Anisotropic Goal-Oriented Mesh Adaptation for Time Dependent Problems

F. Alauzet[1], A. Belme[2], and A. Dervieux[2]

[1] INRIA Roquencourt, Domaine de Voluceau, 78150 Rocquencourt, France
Frederic.Alauzet@inria.fr
[2] INRIA Sophia Antipolis, 2004 route de Lucioles, 06902 Sophia-Antipolis, France
Anca.Belme@inria.fr Alain Dervieux@inria.fr

Summary. We present a new algorithm for combining an anisotropic goal-oriented error estimate with the mesh adaptation fixed point method for unsteady problems. The minimization of the error on a functional provides both the density and the anisotropy (stretching) of the optimal mesh. They are expressed in terms of state and adjoint. This method is used for specifying the mesh for a time sub-interval. A global fixed point iterates the re-evaluation of meshes and states over the whole time interval until convergence of the space-time mesh. Applications to unsteady blast-wave are presented.

Key words: Unsteady compressible flow, goal-oriented mesh adaptation, anisotropic mesh adaptation, adjoint, metric

Introduction

Engineering problems frequently require computational fluid dynamics (CFD) solutions with functional outputs of specified accuracy. The computational resources available for these solutions are often limited and errors in solutions and outputs are difficult to control. CFD solutions may be computed with an unnecessarily large number of mesh vertices (and associated high cost) to ensure that the outputs are computed within a required accuracy.

One of the powerful methods for increasing the accuracy and reducing the computational cost is mesh adaptation, the purpose of which is to control the accuracy of the numerical solution by changing the discretization of the computational domain according to mesh size and mesh directions constraints.

Pioneering works have shown a fertile development of Hessian-based or metric-based methods [12, 9] which rely on an ideal representation of the *interpolation error* and of the *mesh*. The "multiscale" version relies on the optimization of the $\mathbf{L}^p$ norm of the interpolation error [14]. It allows to take into account the discontinuities with higher-order convergence [18]. However, these methods are limited to the minimization of some interpolation errors for some solution fields, the "sensors", and do not take into account the PDE being solved. If for many applications, this simplifying

standpoint is an advantage, there are also many applications where Hessian-based mesh adaptation is far from optimal regarding the way the degrees of freedom are distributed in the computational domain.

On the other side, *goal-oriented* mesh adaptation focuses on deriving the best mesh to observe a given output functional. Goal-oriented methods result from a series of papers dealing with *a posteriori estimates* (see e.g. [20, 6, 11, 21]). Extracting informations concerning mesh anisotropy from an *a posteriori* estimate is a difficult task. Starting from *a priori estimates*, Loseille *et al.* proposed in [17] a fully anisotropic goal-oriented mesh adaptation technique for steady problems. This latter method combines goal-oriented rationale and the application of Hessian-based analysis to truncation error.

Mesh adaptation for unsteady flows is also an active field of research and brings an attracting increase in simulation efficiency. Complexity of the algorithms is larger than for steady case: for most flows, the mesh should change during the time interval. Meshes can be moved [5], pattern-split [7, 13], locally refined [4], or globally rebuild [1]. Hessian-based methods are essentially applied with a non-moving mesh system. A mesh adaptation fixed-point method was proposed in [1]. The Hessian criteria at the different time steps of a sub-interval are synthetized into a single criterion for these steps with the metric intersection [1]. A mesh-PDE solver iteration is applied on time sub-intervals. Extension to $\mathbf{L}^p$ error estimator has been proposed in [4].

The objective of this paper is the extension of goal-oriented anisotropic mesh adaptation method of [17] to the unsteady framework introduced in [1].

To this end, several methodological issues need to be addressed. First, similarly to [4], we propose a global fixed-point algorithm for solving the coupled system made, this time of three fields, the unsteady state, the unsteady adjoint state and the adapted meshes. Second, this algorithm needs to be *a priori* analyzed and its convergence rate to continuous solution needs to be optimized. Third, at the computer algorithmic level, it is also necessary to master the computational (memory and time) cost of the new system, which couples a time-forward state, a time-backward adjoint and a mesh update influenced by global statistics.

We start this paper with a formal description of the error analysis in its most general expression, then the application to unsteady compressible Euler flows is presented. In Section 3, we introduce the optimal adjoint-based metric definition and all its relative issues, then in Section 4 we present our mesh adaptation algorithm. We end this paper with numerical experiments for blast wave problems.

