HAL open science

Mean Mesh Adaptation for Efficient CFD Simulations with Operating Conditions Variability

Hugo Dornier, Olivier Le Maitre, Pietro Marco Congedo, Itham Salah El Din, Julien Marty, Sébastien Bourasseau

To cite this version:

Hugo Dornier, Olivier Le Maitre, Pietro Marco Congedo, Itham Salah El Din, Julien Marty, et al.. Mean Mesh Adaptation for Efficient CFD Simulations with Operating Conditions Variability. Computers and Fluids, 2025, 298, pp.106666. ⟨10.1016/j.compfluid.2025.106666⟩. ⟨hal-04808444v2⟩

HAL Id: hal-04808444

https://hal.science/hal-04808444v2

Submitted on 21 May 2025

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.

Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Computers and Fluids 298 (2025) 106666

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Mean mesh adaptation for efficient CFD simulations with operating conditions variability

Hugo Dornier [a,b,*], Olivier P. Le Maître [c], Pietro M. Congedo [b], Itham Salah el Din [a], Julien Marty [a], Sébastien Bourasseau [d]

[a] DAAA, ONERA, Institut Polytechnique de Paris, 92190 Meudon, France
[b] Inria, Centre de Mathématiques Appliquées, Ecole Polytechnique, IPP, Route de Saclay, 91128 Palaiseau Cedex, France
[c] CNRS, Centre de Mathématiques Appliquées, Ecole Polytechnique, IPP, Route de Saclay, 91128 Palaiseau Cedex, France
[d] DAAA, ONERA, Institut Polytechnique de Paris, 92322 Châtillon, France

ARTICLE INFO

Keywords:
Mesh adaptation
Conditions variability
Error measurement

ABSTRACT

When numerically solving partial differential equations, for a given problem and operating condition producing a steady-state, mesh adaptation has proven its efficiency to automatically build a discretization achieving a prescribed error level at low cost. However, with continuously varying operating conditions, such as those encountered in uncertainty quantification, adapting a mesh for each condition and controlling the error level becomes complex and computationally expensive. To enable more effective error and cost control, this work proposes a novel approach to mesh adaptation. The method consists in building a single adapted mesh that aims to minimize the average error for a continuous set of operating conditions. In the proposed implementation, this single mesh is built iteratively, informed by an estimate of the local average interpolation error. The proposed method leverages the iterative nature of mesh adaptation by re-sampling Monte Carlo quadratures to obtain accurate average error estimates over a reduced set of sample conditions, ensuring a low computational cost. This approach is especially effective for localized flow features whose positions change only slightly with operating conditions, such as moving shocks in supersonic flows, as the refinement is confined to smaller areas of the computational domain. The study focuses on evaluating the method's average error convergence, robustness, and computational cost in comparison to state-of-the-art adaptation techniques. Additionally, the sensitivity of the approach to the choice and size of the quadrature, as well as to the error estimation method, is assessed. For this purpose, the methodology is applied to a one-dimensional variable-step solution of the Burgers equation and a two-dimensional Euler scramjet flow with a variable inlet Mach number. The results show that Mean Mesh Adaptation (MMA) achieves error convergence comparable to specific mesh adaptation while reducing the evaluation cost by up to a factor of five (in the scramjet case). This efficiency gain stems from the reduced dependence on the number of sampled conditions, thanks to robust Monte Carlo re-sampling, as well as the shared computational expense of mesh construction across multiple evaluations. Therefore, the proposed method enables computational efficiency while maintaining error control across varying operating conditions within a prescribed parameter variation range.

1. Introduction

In computational fluid dynamics (CFD), geometries under inflow and boundary conditions of interest can result in complex local phenomena and interactions, such as shock and flow separation patterns. An accurate evaluation of the quantities of interest (global performance, local metrics...) is conditioned by the capability of the calculation to capture these patterns with a control on the error. The latter is highly dependent on the computational domain mesh discretization. As the regions of interest can appear in large zones as well as in very local ones, with strong gradients for example, complying to an accurate calculation of both without knowing *a priori* where they should appear would require such a dense mesh that the effort of calculation would loose all its practicality. Furthermore, one may be interested in more than a single inflow condition, rendering such an approach even less achievable.

A solution commonly proposed by the CFD community is to use mesh adaptation ([1–3]) to adjust iteratively and automatically the local mesh density to the flow structures and provide meshes achieving

* Corresponding author at: DAAA, ONERA, Institut Polytechnique de Paris, 92190 Meudon, France.
E-mail address: hugo.dornier@onera.fr (H. Dornier).

https://doi.org/10.1016/j.compfluid.2025.106666
Received 15 November 2024; Received in revised form 4 March 2025; Accepted 1 May 2025
Available online 17 May 2025
0045-7930/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

a prescribed accuracy while limiting the computational cost. Several adaptation strategies and criteria have been developed over the years and applied to problems of variable complexity [4,5].

The present work proposes to develop a new method to extend the concept of mesh adaptation to continuously variable operating conditions. This variability can be encountered in several categories of studies, such as operating domain analysis, where operating conditions are usually spread out and can lead to very different solutions. One of the objectives of uncertainty quantification [6] aims at understanding how uncertainty in input parameters (e.g. material properties, boundary conditions) affects the model predictions. Therefore, it is also required to handle conditional variability, albeit usually with smaller variations and discrepancies. Nevertheless, both can be computationally demanding because of the numerous computations at different conditions they require, and their accuracy is hard to control for all conditions.

To tackle these complexities, several methods have been developed for mesh adaptation in the case of variable conditions. The standard approach consists in adapting the mesh for each condition to evaluate [7,8] which enables an excellent error control. However, the cost of this approach can be high, depending on the refinement needs and method used. This approach also requires a fully automated and robust refinement tool, especially for the selection of the refinement parameters that may be influenced by the particular condition. Therefore, defining a single adapted mesh yielding a controlled error for any condition is desirable to avoid a systematic adaptation. A straightforward approach consists in adapting the mesh to a single representative condition, named nominal condition in the following, and to use it subsequently for all other conditions. This approach is limited and will not achieve low errors for conditions that yield solution structures different from the nominal one. An alternative method was proposed in [9], where an adjoint-based local sensitivity analysis is employed to enrich the refinement criterion from a nominal condition and build a single goal-oriented adapted mesh for all conditions. In their work, the authors demonstrated the effectiveness of the method in controlling the error in a limited neighborhood of the nominal condition. Nevertheless, because the refinement is only informed by the nominal condition and its sensitivity, the method fails when there are strong topological changes in the flow. Another approach was proposed in [10] for anisotropic adaptation. In this approach, using the Mach number hessian, a metric tensor (representing the required local size and orientation of the elements) is computed for several conditions and combined to drive the generation of the next mesh. In practice, a metric intersection technique is employed [11]. Compared to [9] the method in [10] can prove to be more robust. However, the metric is built only on the fixed set of conditions tested, and the resulting mesh may not be suitable for other conditions.

As a result, none of these methods allows controlling at the same time the cost and error level for studies with arbitrary continuously variable conditions. The present work proposes an alternative original methodology based on mesh adaptation resulting in a single adapted mesh for multiple flow conditions to control cost and error, referred to as the *mean mesh adaptation method* or MMA. The strategy consists in building this single adapted mesh to minimize the average error for a whole continuous set of conditions of interest with cost management. The iterative adaptation method developed uses an estimation of the local mean error from a sub-set of prescribed quadrature conditions to mark the cells for refinement. The local error estimator chosen for the study is a well referenced interpolation error estimator [2,12,13], as it is reasonably assumed that the theoretical solution is not known. The use of Monte Carlo-type quadrature to estimate average quantities is particularly suited to MMA, as the iterative nature of mesh adaptation allows re-sampling at each iteration, leading to a progressive improvement in quadrature quality without additional cost. Moreover, the MMA strategy constructs the unique mesh offline, which is computationally efficient, especially when a large number of online evaluations are needed. The offline mesh construction reduces the cost of repeated online evaluations and eliminates the need for expertise in mesh adaptation.

In this paper, the MMA methodology is described in detail along with the criteria used to evaluate its behavior and performance. Then, the effectiveness of the method is demonstrated on a one-dimensional parametric Burgers equation for which the exact theoretical solution is available; this problem allows to perform a complete statistical characterization of the resulting errors and their sensitivity to the adaptation method parameters. As a first step toward the use of this methodology for CFD problems, a two-dimensional supersonic scramjet inlet in an inviscid flow field [14] is then considered. This case is widely used in the mesh adaptation literature [15–17] as it yields complex shock structure and interactions.

2. Mesh adaptation for finite volume method

This section reviews the tools that will be used for the MMA method proposed in the present work.

2.1. Finite volume method

This work relies on Finite Volume (FV) methods [18] for the discretization and resolution of the problem equations. The study is restricted to a bounded d-dimensional ($d \geq 1$) computational domain, Ω, delimited by its boundary $\partial\Omega$. The solution $u(x)$ over this domain is approximated by discretizing Ω into a mesh $\mathcal{H}$ composed of non-overlapping cells $c_i, i \in \{1, \dots, N\}$:

$$\Omega = \overline{\bigcup_{i=1}^{N} c_i}, \quad c_i \cap c_j = \emptyset, \forall 1 \leq i \neq j \leq N.$$

A cell-centered approach is chosen with x_i the center of a given cell c_i. The discrete approximation of $u(x)$ on $\mathcal{H}$, denoted $u^{\mathcal{H}}(x)$, is defined from the values at the cells' center $u_i \doteq u^{\mathcal{H}}(x_i)$. The system of equations, usually non-linear, satisfied by the discrete solution $u^{\mathcal{H}}$ results from the discretization on the mesh of the model equations, *e.g.* conservation laws. The discrete problem is assumed to be well-posed, in the sense that there is a unique solution u^h associated to a given mesh $\mathcal{H}$, and that $u^{\mathcal{H}}$ converges toward u when the cell size h goes to zero. It is worth noting that the MMA method does not depend on a particular FV method and is readily applicable to other discretization approaches, *e.g.* finite elements and finite differences.

2.2. Mesh adaptation

The focus is on models involving complex physical phenomena that yield solutions u with complex spatial structures. These solutions require a fine mesh to properly capture the structures of the field and achieve an approximation error $|u - u^{\mathcal{H}}|$ small enough. As the computational cost of $u^{\mathcal{H}}$ increases with the number of mesh cells N, that is when the cell size h is reduced, a cost-accuracy trade-off question rises. Further, different areas of Ω may require different discretization efforts, and the mesh size must be adapted in space to provide the most suited aforementioned trade-off. Mesh adaptation strategies have been proposed to iteratively adapt the local mesh size to the solution and optimize the cost-accuracy trade-off. These methods are particularly appealing because they do not require a prior knowledge of the solution's structure and have demonstrated their capability to generate very effective adapted meshes, *i.e.* with minimal number of computational cells to achieve a prescribed accuracy.

2.2.1. AMR strategies

An iterative mesh adaptation algorithm is usually based on the analysis, at each iteration, of the solution $u^{\mathcal{H}}$ computed on the current mesh. This analysis provides quantitative insights on the modifications needed to generate a new mesh with better spatial discretization. The mesh adaptation strategies are generally classified in the literature into three categories [3]:

- The r-adaptation [19] consists in deforming the current mesh to optimize the position of the degrees of freedom according to the solution. In this case, the number of elements in the mesh remains constant, so no additional computational cost due to any mesh refinement is to be foreseen. The main drawback of this approach is that it can yield large mesh deformations with accuracy degradation as a result and could also impact the numerical stability of the calculations.
- The h and p-adaptations [20] change the number of degrees of freedom associated to the mesh. The h-adaptation consists in subdividing (isotropically or not) prescribed elements of the mesh according to a specified criterion. The criterion can also help coarsen the mesh by merging neighboring cells. The p-adaptation locally adapts the order of the discretization scheme according to the criterion. The p-method is mainly used with discontinuous Galerkin solvers [21].
- The m-adaptation method generates a new mesh over the entire domain according to a metric derived from the current solution $u^{\mathcal{H}}$. The m-adaptation is the foundation of the anisotropic mesh adaptation methods [1].

All these adaptation methods rely on a criterion, or metric, to guide the adaptation process. We can distinguish several types of criteria, all derived from the solution on the current mesh.

The first category of criteria, known as *feature-based*, is based on the solution structure. As its name shows, these criteria identify the structures of interest from the discrete solution analyses, for example: shocks, expansion fans [22], boundary and shear stress layers [23–25], vortex, recirculation zones [26], or turbulent areas [13,27].

Another family of criteria uses a posteriori error estimations. The most common strategy is based on the interpolation error of $u^{\mathcal{H}}$ to estimate the local error in each computational element. These error-based criteria have been in use since the 1980s and have proven their effectiveness [2,12,28,29]. Other approaches for local error estimation have been proposed in the literature, such as those based on energy norms [30].

Finally, *goal-oriented* criteria use an estimation of the error in user defined quantities of interests. In most cases, goal-oriented criteria use an adjoint method to estimate the local contribution of each computational element to the error in the quantity of interest (*e.g.*, aerodynamic coefficients). The method was introduced into CFD by [31,32].

