritten as an integral over the control-volume boundary  $\partial\Omega_I$  thanks to the Gauss theorem:

$$\frac{d\bar{\phi}^{\Omega_I}}{dt} + \frac{1}{V_I} \int_{\partial\Omega_I} \phi \mathbf{u} \cdot d\mathbf{S} = 0 \quad (5)$$

The exact value of the flux on the boundary  $\partial\Omega_I$  is then estimated numerically and the quality of this approximation depends on the choice of the numerical scheme. The important fact here is that Eq. 5 is an evolution equation for the control-volume average  $\bar{\phi}^{\Omega_I}$  rather than for the nodal value  $\phi_I$  or for any other pointwise value. However, higher-order schemes require higher-order surface integrals than need either higher-order quadrature rules or moments. In any case, pointwise values of  $\phi$  or moments based on Taylor series expansions must be defined. For example, the second-order integral approximation only needs the evaluation of  $\phi$  at the face center. Even this very simple formulation highlights the necessity to recover pointwise values of  $\phi$  from the integrated quantities  $\bar{\phi}^{\Omega_I}$ . This process is known as *deconvolution*.

Deconvolution is an important step in building of high-order Finite Volume approximations. With the standard  $k$ -exact high-order reconstruction, the computation of the polynomial coefficients for approximating the unknowns is possible through the introduction of an extended stencil: the stencil must contain enough control volumes to recover an invertible system where polynomial coefficients are the unknowns. This process is not new: introduced by Barth & Frederickson [6], it was for instance considered by Abgrall [1], Ollivier-Gooch *et al.* [30, 32, 31, 40], Mandal *et al.* [26], ... Generally, the stencil used to compute the scheme contains more control volumes than the required number of polynomial coefficients and the computation of the polynomial coefficients is performed by means of a least-squares method over a larger stencil. Recently, Sejkan *et al.* [39] proposed an improvement of the least-squares method based on the  $H^1$  projection instead of the  $L^2$  projection in order to derive an accurate gradient usable for the viscous flux computation. The Monotone Optimal Order-Detection (MOOD) technique [9, 13, 14] follows the same principle for defining local polynomial approximation and mostly differs with  $k$ -exact approximation by the polynomial basis.

The definition of a large stencil in a massively parallel framework is the main bottleneck to maintain high performance and an alternative approach is the Successive-Correction  $k$ -exact approximation. Introduced by Brenner, Haider, Courbet, Croisille and co-authors for cell-centered finite-volume discretization in several papers [20, 21, 22, 35, 36, 23, 28], the procedure only needs the smallest possible stencil (local cell and direct neighbors by the faces). Starting from the definition of a discrete gradient operator, the procedure consists in analysing the truncation error of the gradient in order to correct it and to increase its accuracy. Then, the discrete gradient operator is applied to the gradient in order to obtain second-order derivatives. Here again, an analysis of the truncation error of the second-order derivatives enables to define a more accurate polynomial representation of the unknowns. In practice, the stencil is reduced to the minimum but the number of fields to exchange in a parallel environment increases. To the author's knowledge, such a correction procedure was never applied to node-centered formulations.

The Compact Least-Squares Finite-Volume (CLSFV) method [43, 44, 45, 46] gives a framework to define a high-order polynomial of the unknown but it differs with  $k$ -exact and MOOD methods by the procedure to build the approximation. Here, starting from the smallest stencil (current cell and its direct neighbors by faces), it is required that the cell averages of the reconstruction polynomial and its various orders of spatial derivatives are conserved on the stencil. This leads to inverting a linear system of equations.

In the context of cell-centered formulations, Betchen and Straatman [7] define a new reconstruction method to deal with accurate gradient and Hessian. Their idea is simple: a third-order accurate representation of unknowns needs a second-order accurate gradient and a first-order accurate Hessian. To obtain these operators, Betchen and Straatman write a system of equations valid over the whole computational mesh and invert it using an under-relaxed block-Jacobi method. Such a method does not cope with our requirements for HPC since inverting the system cannot be performed locally, by a cell-wise formulation.

The aim of this work is to present a framework that can provide a third-order polynomial reconstruction of a field from CV-averaged quantities on any simplicial mesh. From this polynomial reconstruction, it is then straightforward to find accurate approximations of the field, but also of its gradients and its Hessian. It will be shown that it is possible to achieve a second-order accuracy on the field by only considering the direct neighbors of the mesh

vertices. Moreover, with this same connectivity, it is possible to obtain a third-order accuracy in most situations, with the noticeable exception of boundaries. Obtaining third-order accuracy with such a compact stencil is clearly an advantage in the context of HPC applications, where parallel performance is an issue. Moreover, it is possible to extend the present method to recover third-order accuracy near the interfaces. Finally, the extension of the present method to higher-order reconstructions is straightforward from a theoretical point of view. However, the computational cost and the mathematical complexity associated to higher-order methods can become difficult to handle and for this reason, only second- and third-order methods will be presented here.

Another important issue that will be addressed in this work is the quality of the approximations of the classical FV discrete operators on a generic (i.e. not regular) mesh. We will show that in node-centered codes, most discrete gradient operators are not consistent or first-order only in the best case. Similarly, most discrete Hessian operators will usually diverge when the mesh is refined. The present framework proposes to overcome these limitations by exhibiting discrete operators that provide a second-order accuracy on gradient and a first-order accuracy for Hessian on any mesh.

The applications of this deconvolution technique for Node-Centered Finite-Volumes codes are numerous and only a few of them will be presented here to illustrate the ability of this framework to achieve high-order accuracy in any situation:

- High-precision interpolation
- Curvature computation of a levelset function for two phase flows simulations
- High-order numerical schemes for convection.

## 2 A framework for deconvolution of node-centered FVM

### 2.1 Taylor series expansion and grid moments

Assuming a sufficiently smooth scalar field, any value  $\phi(\mathbf{x})$  can be approximated anywhere in the domain by performing a Taylor series expansion around any node  $I$ :

$$\phi(\mathbf{x}) = \phi_I + (\nabla\phi_I) \cdot \Delta_I(\mathbf{x}) + \frac{1}{2} (\nabla\nabla\phi_I) : (\Delta_I(\mathbf{x}) \otimes \Delta_I(\mathbf{x})) + \mathcal{O}(\Delta^3) \quad (6)$$

where  $\Delta_I(\mathbf{x}) = \mathbf{x} - \mathbf{x}_I$ . The  $\otimes$  and  $\nabla$  symbols have their usual meaning: i.e. the tensor product and the gradient operator, respectively. Thus  $\nabla\phi_I$  is the gradient of  $\phi$  estimated at  $\mathbf{x}_I$  and  $\nabla\nabla\phi_I$  is the Hessian of  $\phi$ , also estimated at  $\mathbf{x}_I$ .

In the context of the FVM, the averaging operator defined in Eq. 4 is then applied to this expression to provide a first relation between the nodal and the averaged value of  $\phi$ :

$$\overline{\phi}^{\Omega_I} = \phi_I + (\nabla\phi_I) \cdot \overline{\Delta_I}^{\Omega_I} + \frac{1}{2} (\nabla\nabla\phi_I) : \left( \overline{\Delta_I \otimes \Delta_I}^{\Omega_I} \right) + \mathcal{O}(\Delta^3) \quad (7)$$

where

$$\overline{\Delta_I}^{\Omega_I} = \frac{1}{V_I} \int_{\Omega_I} (\mathbf{x} - \mathbf{x}_I) dV = \mathcal{O}(\Delta) \quad (8)$$

$$\overline{\Delta_I \otimes \Delta_I}^{\Omega_I} = \frac{1}{V_I} \int_{\Omega_I} (\mathbf{x} - \mathbf{x}_I) \otimes (\mathbf{x} - \mathbf{x}_I) dV = \mathcal{O}(\Delta^2) \quad (9)$$

are the first- and second-order moments of the control volume, respectively. These quantities are of key importance in the present method and thus deserve a few comments. First, it can be noticed that  $\overline{\Delta_I}^{\Omega_I}$  is simply the vector between the vertex  $I$  and the mass center of the control-volume. It is thus zero when  $I$  is the center of the control volume. Similarly, the  $\overline{\Delta_I \otimes \Delta_I}^{\Omega_I}$  term is a symmetric rank 2 tensor which is never zero: its eigenvalues are the moments of inertia of the control volume with respect to  $I$ . Then, it should be emphasized that in Eq. 7, both the Taylor expansion and the averaging operation are performed around the same position  $I$ . This is indeed a very special case and by no means a requirement: for example, one can also define  $\overline{\Delta_I}^{\Omega_J}$  by integrating the function  $\Delta_I(\mathbf{x})$  over a different control volume  $\Omega_J$ .

As already mentioned, it is straightforward to extend Eq. 7 to higher order by defining tensors  $\overline{\Delta_I^{\otimes 3}}^{\Omega_J}$ ,  $\overline{\Delta_I^{\otimes 4}}^{\Omega_J}$ , and so on. Of course, the amount of data associated to these tensors increases substantially for higher orders but it can also be noted that these tensors are geometric quantities which can be pre-computed and stored once and for all: the CPU cost associated to their evaluation is thus not an issue, provided that they can be stored in memory. Moreover, these high-rank tensors have the maximal symmetry property: the amount of data which must be stored to describe them remains tractable for practical applications.

Some technical details presenting the computation of these moments are gathered in Appendix A.

## 2.2 Control-Volumes, mesh quality and discrete differential operators

A simple inspection of Eq. 7 shows that the evaluation of the discrete differential operators at node  $I$  and the choice of the control-volume around this node will have a major impact on the deconvolution procedure. The aim of this section is to clarify these two points.

In the context of the Node-Centered approach, the CV are the cells of the dual mesh. Several methods have been described extensively in the literature to build a dual mesh from the primal mesh (barycentric, Voronoï, ...) [29] so this choice must be clearly stated. Here, the barycentric method is used, but the results can be reformulated for other dual meshes without any major difficulty. The case of non-simplicial meshes will not be discussed here as it is always possible to decompose any element into a finite number of simplices in a pre-processing step: the proposed framework can thus be applied to any grid, provided that such a decomposition has been applied beforehand.

Once the primal and dual meshes have been given, it is then possible to introduce the concept of mesh quality: for a given number of elements, a regular mesh will usually provide a better accuracy than a grid containing highly distorted elements. This concept is detailed in Appendix B where some notations and some objective criteria to quantify the lack of regularity of a given grid are introduced.

Many discretizations of the usual differential operators have been proposed in the context of FVM [4, 34, 24]. Among them, the following pair-based formulations for the nodal gradient and Hessian are easily derived by applying the Gauss Theorem to the dual mesh and are often used in CFD codes, either in the simple form presented hereafter or in more complex forms to define the interface quantities [37]. Indeed, the Gauss theorem needs the definition of

an interface quantity that is assumed here to be an average of left and right contributions:

$$\mathcal{G}_I(\phi) = \frac{1}{V_I} \sum_{J \in \mathcal{N}_I} \frac{\phi_I + \phi_J}{2} \times \mathbf{S}_{IJ} \quad (10)$$

or an interface gradient that is approximated by a standard finite difference:

$$\mathcal{H}_I(\phi) = \frac{1}{V_I} \sum_{J \in \mathcal{N}_I} \left( \frac{\phi_J - \phi_I}{\|\mathbf{IJ}\|} \right) \times \frac{1}{2} (\mathbf{n}_{IJ} \otimes \mathbf{S}_{IJ} + \mathbf{S}_{IJ} \otimes \mathbf{n}_{IJ}) \quad (11)$$

where the notations refer to Fig. 1 and where  $\mathcal{N}_I$  is the set of vertices connected to  $I$  by an edge of the mesh, i.e. the direct neighbors of  $I$ . Both formulations can be found in [5], except that we have slightly modified Eq. 11 to ensure the mixed partial derivatives theorem. By noting  $\mathcal{V}$  the set of all vertices of the mesh,  $\phi = (\phi_I)_{I \in \mathcal{V}}$  is a vector containing the values evaluated at each node of the mesh. This quantity can either be pointwise (e.g.  $\phi = (\phi_I)_{I \in \mathcal{V}}$  or  $\Delta_J = (\Delta_J(\mathbf{x}_I))_{I \in \mathcal{V}}$ ) or an average value (e.g.  $\bar{\phi}^\Omega = (\bar{\phi}^{\Omega_I})_{I \in \mathcal{V}}$  or  $\bar{\Delta}_J^\Omega = (\bar{\Delta}_J^{\Omega_I})_{I \in \mathcal{V}}$ ). A special attention must be paid when dealing with moments such as  $\Delta_J$  and  $\bar{\Delta}_J^\Omega$ : the index  $J$  refers to the center of the function and must be kept constant in the gradient/Hessian operators. For the sake of simplicity, everything is presented here for a scalar  $\phi$  but the method can be extended trivially to any tensor.

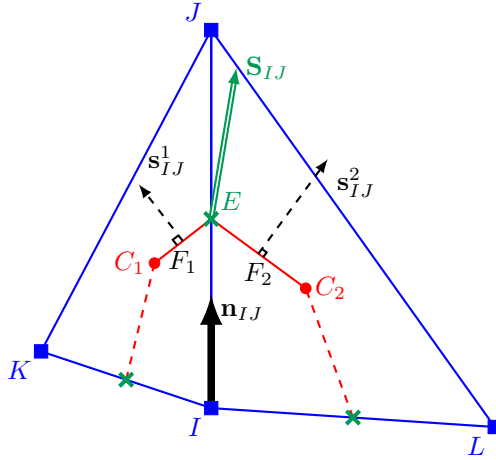


Figure 1: Notations for a 2D control-volume based on the barycentric dual mesh. Here,  $\mathcal{N}_I = \{J, K, L\}$ . Considering the boundary face  $\partial\Omega_{IJ}$  between nodes  $I$  and  $J$  (red line), the full face vector  $\mathbf{S}_{IJ}$  is the sum of the two facelets surface vectors  $\mathbf{s}_{IJ}^1$  and  $\mathbf{s}_{IJ}^2$ . In 3D, the situation is similar but the number of facelets based on an edge is not fixed (generally 5 to 8 facelets by edge). The vector  $\mathbf{n}_{IJ}$  represents the unit vector direction on edge  $IJ$ .

The discrete nodal gradient of Eq. 10 can achieve second-order accuracy on regular meshes, but is only first-order on generic meshes as will be demonstrated below. It can also be highlighted that the adaptation of Eq. 10 to the cell-centered Finite-Volume approach is not straightforward: the mesh interface being generally not located exactly between cell centers, the interface quantity cannot be defined by a simple averaging procedure [41]. Moreover, the approximation of the interface gradient cannot be considered in standard cell-centered Finite-Volume approach since the line passing between cell centers does not generally cross the interface in its center. Similarly, it will also be shown that the approximation of the Hessian given in Eq. 11 usually does not converge on distorted meshes. As a consequence, these operators can not be used in Eq. 7 to achieve a third-order reconstruction of the solution on



distorted meshes. This clearly highlights the necessity to improve the calculation of these two differential operators to perform the high-order deconvolution of the average solution.

### 2.3 Third-order deconvolution procedure

In this section, we detail the methodology that enables to express high-order approximations of  $\phi_I$ ,  $\nabla\phi_I$  and  $\nabla\nabla\phi_I$  from  $\bar{\phi}^\Omega$ . To achieve this goal, some discrete operators for the nodal gradient  $\mathcal{G}_I$  and the nodal Hessian  $\mathcal{H}_I$  must already be available, even if they are not high-order. Their exact expression is actually not very important and we will thus consider generic linear operators written as:

$$\mathcal{G}_I\left(\bar{\phi}^\Omega\right) = \sum_{J \in \mathcal{N}_I^*} \bar{\phi}^{\Omega_J} \mathbf{P}_{IJ}^{\textcircled{1}} \quad (12)$$

$$\mathcal{H}_I\left(\bar{\phi}^\Omega\right) = \sum_{J \in \mathcal{N}_I^*} \bar{\phi}^{\Omega_J} \mathbf{P}_{IJ}^{\textcircled{2}} \quad (13)$$

where  $\mathcal{N}_I^* = \mathcal{N}_I \cup \{I\}$  is the set of direct neighbors of  $I$  augmented with  $I$  itself and  $\mathbf{P}_{IJ}^{\textcircled{1}}$  and  $\mathbf{P}_{IJ}^{\textcircled{2}}$  are generic rank 1 and rank 2 tensors. One can notice that both operators are compact, in the sense that they only imply the direct neighbors of a given node. In the present paper, we used the approximations presented in Eq. 10 and Eq. 11 but other choices would have provided similar results.

To be consistent, it is important that the discrete Hessian satisfies the Schwartz theorem on mixed partial derivatives which simply requires that  $\mathbf{P}_{IJ}^{\textcircled{2}}$  should be symmetrical. Moreover, in what follows, we will assume that:

$$\mathbf{P}^{\textcircled{n}} = \mathcal{O}\left(\Delta^{-n}\right) \quad \text{for } n \in \{1, 2\} \quad (14)$$

which is rather natural to ensure the operators consistency. Additional consistency conditions will be discussed thoroughly hereafter.