1 Formal Error Analysis

Let us introduce a system of PDE's in its variational formulation:

$$\text{Find } w \in \mathcal{V} \text{ such that } \forall \varphi \in \mathcal{V}, \quad (\Psi(w)\,,\,\varphi) = 0 \tag{1}$$

with $\mathcal{V}$ a functional space of solutions. The associated discrete variational formulation then writes:

$$\text{Find } w_h \in \mathcal{V}_h \text{ such that } \forall \varphi_h \in \mathcal{V}_h, \quad (\Psi_h(w_h)\,,\,\varphi_h) = 0 \tag{2}$$

where $\mathcal{V}_h$ is a subspace of $\mathcal{V}$. For a solution w of state system (1), we define a *functional output* as:

$$j \in \mathbb{R} \ ; \ j = (g, w), \tag{3}$$

where (g, w) holds for the following rather general functional output formulation:

$$(g,w) \ = \ \int_0^T \int_\Omega (g_\Omega, w)\, \mathrm{d}\Omega\, \mathrm{d}t \ + \ \int_\Omega (g_T, w(T))\ \mathrm{d}\Omega \ + \ \int_0^T \int_\Gamma (g_\Gamma, w)\ \mathrm{d}\Gamma\ \mathrm{d}t, \tag{4}$$

where g_Ω, g_T, and g_Γ are assumed to be regular enough functions. We introduce the *continuous adjoint* w^*, solution of the following system:

$$w^* \in \mathcal{V} \ , \ \forall \psi \in \mathcal{V} \ , \ \left(\frac{\partial \Psi}{\partial w}(w)\psi, w^* \right) = (g, \psi) \ . \tag{5}$$

The objective here is to estimate the following approximation error committed on the functional:

$$\delta j = j(w) - j(w_h) \ ,$$

where w and w_h are respectively solutions of (1) and (2). It is then useful to choose the test function φ_h as the discrete adjoint state, $\varphi_h = w_h^*$, which is the solution of:

$$\forall \psi_h \in \mathcal{V}_h, \ \left(\frac{\partial \Psi_h}{\partial w_h}(w_h)\psi_h \ , \ w_h^* \right) = (g, \psi_h). \tag{6}$$

We assume that w_h^* is close to the continuous adjoint state w^*. We refer to [17] in which the following *a priori* formal estimate is finally proposed:

$$\delta j \approx ((\Psi_h - \Psi)(w), w^*) \ . \tag{7}$$

The next section is devoted to the application of Estimator (7) to the unsteady Euler model.

2 Unsteady Euler Models

2.1 Continuous state system and Finite Volume formulation

Continuous state system. The 3D unsteady compressible Euler equations are set in the computational space-time domain $\mathcal{Q} = \Omega \times [0, T]$, where T is the (positive) maximal time and $\Omega \subset \mathbb{R}^3$ is the spatial domain. An essential ingredient of our discretisation and of our analysis is the elementwise linear interpolation operator. In order to use it easily, we define our working functional space as $V = \left[H^1(\Omega) \cap \mathcal{C}(\bar{\Omega}) \right]^5$, that is the set of measurable functions that are continuous with square integrable gradient. We formulate the Euler model in a compact variational formulation in the functional space $\mathcal{V} = H^1\{[0, T]; V\}$ as follows:

$$\begin{aligned} &\text{Find } W \in \mathcal{V} \text{ such that } \forall \varphi \in \mathcal{V}, \quad (\Psi(W)\,,\, \varphi) = 0 \\ \text{with} \quad &(\Psi(W)\,,\, \varphi) = \int_\Omega \varphi(0)(W_0 - W(0))\, \mathrm{d}\Omega \ + \ \int_0^T \int_\Omega \varphi W_t \, \mathrm{d}\Omega \, \mathrm{d}t \\ &\qquad + \int_0^T \int_\Omega \varphi \, \nabla \cdot \mathcal{F}(W)\, \mathrm{d}\Omega \, \mathrm{d}t \ - \ \int_0^T \int_\Gamma \varphi \, \hat{\mathcal{F}}(W).\mathbf{n}\, \mathrm{d}\Gamma \, \mathrm{d}t \, . \end{aligned} \tag{8}$$

In the above definition, W is the vector of conservative flow variables and $\mathcal{F}(W) = (\mathcal{F}_1(W), \mathcal{F}_2(W), \mathcal{F}_3(W))$ is the usual Euler fluxes given by:

$$W \;=\; \begin{pmatrix} \rho \\ \rho\, u_i \\ \rho\, e \end{pmatrix} \quad \text{and} \quad \mathcal{F}_j(W) \;=\; \begin{pmatrix} \rho\, u_j \\ \rho\, u_j\, u_i + p\, \delta_{ij} \\ (\rho\, e + p) u_j \end{pmatrix} \quad \text{with } i = \{1,2,3\}\ ,$$

where ρ, u_i, p and e denote respectively the fluid density, i^{th} component of the Cartesian velocity, pressure and total energy. δ_{ij} is the Kronecker delta function. Here, functions φ and W have 5 components, and therefore the product φW holds for $\sum_{k=1..5} \varphi_k W_k$. We have denoted by Γ the boundary of the computational domain Ω, $\mathbf{n}$ is the outward normal to Γ, $W(0)(\mathbf{x}) = W(\mathbf{x}, t)|_{t=0}$ for any $\mathbf{x}$ in Ω, W_0 the initial condition and the boundary flux $\hat{\mathcal{F}}$ contains the different boundary conditions.