2.2.2. Error based adaptation

In this section, the focus is on the adaptation of the mesh $\mathcal{H}$ for a specific condition p. To manage the adaptation, an h-refinement strategy is considered along with a criterion based on the error estimation. The objective is to minimize the global error induced by the discretization while controlling the number of elements N. The global error over Ω relative to the problem's true solution u is given by

$$\mathcal{E}_{\mathrm{u}}^{\mathcal{H}} = \int_{\Omega} \|u(x) - u^{\mathcal{H}}(x)\| \, dx = \sum_{c_i \in \mathcal{H}} \int_{c_i} \|u(x) - u^{\mathcal{H}}(x)\| dx, \tag{1}$$

where $\|\cdot\|$ is a suitable norm. Finding the mesh $\mathcal{H}$ with the smallest number of computational elements and minimizing $\mathcal{E}_{\mathrm{u}}^{\mathcal{H}}$ is a complex optimization problem usually intractable. Instead, an iterative adaptation process that refines some cells into smaller ones is preferred. At each iteration, the solution is computed on the current mesh and a subset of cells is selected for refinement using a flag $f_i \in \{0, 1\}$ to decide if the cell c_i should be refined ($f_i = 1$) or not ($f_i = 0$). To achieve an efficient reduction of the error $\mathcal{E}_{\mathrm{u}}^{\mathcal{H}}$ a criterion is needed to decide the flag's value. The MMA method proposed later is independent of the particular criterion but in the following the case of a criterion based on an estimation of the local error is considered. This criterion is introduced in the next section.

2.2.3. Interpolation error estimator

To set the flag value, a reliable and fast estimation of the local error is required. This study relies on standard interpolation error estimations [2,12], but the authors stress that the subsequent developments on MMA can be extended to more advanced error estimation strategies.

Starting with the one-dimensional case. The solution u is assumed to be sufficiently differentiable. $\Pi_h u$ denotes the interpolation of u over the mesh $\mathcal{H}$ from the values at the cells' centers x_i. In this work, piecewise constant (over the cells) interpolation schemes are used. Using the fundamental theorem of calculus and the Hölder's inequality, the error between the continuous function u and its piecewise constant interpolation satisfies, for $x \in c_i$, the inequality

$$\left|u - \Pi_h u\right|(x) = \left|\int_{x_i}^{x} u'(\xi) d\xi\right| \leq \max_{\xi \in [x_i, x]} |u'(\xi)| \left|\int_{x_i}^{x} d\xi\right| \leq \frac{h_i}{2} \max_{\xi \in [x_i, x]} |u'(\xi)|. \tag{2}$$

In (2), h_i refers to the cell size.

In practice, u is unknown and its derivative u' must be approximated from the values at the computed solution $u^{\mathcal{H}}$. We define $du^{\mathcal{H}}/dx$ as the (piecewise constant) derivative of a piecewise linear reconstruction of $u^{\mathcal{H}}$ over the cell centers. With this estimate of u', the final estimation of the interpolation error ε_i in the cell c_i is:

$$\varepsilon_i \doteq h_i \left|\frac{du^{\mathcal{H}}}{dx}\right|. \tag{3}$$

For higher order interpolation schemes, it is possible to derive similar interpolation error estimators, see [2]. In general, the degree of the n-interpolation error estimate will be based on the $(n+1)$th derivative of the solution.

In dimension $d > 1$, the common practice is to consider one-dimensional interpolation estimators in different directions [2] or to use the maximal value of the higher order derivatives of u in each cell [13]. In the present work, Eq. (3) is applied between a cell c_i and its neighboring cells $c_j \in \mathcal{N}_i$ to compute $\varepsilon_{i,j}$ from (3). Then, the interpolation error estimation is defined as

$$\varepsilon_i = \left(\sum_{c_j \in \mathcal{N}_i} (\varepsilon_{ij})^q\right)^{1/q}, \quad \varepsilon_{i,j} = \frac{h_{ij}}{2} \left|\frac{du^{\mathcal{H}}}{d\tau_{i,j}}\right|, \tag{4}$$

where $h_{i,j}$ and $\tau_{i,j}$ are respectively the distance and the direction between the two neighboring cell's centers. In practice, in the result sections below, a L2-norm ($q = 2$) is used.

2.2.4. Mesh refinement

The estimation of the interpolation error in (3) and (4) is used to estimate the error of $u^{\mathcal{H}}$ in (1) through

$$\mathcal{E}_{\mathrm{u}}^{\mathcal{H}} = \sum_{c_i \in \mathcal{H}} \int_{c_i} \left\|u(x) - u^{\mathcal{H}}(x)\right\| dx \approx \mathcal{E}_{\mathrm{int}}^{\mathcal{H}} = \sum_{c_i \in \mathcal{H}} \int_{c_i} \varepsilon_i dx = \sum_{i=1}^{N} |c_i| \varepsilon_i, \tag{5}$$

where $|c_i|$ is the cell's volume. To reduce the error, one wants to refine cells with highest contributions $|c_i|\varepsilon_i$ to the error. To this end, the cells meant to be refined are identified and flagged. Specifically, $f_i = 1$ is set to the fraction $0 < \alpha < 1$ of cells having the highest contributions, $f_i = 0$ to the others, and refinement is performed. The rationale for this setting of f_i is that, for smooth solutions, the refinement will roughly reduce ε_i by half while dividing the cell's size by half (h_i in (3) and $h_{i,j}$ in (4)). Note that the increase in the number of cells compensates for the reduction of the cell's volume.

For the sake of simplicity, a standard approach for steady state isotropic mesh refinement is considered. It consists in the subdivision of the cells with $f_i = 1$ into 2^d sub-cells. This approach to mesh adaptation results in a hierarchy of nested meshes that can be represented by a tree, providing a data structure which can be exploited to efficiently perform various operations, such as solution interpolation to generate an initial guess of the solution on the new mesh. These initial guesses greatly reduce the amount of computation needed at each iteration to reach the targeted solution convergence and the overall cost of adaptation. In practice, the subdivision phase uses the library `adaptCells` developed at ONERA [33]. The utility can also perform smoothing to limit the differences in neighboring cells' sizes. In addition, a hierarchical solution transfer is performed to initialize each iteration's solution from the previous and ensure fast solution convergence at each iteration. Note that this error-based mesh refinement strategy is transposable to other mesh adaptation methods (such as anisotropic mesh adaptation) as are the subsequent development of Section 3.

3. Mean mesh adaptation method

This section introduces the *Mean Mesh Adaption* (MMA) method which aims at building a single mesh for all considered input conditions, while controlling the resulting averaged error. The presentation starts with the context of variable conditions from which the simplified problem of controlling the average error over a set of continuous operating conditions arises. Then the average error for this problem is formulated and an estimation using quadratures and an interpolation error estimator is built. Special emphasis is placed on the advantages of using a Monte Carlo-type quadrature for MMA, as the iterative nature of adaptation allows for progressively improving these quadratures. The resulting MMA procedure is summarized in Section 3.3. The section concludes with a cost analysis of the proposed method.

3.1. Mesh adaptation with variable conditions

The case of a model problem involving parameters which can vary is considered. As an example, these parameters can be related to boundary conditions or model constants. In the present work, the situations of variable model forms or variable domains problems are not considered. Further, a probabilistic framework is introduced to describe the parameters variability. Specifically, let $(\Theta, \mathcal{A}, \mathbb{P})$ be a probability triplet, with Θ the set of events, $\mathcal{A}$ a sigma-algebra and $\mathbb{P}$ a measure of probability. Let $p(\theta)$ be an input parameter (also called condition) associated to the uncertain variable $\theta \in \Theta$, P be the range of $p(\theta)$ and f_p be the probability density function of $p(\theta)$. As a consequence, the model solution u will also depend on the parameter p and will appear as $u(x, p)$ in the following equations.

A robust approach to controlling the error would aim at constructing the mesh $\mathcal{H}$ with minimal number of cells satisfying the constraint

$$\mathbb{P}\left(\mathcal{E}^{\mathcal{H}}(p) > \mathcal{E}_{\text{tol}}\right) < \epsilon,$$

where $\mathcal{E}^{\mathcal{H}}$ is either the true or approximate error for p, $\mathcal{E}_{\text{tol}}$ is an error tolerance level and $\epsilon > 0$ is a prescribed maximum probability threshold. However, estimating this probability can be very costly, as it would require too many observations requiring the evaluation of many uncertain input conditions, especially for small values of ϵ. Therefore, a more feasible objective is pursued: it consists in minimizing the number of cells to achieve a prescribed average error. This approach is less constraining than enforcing a maximal error with high probability, and can produce a mesh yielding for some conditions errors that are much larger than the averaged one. The management of the resulting error distribution will be examined in the results section to evaluate the relevance of this choice.

3.2. Mean mesh adaptation

First of all, the expectation of a random variable $g(p)$ is defined as

$$\mathbb{E}_p\left[h\right] = \int_{\Theta} g(p(\theta)) d\mathbb{P}(\theta) = \int_{P} g(p) f_p(p) dp.$$

Extending (5) to the variable parameter case, the average error becomes

$$\begin{aligned}\mathbb{E}_p\left[\mathcal{E}_{\text{u}}^{\mathcal{H}}\right] &= \sum_{c_i \in \mathcal{H}} \mathbb{E}_p\left[\int_{c_i} \left\| u(x,\cdot) - u^{\mathcal{H}}(x,\cdot)\right\| dx\right] \approx \mathbb{E}_p\left[\mathcal{E}_{\text{int}}^{\mathcal{H}}\right] \\ &= \sum_{i=1}^{N} |c_i| \, \mathbb{E}_p\left[\varepsilon_i(\cdot)\right], \end{aligned} \tag{6}$$

where $\varepsilon_i(p)$ is given by the estimated error based on $u^{\mathcal{H}}(x, p)$. Eq. (6) shows that the average error is given by the sum over all the cells of the averaged estimated error $\varepsilon_i(p)$. Details of the estimation of these averaged interpolation errors and the resulting refinement strategy to minimize $\mathbb{E}_p\left[\mathcal{E}_{\text{int}}^{\mathcal{H}}\right]$ for the smallest number of cells are provided hereafter.

In general, the average of $\varepsilon_i(p)$ is not known in closed form. Therefore it is proposed to estimate $\mathbb{E}_p\left[\varepsilon_i\right]$ by means of quadratures of the form

$$\mathbb{E}_p\left[\varepsilon_i\right] \approx \widehat{\varepsilon}_i \doteq \sum_{k=1}^{N_{\text{p}}} w_k \varepsilon_i(p_k), \tag{7}$$

where $(p_k, w_k)_{k=1,\ldots,N_{\text{p}}}$ are respectively the quadrature points and weights. In the following $\mathcal{P}_{\text{c}}$ represents the set of quadrature points. For the choice of the quadrature rule, Gauss quadratures [34] have been selected and are convenient for standard densities f_p and limited d_p, leveraging sparse tensorization of one-dimensional quadrature rules. Gauss–Legendre quadrature will be tested in the examples hereafter. Deterministic quadratures are limited by the curse-of-dimensionality, thus stochastic quadratures are also considered, namely Monte Carlo (MC) integration [35]. Another motivation for using MC integration is the possible lack of smoothness in $\varepsilon_i(p)$ that can drastically deteriorate the convergence rate of the deterministic quadrature which would require an overwhelming number of points even in dimension 1. Further, an accurate estimation of $\widehat{\varepsilon}_i$ is not mandatory as the objective is to rank the cells from their potential error reduction when refined. In addition, the adaptation process being iterative, it is quite robust to imprecise ordering of the cells' average error. Indeed, missed cells still have the opportunity to be refined during subsequent iterations. In any case, the same strategy as for the case of a single condition is followed, flagging cells with $|c_i|\widehat{\varepsilon}_i$ value relatively high enough, according to the prescribed fraction α.

3.3. Mean mesh adaptation algorithm

The adaptation process has been developed following the MMA method described above. The full MMA algorithm is provided in Algorithm 1.

Even though it has not been evaluated in the present work, the algorithm enables the selection of the quadrature points p_k at each adaptation iteration which allows to adapt dynamically the order of the Gauss quadrature. It also enables the change of the conditions of the MC quadrature points, even for a fixed N_{p}. Changing the MC points at each iteration has a very beneficial impact on the behavior of the MMA method as it prevents the adaptation to get fixed on the errors associated with the conditions in $\mathcal{P}_{\text{c}}$, at the expense of the errors at other conditions. Randomizing the conditions in $\mathcal{P}_{\text{c}}$ at it each iteration also allows a reduction of N_{p}. This behavior has been investigated and presented in the results section.