Applying the volume averaging, discrete gradient and discrete Hessian operators to the Taylor expansion of  $\phi$  (Eq. 6), and using the linearity of these operators, leads to:

$$\begin{aligned} \bar{\phi}^{\Omega_I} &= \phi_I \bar{1}^{\Omega_I} + (\nabla\phi_I) \cdot \overline{\Delta_I}^{\Omega_I} \\ &+ \frac{1}{2} (\nabla\nabla\phi_I) : \left(\overline{\Delta_I \otimes \Delta_I}^{\Omega_I}\right) + \mathcal{O}\left(\Delta^3\right), \end{aligned} \quad (15)$$

$$\begin{aligned} \mathcal{G}_I\left(\bar{\phi}^\Omega\right) &= \phi_I \mathcal{G}_I\left(\bar{1}^\Omega\right) + (\nabla\phi_I) \cdot \mathcal{G}_I\left(\overline{\Delta_I}^\Omega\right) \\ &+ \frac{1}{2} (\nabla\nabla\phi_I) : \mathcal{G}_I\left(\overline{\Delta_I \otimes \Delta_I}^\Omega\right) + \mathcal{O}\left(\Delta^2\right), \end{aligned} \quad (16)$$

$$\begin{aligned} \mathcal{H}_I\left(\bar{\phi}^\Omega\right) &= \phi_I \mathcal{H}_I\left(\bar{1}^\Omega\right) + (\nabla\phi_I) \cdot \mathcal{H}_I\left(\overline{\Delta_I}^\Omega\right) \\ &+ \frac{1}{2} (\nabla\nabla\phi_I) : \mathcal{H}_I\left(\overline{\Delta_I \otimes \Delta_I}^\Omega\right) + \mathcal{O}\left(\Delta\right), \end{aligned} \quad (17)$$

It can be seen that some new high-rank tensors are involved in Eq. 16 and Eq. 17. These tensors are built by applying the discrete gradient and Hessian operators to the vectors  $\overline{\Delta_I}^\Omega$  and  $\overline{\Delta_I \otimes \Delta_I}^\Omega$ . From a practical point of view, and thanks to Eq. 12 and Eq. 13, it is sufficient to evaluate  $\overline{\Delta_I}^{\Omega_J}$  and  $\overline{\Delta_I \otimes \Delta_I}^{\Omega_J}$  for  $J \in \mathcal{N}_I^*$ , and then to compute some linear combination of these quantities at each node of the mesh.

Eq. 15, Eq. 16 and Eq. 17 can be rewritten all together as:

$$\mathbf{L}_I = \mathbf{C}_I^{\mathcal{O}3} \cdot \mathbf{H}_I + \mathbf{E} \quad (18)$$

where:

$$\mathbf{L}_I = \begin{bmatrix} \overline{\phi}^{\Omega_I} \\ \mathcal{G}_I(\overline{\phi}^{\Omega}) \\ \mathcal{H}_I(\overline{\phi}^{\Omega}) \end{bmatrix}, \quad \mathbf{H}_I = \begin{bmatrix} \phi_I \\ \nabla \phi_I \\ \nabla \nabla \phi_I \end{bmatrix} \quad \text{and} \quad \mathbf{E} = \begin{bmatrix} \mathcal{O}(\Delta^3) \\ \mathcal{O}(\Delta^2) \\ \mathcal{O}(\Delta) \end{bmatrix} \quad (19)$$

are the vectors of low-order approximations, exact nodal values and errors respectively. The 3<sup>rd</sup>-order convolution matrix  $\mathbf{C}_I^{\mathcal{O}3}$  is then:

$$\mathbf{C}_I^{\mathcal{O}3} = \begin{bmatrix} \overline{1}^{\Omega_I} & \overline{\Delta}_I^{\Omega_I} & \frac{1}{2} \overline{\Delta}_I \otimes \overline{\Delta}_I^{\Omega_I} \\ \mathcal{G}_I(\overline{1}^{\Omega}) & \mathcal{G}_I(\overline{\Delta}_I^{\Omega}) & \frac{1}{2} \mathcal{G}_I(\overline{\Delta}_I \otimes \overline{\Delta}_I^{\Omega}) \\ \mathcal{H}_I(\overline{1}^{\Omega}) & \mathcal{H}_I(\overline{\Delta}_I^{\Omega}) & \frac{1}{2} \mathcal{H}_I(\overline{\Delta}_I \otimes \overline{\Delta}_I^{\Omega}) \end{bmatrix} \quad (20)$$

The previous notations rely on a consistent unfolding of the various high-rank tensors that appear in Eq. 19 and Eq. 20 to form a classical linear system. This unfolding is actually trivial and will not be commented any further here, except for the fact that it must take into account the various symmetries of the tensors to ensure that the matrix  $\mathbf{C}_I^{\mathcal{O}3}$  has the maximal-rank property.

This matrix is of key importance in the present work and deserves some additional comments. First it must be noted that the ideal situation is the case where  $\mathbf{C}_I^{\mathcal{O}3} = \mathbf{I}$ : in this case the discrete and continuous variables are identical up to the theoretical order indicated by the vector  $\mathbf{E}$ . In this situation, no deconvolution is needed as the discrete operators can be used directly to perform a polynomial representation of the solution in a given control volume. The main point of the present article is to realize that such an ideal situation is never achieved, even on regular meshes. Moreover, on distorted meshes, this matrix can depart greatly from identity and the discrepancy between  $\mathbf{L}$  and  $\mathbf{H}$  can become very important. In any case, one has to solve Eq. 18 to obtain high-order approximations of  $\mathbf{H}$  from the available data, i.e.  $\mathbf{L}$ .

Considering a single line of the matrix, it can be noticed that the consistency between the discrete and the continuous part is achieved when the elements located at the left of the diagonal of  $\mathbf{C}_I^{\mathcal{O}3}$  are zero and when the diagonal term is one. Assuming that these consistency constraints are fulfilled, the order of the approximation is then given by the position of the first non-zero term on the right of the diagonal.

Several comments can be formulated at this point. First, from the definition of the averaging operator,  $\overline{1}^{\Omega_I} = 1$ , and thus  $\overline{\phi}^{\Omega_I}$  is always a consistent approximation of  $\phi_I$ . Another desirable (but not mandatory) property is that the discrete gradient and Hessian satisfy the following constraints:

$$\mathcal{G}_I(\overline{1}^{\Omega}) = 0 \quad \text{and} \quad \mathcal{H}_I(\overline{1}^{\Omega}) = 0 \quad (21)$$

This is for example the case for operators given in Eq. 10 and Eq. 11. These conditions state that the discrete gradient and Hessian operators applied to a constant field must be zero, which are rather natural requirements. Moreover these two requirements imply a very nice property: the reconstructed gradient and hessian are then decoupled from the reconstructed nodal value. As a consequence, it is sufficient to consider a reduced linear system, i.e. without the first line and column of the convolution matrix, to perform the numerical reconstruction procedure. Finally, the nodal value is obtained in a post-processing step to ensure that the reconstruction conserves exactly

the average value. In what follows, the full system is considered for the sake of clarity but numerically, only the reduced linear system is inverted, following the procedure that is described below.

The important special case of regular meshes is now considered. In the present paper, a mesh is said to be regular around vertex  $I$  if the condition  $\overline{\Delta}_J^{\Omega_J} = 0$  holds for all  $J \in \mathcal{N}_I^*$ . In this situation it can be shown that for the discrete operators considered in Eq. 10 and Eq. 11, many terms cancel in the deconvolution matrix and then:

$$\mathcal{H}_I \left( \overline{\phi}^{\Omega} \right) = \nabla \nabla \phi_I + \mathcal{O}(\Delta^h) \quad (22)$$

$$\mathcal{G}_I \left( \overline{\phi}^{\Omega} \right) = \nabla \phi_I + \mathcal{O}(\Delta^2) \quad (23)$$

$$\overline{\phi}^{\Omega_I} = \phi_I + \frac{1}{2} \mathcal{H}_I \left( \overline{\phi}^{\Omega} \right) : \left( \overline{\Delta}_I \otimes \overline{\Delta}_I^{\Omega_I} \right) \quad (24)$$

where  $h = 2$  in 1D and  $h = 0$  otherwise. Equation 24 shows that  $\overline{\phi}^{\Omega_I}$  is a second-order approximation of  $\phi_I$  in the case of regular meshes.

On the other hand, it can also be shown that if the mesh is not regular those discrete operators are usually not consistent. More specifically, in this situation,  $\mathcal{G}_I \left( \overline{\Delta}_I^{\Omega} \right) \neq \mathbf{0}^{\text{②}}$  and  $\mathcal{H}_I \left( \overline{\Delta}_I^{\Omega} \right) \neq \mathbf{0}^{\text{③}}$  and finally  $\overline{\phi}^{\Omega_I}$  is only a first-order approximation of  $\phi_I$ .

It is now possible to define the following new operators, simply by inverting the system defined in Eq. 18, provided that the matrix  $\mathbf{C}_I^{\mathcal{O}3}$  has the full rank property:

$$\begin{bmatrix} \phi_I^{\mathcal{O}3} \\ \mathbf{G}_I^{\mathcal{O}2} \\ \mathbf{H}_I^{\mathcal{O}1} \end{bmatrix} = (\mathbf{C}_I^{\mathcal{O}3})^{-1} \cdot \begin{bmatrix} \overline{\phi}^{\Omega_I} \\ \mathcal{G}_I \left( \overline{\phi}^{\Omega} \right) \\ \mathcal{H}_I \left( \overline{\phi}^{\Omega} \right) \end{bmatrix} \quad (25)$$

where  $\phi_I^{\mathcal{O}3}$  is a third-order approximation of  $\phi_I$ ,  $\mathbf{G}_I^{\mathcal{O}2}$  a second-order approximation of  $\nabla \phi_I$  and  $\mathbf{H}_I^{\mathcal{O}1}$  a first-order approximation of  $\nabla \nabla \phi_I$ , respectively.

Of course it is possible to downgrade the quality of this deconvolution to obtain a second-order approximation of  $\phi$  and a first-order approximation  $\nabla \phi$ :

$$\begin{bmatrix} \phi_I^{\mathcal{O}2} \\ \mathbf{G}_I^{\mathcal{O}1} \end{bmatrix} = (\mathbf{C}_I^{\mathcal{O}2})^{-1} \cdot \begin{bmatrix} \overline{\phi}^{\Omega_I} \\ \mathcal{G}_I \left( \overline{\phi}^{\Omega} \right) \end{bmatrix} \quad (26)$$

where the second-order convolution matrix  $\mathbf{C}_I^{\mathcal{O}2}$  is simply:

$$\mathbf{C}_I^{\mathcal{O}2} = \begin{bmatrix} \overline{\mathbf{1}}^{\Omega_I} & \overline{\Delta}_I^{\Omega_I} \\ \mathcal{G}_I \left( \overline{\mathbf{1}}^{\Omega} \right) & \mathcal{G}_I \left( \overline{\Delta}_I^{\Omega} \right) \end{bmatrix} \quad (27)$$

As already shown above, the first-order convolution matrix  $\mathbf{C}_I^{\mathcal{O}1}$  simply reduces to the scalar 1 and thus:

$$\phi_I^{\mathcal{O}1} = \overline{\phi}^{\Omega_I} \quad (28)$$

is a first-order approximation on a generic mesh.

To conclude, if the convolution matrix  $\mathbf{C}_I^{\mathcal{O}3}$  is invertible, the method described above provides a third-order polynomial expansion of the solution around every CV of the computational domain:

$$\phi_I^{\mathcal{O}3}(\mathbf{x}) = \phi_I^{\mathcal{O}3} + \mathbf{G}_I^{\mathcal{O}2} \cdot \Delta_I(\mathbf{x}) + \frac{1}{2} \mathbf{H}_I^{\mathcal{O}1} : (\Delta_I(\mathbf{x}) \otimes \Delta_I(\mathbf{x})) = \phi_I + \mathcal{O}(\Delta^3) . \quad (29)$$

It must be noted that this procedure can be carried out analytically in some simple situations: the corresponding results are given in Appendix C.

## 2.4 Invertibility of the convolution matrix

As already mentioned, the convolution matrix  $C_I$  and the deconvolution matrix  $C_I^{-1}$  only depend on geometrical quantities and can thus be computed and stored once for all in a pre-processing step. Their computation is thus transparent from a performance point of view. The only additional cost associated to the deconvolution is thus a matrix-vector product of very limited size, which can be done efficiently on all modern processors.

The main advantage of the present method should now be clear: building a third-order approximation of the solution even on very distorted meshes only requires to compute two compact operators and to invert a small matrix on each node of the mesh. There is no need to build long-distance stencils, which can be both difficult to implement and numerically inefficient for massively parallel solvers. On the other hand, each node must have enough neighbors so that the convolution matrix is invertible. Indeed, the connectivity of a given node needs to be at least equal to the dimension of the convolution matrix (see Tab. 1 that summarizes the size of the various convolution matrices).

For example, the third-order deconvolution matrix in 3D has 10 independent lines and columns: 1 for the field, 3 for the gradient and 6 for the Hessian (symmetric  $3 \times 3$  matrix). It means that the set  $\mathcal{N}_I^*$  must contain 10 elements, *i.e.* each node must be connected to 9 neighbors.

If this condition is not respected (for example, the nodes lying on the boundaries usually do not have that many neighbors), several solutions can be proposed:

- it is always possible to perform a  $2^{nd}$ -order deconvolution at the concerned node: on a simplicial mesh of dimension  $d$ , the cells have  $d + 1$  vertices and each vertex is connected to the other  $d$  vertices which are thus direct neighbors. This implies a local loss of accuracy, but this is usually not a problem since it concerns a limited number of nodes. Moreover, most numerical methods have an order decrease near the boundaries due to the necessary upwinding.
- another possibility is to use the polynomial reconstruction performed on some neighboring nodes, if they have themselves a sufficient connectivity.
- a last technique could consist in blending low-order local reconstructions into a higher-order one, as performed in the  $P^n - P^m$  method (see [16] as the first reference on this technique).

The reason why the present framework can only be applied to node-centered Finite Volumes should now be clear: in the cell-centered method, the number of direct neighbors is simply the number of faces of the simplex which is always equal to  $d + 1$ . As indicated by Tab. 1, this connectivity only allows to achieve a second-order

dimension ( $d$ )	1D	2D	3D
$C_I^{\mathcal{O}^1}$	1	1	1
$C_I^{\mathcal{O}^2}$	2	3	4
$C_I^{\mathcal{O}^3}$	3	6	10
$C_I^{\mathcal{O}^n}$	$n$	$\frac{n(n+1)}{2}$	$\frac{n(n+1)(n+2)}{6}$

Table 1: Size of the convolution matrices in 1D, 2D and 3D (taking the various symmetries into account).

accuracy on the scalar field.

## 2.5 Link with the Least-Squares approach

It must be pointed out that there are many similarities between our approach and the Weighted Least-Squares method proposed by Ollivier-Gooch [32]. Indeed, let us introduce the matrix of the moments around node  $I$  :

$$\mathbf{R}_I = \begin{bmatrix} \overline{1}^{\Omega_I} & \overline{\Delta}_I^{\Omega_I} & \frac{1}{2} \overline{\Delta}_I \otimes \overline{\Delta}_I^{\Omega_I} \\ \overline{1}^{\Omega_{J_1}} & \overline{\Delta}_I^{\Omega_{J_1}} & \frac{1}{2} \overline{\Delta}_I \otimes \overline{\Delta}_I^{\Omega_{J_1}} \\ \vdots & \vdots & \vdots \\ \overline{1}^{\Omega_{J_N}} & \overline{\Delta}_I^{\Omega_{J_N}} & \frac{1}{2} \overline{\Delta}_I \otimes \overline{\Delta}_I^{\Omega_{J_N}} \end{bmatrix} \quad (30)$$

It is a matrix of dimension  $(N + 1) \times M$  where  $N$  stands for the number of neighbours of node  $I$  and  $M$  is the size of the convolution matrix as detailed in Tab. 1. This matrix can be used to write the node-averaged Taylor expansion around the node  $I$ :

$$\begin{bmatrix} \overline{\phi}^{\Omega_I} \\ \overline{\phi}^{\Omega_{J_1}} \\ \vdots \\ \overline{\phi}^{\Omega_{J_N}} \end{bmatrix} = \mathbf{R}_I \begin{bmatrix} \phi_I \\ \nabla \phi_I \\ \nabla \nabla \phi_I \end{bmatrix} + \mathcal{O}(\Delta^3) \quad (31)$$

We also introduce the low-order operator matrix (notations are taken from Eq. 12 and Eq. 13:

$$\mathbf{P}_I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \mathbf{P}_{II}^{\textcircled{1}} & \mathbf{P}_{IJ_1}^{\textcircled{1}} & \dots & \mathbf{P}_{IJ_N}^{\textcircled{1}} \\ \mathbf{P}_{II}^{\textcircled{2}} & \mathbf{P}_{IJ_1}^{\textcircled{2}} & \dots & \mathbf{P}_{IJ_N}^{\textcircled{2}} \end{bmatrix} \quad (32)$$

which is used to build the low order operators from neighbouring nodes as:

$$\begin{bmatrix} \overline{\phi}^{\Omega_I} \\ \mathcal{G}_I \left( \overline{\phi}^{\Omega} \right) \\ \mathcal{H}_I \left( \overline{\phi}^{\Omega} \right) \end{bmatrix} = \mathbf{P}_I \begin{bmatrix} \overline{\phi}^{\Omega_I} \\ \overline{\phi}^{\Omega_{J_1}} \\ \vdots \\ \overline{\phi}^{\Omega_{J_N}} \end{bmatrix} \quad (33)$$

With these notations, our convolution matrix is simply:

$$\mathbf{C}_I = \mathbf{P}_I \mathbf{R}_I \quad (34)$$

while the matrix that arises in the Weighted Least-Squares formalism is:

$$\mathbf{S}_I = \mathbf{R}_I^T \Lambda_I \mathbf{R}_I \quad (35)$$

where  $\Lambda_I$  is a diagonal matrix introduced by Barth [5] to lessen the weight of farthest nodes in the reconstruction.