Discrete state system. As a spatially semi-discrete model, we consider the Mixed-Element-Volume formulation [8]. As in [17], we reformulate it under the form of a finite element variational formulation, this time in the unsteady context. We assume that Ω is covered by a finite-element partition in simplicial elements denoted K. The mesh, denoted by $\mathcal{H}$ is the set of the elements. Let us introduce the following approximation space:

$$V_h \;=\; \Big\{ \varphi_h \in V \;\Big|\; \varphi_{h\,|K} \text{ is affine } \forall K \in \mathcal{H} \Big\}, \text{ and } \mathcal{V}_h \;=\; H^1\{[0,T]; V_h\} \subset \mathcal{V}.$$

Let Π_h be the usual $\mathcal{P}^1$ projector:

$$\Pi_h :\ V \to\ V_h \ \text{ such that } \ \Pi_h \varphi(\mathbf{x}_i) = \varphi(\mathbf{x}_i),\ \forall\ \mathbf{x}_i \text{ vertex of } \mathcal{H}.$$

We extend it to time-dependent functions:

$$\Pi_h :\ H^1\{[0,T]; V\} \to\ \mathcal{V}_h \ \text{ such that } \ (\Pi_h \varphi)(t) = \Pi_h(\varphi(t))\ ,\ \forall\ t \in\ [0,T].$$

The weak discrete formulation writes:

$$\text{Find } W_h \in \mathcal{V}_h \text{ such that } \forall \varphi_h \in \mathcal{V}_h, \quad (\Psi_h(W_h)\ ,\ \varphi_h) = 0,$$

$$\text{with: } \ (\Psi_h(W_h)\ ,\ \varphi) = \int_\Omega \varphi(0)(\Pi_h W_h(0) - W_{0h})\,\mathrm{d}\Omega \ +\ \int_0^T \int_\Omega \varphi\ \Pi_h W_{h,t}\,\mathrm{d}\Omega\,\mathrm{d}t$$

$$+ \int_0^T \int_\Omega \varphi \nabla\cdot \mathcal{F}_h(W_h)\,\mathrm{d}\Omega\,\mathrm{d}t - \int_0^T \int_\Gamma \varphi \hat{\mathcal{F}}_h(W_h).\mathbf{n}\,\mathrm{d}\Gamma\,\mathrm{d}t + \int_0^T \int_\Omega \varphi\, D_h(W_h)\,\mathrm{d}\Omega\,\mathrm{d}t$$

with $\mathcal{F}_h = \Pi_h \mathcal{F}$ and $\hat{\mathcal{F}}_h = \Pi_h \hat{\mathcal{F}}$. The D_h term accounts for the numerical diffusion. In the present study, we only need to know that for smooth fields, the D_h term is a third order term with respect to the mesh size. For shocked fields, monotonicity limiters become first-order terms.

2.2 Continuous Adjoint system and discretization

Continuous adjoint system. We refer here to the continuous adjoint system introduced previously:

$$W^* \in \mathcal{V}\ ,\ \forall \psi \in \mathcal{V}\ :\ \left(\frac{\partial \Psi}{\partial W}(W)\psi, W^* \right) - (g, \psi) \;=\; 0. \tag{9}$$

We recall that (g, ψ) is defined by (4). Replacing $\Psi(W)$ by its Formulation (8) and integrating by parts, we get:

$$\begin{aligned}\left(\frac{\partial \Psi}{\partial W}(W)\psi, W^*\right) = &\int_\Omega \left(\psi(0)W^*(0) - \psi(T)W^*(T)\right) \mathrm{d}\Omega \\ &+ \int_0^T \int_\Omega \psi \left(-W_t^* - \left(\frac{\partial \mathcal{F}}{\partial W}\right)^* \nabla W^*\right) \mathrm{d}\Omega \, \mathrm{d}t \\ &+ \int_0^T \int_\Gamma \psi \left[\left(\frac{\partial \mathcal{F}}{\partial W}\right)^* W^*.\mathbf{n} + \left(\frac{\partial \hat{\mathcal{F}}}{\partial W}\right)^* W^*.\mathbf{n}\right] \mathrm{d}\Gamma \, \mathrm{d}t \ . \quad (10)\end{aligned}$$

The adjoint Euler equations is a system of advection equations, where the temporal integration goes backwards, *i.e.,* in the opposite direction of usual time. Thus, when solving the unsteady adjoint system, one starts at the end of the flow run and progresses back until reaching the start time.