When $\mathcal{P}_{\text{c}}$ is set, the solutions $u^{\mathcal{H}}(x, p_k)$ are computed and collected to estimate the local average error $\widehat{\varepsilon}_i$ using (7). In addition, the average error is estimated at each iteration using

$$\mathbb{E}_p\left[\mathcal{E}_{\text{int}}^{\mathcal{H}}\right] \approx \widehat{\mathcal{E}_{\text{int}}^{\mathcal{H}}} \doteq \sum_{c_i \in \mathcal{H}} \sum_{k=1}^{N_{\text{p}}} w_k |c_i| \, \varepsilon_i(p_k). \tag{8}$$

Algorithm 1 MMA adaptation

Require: $\mathcal{H}_0$, $\mathcal{E}_{\text{tol}}$
  $j \leftarrow 1$
  converged ← False
  while converged=False **do**
    Set N_{p} quadrature points and weights $(p_k, w_k)_{k=1,\dots,N_{\text{p}}}$
    for $k := 1$ to N_{p} **do**
      solve for $u^{\mathcal{H}_j}(p_k)$ on $\mathcal{H}_j$
    end for
    for each cell c_i compute local average error $\widehat{\varepsilon}_i$ ▷ Use equation (7)
    compute average error, $\widehat{\mathcal{E}_{\text{int}}^{\mathcal{H}_j}}$ ▷ Use equation (8)
    if $\widehat{\mathcal{E}_{\text{int}}^{\mathcal{H}_j}} < \mathcal{E}_{\text{tol}}$ **then**
      converged ← True
    else
      Set the flag f_i ▷ Use section Section 2.2.4 strategy
      Subdivide flagged cells, $\mathcal{H}_{j+1}$ ▷ With `adaptCells`
      $j \leftarrow j + 1$
    end if
  end while

The adaptation process stops when the estimated average error falls under a certain user-defined tolerance value $\mathcal{E}_{\text{tol}}$. Otherwise, the flags are set and the mesh is refined accordingly. This step may include advanced features, such as mesh smoothing and body fitting node adjustment. In any case, the framework is identical to the framework for the one condition adaptation, except for the evaluation of the N_{p} solutions, possibly performed in parallel, and the averaging of their local error estimates. When the conditions in $\mathcal{P}_{\text{c}}$ are kept unchanged from an iteration to the next, one can store the corresponding solutions $u^{\mathcal{H}}(x, p_k)$ to initialize the solve on the next adapted mesh. When, on the contrary, the conditions in $\mathcal{P}_{\text{c}}$ are resampled at every iteration, an overhead may be induced by lacking the knowledge of the solutions on the previous mesh. To mitigate this overhead, an initialization strategy is implemented. It consists in initializing the solver with the closest (in terms of Euclidean distance in P) solution on the previous mesh.

3.4. Performance evaluation

As one of the main goals of this study is to control the mean error on continuous sets of conditions, the strategy developed to accurately compare errors is based on the computation of the *a posteriori* true average error. Extending on Eq. (6) leads to the following formulation:

$$\mathbb{E}_p\left[\mathcal{E}_{\text{u}}^{\mathcal{H}}\right] = \sum_{c_i \in \mathcal{H}} \mathbb{E}_p\left[\int_{c_i} \left\| u(x,\cdot) - u^{\mathcal{H}}(x,\cdot)\right\| dx\right] \approx \widehat{\mathcal{E}_{\text{u}}^{\mathcal{H}}}$$
$$\doteq \sum_{c_i \in \mathcal{H}} \sum_{k=1}^{N_{\text{p}}} |c_i| \left\| u(x,\cdot) - u^{\mathcal{H}}(x,\cdot)\right\| w_k. \tag{9}$$

Here, $\widehat{\mathcal{E}_{\text{u}}^{\mathcal{H}}}$ is the estimation with a quadrature of the true average error $\mathbb{E}_p\left[\mathcal{E}_{\text{u}}^{\mathcal{H}}\right]$. The true average error is computed using accurate reference solutions u that can be exact analytical solutions of the problem or discrete solutions on very fine grids (at least as deep as the evaluated grid). In practice, the true average error on a given mesh is estimated with $\widehat{\mathcal{E}_{\text{u}}^{\mathcal{H}}}$ using a set of validation quadrature points of sufficient size for the measurement to be converged enough on every iteration's mesh compared to the MMA construction quadrature. In this work, the quadrature used for the true average error evaluation is the Monte Carlo quadrature with uniform sampling and equal weights.

3.5. Cost analysis

In this section, the goal is to provide simple computational cost estimates for the different adaptation methods studied, enabling a better understanding of the advantages and limitations of each. Only the costs of the mean and deterministic methods are compared, as nominal adaptation is a specific case of mean adaptation (with $N_{\text{p}} = 1$). Two types of cost are distinguished: evaluation cost and construction cost. The primary cost is the evaluation cost, which represents the actual computational expense required to compute the solution for a given condition with each method. This evaluation cost applies to each of the N_{e} evaluations that must be performed. The secondary cost, construction cost, pertains to MMA and represents the offline cost of building the mean mesh. For smaller values of N_{e}, construction cost is significant to the total cost of MMA. However, as N_{e} increases, this fixed cost becomes negligible. Therefore, since our focus is on developing a method to reduce the cost of mesh adaptation for controlling the average error, the construction cost is neglected and only the evaluation costs are compared for MMA and standard specific mesh adaptation (SMA) methods. The relevance of this assumption is quantitatively discussed in Section 4.2.4 through a representative test case.

By examining only the evaluation costs, the ratio between T_{MMA} and T_{SMA} provides a benchmark for the potential cost advantage of the mean method. Specifically, as $N_{\text{e}} \to \infty$, if $T_{\text{MMA}} < T_{\text{SMA}}$, then the mean method can outperform the deterministic one, though only for a sufficiently large number of evaluations. Otherwise, the mean method will not surpass the deterministic method.

Finally, two additional notations are introduced. The cost of solving the CFD problem for a given condition on a specific mesh, denoted T_{cfd}, and the cost of all intermediary adaptation operations (such as remeshing, criterion calculation, projection, etc.), referred to as T_{ad}.

During any mesh adaptation, the mesh size at iteration $k > 0$, N_k, depends only on the initial mesh size and the refined fraction (fixed in our case). Thus, $N_k = N_0 \prod_{i=1}^{k} G_\alpha = N_0 {G_\alpha}^k$. Where the parameter G_α is the increase in mesh size at a given iteration depending on the refined fraction α with the following relation $G_\alpha = (1 + (2^d - 1)\alpha), d \geq 1$. In this work, the cost model considered is of the form: $T = C.N^\gamma$, $\gamma \geq 1$. This formulation links a given iteration's CFD and adaptation costs ($T_{\text{cfd},k}$ and $T_{\text{ad},k}$) to the iteration's mesh size N_k and ultimately to the refined fraction. These costs are thus written:

$$T_{\text{cfd},k} = C_{\text{cfd}} {G_\alpha}^{\gamma_{\text{cfd}}.k} \quad T_{\text{ad},k} = C_{\text{ad}} {G_\alpha}^{\gamma_{\text{ad}}.k}. \tag{10}$$

This form introduces the γ_{cfd} and γ_{ad} parameters to adjust the scaling performance. High values of γ will model codes with bad scaling performances while values close to 1 will represent codes with perfect scalability. In addition, the initial mesh size N_0 is implicitly taken into account in the cost model constants C_{cfd} and C_{ad}.

Considering $n_{\text{MMA}}(\mathcal{E}_{\text{tol}})$ the number of adaptation iterations which are required to build a mean mesh suited for a prescribed error level $\mathcal{E}_{\text{tol}}$, the average evaluation cost $T_{\text{MMA}}(\mathcal{E}_{\text{tol}})$ on the mean mesh writes as follows:

$$T_{\text{MMA}}(\mathcal{E}_{\text{tol}}) = C_{\text{cfd}} {G_\alpha}^{\gamma_{\text{cfd}} \cdot n_{\text{MMA}}(\mathcal{E}_{\text{tol}})}. \tag{11}$$

For SMA the number of adaptation iterations depends on the condition considered and thus writes $n_{\text{SMA}}(\mathcal{E}_{\text{tol}}, p)$. The associated cost is :

$$T_{\text{SMA}}(\mathcal{E}_{\text{tol}}) = C_{\text{cfd}} \mathbb{E}_p\left[\sum_{k=0}^{n_{\text{SMA}}(\mathcal{E}_{\text{tol}},p)} {G_\alpha}^{\gamma_{\text{cfd}}.k}\right] + C_{\text{ad}} \mathbb{E}_p\left[\sum_{k=0}^{n_{\text{SMA}}(\mathcal{E}_{\text{tol}},p)} {G_\alpha}^{\gamma_{\text{ad}}.k}\right]. \tag{12}$$

An efficiency estimation is built to compare both methods' performance using $T_{\text{MMA}}(\mathcal{E}_{\text{tol}})$ and $T_{\text{SMA}}(\mathcal{E}_{\text{tol}})$:

$$\eta(\mathcal{E}_{\text{tol}}) = \frac{T_{\text{SMA}}(\mathcal{E}_{\text{tol}})}{T_{\text{MMA}}(\mathcal{E}_{\text{tol}})}. \tag{13}$$

Under the following hypothesis:

1. The adaptation cost is small compared to the CFD cost: $C_{\text{ad}} \ll C_{\text{cfd}}$.

2. The variations of $n_{\mathrm{SMA}}(\mathcal{E}_{\mathrm{tol}}, p)$ are small and : $C_{\mathrm{cfd}}\mathbb{E}_p\left[\sum_{k=0}^{n_{\mathrm{SMA}}(\mathcal{E}_{\mathrm{tol}},p)} G_\alpha^{\gamma_{\mathrm{cfd}}\cdot k}\right] \approx C_{\mathrm{cfd}} \sum_{k=0}^{\overline{n_{\mathrm{SMA}}(\mathcal{E}_{\mathrm{tol}})}} G_\alpha^{\gamma_{\mathrm{cfd}}\cdot k}$.

The efficiency η can be approximated in two parts as :

$$\eta(\mathcal{E}_{\mathrm{tol}}) \approx G_\alpha^{\gamma_{\mathrm{cfd}}(\overline{n_{\mathrm{SMA}}(\mathcal{E}_{\mathrm{tol}})}-n_{\mathrm{MMA}}(\mathcal{E}_{\mathrm{tol}}))} + \sum_{k=1}^{\overline{n_{\mathrm{SMA}}(\mathcal{E}_{\mathrm{tol}})}-1} G_\alpha^{\gamma_{\mathrm{cfd}}(k-n_{\mathrm{MMA}}(\mathcal{E}_{\mathrm{tol}}))}. \quad (14)$$

This formulation allows us to analyze the method's principal cost characteristics and their influence on η. Indeed, when $n_{\mathrm{MMA}}(\mathcal{E}_{\mathrm{tol}}) \leq \overline{n_{\mathrm{SMA}}(\mathcal{E}_{\mathrm{tol}})}$ the first term is always greater than one, therefore $\eta(\mathcal{E}_{\mathrm{tol}}) \geq 1$ and MMA is more efficient than SMA. Conversely, when $n_{\mathrm{MMA}}(\mathcal{E}_{\mathrm{tol}}) > \overline{n_{\mathrm{SMA}}(\mathcal{E}_{\mathrm{tol}})}$ no direct conclusions can be made on η as it's value will depend on the difference between the number of adaptation iterations of both methods and on γ_{cfd}. Nevertheless, small $\overline{n_{\mathrm{SMA}}(\mathcal{E}_{\mathrm{tol}})}$ compared to $n_{\mathrm{MMA}}(\mathcal{E}_{\mathrm{tol}})$ and big scaling γ_{cfd} will reduce efficiency. Finally, this estimation does not take into account adaptation costs T_{ad} (hypothesis 1) which will have a positive impact on η as it only increases T_{SMA}.

4. Application and evaluation of MMA

In the following sections, the performance of MMA in terms of error and cost is evaluated and compared to uniform refinement as well as existing adaptation methods used in the context of continuously variable conditions, using the evaluation methods described in Sections 3.4 and 3.5. To this end, the following adaptation strategies have been implemented for comparison with the MMA method:

- **Uniform refinement (UR)** which subdivides all cells of the mesh at each adaptation iteration (which is equivalent to $\alpha = 1$), the error convergence with this method should reflect the order of the numerical method employed, therefore it is used as a reference for error comparisons.
- **Specific mesh adaptation (SMA)** for which a specific adaptation for each condition to be evaluated is performed. With this method, the error level is well controlled, thus it will be used as an error convergence best case scenario.
- **Nominal mesh adaptation (NMA)** which uses the mesh adapted for a single specific flow condition (referred as the nominal condition) and calls the resulting mesh to tackle any other condition calculations. While the cost of this method stays low, the quality of the results cannot be ensured for most conditions. Hence error levels higher than those of SMA are expected.

The results are presented on two finite volumes problems of increasing complexity: A one-dimensional parametric Burgers equation and a two-dimensional Supersonic Scramjet configuration solved with Euler equations.