There is thus a clear link between the two approaches: the initial statement (Eq. 31) is identical but the way to solve this over-determined linear system is different. In our framework, we state that:

$$\begin{bmatrix} \phi_I \\ \nabla \phi_I \\ \nabla \nabla \phi_I \end{bmatrix} = (\mathbf{P}_I \mathbf{R})^{-1} \mathbf{P}_I \begin{bmatrix} \overline{\phi}^{\Omega_I} \\ \overline{\phi}^{\Omega_{J_1}} \\ \vdots \\ \overline{\phi}^{\Omega_{J_N}} \end{bmatrix} \quad (36)$$

while the LS formalism solves for:

$$\begin{bmatrix} \phi_I \\ \nabla \phi_I \\ \nabla \nabla \phi_I \end{bmatrix} = (\mathbf{R}_I^T \Lambda_I \mathbf{R}_I)^{-1} \mathbf{R}_I^T \begin{bmatrix} \overline{\phi}^{\Omega_I} \\ \overline{\phi}^{\Omega_{J_1}} \\ \vdots \\ \overline{\phi}^{\Omega_{J_N}} \end{bmatrix} \quad (37)$$

A major difference between the two frameworks is the condition number of the matrices  $\mathbf{C}_I$  and  $\mathbf{S}_I$  that need to be inverted to build the reconstruction. By a simple dimensional analysis, it can be seen that the condition number of our convolution matrix  $\mathbf{C}_I$  is dimensionless: it only depends on the mesh quality but not on the mesh size. This is not the case of the  $\mathbf{S}_I$  matrix: its eigenvalues have physical dimensions  $1, \Delta^2, \Delta^4, \dots$ . This point is of paramount importance from a numerical point of view and, to the authors' understanding, it is a clear advantage of the present method over the classical Least-Square approach.

As an illustration, the Fig. 2 presents the condition number of the convolution matrix for various 1D, 2D and 3D meshes (with very different physical sizes and with large amount of perturbation). It clearly appears that this condition number is better on regular meshes than on distorted meshes but is always of the order of unity even on quite irregular grids.

Both regular and distorted meshes are used to evaluate the performance of the method: those meshes are identified thanks to the notation  $\mathcal{M}_b^d$ , where  $d$  is the dimension of the problem and  $b$  is a parameter that indicates the perturbation of the grid, as explained in Appendix B ( $b = 0$  indicates a regular mesh and the more  $b$  increases, the more the mesh is deformed).

In both 2D and 3D, some meshes have also been generated by using dedicated commercial software. Those meshes are identified thanks to the notation  $\mathcal{M}_{soft}^d$ .

## 2.6 Extension to higher orders

First, the extension of Eq. 6 to the order  $k$  is straightforward. Then, some discrete differentials operators also up to order  $k$  must be available. Those operators can be obtained by composing the low-order first and second derivatives as many times as needed, for example:

$$\mathcal{D}^{(\text{odd})} = \mathcal{G} \circ \mathcal{H} \circ \dots \circ \mathcal{H} \quad (38)$$

$$\mathcal{D}^{(\text{even})} = \mathcal{H} \circ \mathcal{H} \circ \dots \circ \mathcal{H} \quad (39)$$

Of course, this implies that larger and larger stencils will be involved but this is anyway needed to ensure the invertibility of the convolution matrix. Then, one has to build and to invert the convolution matrix  $\mathbf{C}_I^{\mathcal{O}^k}$ . For this, the grid moments up to  $\overline{\Delta_I^{\otimes k-1} \Omega_J}$  must also be pre-computed and the previously introduced differential operators must be applied to them. When  $k$  is greater than 3, even if still feasible, this can become challenging from a computing point of view, as it implies to deal with large amount of data for each vertex of the mesh. Moreover, achieving third-order accuracy is usually sufficient for many practical applications.

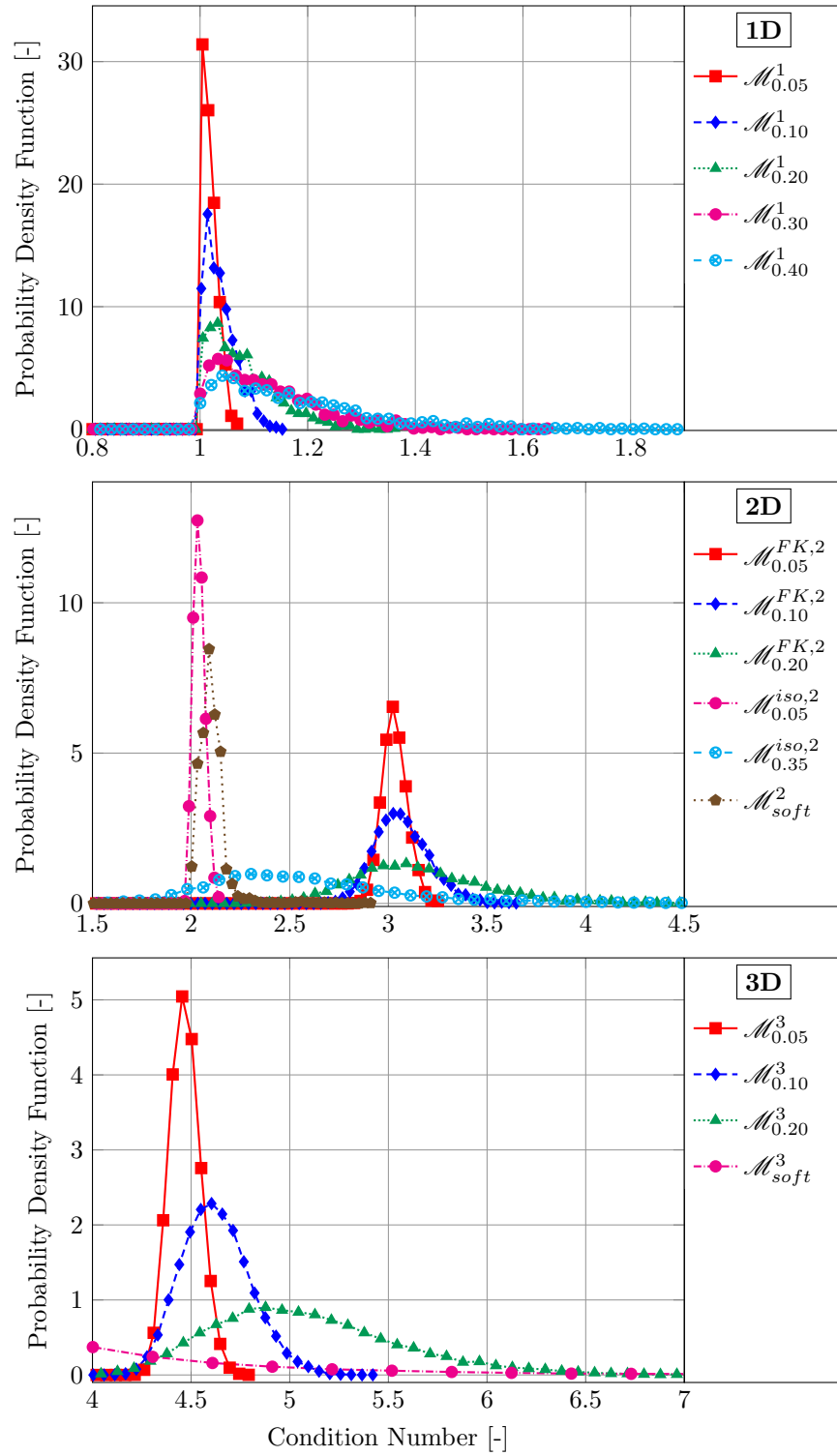


Figure 2: Probability Density Function of the condition number of the convolution matrix for various meshes

### 3 Verification and applications

#### 3.1 Numerical verification of the accuracy of the deconvolution

The present section describes the tests that are used to assess the accuracy of the deconvolution procedure described above. In these tests, a 1D bump function is used, *i.e.* a compactly supported  $C^\infty$  function, which depends only on  $x$ :

$$\phi(x) = \begin{cases} \exp\left(\frac{1}{R_0^2}\right) \exp\left(\frac{1}{x^2 - R_0^2}\right) & \text{if } |x| < R_0 \\ 0 & \text{else} \end{cases} \quad (40)$$

In all presented results, the domain length is  $L_x = 1$  and  $R_0 = 0.4$ . The convergence order is studied by varying the number of cells  $N_x$  in the principal direction. In 2D and in 3D, the domain width/height is always set to 30 cells to keep a reasonable CPU cost on the finest meshes: this is not an issue since the solution mostly remains 1D. Moreover, additional tests with full 2D and 3D solutions have not shown any noticeable difference.

To apply the deconvolution procedure, the average value of the scalar  $\bar{\phi}^{\Omega_I}$  over each CV of the mesh must be available at the initial time:

- A first possibility to obtain this average field is to compute it analytically by integrating the bump function on each CV. However, depending on the signal expression, this can be cumbersome, especially in 2D and 3D.
- Another option is to compute this field from Eq. 15: this requires to know the analytical expression of  $\phi$ ,  $\nabla\phi$  and  $\nabla\nabla\phi$  and the first and second order moments. This method will provide a third-order accurate approximation of the analytical average value and is used in the present situation.
- In the general case, when the nodal data is known but not its successive derivatives, it is not a trivial task to compute the nodal gradient up to the second order and the nodal Hessian up to the first order, to ensure a third order reconstruction of the average data.

In this last situation, it is still possible to apply the discrete gradient and Hessian operators to Eq. 6 at point  $I$ :

$$\mathcal{G}_I(\phi) = \nabla\phi_I \cdot \mathcal{G}_I(\Delta_I) + \nabla\nabla\phi_I : \mathcal{G}_I(\Delta_I \otimes \Delta_I) + \mathcal{O}(\Delta^2) \quad (41)$$

$$\mathcal{H}_I(\phi) = \nabla\phi_I \cdot \mathcal{H}_I(\Delta_I) + \nabla\nabla\phi_I : \mathcal{H}_I(\Delta_I \otimes \Delta_I) + \mathcal{O}(\Delta) \quad (42)$$

For the previous equations to hold, the assumptions  $\mathcal{G}_I(1) = 0$  and  $\mathcal{H}_I(1) = 0$  must be verified (Eq. 21) but this is not an issue as already explained. Following the same path as for the averaged quantities (see Eq. 25), this can be re-written as a linear system:

$$\begin{bmatrix} \mathbf{G}_I^{\mathcal{O}2} \\ \mathbf{H}_I^{\mathcal{O}1} \end{bmatrix} = \left(\widetilde{\mathbf{C}}_I^{\mathcal{O}3}\right)^{-1} \cdot \begin{bmatrix} \mathcal{G}_I(\phi) \\ \mathcal{H}_I(\phi) \end{bmatrix} \quad (43)$$

where the pointwise convolution matrix  $\widetilde{\mathbf{C}}_I^{\mathcal{O}3}$

$$\widetilde{\mathbf{C}}_I^{\mathcal{O}3} = \begin{bmatrix} \mathcal{G}_I(\Delta_I) & \frac{1}{2}\mathcal{G}_I(\Delta_I \otimes \Delta_I) \\ \mathcal{H}_I(\Delta_I) & \frac{1}{2}\mathcal{H}_I(\Delta_I \otimes \Delta_I) \end{bmatrix} \quad (44)$$

does not require the knowledge of the CV moments but only evaluation of the function  $\Delta_I(\mathbf{x}) = \mathbf{x} - \mathbf{x}_I$  for all  $J \in \mathcal{N}_I^*$ . The evaluation of this pointwise convolution matrix  $\widetilde{\mathbf{C}}_I^{\mathcal{O}3}$  is actually very similar to the evaluation of  $\mathbf{C}_I^{\mathcal{O}3}$  and will not be commented any further.



Once  $\bar{\phi}^{\Omega_I}$  is available (by any of the aforementioned method), the discrete differential operators Eq. 10 and Eq. 11 are applied to this averaged field and then the high-order approximations  $\phi_I^{\mathcal{O}3}$ ,  $\mathbf{G}_I^{\mathcal{O}2}$  and  $\mathbf{H}_I^{\mathcal{O}1}$  are obtained by solving linear system of Eq. 25.

The accuracy of the resulting quantities is then evaluated by computing  $L^1$ -,  $L^2$ - and  $L^\infty$ -norms of errors between the theoretical values and the approximations. As expected, the convergence order is independent of the specific norm chosen to measure the error. Thus, for the sake of conciseness, only results corresponding to  $L^2$ -norm are represented hereafter. For example, the error on the deconvoluted scalar field reads:

$$\|\mathcal{E}(\phi)\|_2 = \sqrt{\frac{1}{V_\Omega} \sum_{I=1}^{N_\Omega} (\phi_I^{\text{ex}} - \phi_I^{\mathcal{O}3})^2 V_I} \quad (45)$$

where  $N_\Omega$  is the number of nodes in the domain  $\Omega$ . For the differential operators, the  $L^2$ -norm of error is evaluated for each component and then the Euclidean norm and the Frobenius norm are computed for gradient and Hessian, respectively.

Fig. 3a shows that the classical Hessian operator converges in 1D whereas it is either  $\mathcal{O}(1)$  on regular meshes or even  $\mathcal{O}(\Delta^{-1})$  on distorted 2D and 3D meshes. More precisely:

- the error on the diagonal terms is  $\mathcal{O}(\Delta^2)$  on regular meshes and  $\mathcal{O}(\Delta)$  in the general case.
- the error on the off-diagonal terms is  $\mathcal{O}(1)$  on regular meshes and  $\mathcal{O}(\Delta^{-1})$  in the general case.

That explains the specific behavior observed on the 1D results, as there is no off-diagonal terms in this situation. Regarding the deconvoluted Hessian, Fig. 3b shows that it is  $\mathcal{O}(\Delta^2)$  on regular meshes and  $\mathcal{O}(\Delta)$  on distorted meshes, whatever the problem dimension.

Concerning the approximation of the gradient, Fig. 4 shows that, even if the classical operator is  $\mathcal{O}(\Delta^2)$  on regular meshes, its convergence decreases to  $\mathcal{O}(\Delta)$  in the general case. Here again, the deconvoluted gradient remains  $\mathcal{O}(\Delta^2)$ , no matter the grid quality.

Finally, as illustrated in Fig. 5, the averaged scalar  $\bar{\phi}^{\Omega_I}$  approximates the pointwise value up to  $\mathcal{O}(\Delta^2)$  on regular meshes and  $\mathcal{O}(\Delta^1)$  on distorted meshes. By using the new framework, the pointwise reconstruction of the scalar is  $\mathcal{O}(\Delta^4)$  and  $\mathcal{O}(\Delta^3)$  on regular and distorted meshes respectively.