4.1. Burgers case

The generalized 1D Burger's unforced convection–diffusion equation has been selected as a relevant test case for its analytical solution properties [36].

$$\frac{du}{dt} + u\frac{du}{dx} - \nu\frac{d^2u}{dx^2} = 0, \quad (15)$$

The computational domain and the following boundary conditions have been chosen:

$$x \in [0;1], \quad u(x=0,t)=1, \quad u(x=1,t)=-1. \quad (16)$$

With these conditions, the monotone final solution highlights a strong gradient in a small zone around a position x_s, this size being controlled by the viscosity parameter, ν (see Fig. 1). The initial condition $u(x, t = 0)$ is set using the following formulation:

$$u(x,t=0) = \begin{cases} 1 & \text{if } x \leq x_s \\ -1 & \text{else} \end{cases} \quad (17)$$

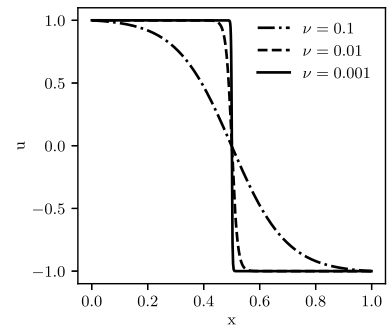


Fig. 1. Studied exact solution of the Burgers equation for different viscosity values. ($x_s = 0.5$).

In an infinite domain setting, the strong gradient final position is precisely centered on x_s at convergence. Here, because the computation is performed on a finite domain, a slight flux imbalance can be created at the domain boundaries convecting away from x_s the strong gradient. As small values of ν will be considered in this work, the time scale of this phenomenon is several orders of magnitude longer than the initial step diffusion, and is therefore negligible for the targeted convergence level.

The ν value chosen for the tests is 0.001 to generate a strong localized gradient in the computational domain. The problem is numerically solved using a cell-centered finite volume method along with an implicit Euler temporal integration which time-step is computed by enforcing a CFL number. The convective fluxes are discretized using a Roe scheme [37] while the diffusive viscous fluxes are discretized with a centered scheme.

The starting mesh for every adaptation is coarse with 25 uniformly distributed cells over the $[0, 1]$ range and the non-linear system is solved using the *PETSc* library [38] until reaching a residual target of 10^{-9}.

To perform accurate error computations, the numerical solutions are compared to the analytical solution of the problem under the assumption that the strong gradient zone stays far from the domain's boundaries and that ν is small. This solution writes as follows:

$$u(x,p) = -\tanh\left(\frac{(x-x_s)}{2\nu}\right). \quad (18)$$

In Figs. 2(a) a consistent convergence of the solutions toward the reference analytical solution is observed on uniformly refined grids.

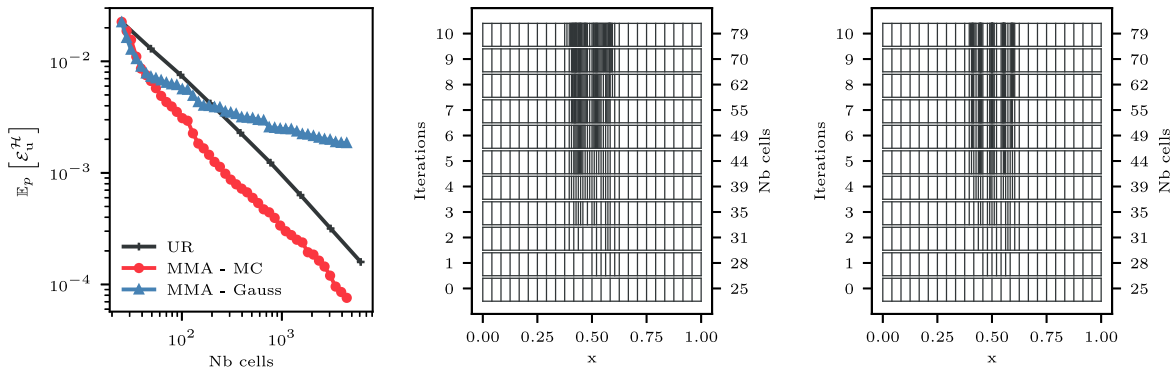
4.1.1. Experimental setup

To compute the mean mesh criterion and validate the results, the positions of the strong gradient zone, x_s are chosen uniformly spread between 0.4 and 0.6 following the law $x_s = 0.4 + 0.2p$, $p \sim \mathcal{U}(0,1)$.

The true average error ($\mathbb{E}_p\left[\mathcal{E}_{\mathrm{u}}^{\mathcal{H}}\right]$) and the cost are estimated using Eq. (9), with 500 Monte Carlo quadrature points, at each adaptation iteration and for each method. For each method, this requires evaluating 500 solutions at each iteration on the current mesh, along with an additional 500 reference solutions computed using the analytical relationship provided above.

4.1.2. Impact of the quadrature method

As explained in the previous sections, the MMA method requires to choose a quadrature to approximate the averaged error over the cells. In a first experiment, the convergence of the average error is compared with the number of cells in the mesh for quadrature sets of size $N_{\mathrm{p}} = 5$ and a refinement fraction $\alpha = 0.14$. The cases of Gauss and MC quadratures are contrasted, together with the evolution of the error for the UR method. For the MC quadrature the random conditions p_k are sampled using an uniform law with uniform weights $w_k = \frac{1}{N_{\mathrm{p}}}$ The results are shown on Fig. 2(a).



(a) Average errors as functions of the number of cells (Nb cells) in the mesh.

(b) Grids with MC quadrature.

(c) Grids with Gauss quadrature

Fig. 2. Comparison of average error and mesh evolution with different MMA quadratures (with $\alpha = 0.14$ and $N_p = 5$).

The first observation is that the uniform refinement yields a first order convergence, as can be expected. The two MMA variants perform initially better than the uniform refinement, with a slight advantage to the Gauss quadrature during the first few iterations. Subsequently, the MC quadrature continues to out-perform the uniform refinement, while the decrease of error is seen to slow down for the Gauss quadrature. Eventually, the averaged error for the Gauss quadrature becomes larger than for the uniform refinement, when the MC quadrature seems to converge at order one.

To understand this behavior, Figs. 2(b) and 2(c) illustrate the sequences of meshes generated by the two quadratures for the first 10 iterations. For both methods, the refinement is located in the central area ($x \in [0.4, 0.6]$) of the domain corresponding to the domain of variability of x_s and where the steepest gradient of u is expected. The MC quadrature which resamples the $N_p = 5$ conditions at each iteration leads to a rather uniform refinement of this central area. In contrast, the Gauss quadrature keeps concentrating the refinement effort around the particular locations corresponding to the location of the maximum gradients (x_s) for the fixed $N_p = 5$ Gauss points and disregards the refinement in between. This behavior is related to the limitation of the Gauss quadrature to accurately approximate the discontinuous integrands with the limited number of fixed quadrature points it uses. A more precise approximation would require more fixed Gauss quadrature points inducing a higher cost. This is amplified when using lower refinement fraction values and by the fact that there are no constraints on neighboring cell size ratios. The MC quadrature would suffer from the same behavior if the quadrature conditions were not resampled at each iteration of the MMA method effectively improving the quality of the quadrature. Further, increasing N_p delays the stagnation of the error for the Gauss quadrature (not shown), but does not improve much the error of the MC quadrature (see Fig. 3(a)). It can be seen that asymptotically, the MC quadrature needs about 1/5 the number of cells of the uniform refinement to achieve a comparable error level, a ratio corresponding roughly to the fraction of the domain area requiring adaptation.

Because of the finding of these experiments, only the MC quadrature is kept in the rest of the section.

4.1.3. Performance of the adaptation approaches

Now that the quadrature for MMA has been down-selected, its performance against other adaptation methods is investigated. Fig. 3(a) shows the evolution of the average error during the adaptation process for the MMA, UR, SMA and NMA methods. The refined fraction α is kept to 0.14 and the number of quadrature conditions N_p is set to 20 to ensure a good convergence and alleviate the small oscillations observed with $N_p = 5$. Additionally, a 95% confidence interval on the average error has been computed by bootstrapping at each adaptation iteration and for each method to assess the results' dependence on the number of validation conditions. The maximum value of the confidence interval relative to the average is reported in Table 1.

Table 1
Mean Bootstrap 95% confidence interval spread, in percent of average error, for each method.

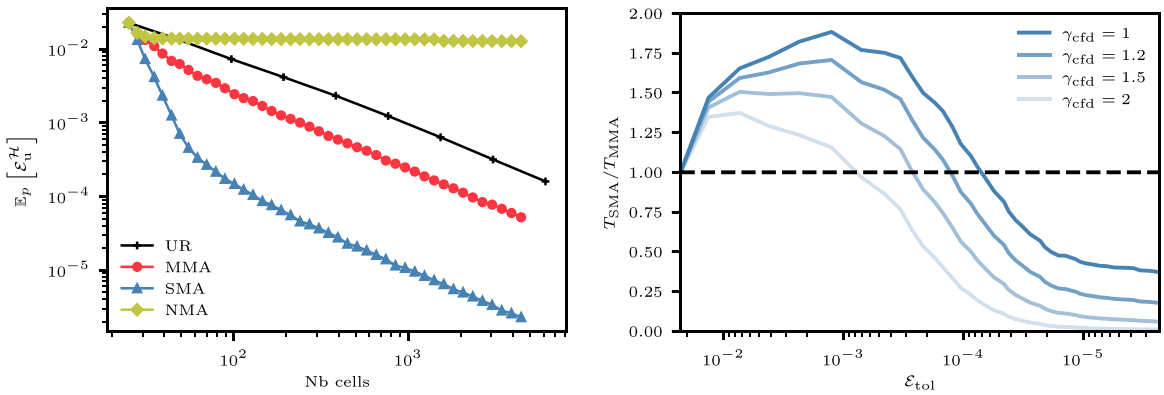
	UR	MMA	SMA	NMA
Mean error on $\mathbb{E}_p\left[\mathcal{E}_u^{\mathcal{H}}\right]$ (%)	1.977	3.078	0.832	6.429

For all the tested methods, the relative spread of the confidence interval on the average is only a few percent, while the differences between the methods observed in Fig. 3(a) span several orders of magnitude. Therefore, the observed discrepancies between methods are not due to the validation strategy, but rather reflect the distinct behaviors of the methods themselves.

The error convergence trend remains the same when increasing N_p. Consequently, N_p could be kept small to reduce construction cost for cost intensive cases. Nevertheless, with a greater value of N_p the error convergence is smoother meaning that the error reduction is less influenced by individual quadrature conditions. Indeed, because the average error estimator is averaged over more quadrature conditions, the error variations induced by individuals have a smaller impact on the mean estimation. These results foreshadow a good robustness of the method that will be investigated in detail in a next section.

With the SMA method, the error is very significantly reduced at the beginning of the process, due to the small size of the zone of interest for each adaptation (one at a time). Then, a gradual return to order 1 decrease is observed as the proportion of cells in the zone of interest increases. To achieve an error level of 10^{-3}, the method requires 50 cells in each of the meshes, which is five times less than for a mean mesh. However, this number needs to be considered in perspective with the cost of such an approach, as each of the subsequent evaluations of conditions requires a complete mesh adaptation. Therefore, this method is not feasible for evaluating a large number of conditions (see next section). Although the first two adaptation iterations of the NMA method allow for an error reduction equivalent to the other methods, due to yet coarse meshes, the progressive specification of the mesh on a single operating condition limits the achievable progress and the average error reaches a plateau. Sharing a mesh among multiple conditions is therefore a significant source of error. Thus, the NMA method, although inexpensive in a multi-condition context, has limited usefulness. The issues identified with both of these approaches further highlight the importance of designing a low-cost and accurate adaptation method in a multi-condition context.

In addition to the previous error analysis, to better understand how the MMA and SMA methods compare, their computational cost is



(a) Evolution of the average error with number of cells (Nb cells). (b) Evolution of the modeled evaluation cost efficiency of MMA compared to SMA as a function of error target

Fig. 3. Comparison of uniform refinement, MMA, SMA and NMA methods ($\alpha = 0.14$ and for MMA $N_p = 20$ and quadrature is MC).

analyzed. Using the methodologies developed in the previous section, the ratio between the cost per evaluation to reach a specific error level (efficiency) has been computed for the MMA and SMA methods. For the Burgers case this ratio is represented against the error level on Fig. 3(b) for different solver scaling (γ_{cfd}). Here γ_{cfd} is chosen to vary between 1 and 2, 1 being the best case scenario as the convergence time per element is not increasing, while 2 represents the worst scaling scenario. The observations made with our 1D Burgers solver here would be close to a scaling of 1. For errors levels above 5.10^{-5} the observed efficiency, for $\gamma_{cfd} = 1$, is greater than one. Therefore, if the objective is to make a mesh suited for a higher average error level, the cost of adapting the mesh for each condition will be higher than the cost of evaluating the solution for this condition on the equivalent mean mesh. The position of the efficiency intersection point depends on the chosen adaptation parameters because α directly influences the slope of the error convergence for the SMA adaptation. The scaling γ_{cfd} also influences the intersection point position, as it gets bigger the majority of the cost gets concentrated on the biggest meshes, thus the cost of computing on the mean mesh gets higher than the cost of computing each of the smaller successive SMA meshes.

It is clear that the Burger case chosen here is very challenging for the MMA process because of localized solutions and large studied range of conditions. Nevertheless, the method is capable of good cost reductions as long as the target error level is not too low and the solver scales well with the mesh size. In any case, the method still benefits from its simplicity, as it allows to control the average error of the solutions without having to parametrize and run full adaptation processes once it is built. Finally, only the cost of the solution computation has been considered here without any overhead costs which could penalize SMA compared to MMA making the latter a better choice.

4.1.4. Robustness of MMA

The convergence of $\mathbb{E}_p\left[\mathcal{E}_u^{\mathcal{H}}\right]$ is a key information in analyzing the quality of meshes resulting from MMA process. Indeed, if the number of quadrature conditions evaluated to build an adapted mesh is high, then its performance should converge towards the ideal mean mesh (uniformly refined over the interval $[0.4; 0.6]$). Thus, the role of N_p is decisive, as the higher it is, the more representative the obtained mean mesh is. Yet, due to the nature of the mesh adaptation process with MC quadrature, the number of iterations also has a significant impact on the effective number of quadrature conditions. As $\mathcal{P}_c$ is changed at each adaptation iteration, the effective number of quadrature conditions is multiplied by the number of iterations. However, to achieve a target error level or a given number of cells, it is the refined fraction α that determines the number of iterations. The effective number of quadrature conditions is in fact proportional

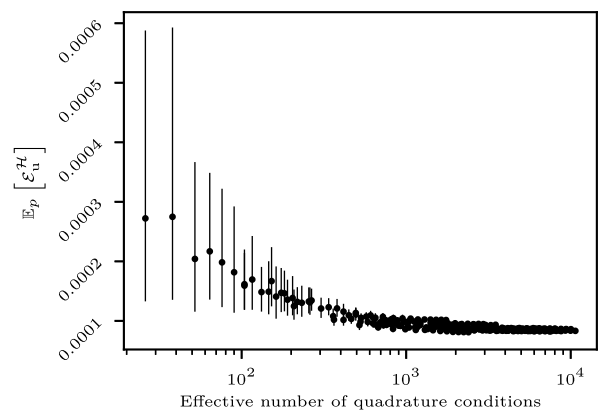
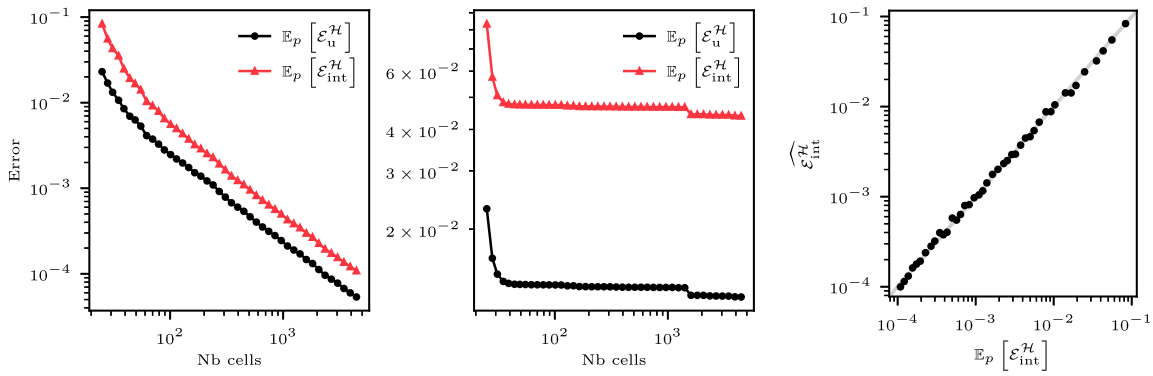


Fig. 4. Average and 95% confidence interval of the error on 40 realizations of the $N_p \in [2; 92], n_{MMA} \in [12; 102]$ design of experiments, using the MMA method with MC quadrature.

to $\frac{N_p}{\ln(1+\alpha)}$(Indeed, $N_{p_{eff}} = N_p.\frac{\ln(N_{n_{MMA}})-\ln(N_0)}{\ln(1+\alpha)} \propto \frac{N_p}{\ln(1+\alpha)}$). In order to quantify the impact of the effective number of quadrature conditions on MMA and better understand its influence on the process robustness, a set of numerical experiments is conducted. The adaptation processes are stopped when the refined mesh reaches a given size, $N = 2500$ and the error on this mesh is recorded. To obtain comparable results, the mesh resulting from each adaptation process must be as close as possible to 2500 cells. This is ensured by determining the parameter α based on a desired mean number of adaptation iterations n_{MMA}. The resulting experimental design is therefore: $N_p \in [2; 92], n_{MMA} \in [12; 102]$ (the range of variation of α is close to $[0.05; 0.5]$). Additionally, the experiments are repeated 40 times to analyze the variability from one adaptation to another induced by the randomness of $\mathcal{P}_c$, for the same parametrization.

Fig. 4 summarizes the average measured error over the 40 realizations of the experiments as a function of the effective number of quadrature conditions, as well as the associated 95% confidence intervals. On Fig. 4, as the number of effective quadrature conditions increases, the average error converges towards a fixed value. At the same time, its confidence interval is also greatly reduced. With a small number of effective quadrature conditions, for example 24 (the lowest tested), a mesh of 2500 cells created by mean adaptation achieves, with 95% confidence, an average error ranging from 1.1×10^{-4} to 6×10^{-4}, an interval 120 times wider than for a high number of effective quadrature conditions (around 10000). Moreover, on average, the level of error reached when the number of effective quadrature conditions is



(a) Evolution with number of cells (Nb cells) of the true average error and it's estimation with the interpolation error for the MMA (left) and NMA (right) methods.

(b) Comparison of the average interpolation error and its estimation on the quadrature conditions.

Fig. 5. Evaluation of the average error estimation (with $\alpha = 0.14$, $N_{\mathrm{p}} = 20$ and MC quadrature).

low is 3 times higher than for a high number of effective quadrature conditions. Therefore, in addition to influencing the variability of the meshes obtained after the adaptation process, the number of effective quadrature conditions also impacts the achievable error level.

This result confirms the hypothesis formulated earlier, the average error tends towards a minimum value as the number of quadrature conditions increases. The meshes created under these conditions therefore tend towards a density that minimizes the error expectation and consequently towards the ideal average mesh, with a high level of confidence. To design good mean meshes, one needs to maximize the number of effective quadrature conditions by choosing a very large value for N_{p} and/or a very small value for α. However, drawing a large number of quadrature conditions involves a large number of computations and therefore a high cost to build the mesh, making MMA difficult to apply.

4.1.5. Use of interpolation error as an average error estimator

As an analytical solution may not be available for the problem at hand, one would need to rely on the interpolation error estimator to evaluate the quality of the meshes. To ensure this approach will not misrepresent the error behavior, the error from the analytical solution $\mathbb{E}_p\left[\mathcal{E}_{\mathrm{u}}^{\mathcal{H}}\right]$ has been compared to the interpolation error estimate $\mathbb{E}_p\left[\mathcal{E}_{\mathrm{int}}^{\mathcal{H}}\right]$ for the MMA and NMA adaptation methods in Fig. 5(a).

In the case of the MMA method, the convergence for both errors is similar and the error estimated by interpolation behaves as an upper bound to the true value as could have been expected. This result shows that error in the Burgers solution considered is dominated by interpolation errors. Indeed, the shock position is well defined from the beginning of the adaptation (on the coarsest mesh) and will not drift significantly because of the boundary conditions. Even in the unfavorable nominal case for which the refinement is not adapted to most conditions, the interpolation errors dominate, thus the error estimated by interpolation bounds the true value. For more complex applications, the interpolation error might not be as dominant because feature placement can vary with the resolution and can be interdependent, making the approach misleading. At this stage no definitive conclusion can be drawn.

4.1.6. Computation of the interpolation error on the quadrature points

The previous results have shown that the interpolation error is a good, low cost, error estimate for the MMA method. Nevertheless, the average error is still estimated on a large independent set that needs to be computed in addition to the quadrature conditions for validation studies. To alleviate this additional cost, the average error during the adaptation process needs to be directly computed on the successive quadrature sets conditions. In order to validate this approach, the average error using the interpolation error estimator has been computed on the validation set and on the successive quadrature sets. Both methods are compared on Fig. 5(b).

When performing the error measurement on the quadrature set, no additional cost is required since the error is the same as the one used for refinement. Also, this allows to evaluate on the fly the solution's quality to inform a stop criterion for the adaptation. A strong agreement is observed, meaning that estimating the error on the quadrature set should not deteriorate the observations. This also emphasizes the fast convergence of the MMA error estimator in the tested case to build a flag function. Finally, because the refinement is performed after the error estimate computation during an adaptation step, there is no risk of contamination effects from refining the mesh for the same conditions as the ones used in the estimation.

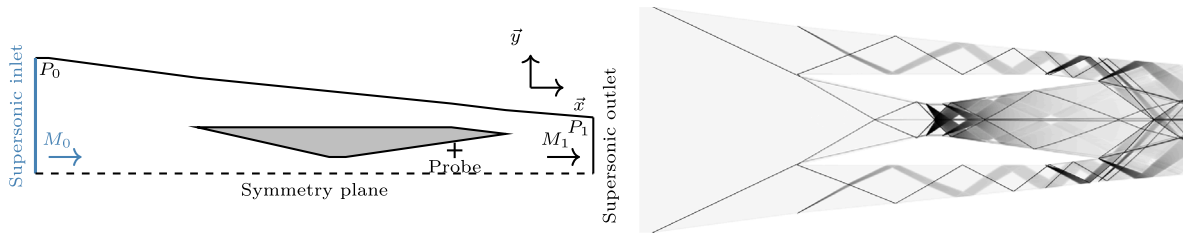
4.2. Scramjet case

In this second section, the MMA method is analyzed on a higher complexity two-dimensional fluid dynamics problem. A 2D scramjet inlet configuration. This device is meant to adapt supersonic and hypersonic upstream conditions and compress the flow field to create optimal conditions for a combustion chamber downstream. Thus, it operates at a supersonic regime generating complex shock interaction patterns.

The specific scramjet inlet configuration considered is a symmetric two-strut inlet originally introduced by [14] in 1981 and widely used in the mesh adaptation literature [15–17,39]. Its geometry along with the computational domain configuration is presented on Fig. 6(a).

The domain is divided into 4 parts: the upstream section, a central duct, two outer top and bottom (by symmetry) ducts, and a downstream section where the duct flows merge. In addition, a probing point is placed in the diverging section of the central duct to monitor the flow.

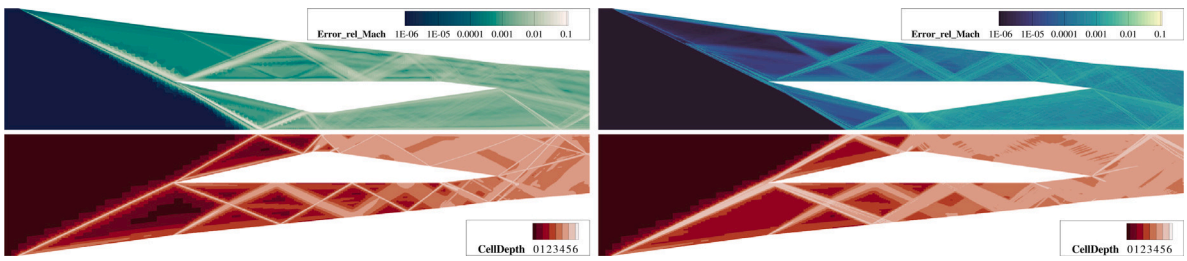
Shock boundary layer interactions and flow separations are common in scramjet flows and can have a huge impact on their behavior. Nevertheless, to focus the study on shocks resolution, the scramjet's flow will be modeled using Euler equations closed for a perfect gas. For the whole study, the gas coefficient is fixed to 1.4. At the nominal condition, the flow enters the domain at $M_0 = 3$ and creates oblique shocks, expansion fans and complex interactions in both the central and outer ducts. As can be seen on Fig. 6(b). Symmetric oblique shocks are induced by the upstream section reduction, and meet the struts leading edge. In the outer ducts, their reflection bounces on the walls up to the outlet. The successive section modifications on these ducts create several expansion fans and oblique shocks that interact with each other in the downstream section. In the central duct, the upstream oblique shocks meet and reflect in the converging section. At the throat, new expansion fans and



(a) Computational domain configuration.

(b) Numerical Schlieren $M_0 = 3$.(symmetrized)

Fig. 6. Scramjet problem.



(a) Nominal adapted mesh for $M_0 = 3$ (384K cells): local average error (top) and refinement depth (bottom).

(b) MMA mesh (470K cells): local average error (top) and mesh refinement depth (bottom).

Fig. 7. Comparison of the meshes and local average errors for the NMA and MMA methods. The adaptations use 10 iterations from the initial mesh and $\alpha = 0.3$ (for MMA $N_p = 20$ and the quadrature is MC).

shocks are created, resulting in interacting reflected structures in the diverging and downstream sections. As can be seen on Fig. 6(b) the union of the central and outer flows along with trailing edge structures make the downstream section flow highly complex and interdependent.