### 3.2 Application to high-order interpolation

The availability of a high-order polynomial reconstruction (Eq. 29) offers the possibility to perform an accurate interpolation of a field at any location in the domain. It permits for example to interpolate precisely any fluid properties from an Eulerian phase towards Lagrangian particles. Another obvious application of this high-order deconvolution framework is to perform accurate mesh-to-mesh interpolation in the context of dynamic mesh adaptation [2] for example. This last point is of particular importance: dynamic mesh adaptation implies a very frequent remeshing and any spurious numerical noise can accumulate over time and produce unexpected discrepancies if low-accuracy interpolation is used. In the context of dynamic mesh adaptation, a conservative and monotonicity preserving interpolation was shown to overcome standard interpolation technique [3] but is still limited to second-order of accuracy

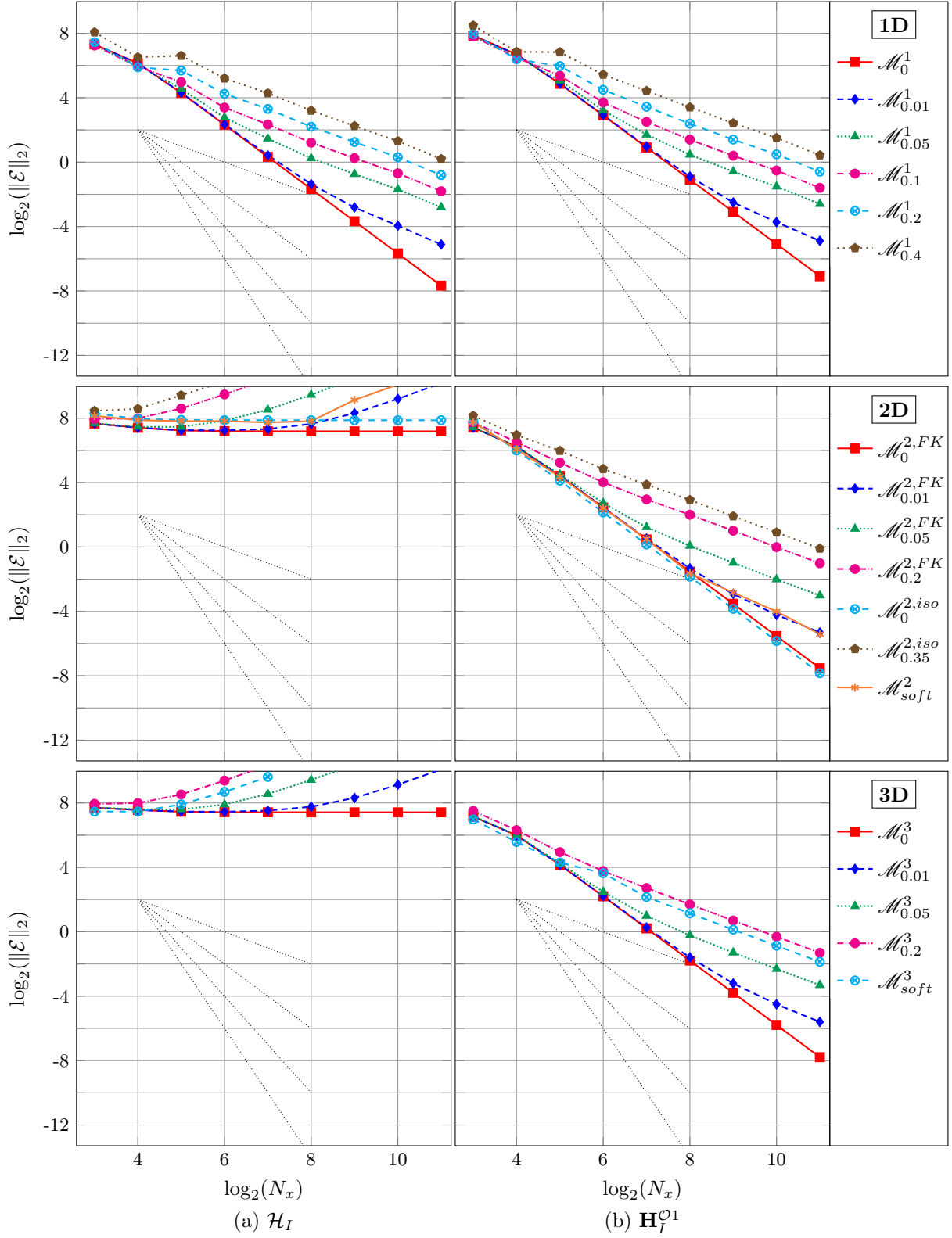


Figure 3: Errors on Hessian for different mesh quality, without deconvolution (a) and with deconvolution (b). Meshes quality notations are detailed in Appendix B. Black dotted lines correspond to convergence order 1, 2, 3 and 4 respectively.

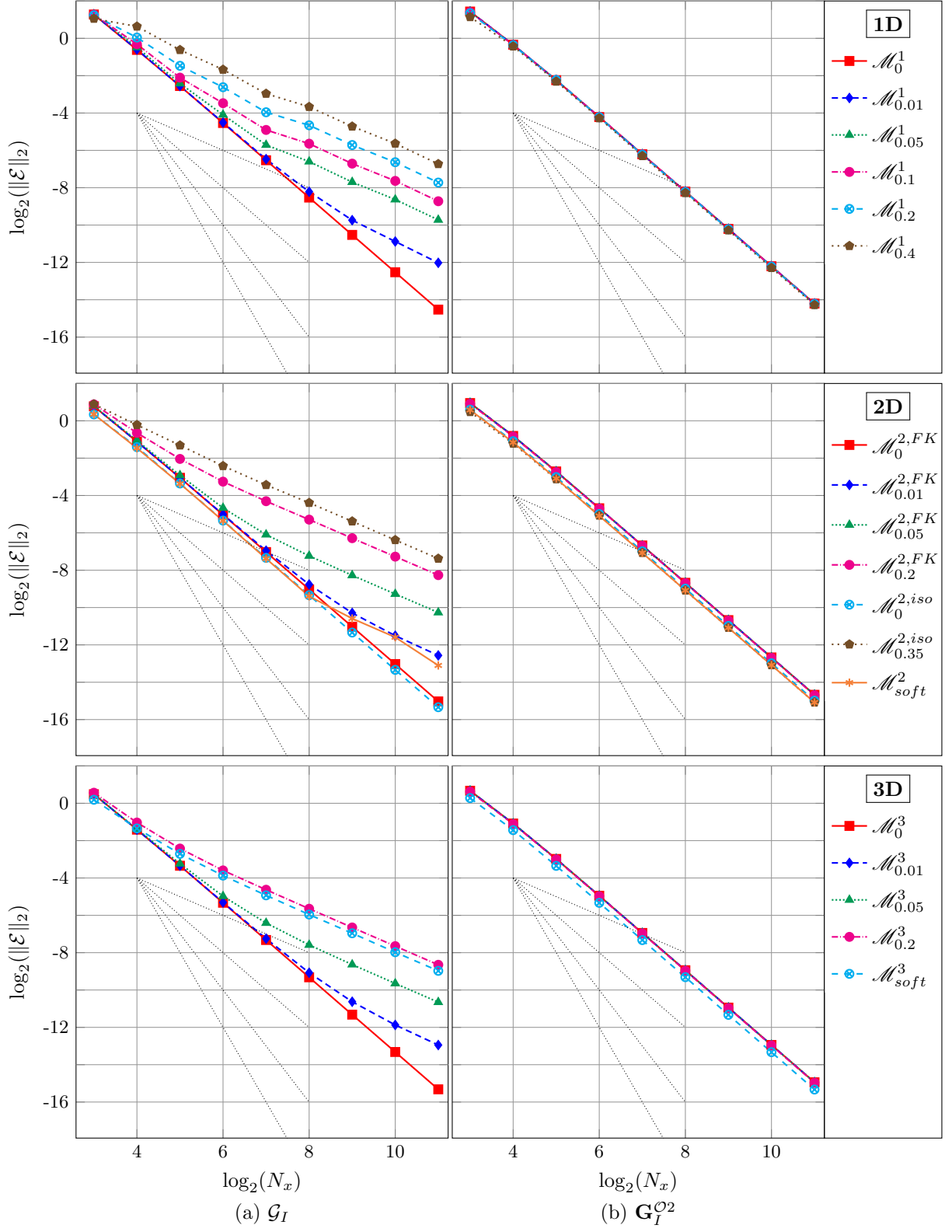


Figure 4: Errors on gradient for different mesh quality, without deconvolution (a) and with deconvolution (b). Meshes quality notations are detailed in Appendix B. Black dotted lines correspond to convergence order 1, 2, 3 and 4 respectively.

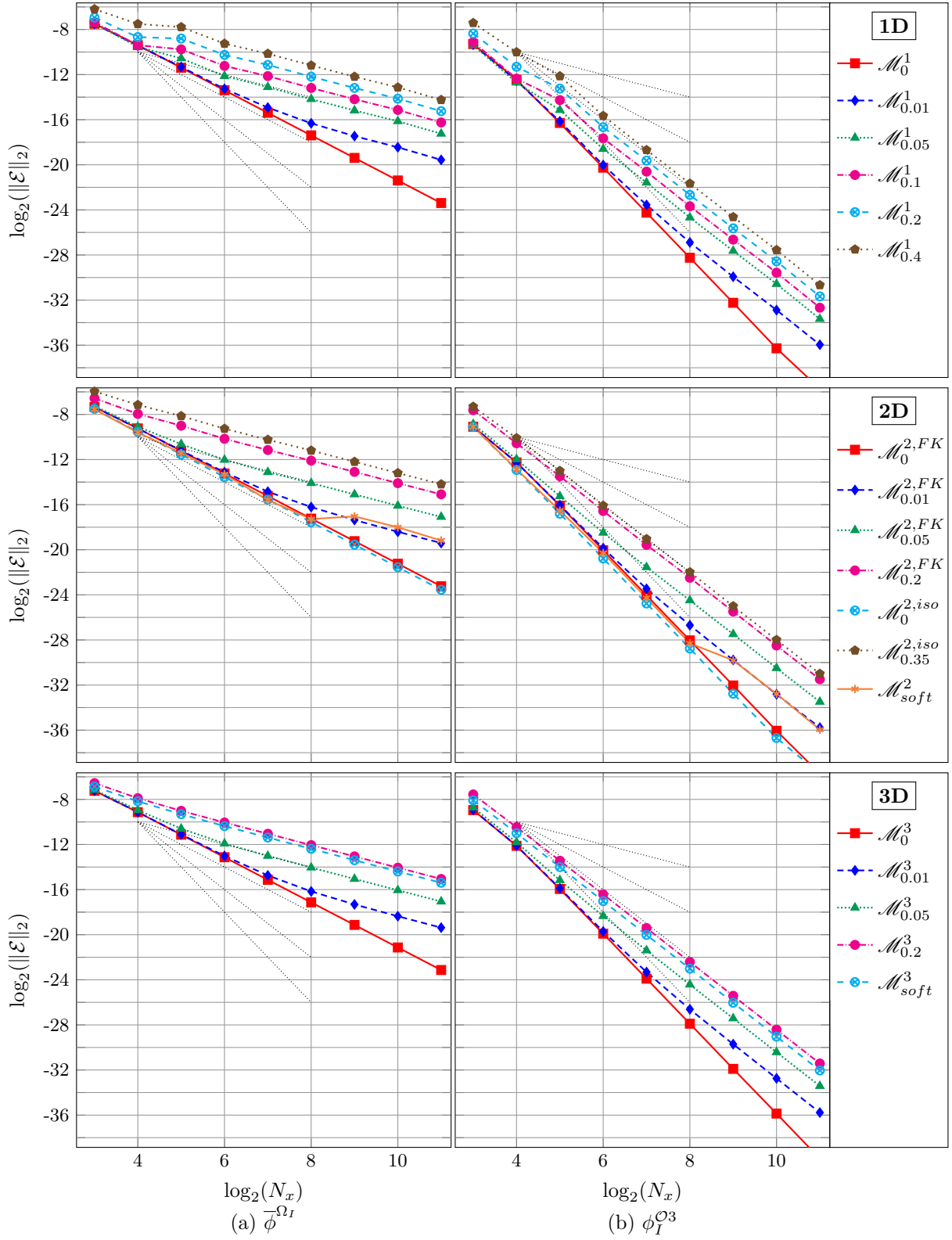


Figure 5: Errors on scalar for different mesh quality, without deconvolution (a) and with deconvolution (b). Meshes quality notations are detailed in Appendix B. Black dotted lines correspond to convergence order 1, 2, 3 and 4 respectively.

and work is still needed to consider higher accuracy. For completeness, it must also be mentioned that an alternative to interpolation with polynomials is interpolation using the Radial Basis Functions. The literature is large but recent applications concern the treatment of non conforming grid interface for high-order finite-volume schemes [25] or the coupling computations using non conforming grids and potentially different discretization techniques [17].

The interest of the proposed framework is illustrated by applying the high-order interpolation method proposed by D el eze *et al.* [11, 10]. This method is an extension of the Hermite interpolation method to 2D and 3D simplices which relies on the construction of a local finite-element basis in each cell of the mesh. This leads to interpolation functions which are  $C^1$ -differentiable everywhere, and  $C^2$ -differentiable almost everywhere and that can achieve third-order convergence order. It is particularly interesting as it ensures the continuity of the fluid properties (and of their first derivative) when Lagrangian particles cross an element boundary. To achieve these desirable properties, the D el eze method requires to have the value of the field and its gradient at each vertex of the mesh with respectively 3<sup>rd</sup> and 2<sup>nd</sup> order accuracy.

To illustrate this application, the bump function presented in section 3.1 is interpolated from a source mesh consisting of a manufactured distorted grid using perturbation parameter  $b = 0.2$  (see Appendix B) towards a regular destination grid, *i.e.*  $b = 0$ . Both grids contain the same number of nodes (which do not coincide) and the convergence order is studied by refining the source and the destination grid simultaneously.

Fig. 6 shows the  $L^2$ -norm of the error on the solution obtained on the destination grid by the interpolation method proposed by D el eze. This figure shows three different datasets:

- The curve labeled *analytical* is obtained by using the analytical values for the data and its gradient in the formula of D el eze. As expected, a 3<sup>rd</sup> order accuracy is obtained.
- The *low-order* curve corresponds to the results obtained when replacing the analytical values of  $\phi_I$  and  $\nabla\phi_I$  by the averaged data  $\bar{\phi}^{\Omega_I}$  and its low-order gradient  $\mathcal{G}_I(\bar{\phi}^{\Omega})$ . It can be seen than in this situation, the method degenerates to 1<sup>st</sup>-order.
- Finally, the *high-order* set is obtained by using the deconvoluted values  $\phi_I^{\mathcal{O}3}$  and  $\mathbf{G}_I^{\mathcal{O}2}$ : in this situation, the 3<sup>rd</sup>-order theoretical convergence is recovered, with however a lower precision than the first method which uses the analytical values.

### 3.3 Application to levelset curvature computation

In many multiphase applications, the interface is treated by the mean of the levelset method [33]: in this case, an accurate evaluation of geometrical quantities is requested to ensure the consistency of the simulation. For example, computing the pressure drop between the two phases with Young-Laplace equation requires to evaluate the curvature of the interface. However, computing a consistent curvature on a distorted mesh is a common issue in multiphase flows [12, 8, 27].

In what follows, the levelset function  $\phi$  can represent either the hyperbolic tangent or the signed distance function depending on the considered levelset method. In both cases, the normal to the interface is defined as:

$$\mathbf{n} = \frac{\nabla\phi}{\|\nabla\phi\|}. \quad (46)$$

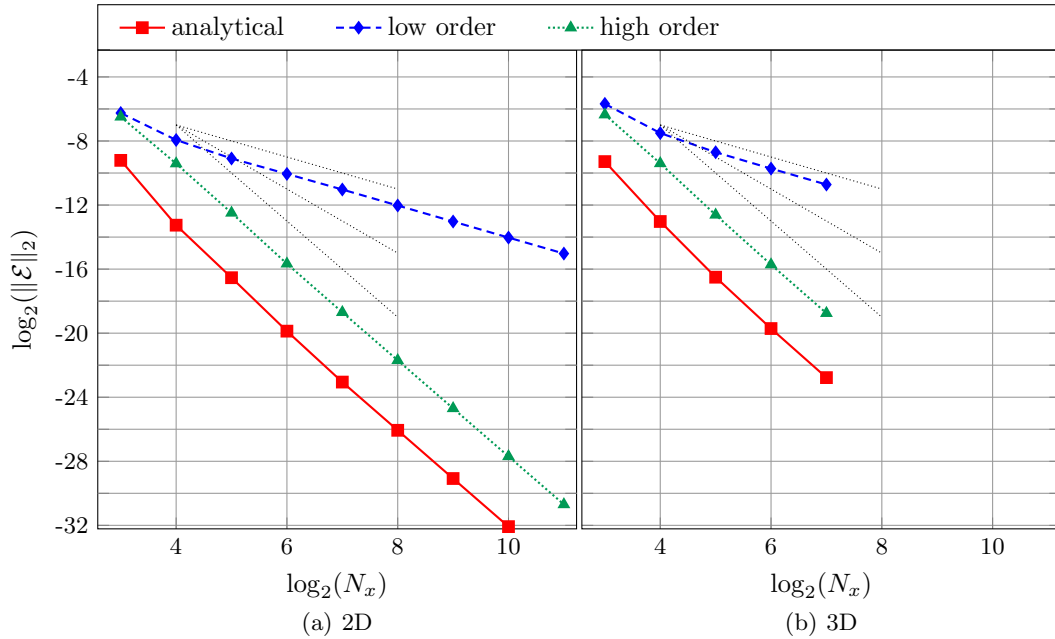


Figure 6: Relative  $L^2$ -norm of error on interpolation by using Déléze methodology presented in section 3.2. Black dotted lines correspond to convergence order 1, 2 and 3 respectively.

and the the total curvature of the interface, *i.e.* the sum of the main curvatures, reads:

$$\mathcal{K} = -\nabla \cdot \mathbf{n}. \quad (47)$$

As summarized by Goldman [18], various formulations are available to evaluate numerically this curvature. All these formulations are equivalent when considering continuous differential operators but can show significant differences when evaluated numerically. In this paper we focus on two formulations.