The Euler equations for the case are solved by *elsA* [40] a finite volume and cell centered solver (ONERA-SAFRAN property). For the finite volume discretization of the equations, the Roe scheme [37] is used, it is well suited for configurations with shocks since it creates minimal numerical dissipation. While the cases are solved using an implicit backward Euler time integration and a LU-SSOR type linear solver. For the computations, the time step is locally enforced by using a target CFL number at each iteration, and the computation is considered converged when a residual reduction of five orders of magnitude is achieved, or when a maximum of 15,000 solver iterations is reached. The mesh adaptation strategy used in this paper works only on unstructured grids, therefore all the calculations are performed using *elsA*'s unstructured capabilities.

4.2.1. Experimental setup

The continuous parameter that is chosen to adjust the flow conditions is the inlet Mach number M_0. It affects the inflow condition and greatly affects the shock structures' position and interactions in the domain. The range of Mach number considered in this study goes from $M_0 = 2.9$ to $M_0 = 3.3$. The lowest bound has been chosen to avoid the formation of a strong shock at the central duct throat which results in a significant flow regime change.

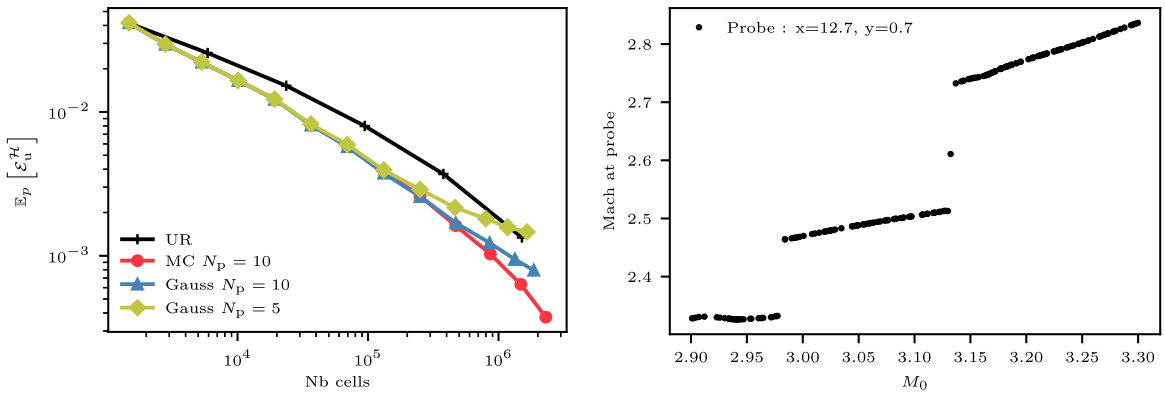
The error measurements to evaluate the adaptation methods on this case are computed using a thin uniform reference mesh since there is no analytical solution available. This mesh has been uniformly subdivided six times (depth of 6) to reach a size around 6M cells. This size has been chosen mainly for computational cost reasons. For the error measurements to be consistent, the cell size of the meshes that are being evaluated must remain smaller than the cell size of the reference, so that all meshes are embedded in the reference mesh. Therefore, in what follows, the depth of all adaptations is limited to 6 levels (equivalent to the reference). In addition, it has been ensured that the error level reached on the thin reference meshes is equivalent for all tested conditions.

In this case, the error measurement is estimated using 200 Monte Carlo samples for each iteration of each method. As with the Burgers case, this represents a significant computational effort, as 200 computations must be performed on each iteration's mesh, along with an additional 200 computations on the reference thin mesh.

4.2.2. Local error and mesh refinement

Before presenting in details the performances and characteristics of the newly developed MMA method, and to better understand the accuracy advantages of MMA compared to NMA, the local error relative to the nominal inlet Mach number along with the refinement depth have been plotted on Figs. 7(a) and 7(b) for the final meshes of each method.

The Fig. 7(a) highlights the important discrepancies in the behavior of the MMA and NMA methods. With the NMA method the error reaches up to 10% of the nominal value in the upstream oblique shock region and converging section of the central duct. It is consistently above 1% in most of the central and outer ducts as well as in the downstream section. Localized lower error values in this high error region are due to the deeply refined features of the nominal condition (especially on the upstream oblique shock). NMA especially fails in the central duct diverging section and the downstream section where the flow topology undergoes drastic changes. In this area, the refinement is almost as deep as with MMA meaning that the error here comes mainly from the misplacement of the flow features induced by upstream shortcomings of the mesh. For these reasons, it is clear that the NMA method is unable to handle important conditional variabilities and is not performing for flow topologies far from the nominal one. The error map resulting from MMA also highlights some interesting features. First, the upstream section error level is close to zero as its uniformity leads to expect, resulting in low refinement depths. Then, low error levels are measured in the upstream oblique shock region since this shock is not very sensitive to the flow conditions, fine refinement has been possible in the regions it covers (see Fig. 7(b)) resulting in good error control. In the outer ducts, a refined zone is observed around the reflections of the upstream oblique shock and around the reflections of an expansion fan. With the successive reflections, the refinement band from the oblique shock gets larger, covering an increasingly large area



(a) Average error depending on the number of cells (Nb cells) for the different MMA quadratures (with $\alpha = 0.3$).

(b) Local Mach at the probe point as a function of the inflow Mach number.

Fig. 8. Influence of the MMA quadrature and local Mach number regularity.

of variation of the features in this region. Overall, where the sensitivity of the phenomena to the flow conditions increases, the measured error level gets higher. Especially in the downstream and the central duct diverging sections. In these regions the mesh depth is almost uniform and has not yet reached the same level as in the upstream shock region. The error is well managed by the MMA method and the relative error does not get higher than 1%. It is also relatively uniform as no strong error variations are observed on the challenging central duct and downstream regions.

Additionally, there is a complementary footprint on the mesh depth and error map left by the random sampling of the previous adaptation iterations. Especially after the first reflection of the upstream oblique shock. The over-representation of specific conditions at this location indicates a lack of convergence of $\widehat{\varepsilon}_i$ at the last iteration. Since the quadrature set is renewed at each adaptation iteration, this lack of regularity is meant to gradually disappear.

Finally, the sensitivity of the local phenomena with respect to the variable conditions seems to be an important aspect driving the MMA performance. Indeed, where numerous phenomena sensitive to the varying conditions are known to occur, like in the central duct diverging section and the downstream section, to control the average error, most of the cells must be refined, resulting in almost uniform refinement. Whereas in the outer ducts and the upstream section, the areas with strong average error are localized and the refinement can be non-uniform. The main benefits brought by MMA on this configuration are therefore upstream of the domain and the outer ducts where the refinement remains localized. In general, the MMA method performance compared to UR is conditioned by the overall sensitivity of the flow to the conditional variability imposed.

4.2.3. Impact of the quadrature method

In 1D, the Burgers case analysis has shown that using a Gauss quadrature leads to error stagnation due to over-adaptation on the sample conditions, as there is no sampling update at each adaptation iteration. In order to verify this behavior on a higher dimension and a more complex case, the error convergence on the scramjet has been reported on Fig. 8(a) for the MC quadrature (with $N_p = 10$) and the Gauss quadrature (with $N_p = 5$ and $N_p = 10$), along with the UR results.

Using Gauss and MC sampling yields comparable error levels until the 9th step for $N_p = 10$ and the 6th step for $N_p = 5$, after this point the error convergence is significantly slowed down for Gauss sampling. The nature of the local solution evolution when the condition changes, plotted at the probe point (see Fig. 8(b)) explains this behavior. The shocks in the domain are localized phenomena that induce steep Mach number variations when M_0 changes. Because of that, the interpolation error is composed of spikes, impossible to accurately and reliably

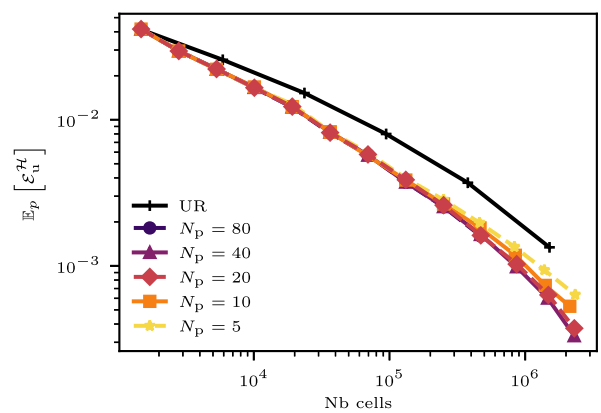


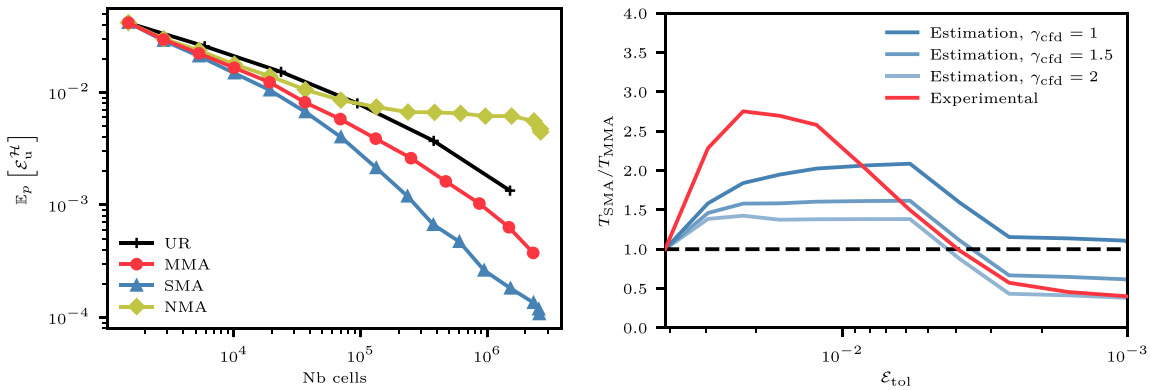
Fig. 9. Average errors depending on the number of cells (Nb cells) for different quadrature set size N_p (with $\alpha = 0.3$ and MC quadrature).

capture with a Gauss sampling. In this case, the large value for α chosen and the small position variation of some important features has allowed the Gauss sampling to keep up with the MC sampling longer than in the Burgers case.

On the other hand, not changing the quadrature set at each iteration with the Gauss sampling has resulted in significantly better initialization states and faster computational convergence. That makes it an interesting method to reach less demanding error levels faster. As the robustness of the method is a main focus of this work, MC sampling is kept for the rest of the scramjet study.

The size of the initial mesh for the scramjet (1500 cells) is of the same order as the size of the last adapted meshes from the Burgers case. Also, because the case is 2D and the adaptation isotropic and hierarchical, each cell is subdivided into 4 cells (instead of 2) when flagged. Therefore, the cost of the refinement is higher making it even more critical to find the best compromise between cost and accuracy when choosing N_p. That is why the error convergence from five different values of N_p has been compared on Fig. 9.

Overall, all the configurations tested give close results with limited discrepancies for the last adaptions with $N_p = 5$ and $N_p = 10$. It should be noticed that the sensitivity to N_p is negligible between 40 and 80 as curves are superimposed. The high α value chosen along with the good robustness of the random adaptation process make the required number of effective quadrature conditions lower for this configuration. In the following, the number of quadrature conditions has been fixed to $N_p = 20$, to keep the cost under control while achieving high quality mean meshes.



(a) Evolution of the average error with number of cells (Nb cells). (b) Evolution of the modeled and measured evaluation cost efficiency of MMA compared to SMA as a function of error target

Fig. 10. Comparison of UR, MMA, SMA and NMA methods ($\alpha = 0.3$ and for MMA $N_p = 20$ and quadrature is MC).

Table 2
Mean Bootstrap 95% confidence interval spread, in percent of average error, for each method.

	UR	MMA	SMA	NMA
Error on $\mathbb{E}_p\left[\mathcal{E}_u^{\mathcal{H}}\right]$ (%)	0.620	2.085	1.297	4.013

Table 3
Construction ($T_{construct}$) and evaluation cost measured experimentally for MMA and SMA methods, target error 1.2e−2.

	$T_{construct}$ (CPUh)	T_{cfd} (CPUh.p^{-1})	T_{ad} (CPUh.p^{-1})
MMA	32.5	0.22	0.0
SMA	0	0.56	0.42

4.2.4. Performance of the adaptation approaches

The performance of the MMA method is compared to the UR, SMA and NMA methods similarly to the 1D Burgers study. The Fig. 10(a) shows the evolution of the average error, with $N_p = 20$ and $\alpha = 0.3$ for each of these methods. In addition, since the number of validation conditions is limited (200), a 95% confidence interval on the average error has been computed by bootstrapping at each adaptation iteration and for each method. The maximum value relative to the average is reported in Table 2.

For each of the tested methods, the error on the average error is within a range of less than 4% around the mean, while the differences between methods observed in Fig. 10(a) are at least one order of magnitude larger. Therefore, the observed differences are not due to the validation strategy, but rather result from actual discrepancies between the methods.

Therefore, the analysis of Fig. 10(a) confirms that the worst performing method is NMA. During the first iterations, it leads to similar error reduction levels as the other methods, but from the fifth iteration, the error reaches a plateau almost until the last iteration. This plateau is due to the specification of the mesh for the nominal condition ($M_0 = 3$). The depth of the mesh is limited on the studied case. Therefore, as more and more high error cells get to the maximum depth, the amount of refined cells decreases. This amount ultimately goes to zero when 30% (the refined fraction chosen) of the biggest error contributors get to the maximum depth. In the case of the NMA, the cells with the biggest error, localized on the phenomena of the nominal flow condition, are first to reach the limit. Progressively, the other cells with the refined fraction limits are the only ones left to be refined. Due to this mechanism at the end of the adaptation, while the mesh size does not increase significantly, the average error is still reduced, resulting in a fast error convergence for the last adaptation iterations. This is of short duration since the mesh converges fast toward its final state.

Since the SMA adaptation process produces a mesh for every condition, the error level is always controlled and therefore the average error is lower than for the other methods. In fact it is impossible for the MMA and UR methods to do better since creating a single mesh for different flow conditions will necessary lead to mesh size or error convergence inefficiencies as adapting for all conditions means that useless resolution is introduced for each condition and adapting on only one means that no resolution is dedicated to most conditions.

As previously discussed on the Burgers case, because both SMA and MMA methods manage to consistently reduce the error during the adaptation process, the main influence for choosing one or the other comes from the cost to reach a specific error level. To this end, the ratio between the evaluation cost of the MMA and SMA methods (efficiency) has been represented against the target error level on Fig. 10(b). It has been measured experimentally and modeled for different solver scaling using the method proposed in Section 3.5. For both, the adaptation cost has been ignored ($T_{ad} = 0$) since the remeshing, initialization and prolongation phases developed for this paper are not as efficiently implemented as the CFD solver *elsA* making their comparison unrepresentative as it can be seen on Table 3. Nevertheless, neglecting the adaptation cost is a conservative choice since it lowers SMA's cost, without changing MMA's. For the target error presented in Table 3, the cost advantage of MMA ranges from 2.5 times to nearly 4.5 times when considering both costs.

To reach a specific error level, the MMA mesh is always bigger than the SMA. Indeed, as the shocks get more localized with the refinement a large area still needs to be refined to create a single mean mesh, like in the central duct diverging section (see Fig. 7(b)), whereas the SMA method is able to keep refinement localized along the adaptation (see Fig. 7(a)). Therefore, the observed efficiency comes from the difference between solving the CFD problem once per evaluation on a big mesh (MMA) and the cost of adaptation, several solves, for each condition on smaller meshes (SMA).

With the model, a strong dependency of the efficiency on the solver scaling parameter is observed. For an optimal solver scaling ($\gamma_{cfd} = 1$) MMA is expected to always be more efficient than SMA. Increasing it reduces the MMA advantage zone to higher errors. This is mainly due to the bigger meshes used for MMA evaluations. Indeed, with high solver scaling, solving on big meshes gets comparatively more expensive than solving multiple times on small meshes.

In the experiments, MMA is more efficient than SMA for high error targets until 4.10^{-3}. This is consistent with what is modeled using solver scaling between 1.5 and 2. Nevertheless, for high error targets the observed efficiency is almost double that of the modeled one. This comes from the fact that the model only considers the cost of solving the CFD problem but does not consider any overhead costs like mesh partitioning costs that are inherent to the CFD solver. These

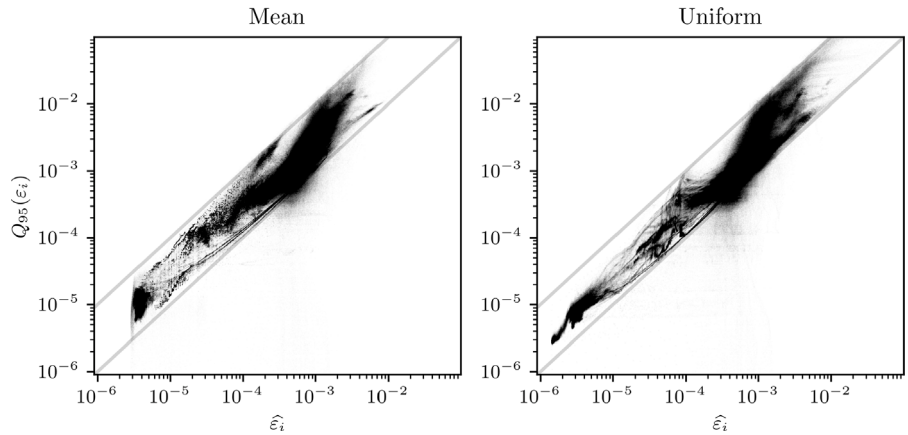


Fig. 11. 95% quantile of the local error with respect to the corresponding average value. For a MMA grid (left) and a uniform grid (right) achieving the same average error target (for MMA $\alpha = 0.3$, $N_p = 20$ and quadrature is MC).

overheads benefit the MMA efficiency because they are only computed once with MMA while they are repeated for each SMA iteration. As the error target gets lower, the solving cost gets higher, therefore, these overheads become negligible and the costs converge to the same values.

As explained in Section 3.5, the objective of methods like MMA is to reduce the cost of conducting studies that require a large number of condition evaluations. This is why the focus is on average cost and construction cost is neglected throughout the study. Considering that the construction cost is negligible when it represents less than 5% of the evaluation cost gains of MMA the values reported in Table 3 show that, for the targeted error level, construction cost becomes negligible for 855 or more condition evaluations. Therefore, for uncertainty quantification studies, where the convergence of statistics may require more than a thousand evaluations, construction costs are indeed negligible.

To summarize, when compared to SMA, MMA should be used when targeting reasonable error levels. The specific error level from which it becomes more efficient to use SMA is mainly dependent on the flow topology variability, as it will set the mean mesh size for a given error level. The solver scaling is also an important parameter as poorly scaling solvers will reduce MMA efficiency. Finally, adaptation and solver overhead should also be considered as their positive impact on MMA efficiency is non-negligible for moderate error levels.

4.2.5. Detailed error analysis

The MMA process as it is implemented here is aimed at minimizing the estimated average error. Nevertheless, since the goal is to build a mesh adapted for a continuous range of parameters, this could be insufficient in some cases. Indeed, while the error is guaranteed to be improved on average, some outliers could see higher local error levels. To check that MMA does not exhibit such behaviors, a 95% quantile of the measured error relative to the nominal inlet Mach number is computed. It is displayed with respect to the corresponding average error in each cell of the domain, for the MMA (left) and an equivalent uniform grid (depth 5) (right) on Fig. 11.

For both the MMA and the UR mesh, the quantile remains confined within a band of one order of magnitude above the mean value. Therefore, the local refinement choices of the MMA method do not induce higher error outliers than an uniform refinement. This can be explained by the fact that for each quadrature condition used to build the mean mesh, the refinement is in general wider than the targeted features, mainly because of smoothing and the high refined fraction chosen. Consequently, in most of the domain the refined area is continuous (see 7(b)) and almost every condition gets sufficient refinement where needed.

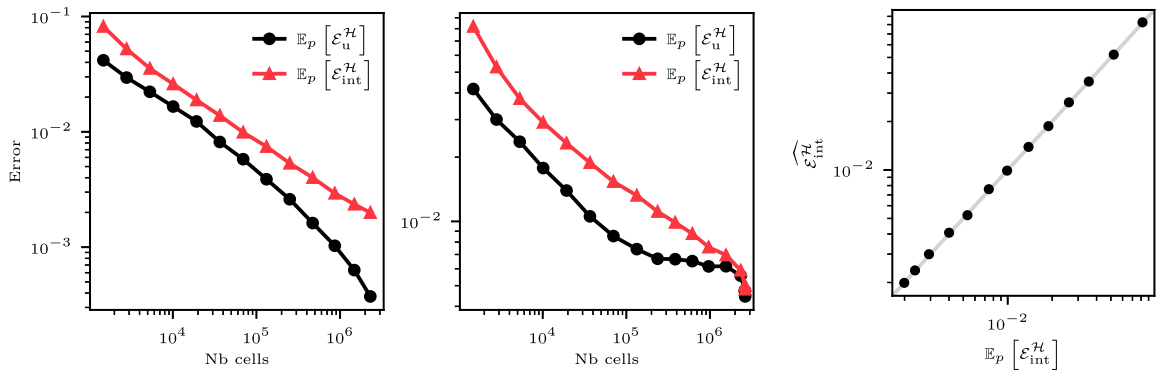
The error measurement method used for the scramjet configuration has several limitations. First, it is expensive to compute, because the flow for each of the validation samples needs to be computed on the reference grid of 6.6M cells, and also at every step of the adaptations process to evaluate. Then, the fine reference mesh allows for small but not negligible errors, meaning that convergence of the proposed error measurement does not mean convergence toward the true error. Finally, the reference limits the refinement depth of the adaptations and therefore their ability to reduce the error. To prevent these issues, the use of interpolation error as an error convergence estimator is investigated. On Fig. 12(a) the errors using the reference solution and the interpolation error estimate have been compared for the MMA and NMA adaptation methods.

The interpolation error estimation is an upper bound of the error measured from the reference. This is the preferred behavior for this kind of estimates as one would rather have the error target guaranteed at the expense of a more refined mean mesh. In addition, the difference increases at the end of the adaptation process as the interpolation error keeps a constant convergence order while the true error convergence accelerates. This divergence is due to the fact that the observed solution tends toward the reference solution, which is yet not the analytical one, as the cells get refined to a depth of 6. Finally, the good estimate made by the interpolation estimator means that during the MMA process the true error is on average dominated by interpolation errors. Therefore, the interpolation error can be used as an average error convergence estimator for the MMA method instead of the expensive and reference sensitive error measurement used before.

For the NMA adaptation, while both error estimators start with comparable behaviors, at the end of the adaptation process the measurement against reference highlights a plateau that is not captured by the interpolation error estimator. Therefore, the interpolation error is not the dominant error source at the end of the NMA adaptation process. This means that the error mainly comes from other sources. Thus, in addition to not being well resolved, the features far from the nominal condition are also misplaced and/or misrepresented. In this case, the mesh depth still increases in some areas of interest, effectively reducing the average interpolation error, which therefore does not feature the plateau that the misrepresentation of most features should impose.

Finally, the same conclusion that has been made on the Burgers case is verified: in the presence of conditional variability an adaptation method that fails to refine all the areas of interest, such as the NMA adaptation, does not guarantee dominance of the interpolation error on average and thus the effectiveness of the interpolation error as an error convergence estimator.

The MMA method has shown to be robust enough to make the interpolation error estimator a viable option for convergence evaluation. Now, to evaluate the possibility of assessing the convergence during



(a) Evolution with number of cells (Nb cells) of the true average error and it's estimation with the interpolation error for the MMA (left) and NMA (right) methods.

(b) Comparison of the average interpolation error and its estimation on the quadrature conditions.

Fig. 12. Evaluation of the average error estimation (with $\alpha = 0.3$, $N_p = 20$ and MC quadrature).

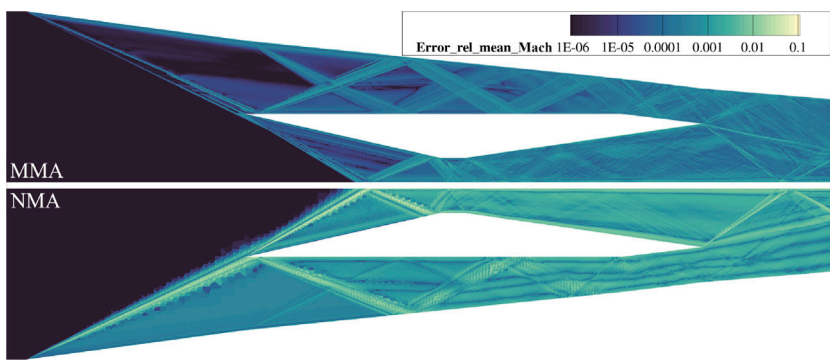


Fig. 13. Comparison of the local relative error on the average of the Mach number for the NMA and MMA methods. The adaptations use 10 iterations from the initial mesh and $\alpha = 0.3$ (for MMA $N_p = 20$ and the quadrature is MC).

the adaptation process without additional computation, the interpolation estimation of the error on the quadrature and the validation set have been compared on Fig. 12(b). A good agreement between both approaches is found during the whole adaptation process. This validates the behavior observed with the Burgers case and consolidates the confidence one can have in estimating the error with this technique.

4.2.6. Uncertainty quantification study

MMA is designed for studies requiring a high number of online evaluations, such as Uncertainty Quantification (UQ) studies. In UQ analysis, accurately estimating statistics of interest often demands numerous evaluations. Here, an uncertainty quantification analysis of the average Mach number has been conducted using different methodologies. Specifically, the statistics are computed using the same 200-condition validation set as in the previous analysis.