The first one is a two-step method, based on the preliminary computation of the normal following Eq. 46. Then, its divergence is computed as the trace of the gradient of the normal:

$$\mathcal{K}_{nc} = -\text{tr}(\nabla \mathbf{n}) = -\text{tr} \left( \nabla \frac{\nabla \phi}{\|\nabla \phi\|} \right) \quad (48)$$

This method requires two applications of the gradient operator to the levelset function and is thus non-compact (as indicated by the subscript  $nc$ ). The second formulation is expressed as a function of the gradient and the Hessian of the levelset function.

$$\mathcal{K}_c = \frac{\nabla \nabla \phi}{\|\nabla \phi\|} : \left( \frac{\nabla \phi \otimes \nabla \phi}{\|\nabla \phi\|^2} - \mathbb{I}^{(2)} \right) \quad (49)$$

which requires to evaluate the discrete gradient and Hessian to compute the curvature but only implies the direct neighbors: this method is thus named compact, as indicated by the subscript  $c$ .

Let us point out that with both formulations, due to the discrete operators definition, the curvature is known at the mesh vertices. On the other hand, most numerical methods that solve the Laplace equation on the pressure need to evaluate the pressure drop at interface position, which generally does not coincide to a node position.

In the case of classical FVM framework, regardless the formulation employed (Eq. 48 or Eq. 49), the curvature

is linearly interpolated at interface location  $\Gamma$  with the usual formula:

$$\mathcal{K}_\Gamma = \theta_{IJ}\mathcal{K}_I + (1 - \theta_{IJ})\mathcal{K}_J, \quad (50)$$

where:

$$\theta_{IJ} = \frac{\|\Delta_I(\mathbf{x}_\Gamma)\|}{\|\Delta_I(\mathbf{x}_J)\|}, \quad (51)$$

In this last equation,  $\Delta_I(\mathbf{x}_\Gamma)$  is the vector between node  $I$  and interface  $\Gamma$  and  $\Delta_I(\mathbf{x}_J)$  is the vector between node  $I$  and node  $J$ .

When using the new high-order FVM framework, the deconvoluted gradient and Hessian are first evaluated on both sides of each edge  $[IJ]$  that crosses the interface, *i.e.* at node  $I$  and  $J$ , with Eq. 25. Then their values at the interface position  $\Gamma$  are computed by performing a Taylor series expansions with respect to  $I$  and  $J$ :

$$\begin{cases} \mathbf{G}_{\Gamma,I}^{\mathcal{O}2} = \mathbf{G}_I^{\mathcal{O}2} + \Delta_I(\mathbf{x}_\Gamma) \cdot \mathbf{H}_I^{\mathcal{O}1} + \mathcal{O}(\Delta^2) \\ \mathbf{G}_{\Gamma,J}^{\mathcal{O}2} = \mathbf{G}_J^{\mathcal{O}2} + \Delta_J(\mathbf{x}_\Gamma) \cdot \mathbf{H}_J^{\mathcal{O}1} + \mathcal{O}(\Delta^2) \end{cases} \quad (52a)$$

$$\quad (52b)$$

$$\begin{cases} \mathbf{H}_{\Gamma,I}^{\mathcal{O}1} = \mathbf{H}_I^{\mathcal{O}1} + \mathcal{O}(\Delta) \\ \mathbf{H}_{\Gamma,J}^{\mathcal{O}1} = \mathbf{H}_J^{\mathcal{O}1} + \mathcal{O}(\Delta) \end{cases} \quad (53a)$$

$$\quad (53b)$$

Finally, the values  $\mathbf{G}_\Gamma^{\mathcal{O}2}$  and  $\mathbf{H}_\Gamma^{\mathcal{O}1}$  that are used in Eq. 48 or in Eq. 49 to evaluate the curvature at the interface are computed with the following linear interpolation:

$$\begin{cases} \mathbf{G}_\Gamma^{\mathcal{O}2} = \theta_{IJ}\mathbf{G}_{\Gamma,I}^{\mathcal{O}2} + (1 - \theta_{IJ})\mathbf{G}_{\Gamma,J}^{\mathcal{O}2} \\ \mathbf{H}_\Gamma^{\mathcal{O}1} = \theta_{IJ}\mathbf{H}_{\Gamma,I}^{\mathcal{O}1} + (1 - \theta_{IJ})\mathbf{H}_{\Gamma,J}^{\mathcal{O}1} \end{cases} \quad (54a)$$

$$\quad (54b)$$

To illustrate the accuracy of both formulations, the curvature of a spherical bubble of radius  $R$  is computed for various mesh resolutions both in 2D and in 3D, for regular ( $b = 0$ ) and distorted ( $b = 0.2$ ) grids. Note that the coarser mesh  $N_x = 8$  corresponds to  $R \simeq 2.66 \Delta_x$ . In each situation, the error is computed with the classical and the deconvoluted discrete operators presented in section 2 and the results are gathered in Fig. 7a and Fig. 7b. More precisely, these figures present the  $L^2$ -norm of the relative error on curvature which is evaluated by comparing the discrete value along each edge to the analytical result  $\mathcal{K}^0 = (d - 1)/R$ :

$$\|\mathcal{E}(\mathcal{K})\|_2 = \sqrt{\frac{1}{N_e} \sum_{e=1}^{N_e} \left( \frac{\mathcal{K}_e - \mathcal{K}^0}{\mathcal{K}^0} \right)^2} \quad (55)$$

where  $N_e$  is the number of edges of the mesh that intersect the interface.

As illustrated in Fig. 7a, the non-deconvoluted differential operators can provide a  $\mathcal{O}(\Delta^2)$  convergence order on regular meshes when combined with the divergence formula (Eq. 47). But when the same methodology is applied to a distorted grid, the numerical evaluation of the curvature fails to converge and is  $\mathcal{O}(1)$ . The compact formulation is even worse: it only provides an  $\mathcal{O}(1)$  error on regular grids and a  $\mathcal{O}(\Delta^{-1})$  error on distorted meshes (the error increases as the mesh is refined).

Alternatively, the Fig. 7b shows that, when using the new deconvoluted operators, both formulations lead to an  $\mathcal{O}(\Delta^2)$  convergence on regular meshes and an  $\mathcal{O}(\Delta)$  convergence on distorted grids.

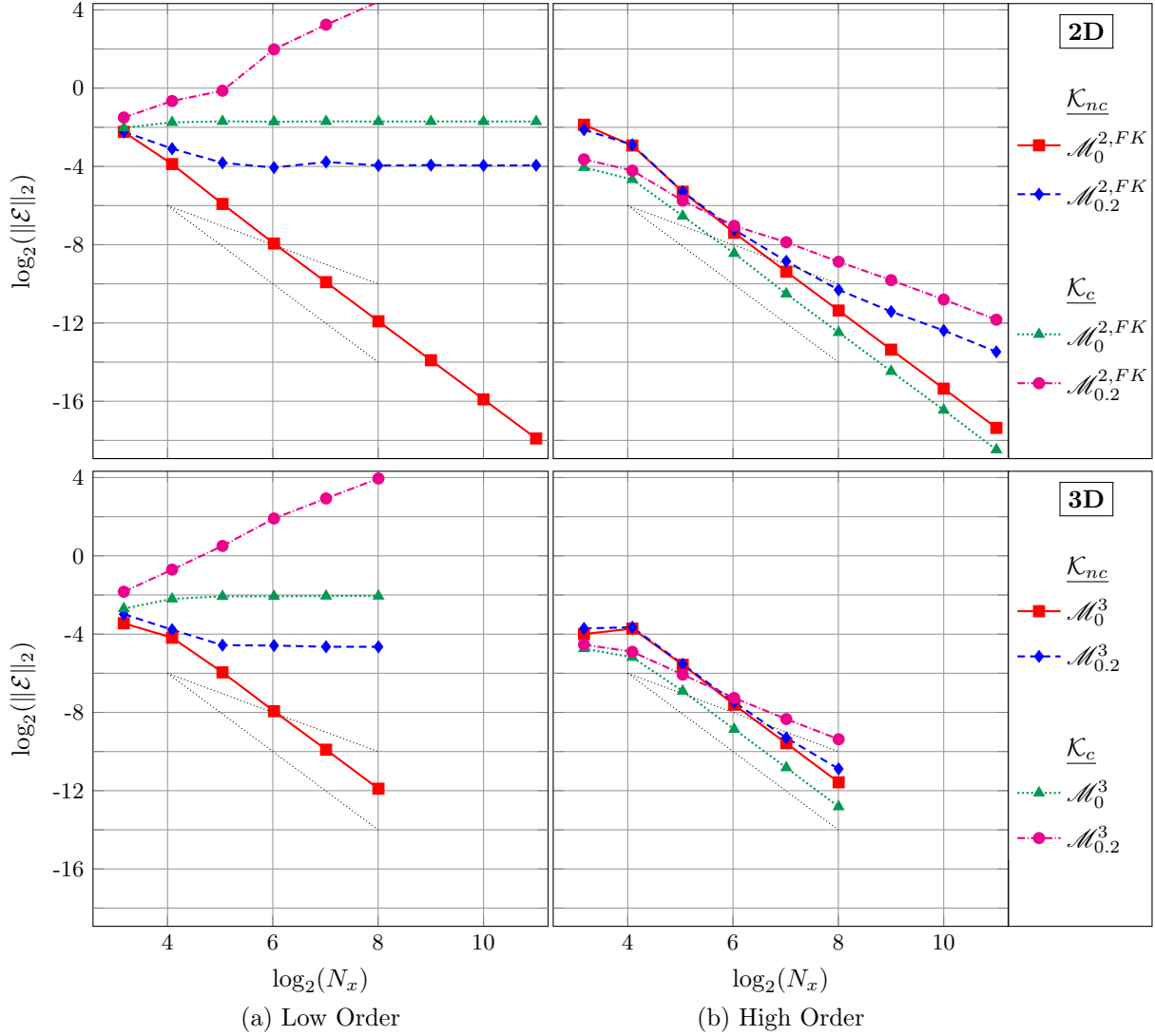


Figure 7: Errors on curvature computation using both formulations presented in section 3.3. Black dotted lines correspond to convergence order 1, 2.



### 3.4 Application to flux computation for convection problems

Thanks to the deconvolution procedure exposed in section 2.3, it is possible to compute a high-order approximation of the solution everywhere in a given CV and more particularly at its boundaries. The objective of this section is to show that it is possible to use this polynomial expansion to compute the convection fluxes on the CV boundaries. This is simply achieved by performing the exact integration of this polynomial over the boundary faces. This can be done either by a Gauss quadrature or by using the surface moments on boundary faces. Both methods give the same result: the later method is less common but more straightforward here as it involves moments, just like the deconvolution technique.

Starting from Eq. 5, we are looking for an accurate evaluation of the flux  $\int_{\partial\Omega_I} \phi \mathbf{u} \cdot d\mathbf{S}$ . This integral is performed over the full CV boundary  $\partial\Omega_I$  which can always be split into a set of smaller simplicial pieces, called facelets hereafter:

$$\partial\Omega_I = \bigcup_f \partial_f \quad (56)$$

Thanks to the linearity of the integration, it is thus sufficient to be able to compute this integral on a generic simplicial facelet and then to sum up all the contributions to have to the full flux across the CV.

We will restrict the discussion to the evaluation of the convective flux with a constant velocity to avoid any problems arising from the discretization of the velocity field. The case of variable velocity can be addressed by two different level of approximations:

- the easiest and most straightforward solution is to use the value of the velocity at the center of each facelet. If a linear interpolation is used, this method will provide a second-order approximation of the velocity.
- another option is to build a polynomial expansion of the velocity, just as for the scalar, and to compute the flux from the product of these two polynomials. Only the terms up to order 3 are needed to provide a third-order approximation of the flux.

First, we express that the velocity is constant over the facelet  $f$  and that this facelet has a constant normal vector  $\mathbf{n}_f$  everywhere (because it is a simplex) so that the contribution of the facelet  $f$  to the global residual at node  $I$  writes:

$$\mathcal{F}_{I,f} = \int_{\partial_f} \phi \mathbf{u} \cdot d\mathbf{S} = \mathbf{u} \cdot \mathbf{n}_f \int_{\partial_f} \phi dS \quad (57)$$

Then we can replace  $\phi$  by its third-order approximation (Eq. 29):

$$\mathcal{F}_{I,f} = \mathbf{u} \cdot \mathbf{n}_f \int_{\partial_f} \left( \phi_I^{\mathcal{O}3} + \mathbf{G}_I^{\mathcal{O}2} \cdot \Delta_I(\mathbf{x}) + \frac{1}{2} \mathbf{H}_I^{\mathcal{O}1} : (\Delta_I(\mathbf{x}) \otimes \Delta_I(\mathbf{x})) \right) dS + \mathcal{O}(\Delta^{2+d}) \quad (58)$$

We can now introduce the facelet surface (both scalar and vector):

$$S_f = \int_{\partial_f} dS \quad \text{and} \quad \mathbf{S}_f = S_f \mathbf{n}_f \quad (59)$$

and the various facetlet moments with respect to  $I$ :

$$\bar{1}^f = \frac{1}{S_f} \int_{\partial_f} 1 \, dS = 1 \quad (60)$$

$$\overline{\Delta_I}^f = \frac{1}{S_f} \int_{\partial_f} \Delta_I(\mathbf{x}) \, dS \quad (61)$$

$$\overline{\Delta_I \otimes \Delta_I}^f = \frac{1}{S_f} \int_{\partial_f} \Delta_I(\mathbf{x}) \otimes \Delta_I(\mathbf{x}) \, dS \quad (62)$$

Similarly to the control-volume moments, all these quantities can be computed exactly as a function of the coordinates of the simplex vertices. Now Eq. 58 becomes:

$$\mathcal{F}_{I,f} = \mathbf{u} \cdot \mathbf{S}_f \left( \phi_I^{\mathcal{O}3} + \mathbf{G}_I^{\mathcal{O}2} \cdot \overline{\Delta_I}^f + \frac{1}{2} \mathbf{H}_I^{\mathcal{O}1} : \left( \overline{\Delta_I \otimes \Delta_I}^f \right) \right) + \mathcal{O}(\Delta^{2+d}) \quad (63)$$

It can be seen that the integrated flux is of order  $2 + d$ , as the integration over the CV boundary introduces an additional  $d - 1$  order to the flux evaluation order. The residual will then be computed by dividing this integrated flux by the measure  $V_I$  of the CV and the residual will thus be of second order (for a third-order deconvolution). However, in the context of unsteady simulations, this residual is multiplied by the timestep to obtain the solution at the next discrete time. The error on the solution at the next iteration is thus  $\mathcal{O}(\Delta^2 \times \Delta t)$ . If this timestep is subject to a stability constraint such as the CFL condition, it is proportional to the grid size and the error on the solution is finally  $\mathcal{O}(\Delta^3)$ .

Three different flux evaluations will be examined to assess the performance of the proposed method.

The first one, named  $S$  (for Simple), approximates the convective flux on facetlet  $f$  from node  $I$  with the formula:

$$\mathcal{F}_{I,f}^S = \mathbf{u} \cdot \mathbf{S}_f \bar{\phi}^{\Omega_I} \quad (64)$$

where the usual non-deconvoluted value of the transported variable is used in the integration.

The second one uses the deconvolution limited to order 2 and is named  $D_2$ :

$$\mathcal{F}_{I,f}^{D_2} = \mathbf{u} \cdot \mathbf{S}_f \left( \phi_I^{\mathcal{O}2} + \mathbf{G}_I^{\mathcal{O}1} \cdot \overline{\Delta_I}^f \right) \quad (65)$$

Finally, the method  $D_3$  uses the third-order deconvolution presented above:

$$\mathcal{F}_{I,f}^{D_3} = \mathbf{u} \cdot \mathbf{S}_f \left( \phi_I^{\mathcal{O}3} + \mathbf{G}_I^{\mathcal{O}2} \cdot \overline{\Delta_I}^f + \frac{1}{2} \mathbf{H}_I^{\mathcal{O}1} : \left( \overline{\Delta_I \otimes \Delta_I}^f \right) \right) \quad (66)$$

The purpose of the present work is not to design a specific numerical scheme but to show that it is possible to build a high-order approximation of the flux at the control volume interface. The convective residuals associated to the aforementioned fluxes will be built in very simple ways, by considering either fully upwind (Eq. 67) or fully centered (Eq. 68) fluxes at the facetlets:

$$\mathcal{F}_f^* = \begin{cases} \mathcal{F}_{I,f}^* & \text{if } \mathbf{u} \cdot \mathbf{IJ} > 0 \\ 0 & \text{if } \mathbf{u} \cdot \mathbf{IJ} = 0 \\ \mathcal{F}_{J,f}^* & \text{if } \mathbf{u} \cdot \mathbf{IJ} < 0 \end{cases} \quad (67)$$

$$\mathcal{F}_f^{*c} = \frac{1}{2} (\mathcal{F}_{I,f}^* + \mathcal{F}_{J,f}^*) \quad (68)$$

where  $I$  and  $J$  are the two vertices that share the facetlet  $f$ .