Fig. 13 presents the local relative error in the average Mach number field using the MMA and NMA meshes. Unlike NMA, which exhibits significant error levels reaching up to 30% near key flow features, MMA maintains a consistently lower error, remaining below 10% throughout the domain. This discrepancy is likely due to the robust refinement placement in MMA, which prevents feature misplacement and the subsequent error propagation observed in NMA. Specifically, the localized refinement produced with the nominal mesh (see Fig. 7(a)) compromises overall solution quality by under-refining critical upstream shocks, leading to their misplacement, as discussed in Section 4.2.2. Overall, the MMA methodology demonstrates superior performance in uncertainty quantification (UQ) studies, particularly in controlling error levels across the entire computational domain. However, it is important to note that performing UQ studies with a limited set of conditions, such as the 200 used here, may lead to imprecise results. In this case, the spread of the 95% confidence interval for the estimated statistics, computed via bootstrapping, reaches up to 6% for both MMA and NMA. This issue is especially pronounced for statistics that are difficult to converge. Despite this, MMA effectively controls error and ensures robust refinement placement, making it a reliable choice for UQ studies. Additionally, the use of unique meshes in MMA enables direct computation of local statistics without requiring interpolation, further enhancing both computational efficiency and accuracy.

5. Conclusions and perspectives

A new approach, the mean mesh adaptation, has been developed in the context of continuous conditional variability and has been applied to error-based mesh adaptation. The method consists in building a single mean mesh from the average estimated error of several sample conditions. Its evaluation on finite volume CFD test cases, in 1 and 2 dimensions, has shown the ability of the method to robustly control the average error at a reasonable cost. It has been observed that using a Monte Carlo renewed quadrature for the average error computation allowed to choose small and cost-effective construction sets with minimal impact on the error. Indeed, in the scramjet test case, it was observed that as few as five conditions per adaptation iteration were sufficient to achieve good error convergence. Moreover, this sampling strategy guarantees that the adapted mesh converges toward the desired mean mesh as enrichment allows the estimator to progressively converge toward the true average error. Results on the scamjet test case have shown up to 5 times cost reduction compared to SMA for the same error target. Therefore, for reasonable error targets MMA allows for significant cost reductions when compared to UR and SMA adaptations strategies and is not subject to error stagnation like NMA. This makes

it a good addition to the existing mesh adaptation method for variable conditions existing in the literature [7,9,10].

In addition, once a mean mesh is built for a given error target, its use for condition evaluation is fast and straightforward without any meshing expertise requirements. The MMA method has also some implementation advantages, the average error level can be estimated on the fly without any additional expense , as it has been consistently observed on both test cases, and the method's structure is inherently parallel, making it well-suited for HPC frameworks.

Nevertheless, MMA still faces some challenges. Creating good initialization states for fast solver convergence is complex and impacts the method overall cost. Our investigations have also shown that MMA becomes less cost efficient when targeting very low average errors and when working with high conditional variability like in the central duct of the scramjet test case. Thus, MMA is particularly well-suited for problems involving local features that move within a confined region. However, the method is still guaranteed to control the average error across any range of parametric and feature variations. Therefore, cases like the scramjet, where some features exhibit minor variations while others undergo significant changes, are still suitable for this approach, as long as the majority of the features exhibit limited movement.

As a perspective, the extension of the method to unsteady simulation could be considered. Introducing time dependence would require reformulating both the problem and its objectives, raising a crucial question: does constructing a single mesh still make sense in this context? In unsteady configurations, mesh adaptation is typically applied at each time step, leading to a sequence of adapted meshes rather than a single one. An extension of the Mean Mesh Adaptation (MMA) approach could involve generating a collection of unique meshes for different time steps. However, this would necessitate developing a new methodology to construct and integrate these meshes effectively over time. As in the steady-state case, the efficiency of the generated meshes would depend on the spatial distribution of flow features at each time step.

Future work will be dedicated to the development of new methodologies that are more adapted to high conditional variability and high accuracy and that are not necessarily based on a single common mesh. To do so, the construction of collections of adapted meshes from which to infer the best mesh for a given condition is considered. Optimal transport based mesh adaptation is also investigated to allow for direct mesh adaptation at unknown conditions.

CRediT authorship contribution statement

Hugo Dornier: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Olivier P. Le Maître:** Writing – review & editing, Supervision, Methodology, Investigation, Formal analysis, Conceptualization. **Pietro M. Congedo:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization. **Itham Salah el Din:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization. **Julien Marty:** Writing – review & editing, Supervision, Methodology, Data curation, Conceptualization. **Sébastien Bourasseau:** Writing – review & editing, Supervision, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partially supported by the SONICE project which is a France Relance project co-funded by the French Civil Aviation Authority (DGAC). France Relance benefits from EU funding via the NextGenerationEU initiative.

Data availability

The authors do not have permission to share data.

References

[1] Alauzet F, Loseille A. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. Computer-Aided Des 2016;72:13–39.
[2] Habashi WG, Dompierre J, Bourgault Y, Ait-Ali-Yahia D, Fortin M, Vallet M-G. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver-independent CFD. Part I: General principles. Internat J Numer Methods Fluids 2000;32(6):725–44. http://dx.doi.org/10.1002/(SICI)1097-0363(20000330)32:6<725::AID-FLD935>3.0.CO;2-4.
[3] Baker TJ. Mesh adaptation strategies for problems in fluid dynamics. Finite Elem Anal Des 1997;25:243–73.
[4] Alauzet F, Frazza L, Papadogiannis D. Periodic adjoints and anisotropic mesh adaptation in rotating frame for high-fidelity RANS turbomachinery applications. J Comput Phys 2022;450:110814.
[5] Vivarelli G, Qin N, Shahpar S, Radford D. Efficient adjoint-based mesh adaptation applied to turbo-machinery flows. In: Proceedings of ASME turbo expo 2018. 2018.
[6] Sullivan TJ. Introduction to uncertainty quantification. In: Texts in applied mathematics, (63). Cham Heidelberg New York Dordrecht London: Springer; 2015, http://dx.doi.org/10.1007/978-3-319-23395-6.
[7] Van Langenhove J, Lucor D, Alauzet F, Belme A. Goal-oriented error control of stochastic system approximations using metric-based anisotropic adaptations. J Comput Phys 2018;374:384–412. http://dx.doi.org/10.1016/j.jcp.2018.07.044.
[8] Capriati M, Cortesi A, Magin TE, Congedo PM. Stagnation point heat flux characterization under numerical error and boundary conditions uncertainty. Eur J Mech / B Fluids 2022;95:221–30.
[9] Palacios F, Duraisamy K, Alonso JJ, Zuazua E. Robust grid adaptation for efficient uncertainty quantification. AIAA J 2012;50(7):1538–46. http://dx.doi.org/10.2514/1.J051379.
[10] Barral N, Taddei T, Tifouti I. Registration-based model reduction of parameterized PDEs with spatio-parameter adaptivity. J Comput Phys 2024;499:112727. http://dx.doi.org/10.1016/j.jcp.2023.112727.
[11] Frey PJ, Alauzet F. Anisotropic mesh adaptation for CFD computations. Comput Methods Appl Mech Engrg 2005;194:5068–82.
[12] Peraire J, Vahdati M, Morgan K, Zienkiewicz O. Adaptive remeshing for compressible flow computations. J Comput Phys 1987;72(2):449–66. http://dx.doi.org/10.1016/0021-9991(87)90093-3.
[13] Benard P, Balarac G, Moureau V, Dobrzynski C, Lartigue G, d'Angelo Y. Mesh adaptation for large-eddy simulations in complex geometries. Int J Numer Methods Fluids, Wiley 2016;719–40.
[14] Kumar A. Numerical analysis of the scramjet-inlet flow field by using two-dimensional Navier-Stokes equations. Technical report, (NASA-TP-1940). NASA Langley Research Center; 1981.
[15] Castro-Díaz MJ, Hecht F, Mohammadi B, Pironneau O. Anisotropic unstructured mesh adaption for flow simulations. Internat J Numer Methods Fluids 1997;25(4):475–91. http://dx.doi.org/10.1002/(SICI)1097-0363(19970830)25:4<475::AID-FLD575>3.0.CO;2-6.
[16] Alauset F. Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en mécanique des fluides [Ph.D. thesis], Montpellier II; 2003.
[17] Loseille A, Dervieux A, Frey P, Alauzet F. Achievement of global second order mesh convergence for discontinuous flows with adapted unstructured meshes. AIAA J 2007.
[18] Versteeg HK, Malalasekera W. An introduction to computational fluid dynamics: The finite volume method. 2nd ed.. Harlow, England ; New York: Pearson Education Ltd; 2007.
[19] Gnoffo PA. A vectorized, finite-volume, adaptive-grid algorithm for Navier-Stokes calculations. Appl Math Comput 1982;10–11:819–35. http://dx.doi.org/10.1016/0096-3003(82)90224-7.
[20] Berger MJ, Jameson A. Automatic adaptive grid refinement for the Euler equations. AIAA J 1985;23(4):561–8. http://dx.doi.org/10.2514/3.8951.
[21] Kompenhans M, Rubio G, Ferrer E, Valero E. Comparisons of P-adaptation strategies based on truncation- and discretisation-errors for high order discontinuous Galerkin methods. Comput & Fluids 2016;139:36–46. http://dx.doi.org/10.1016/j.compfluid.2016.03.026.
[22] Lovely D, Haimes R. Shock detection from computational fluid dynamics results. In: Proceedings of the 14th AIAA computational fluid dynamics conference. 1999.
[23] Haimes R, Kenwright D. On the velocity gradient tensor and fluid feature extraction. AIAA J 1999.
[24] Deck S, Renard N. Towards an enhanced protection of attached boundary layers in hybrid RANS/LES methods. JComput Phys 2020. Elsevier.
[25] Zhou L, Zhao R, Shi X-P. Entropy: Theory and new insights. Ricardo Beltran-Chacon; 2020,

[26] Liu C, Gao Y-s, Dong X-r, Wang Y-q, Liu J-m, Zhang Y-n, et al. Third generation of vortex identification methods: Omega and Liutex/Rortex based systems. J Hydrodyn 2019;31(2):205–23. http://dx.doi.org/10.1007/s42241-019-0022-4.

[27] Grenouilloux A, Balarac G, Leparoux J, Moureau V, Lartigue G, Bénard P, et al. On the use of kinetic-energy balance for the feature-based mesh adaptation applied to large-Eddy simulation in complex geometries. In: Proceedings of ASME turbo expo 2022. 2022.

[28] Löhner R, Morgan K, Zienkiewicz O. An adaptive finite element procedure for compressible high speed flows. Comput Methods Appl Mech Engrg 1985;51(1–3):441–65. http://dx.doi.org/10.1016/0045-7825(85)90042-8.

[29] Oden JT, Strouboulis T, Devloo P. Adaptive finite element methods for high-speed compressible flows. Internat J Numer Methods Fluids 1987;7(11):1211–28. http://dx.doi.org/10.1002/fld.1650071105.

[30] Wu J, Zhu JZ, Szmelter J, Zienkiewicz OC. Error estimation and adaptivity in Navier-Stokes incompressible flows. Comput Mech 1990;6(4):259–70. http://dx.doi.org/10.1007/BF00370106.

[31] Giles M, Pierce N. Improved lift and drag estimates using adjoint Euler equations. In: 14th computational fluid dynamics conference. Norfolk,VA,U.S.A.: American Institute of Aeronautics and Astronautics; 1999, http://dx.doi.org/10.2514/6.1999-3293.

[32] Venditti DA, Darmofal DL. Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. J Comput Phys 2003;187(1):22–46. http://dx.doi.org/10.1016/S0021-9991(03)00074-3.

[33] Benoit C, Péron S, Landier S. Cassiopee: A CFD pre- and post-processing tool. Aerosp Sci Technol 2015;45:272–83.

[34] Abramowitz M, Stegun IA, editors. Handbook of mathematical functions: With formulas, graphs, and mathematical tables, In: Dover books on mathematics, 9. Dover print.; [Nachdr. der Ausg. von 1972]. New York, NY: Dover Publ; 2013.

[35] Liu JS. Monte Carlo strategies in scientific computing. In: Springer series in statistics, New York, NY: Springer New York; 2004, http://dx.doi.org/10.1007/978-0-387-76371-2.

[36] Hopf E. The partial differential equation ut + uux=Mxx. Comm Pure Appl Math 1950;3(3):201–30. http://dx.doi.org/10.1002/cpa.3160030302.

[37] Roe P. Approximate Riemann solvers, parameter vectors, and difference schemes. J Comput Phys 1981;43(2):357–72. http://dx.doi.org/10.1016/0021-9991(81)90128-5.

[38] Balay S, Abhyankar S, Adams M, Benson S, Brown J. PETSc/TAO users manual. Technical report, (ANL-21/39-Rev.3.19, 1968587, 181821). 2023, http://dx.doi.org/10.2172/1968587.

[39] Van Langenhove J. Adaptive control of deterministic and stochastic approximation errors in simulations of compressible flow [Ph.D. thesis], Paris: Université Pierre et Marie Curie - Paris VI; 2017.

[40] Cambier L, Heib S, Plot S. The ONERA elsA CFD software : Input from research and feedback from industry. Mech Ind EDP Sci 2013;14(3):159–74.