The 6 methods introduced above ( $S^c$ ,  $S^u$ ,  $D_2^c$ ,  $D_2^u$ ,  $D_3^c$  and  $D_3^u$ ) have been applied to the convection of the bump function described in section 3.1. The results presented below show the discrepancies between the numerical solution and the analytical solution after the signal has been convected for one period. Figures 8, 9 and 10 show the  $L^2$ -norm of the errors for the same 1D, 2D and 3D domains than those described in section 3.1.

All three figures share the same trend except for one specific feature that will be discussed in detail below. It can be first noticed that the  $S$  schemes (first lines of the figures) exhibit their usual features: when used in conjunction with a centered formulation ( $S^c$ ), it is second-order on regular meshes with a loss of accuracy on distorted meshes. On the other hand, the upwind formulation ( $S^u$ ) always produces a first-order convergence, whatever the mesh quality. Moreover, as it will be illustrated below, it introduces a large amount of numerical dissipation.

The  $D_2$  scheme (second lines of the figures) shows a second-order convergence in all situations, whatever the mesh quality or the amount of upwinding. This is the expected behaviour: the method was demonstrated to be second-order in all situations. Moreover, it appears that just like in the  $S$  schemes, the  $D_2$  scheme is stable even in the centered case and does not require any flux limiting technique.

Finally, the  $D_3$  scheme (third lines of the figures) shows a specific behavior that was not observed with the  $S$  and  $D_2$  schemes. When used with a centered approach ( $D_3^c$ ), it is unstable in 2D and in 3D (on distorted grids only): this is illustrated by the error which increases when the mesh is refined, on the lower left graph of Fig. 10. However, it can be observed that these instabilities totally disappear when an upwind formulation ( $D_3^u$ ) is used. This last choice still provides a third-order convergence on all meshes for 1D, 2D and 3D domains. It can also be observed that before becoming unstable, the  $D_3^c$  approach offers a fourth-order convergence. As shown recently by [19], once a high-order evaluation of the flux is available, it is always possible to apply a convex-limiting technique to mix low-order and high-order approximations to guarantee both the precision and stability of the numerical scheme: such a technique could be used to mix  $\mathcal{F}^{D_3^c}$  and  $\mathcal{F}^{D_3^u}$  to have fourth-order convergence where the mesh is regular and third-order on distorted meshes.

To get a better idea of the improvement provided by the new formulation, the solution after one turn-over time are represented in Fig 11 (for the 1D case only). The left side of the figure is obtained with the centered formulation while the right side is obtained with the upwind scheme. The upper part corresponds to a regular mesh and the lower part represents the solution on a distorted mesh (this can be seen directly on the position of the nodes). Both meshes are composed of 32 cells.

The first observation is that the  $S$  schemes behaves poorly in all situations except for the  $S^c$  formulation on regular grids. On a distorted mesh, large spurious oscillations are observed near the gradients. On the other hand, when using the  $S^u$  formulation, the signal is strongly dissipated: the peak value of the bump has been divided by a factor of 2.

The second major observation is that both  $D_2$  and  $D_3$  schemes are very insensitive to either the mesh quality or the amount of upwinding. In all cases, the transported signal is much closer to the exact solution, more particularly for the  $D_3$  scheme. All these observations are confirmed by the values of the  $L^2$ -norm of the errors associated to each signal, which are reported in the legend box of each figure.

As already mentioned, the deconvolution and flux reconstruction procedure presented here has a computational overhead compared to simpler methods such as the  $S$  scheme. However, this additional CPU remains low, and it is

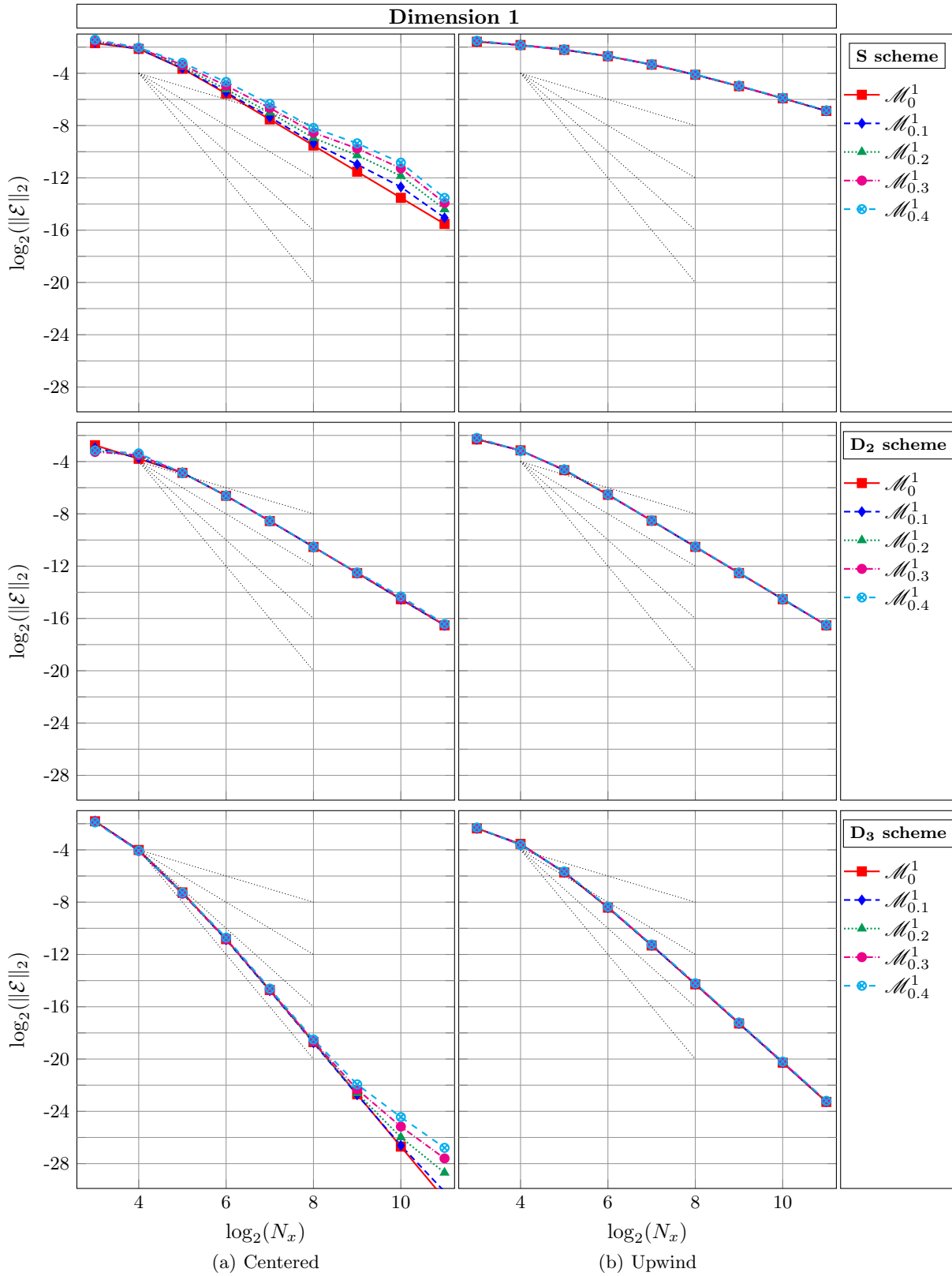


Figure 8: Errors after one turn-over in 1D using the  $S$ ,  $D_2$  and  $D_3$  schemes. Dotted lines correspond to convergence order 1, 2, 3 and 4 respectively.

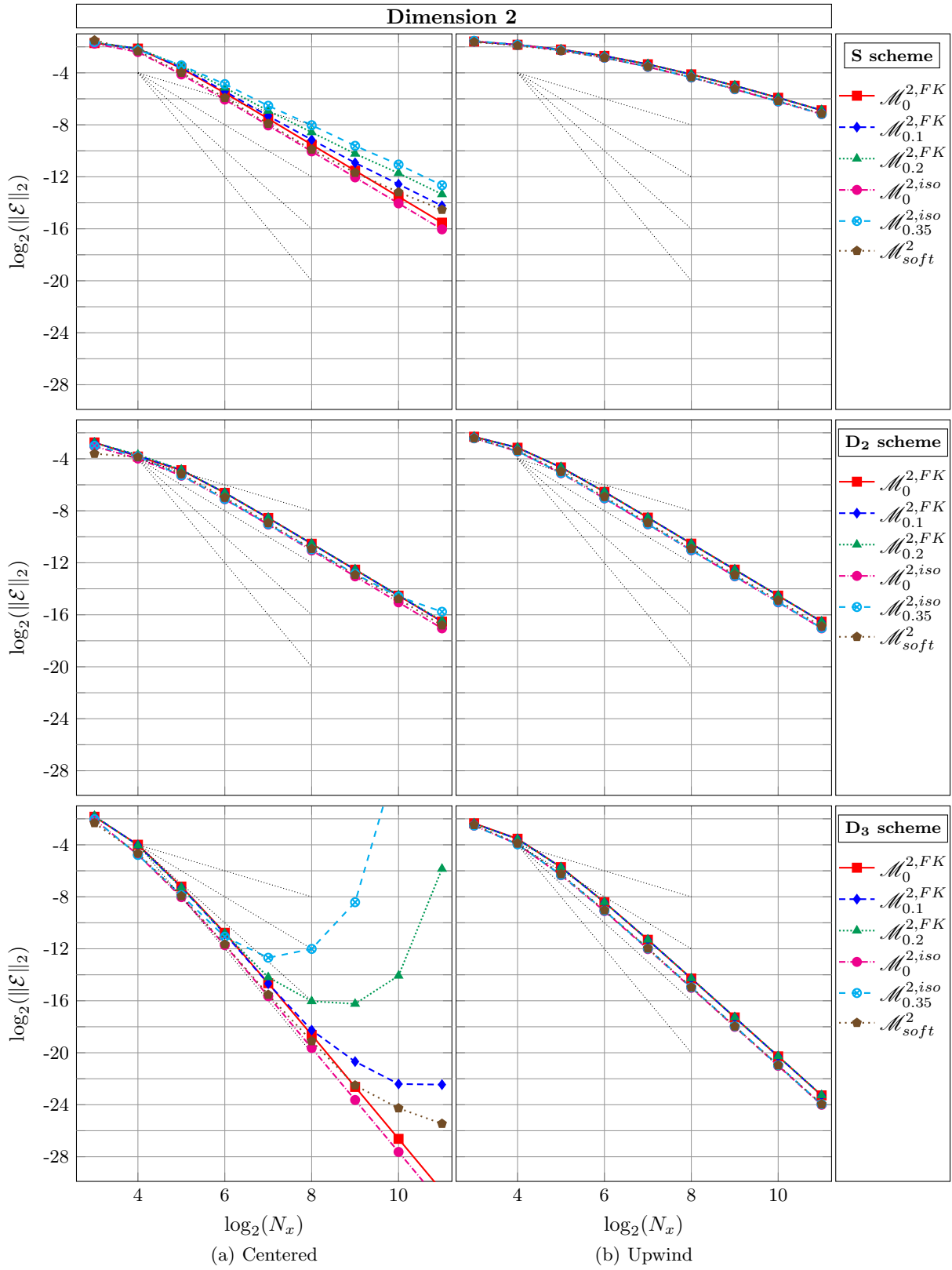


Figure 9: Errors after one turn-over in 2D using the  $S$ ,  $D_2$  and  $D_3$  schemes. Dotted lines correspond to convergence order 1, 2, 3 and 4 respectively.

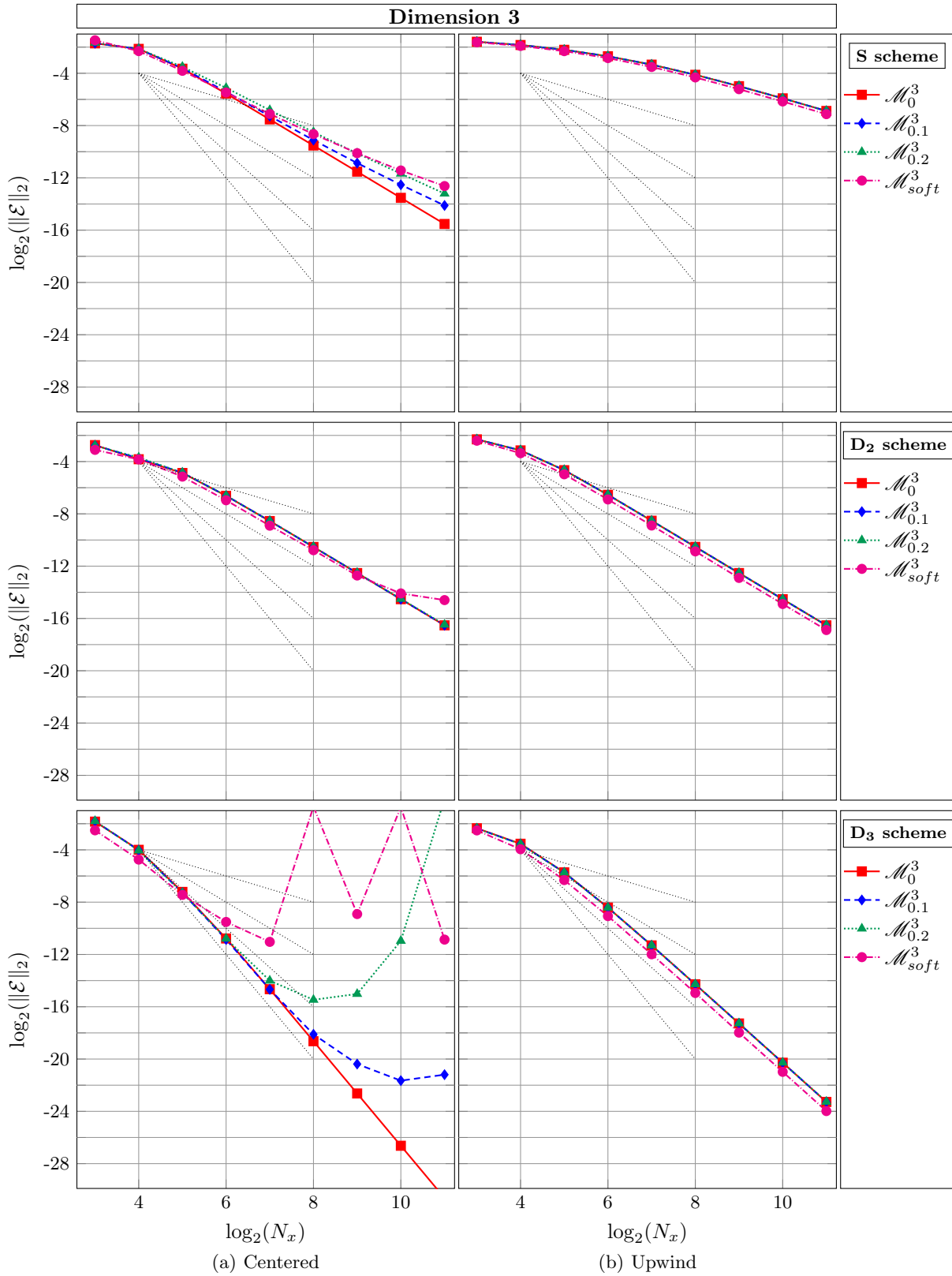


Figure 10: Errors after one turn-over in 3D using the  $S$ ,  $D_2$  and  $D_3$  schemes. Dotted lines correspond to convergence order 1, 2, 3 and 4 respectively.

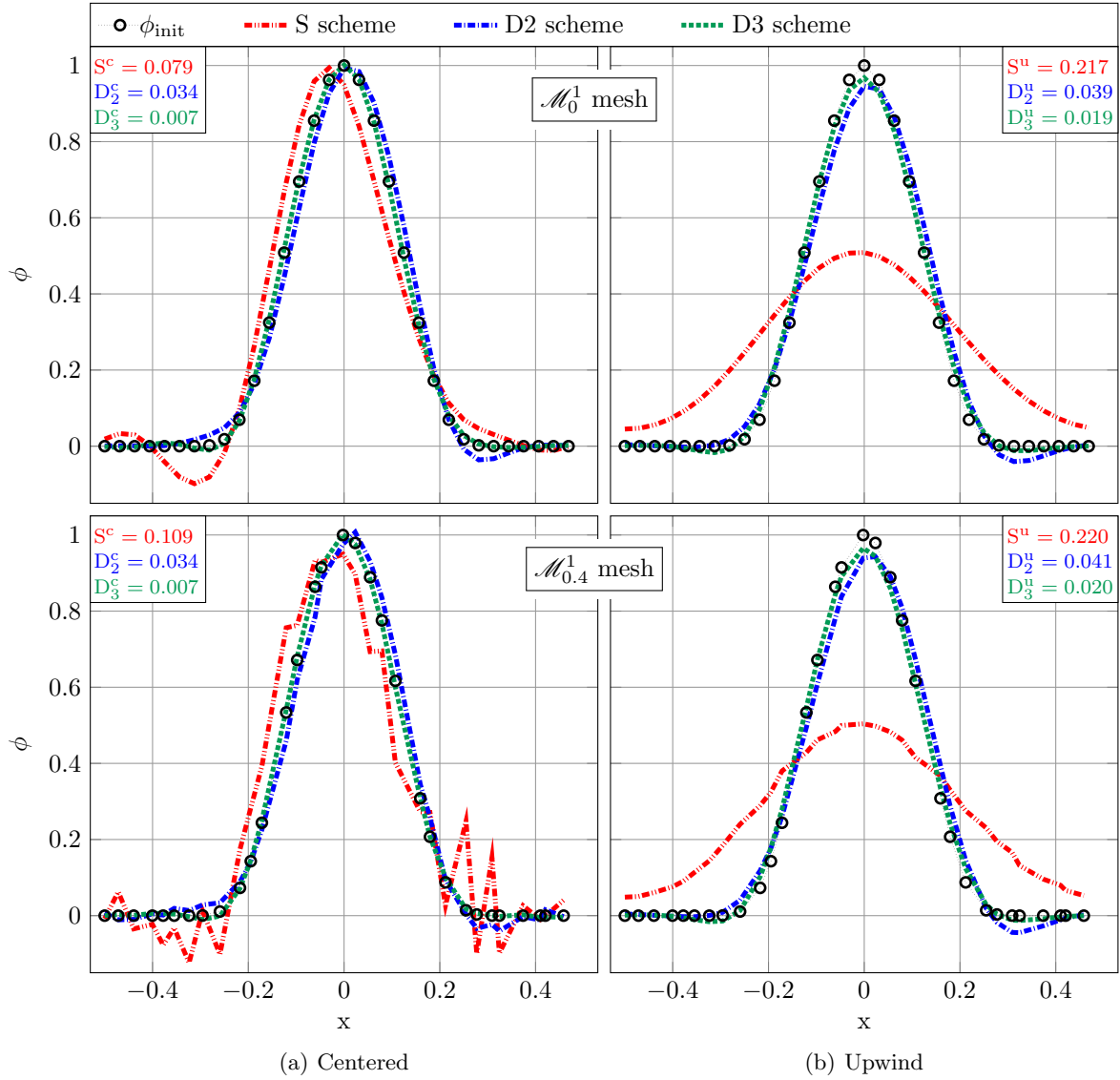


Figure 11: Signal after one turn-over time in 1D using  $N_x = 32$  on regular (left) and distorted (right) meshes.  $L^2$ -norm of error is indicated in upper corner for each scheme. Symbols correspond to node positions.

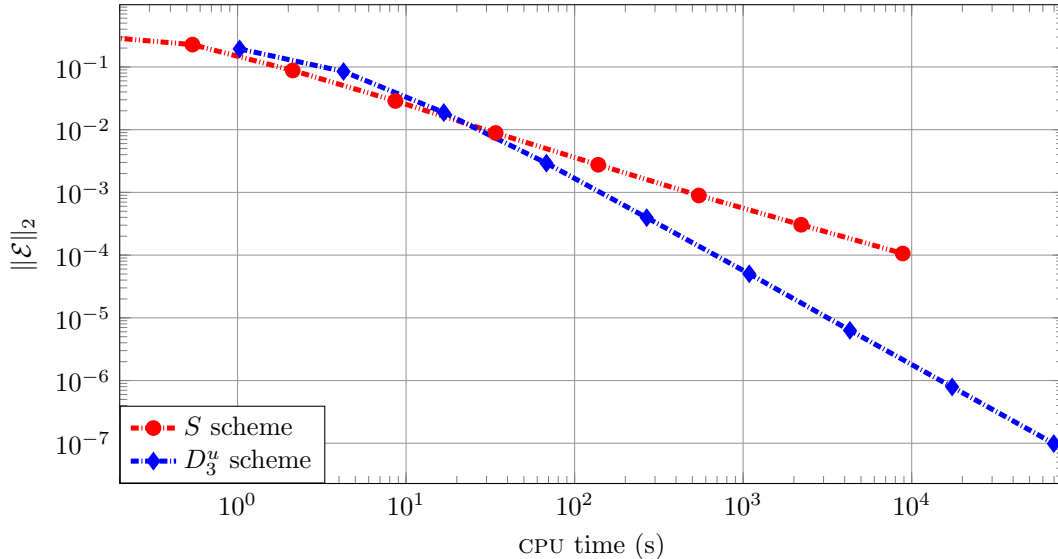


Figure 12: Error after one turnover plotted as a function of the CPU time needed to obtain the solution. Each point corresponds to a different mesh resolution. The data were collected on a 3D domain with distorted grids ( $\mathcal{M}_{0,2}^3$ ).

rapidly counterbalanced by the better accuracy of the method. This is illustrated by Fig. 12 which shows the error evolution with respect to the CPU time needed to obtain the solution for the 3D convection problem. Each point of the figure has been obtained with an  $\mathcal{M}_{0,2}^3$  grid but with different resolutions: the leftmost points corresponds to coarse grids while the rightmost points have been obtained on fine meshes. It can be observed that the  $D_3^u$  scheme can achieve a given accuracy much faster than the low order  $S^c$  scheme.

## 4 Conclusion

Increasing the order of convergence of Finite Volume schemes while keeping an efficient and compact formulation is highly challenging. The development of such schemes requires a rigorous framework which enables to control all the discretization errors. One of these errors comes from the fact that FV methods solve for the averaged quantities, which need to be deconvoluted to build local higher-order polynomial reconstructions of these quantities. Several methods were proposed in the past to deal with this issue, such as the  $k$ -exact formulation (standard and successive correction formulations), the MOOD procedure or the Compact Least-Squares Finite-Volume method. These procedures were mainly studied in the context of the cell-centered FV formulation and to the authors' knowledge, less attention was paid to the node-centered FV approach.

In this paper, a deconvolution method is presented for the node-centered FV approach. It was shown that the node-centered formulation gives a general framework to attain a third-order reconstruction, using only the direct neighbors of a control volume in an unstructured grid composed of simplices. The reconstruction needs to solve a linear system  $Ax = b$  of size 10x10 (in 3D) or 6x6 (in 2D) or 3x3 (in 1D) on each node of the mesh, where  $A$  depends on local metrics only and thus can be computed and inverted in a pre-processing step. As a consequence, the CPU overhead associated to the proposed deconvolution is very limited. Near a boundary, the number of mesh vertices



linked to a boundary node is not large enough to attain a third-order of accuracy but a second-order reconstruction can still be obtained. If needed, third-order can also be achieved at boundaries by using the polynomial expansion from neighbouring nodes.

Several applications from different physical domains were proposed and analyzed, namely high-order interpolation, levelset curvature computation and flux computation for hyperbolic transport equations. It is shown that the proposed technique is efficient and gives results of better quality than the standard scheme. Focusing on the Navier-Stokes equations, another analysis should be performed in the near future to treat the diffusion scheme as well as the case of variable transport coefficients.

## Acknowledgement

Dr. Guillaume Puigt is partially supported by LMA2S (Laboratoire de Mathématiques Appliquées à l'Aéronautique et au Spatial), the Applied Mathematics Lab of ONERA. We thank PSPC Innov'Hydro for funding M. Bernard, GENCI (Grant 2A00611) and CRIANN for providing the computing resources for the various test cases.

## A Grid moments computation

The technical details to compute the moments of a generic simplex are briefly explained hereafter. Let us consider a generic  $d$ -dimensional simplex, whose vertices coordinates are noted  $\mathbf{x}_1, \dots, \mathbf{x}_{d+1}$ . We can first compute its volume with the classical formula:

$$V = \int_{\Omega} dV = \frac{1}{d!} |\mathbf{X}| \quad (69)$$

where  $\mathbf{X}$  is a  $d \times d$  matrix formed by the coordinates of  $d$  edges of the simplex that all share a common vertex, for example:

$$\mathbf{X} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_3 - y_1 & y_4 - y_1 \\ z_2 - z_1 & z_3 - z_1 & z_4 - z_1 \end{bmatrix} \quad \text{for } d = 3 \quad (70)$$

By adapting the formulas given by [38] for a tetrahedron, it is then possible to derive the first two moments of any simplex, with respect to the origin.

$$\mathbf{M}_O^1 = \frac{1}{V} \int_{\Omega} \mathbf{x} dV \quad (71)$$

$$= \frac{1}{d+1} \sum_{i=1}^{d+1} \mathbf{x}_i \quad (72)$$

$$\mathbf{M}_O^2 = \frac{1}{V} \int_{\Omega} \mathbf{x} \otimes \mathbf{x} dV \quad (73)$$

$$= \frac{1}{(d+1)(d+2)} \left[ \left( \sum_{i=1}^{d+1} \sum_{j=1}^{d+1} \mathbf{x}_i \otimes \mathbf{x}_j \right) + \left( \sum_{i=1}^{d+1} \mathbf{x}_i \otimes \mathbf{x}_i \right) \right] \quad (74)$$

Deriving similar expressions for higher order moments, although tedious, is not a problem and can be conducted for any simplex.

The next ingredient that is needed to compute the moments of a simplex with respect to any point  $I$  is the translation formulas, which are given here only for orders 1 and 2 but which can be derived to any order without any difficulty with the binomial formula in a non-commutative ring:

$$\mathbf{M}_I^1 = \mathbf{M}_O^1 - \mathbf{x}_I \quad (75)$$

$$\mathbf{M}_I^2 = \mathbf{M}_O^2 - \mathbf{x}_I \otimes \mathbf{M}_O^1 - \mathbf{M}_O^1 \otimes \mathbf{x}_I + \mathbf{x}_I \otimes \mathbf{x}_I \quad (76)$$

Finally, thanks to the linearity of the integration operator, the full control-volume moments can be computed by first performing a simplicial decomposition of the CV and then by summing up the moments of the elementary simplices which compose the control-volume (as long as they are computed with respect to the same position,  $I$  in the present case).

## B Mesh Quality

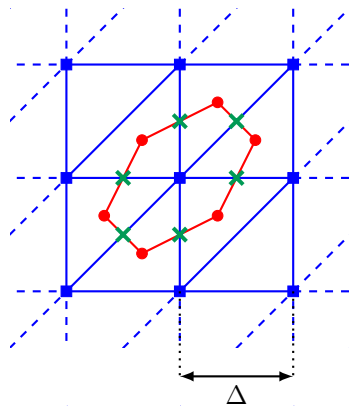
In the present work, numerical experiments are performed to assess the quality of various operators, i.e. their order of convergence, both on regular and distorted simplicial meshes. The specific procedure that was used to build these meshes is detailed below.

In 1D, the procedure is straightforward: the 1D domain of size  $L$  is divided into  $N$  segments of identical size to produce a regular mesh whose cells have an equal size of  $\Delta = \frac{L}{N}$ . The 1D distorted meshes are then constructed by applying a perturbation to each node of this regular mesh. For the node  $I$ , the amplitude of this perturbation is equal to  $r_I \Delta$  where  $r_I$  is a random variable with a uniform distribution on  $[-b/2; +b/2]$ , where  $b$  is a perturbation parameter which is fixed for a given mesh. Those meshes will be noted  $\mathcal{M}_b^1$ .

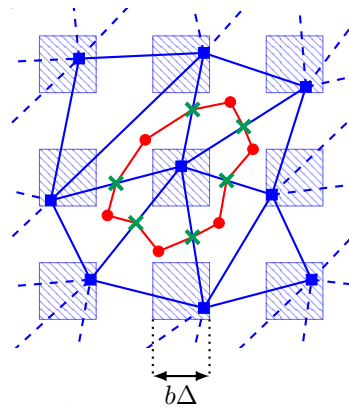
In 2D, a regular triangular mesh is built by performing a Friedrichs and Keller triangulation of a Cartesian mesh (see Fig. 13-a). Following the same procedure as in 1D, this grid is then perturbed by randomly shifting its vertices inside a square box of size  $b\Delta$  surrounding each node, as depicted in Fig. 13-b. Those meshes will be noted  $\mathcal{M}_b^{2,FK}$ . Just by stretching and shifting the  $\mathcal{M}_0^{2,FK}$  mesh described above (but without modifying the connectivity), it is also possible to build a mesh only composed of equilateral triangles (Fig. 13-c). In this situation, the control volume is a regular hexagon: this is the most isotropic CV that can be obtained with this connectivity. This mesh can also be perturbed by the technique described above and those meshes will be noted  $\mathcal{M}_b^{2,iso}$ . An example of  $\mathcal{M}_{0.35}^{2,iso}$  mesh is shown in Fig. 14 to illustrate the type of mesh which is concretely used.

Based on the same idea, the 3D tetrahedral meshes that are used in the present paper are built in two steps. First, a Cartesian hexahedral grid is built and its vertices are shifted randomly into a cube of size  $b\Delta$ . Then, each hexahedron is divided into six tetrahedra following the method proposed by Dompierre *et al.* [15] (see Fig. 13-d). Those meshes will be noted  $\mathcal{M}_b^3$ .

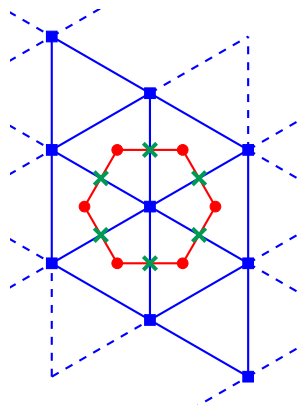
It must be noticed that the perturbation parameter  $b$  can not exceed a certain value (which depends on the initial mesh) to ensure that all the cells have a positive volume.



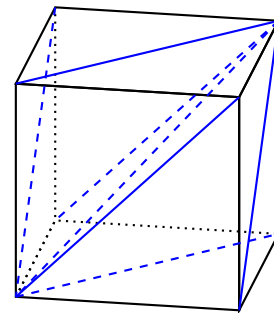
(a)  $\mathcal{M}_0^{2,FK}$



(b)  $\mathcal{M}_b^{2,FK}$



(c)  $\mathcal{M}_0^{2,iso}$



(d)  $\mathcal{M}_0^3$  : hexahedron splitting

Figure 13: Examples of manufactured meshes used for the convergence order tests. (a), (b) and (c) correspond to 2D situations with the same coloring conventions as above while (d) represents the procedure used to partition an hexahedron into six tetrahedrons.

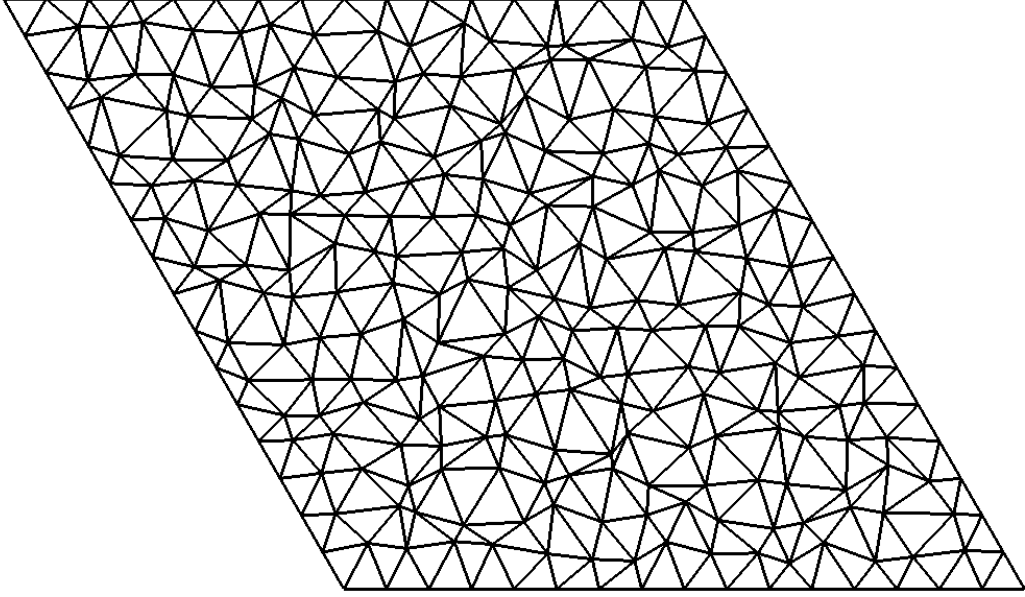


Figure 14: Example of  $\mathcal{M}_{0.35}^{2,iso}$  manufactured mesh with  $N_x = 16$ .

## C Analytical evaluation of the convolution matrix

In some simple cases, the convolution / deconvolution procedure can be performed analytically. This is particularly interesting for a distorted 1D mesh and for the mesh referred to as  $\mathcal{M}_0^{2,iso}$ . The aim of this appendix is to present briefly these two results.

With respect to the notations introduced in Fig. 15, the convolution matrix on a distorted 1D mesh reads:

$$\mathbf{C}_I^{\mathcal{O}3} = \begin{bmatrix} 1 & \frac{h^+ - h^-}{4} & \frac{1}{24} \left[ (h^+)^2 - h^+ h^- + (h^-)^2 \right] \\ 0 & \mathcal{G}_I \left( \overline{\Delta_I}^\Omega \right) & \frac{1}{2} \mathcal{G}_I \left( \overline{\Delta_I \otimes \Delta_I}^\Omega \right) \\ 0 & \mathcal{H}_I \left( \overline{\Delta_I}^\Omega \right) & \frac{1}{2} \mathcal{H}_I \left( \overline{\Delta_I \otimes \Delta_I}^\Omega \right) \end{bmatrix}, \quad (77)$$

with

$$\mathcal{G}_I \left( \overline{\Delta_I}^\Omega \right) = \frac{1}{4(h^+ + h^-)} \left[ h^{++} + 3h^+ + 3h^- + h^{--} \right], \quad (78)$$

$$\mathcal{G}_I \left( \overline{\Delta_I \otimes \Delta_I}^\Omega \right) = \frac{1}{12(h^+ + h^-)} \left( (h^{++})^2 + 5h^+ h^{++} + 7(h^+)^2 - (h^{--})^2 - 5h^{--} h^- - 7(h^-)^2 \right), \quad (79)$$

$$\mathcal{H}_I \left( \overline{\Delta_I}^\Omega \right) = \frac{1}{2h^- h^+ (h^+ + h^-)} \left[ h^- (h^{++} + h^-) - h^+ (h^{--} + h^+) \right], \quad (80)$$

$$\mathcal{H}_I \left( \overline{\Delta_I \otimes \Delta_I}^\Omega \right) = \frac{1}{6h^- h^+ (h^+ + h^-)} \left[ h^- (h^{++})^2 + 5h^- h^+ (h^{++} + h^{--}) + 7h^- h^+ (h^+ + h^-) - (h^-)^3 + h^+ (h^{--})^2 - (h^+)^3 \right]. \quad (81)$$

Its determinant,

$$\det(\mathbf{C}_I^{\mathcal{O}3}) = \frac{(h^- + 2h^+ + h^{++})(h^{--} + 2h^- + h^+)(h^{--} + 2h^- + 2h^+ + h^{++})}{48h^- h^+ (h^- + h^+)}, \quad (82)$$

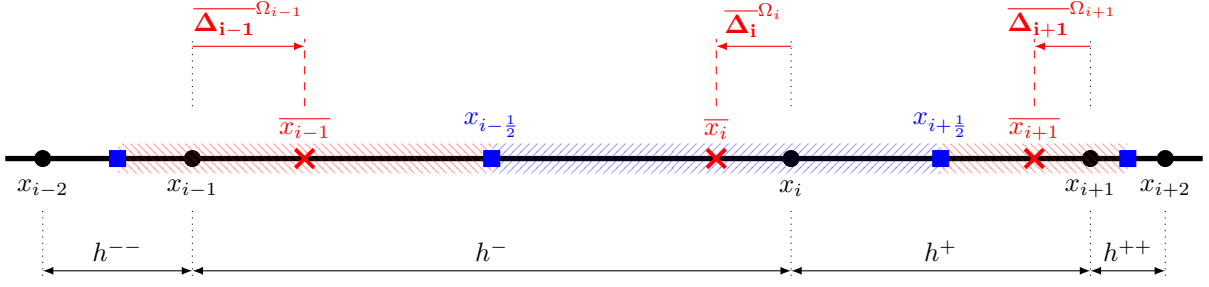


Figure 15: Notations for a distorted 1D barycentric dual mesh.

never vanishes so that  $C_I^{\mathcal{O}3}$  is always invertible in 1D.

On 1D uniform mesh of size  $h$ , the convolution matrix Eq. 77 reduces to the well-known link between the nodal value and the cell averaged value:

$$C_I^{\mathcal{O}3} = \begin{bmatrix} 1 & 0 & \frac{h^2}{24} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (83)$$

For a 2D regular grid composed of equilateral triangles (Fig. 16, the same case that in [32]), the convolution matrix that arises in our framework is:

$$C_I^{\mathcal{O}3} = \begin{bmatrix} 1 & 0 & 0 & \frac{5h^2}{72} & \frac{5h^2}{72} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{3}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{3}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \quad (84)$$

and the resolution of this system leads to:

$$\begin{bmatrix} \phi_I \\ \partial_x \phi_I \\ \partial_y \phi_I \\ \frac{1}{2} \partial_{xx} \phi_I \\ \frac{1}{2} \partial_{yy} \phi_I \\ \partial_{xy} \phi_I \end{bmatrix} = \begin{bmatrix} \frac{246\bar{\phi}_I - 5(\bar{\phi}_{J_1} + \bar{\phi}_{J_2} + \bar{\phi}_{J_3} + \bar{\phi}_{J_4} + \bar{\phi}_{J_5} + \bar{\phi}_{J_6})}{216} \\ \frac{2(\bar{\phi}_{J_1} - \bar{\phi}_{J_4}) + \bar{\phi}_{J_2} + \bar{\phi}_{J_6} - \bar{\phi}_{J_3} - \bar{\phi}_{J_5}}{6h} \\ \frac{\sqrt{3}(\bar{\phi}_{J_2} + \bar{\phi}_{J_3} - \bar{\phi}_{J_5} - \bar{\phi}_{J_6})}{6h} \\ \frac{\bar{\phi}_{J_1} - 2\bar{\phi}_I + \bar{\phi}_{J_4}}{2h^2} \\ \frac{2(\bar{\phi}_{J_2} + \bar{\phi}_{J_3} + \bar{\phi}_{J_5} + \bar{\phi}_{J_6}) - 6\bar{\phi}_I - \bar{\phi}_{J_1} - \bar{\phi}_{J_4}}{6h^2} \\ \frac{\sqrt{3}(\bar{\phi}_{J_2} + \bar{\phi}_{J_5} - \bar{\phi}_{J_3} - \bar{\phi}_{J_6})}{3h^2} \end{bmatrix}. \quad (85)$$

The components of the gradient are identical to the equation 13 of [32] which is obtained with the Least-Square method in the linear reconstruction case.

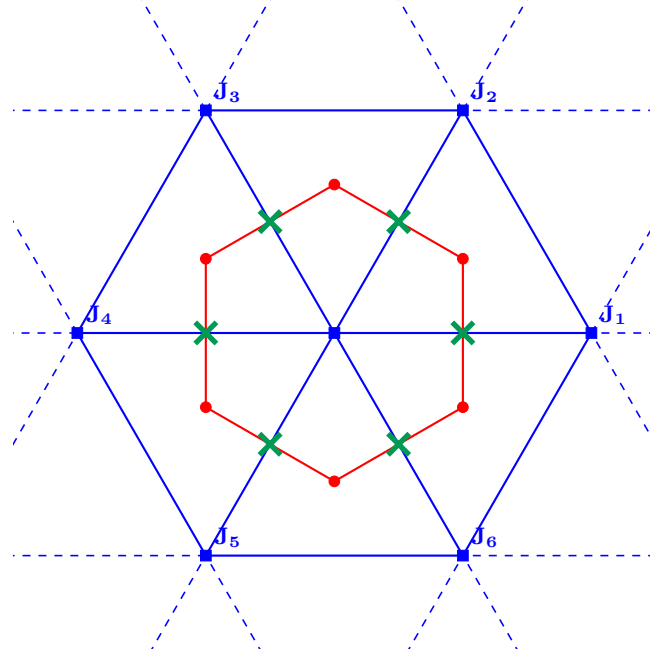


Figure 16: Regular 2D mesh composed of equilateral triangles

## References

- [1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics*, 114:45–58, 1994.
- [2] F. Alauzet. A parallel matrix-free conservative solution interpolation on unstructured tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 299:116–142, feb 2016.
- [3] F. Alauzet and M. Mehrenberger.  $p^1$ -conservative solution interpolation on unstructured triangular meshes. *International Journal for Numerical Methods in Engineering*, 84(13):1552–1588, nov 2010.
- [4] T. J. Barth. A 3-d upwind euler solver for unstructured meshes. In *10th Computational Fluid Dynamics Conference*, page 1548, Honolulu, HI, U.S.A., 1991. Aerospace Research Central (ARC).
- [5] T. J. Barth. Aspects of unstructured grids and finite-volume solvers for the euler and navier-stokes equations. In *NASA report*, Honolulu, HI, U.S.A., 1992. NASA Ames Research Center.

- [6] T. J. Barth and P. O. Frederickson. Higher order solution of the euler equations on unstructured grids using quadratic reconstruction. *AIAA paper*, 90:0013, 1990.
- [7] L. J. Betchen and A. G. Straatman. An accurate gradient and hessian reconstruction method for cell-centered finite volume discretizations on general unstructured grids. *International Journal for Numerical Methods in Fluids*, 62:945–962, 2009.
- [8] R. Chiodi and O. Desjardins. A reformulation of the conservative level set reinitialization equation for accurate and robust simulation of complex multiphase flows. *Journal of Computational Physics*, 343:186–200, 2017.
- [9] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for systems of conservation laws - Multi-dimensional Optimal Order Detection (MOOD). *Journal of Computational Physics*, 230:4028–4050, 2011.
- [10] M. Déléze. *Eléments finis tétraédriques de classe C1 et de degré deux*. PhD thesis, Faculté des Sciences de l’Université de Fribourg, Chemin du Musée, 1700 Fribourg, Suisse, 1979.
- [11] M. Déléze, J.-J. Goël, and B. Meyenhofer. Finite elements of c1-class on a tetrahedron. *International Journal for Numerical Methods in Engineering*, 12(5):787–793, 1978.
- [12] O. Desjardins, V. Moureau, and H. Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395–8416, 2008.
- [13] S. Diot, S. Clain, and R. Loubère. Improved detection criteria for the Multi-dimensional Optimal Order Detection (MOOD) on unstructured meshes with very high-order polynomials. *Computers & Fluids*, 64:43–63, 2012.
- [14] S. Diot, R. Loubère, and S. Clain. The Multidimensional Optimal Order Detection method in the three-dimensional case: very high-order finite volume method for hyperbolic systems. *International Journal for Numerical Methods in Fluids*, 73:3762–392, 2013.
- [15] J. Dompierre, P. Labbé, M.G. Vallet, and R. Camarero. How to subdivide pyramids, prisms and hexahedra into tetrahedra. In *8th International Meshing Roundtable*, pages 195–204, South Lake Tahoe, California, 1999. Major Shared Resource Centers (MSRC) established by Department of Defense (DOD) High Performance Computing Modernization Office (HPCMO) and International Society of Grid Generation (ISGG).
- [16] M. Dumbser, D. S. Balsara, E. F. Toro, and C.-D. Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227:8209–8253, 2008.
- [17] P. Gervasio and A. Quarteroni. The INTERNODES method for non-conforming discretizations of PDEs. *Communications on Applied Mathematics and Computation*, 1(3):361–401, may 2019.
- [18] R. Goldman. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design*, 22(7):632 – 658, 2005. Geometric Modelling and Differential Geometry.



- [19] J.L. Guermond, B. Popov, and I. Tomas. Invariant domain preserving discretization-independent schemes and convex limiting for hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 347:143 – 175, 2019.
- [20] F. Haider. *Discrétisation en maillage non structuré général et applications LES*. PhD thesis, Université Paris 6, 4 Place Jussieu, 75005 Paris, France, 2009.
- [21] F. Haider, P. Brenner, B. Courbet, and J.-P. Croisille. Efficient implementation of high order reconstruction in finite volume methods. In *Finite Volumes for Complex Applications VI Problems & Perspectives*, pages 553–560. Springer, 2011.
- [22] F. Haider, P. Brenner, B. Courbet, and J.-P. Croisille. Parallel implementation of  $k$ -exact finite volume reconstruction on unstructured grids. In *High Order Nonlinear numerical schemes for evolutionary PDEs*, pages 59–75. Springer, 2014.
- [23] F. Haider, B. Courbet, and J.-P. Croisille. A high-order interpolation for the finite volume method: The coupled least squares reconstruction. *Computers & Fluids*, 176:20–39, 2018.
- [24] J.M. Hyman, R.J. Knapp, and J.C. Scovel. High order finite volume approximations of differential operators on nonuniform grids. *Physica D: Nonlinear Phenomena*, 60(1-4):112–138, 1992.
- [25] S. Le Bras, H. Deniau, and C. Bogey. A technique of flux reconstruction at the interfaces of non-conforming grids for aeroacoustic simulations. *International Journal for Numerical Methods in Fluids*, aug 2019.
- [26] J.C. Mandal and S.P. Rao. High resolution finite volume computations on unstructured grids using solution dependent weighted least squares gradients. *Computers & Fluids*, 44:23–31, 2011.
- [27] E. Marchandise, P. Geuzaine, N. Chevaugeon, and J.F. Remacle. A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics. *Journal of Computational Physics*, 225(1):949–974, 2007.
- [28] A. Menasria, P. Brenner, and P. Cinnella. Improving the treatment of near-wall regions for multiple-correction  $k$ -exact schemes. *Computers & Fluids*, 181:116–134, 2019.
- [29] P. Mullen, P. Memari, F. De Goes, and M. Desbrun. Hot: Hodge-optimized triangulations. *ACM Transactions on Graphics*, 30(4):103:1–103:12, 2011.
- [30] C. Ollivier-Gooch. Quasi-eno schemes for unstructured meshes based on unlimited data-dependent least-squares reconstruction. *Journal of Computational Physics*, 133:6–17, 1997.
- [31] C.F. Ollivier-Gooch, A. Nejat, and K. Michalak. Obtaining and verifying high-order unstructured finite volume solutions to the Euler equations. *AIAA Journal*, 47(9):2105–2120, 2009.
- [32] C.F. Ollivier-Gooch and M. Van Altena. A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation. *Journal of Computational Physics*, 181(2):729–752, 2002.

- [33] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.
- [34] I.W. Turner P.A. Jayantha. A comparison of gradient approximations for use in finite-volume computational models for two-dimensional diffusion equations. *Numerical Heat Transfer, Part B: Fundamentals*, 40(5):367–390, 2001.
- [35] G. Pont. *Self adaptive turbulence models for unsteady compressible flows Modèles de turbulence auto-adaptatifs pour la simulation des écoulements compressibles instationnaires*. PhD thesis, ENSAM, Paris, 151 Boulevard de l’Hôpital, 75013 Paris, France, 2015.
- [36] G. Pont, P. Brenner, P. Cinella, B. Maugars, and J.C. Robinet. Multiple-correction hybrid  $k$ -exact schemes for high-order compressible RANS-LES simulations on fully unstructured grids. *Journal of Computational Physics*, 350:45–83, 2017.
- [37] G. Puigt, J.-D. Müller, and V. Auffray. Discretisation of diffusive fluxes on hybrid grids. *Journal of Computational Physics*, 229(5):1425–1447, 2010.
- [38] H.T. Rathod and H.S. Govinda Rao. Integration of polynomials over an arbitrary tetrahedron in euclidean three-dimensional space. *Computers & Structures*, 59(1):55 – 65, 1996.
- [39] C.B. Sejeikan and C.F. Ollivier-Gooch. Improving finite-volume diffusive fluxes through better reconstruction. *Computers & Fluids*, 139(5):216–232, 2016.
- [40] M. Sharbatdar, A. Jalali, and C.F. Ollivier-Gooch. Smoothed truncation error in functional error estimation and correction using adjoint methods in an unstructured finite volume solver. *Computers & Fluids*, 140:406–421, 2016.
- [41] E. Sozer, C. Brehm, and C.C. Kiris. Gradient calculation methods on arbitrary polyhedral unstructured meshes for cell-centered CFD solvers. In *52nd AIAA Aerospace Sciences Meeting, National Harbor, Maryland, 13-17 January, AIAA Paper 2014-1440*. Aerospace Research Central (ARC), 2014.
- [42] B. Van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method. *Journal of Computational Physics*, 32:101–136, 1979.
- [43] Q. Wang, Y.X. Ren, and W. Li. Compact high order finite volume method on unstructured grids I: basic formulations and one-dimensional schemes. *Journal of Computational Physics*, 314:863–882, 2016.
- [44] Q. Wang, Y.X. Ren, and W. Li. Compact high order finite volume method on unstructured grids II: extension to two-dimensional Euler equations. *Journal of Computational Physics*, 314:883908, 2016.
- [45] Q. Wang, Y.X. Ren, J. Pan, and W. Li. Compact high order finite volume method on unstructured grids III: Variational reconstruction. *Journal of Computational Physics*, 337:1–26, 2017.

- [46] Y.-S. Zhang, Y.-X. Ren, and Q. Wang. Compact high order finite volume method on unstructured grids IV: Explicit multi-step reconstruction schemes on compact stencil. *Journal of Computational Physics*, 396:161–192, 2019.