Discrete Adjoint System. Although any consistent approximation of the continuous adjoint system could be built, we choose to build the adjoint of the discrete state defined in (9) in order to be closer to the true error from which the continuous model were derived. Consider the following semi-discrete unsteady compressible Euler model (explicit RK1 time integration):

$$\Psi_h^n(W_h^n, W_h^{n-1}) = \frac{W_h^n - W_h^{n-1}}{\delta t^n} + \Phi_h(W_h^{n-1}) = 0 \quad \text{for } n = 1, ..., N. \quad (11)$$

The time-dependent functional is discretised as follows:

$$j(W_h) = \sum_{n=1}^{N} \delta t^n j^{n-1}(W_h).$$

The problem of minimizing the error committed on the target functional $j(W_h) = (g, W_h)$, subject to the Euler system (11), can be transformed into an unconstrained problem for the following Lagrangian functional:

$$\mathcal{L} = \sum_{n=1}^{N} \delta t^n j^{n-1}(W_h) - \sum_{n=1}^{N} \delta t^n (W_h^{*,n})^T \Psi_h^n(W_h^n, W_h^{n-1})$$

where $W_h^{*,n}$ are the N vectors of the Lagrange multipliers (which are the time-dependent adjoint states). The conditions for an extremum becomes then:

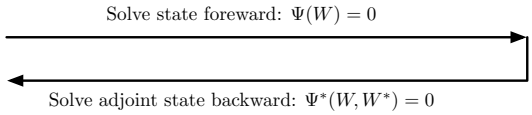
$$\frac{\partial \mathcal{L}}{\partial W_h^{*,n}} = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial W_h^n} = 0, \quad \text{for } n = 1, ..., N.$$

The first condition is clearly verified from relation (11). Thus the Lagrangian multipliers $W_h^{*,n}$ must be chosen such that the second condition of extrema $\frac{\partial \mathcal{L}}{\partial W_h^n} = 0$ is verified. This gives the unsteady discrete adjoint system:

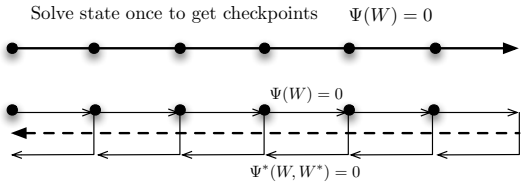
$$\begin{cases} W_h^{*,N} = 0 \\ W_h^{*,n-1} = W_h^{*,n} + \delta t^n \dfrac{\partial j^{n-1}}{\partial W_h^{n-1}} - \delta t^n (W_h^{*,n})^T \dfrac{\partial \Psi_h^{n-1}}{\partial W_h^{n-1}} \end{cases} \quad (12)$$

As the adjoint system runs in reverse time, the first expression in the adjoint system (12) is referred to as adjoint "initialization". Computing $W_h^{*,n-1}$ at time t^{n-1}

requires the knowledge of state W_h^{n-1} and adjoint state $W_h^{*,n}$. Therefore, the knowledge of all states $\left\{W_h^{n-1}\right\}_{n=1,N}$ is needed to compute backward the adjoint state from time T to 0 which involves large memory storage effort.



This drawback can be reduced by out-of-core storage of checkpoints (as shown in the picture below), although it implies a recomputing effort of the state W.



2.3 Numerical example

The simulation of a blast in a 2D geometry representing a city is performed, see Figure 1. A blast-like initialization $W_{blast} = (10, 0, 0, 250)$ in ambient air $W_{air} = (1, 0, 0, 2.5)$ is considered in a small region of the computational domain. We perform a forward/backward computation on a uniform mesh of 22 574 vertices and 44 415 triangles. Output functional of interest j is the quadratic deviation from ambient pressure on target surface S which is a part of the higher building roof (Figure 1):

$$j(W) = \int_0^T \int_S \frac{1}{2}(p(t) - p_{air})^2 \, \mathrm{d}S \, \mathrm{d}t.$$

Figure 2 plots the density isolines of the flow at different times showing several shock waves traveling throughout the computational domain. Figure 3 depicts the associated density adjoint state progressing backward in time. The same computational time is considered for both figures.

The simulation points out the ability of the adjoint to automatically provide the sensitivity of the flow field on the functional. Indeed, at early time of the simulation (top left picture), a lot of information is captured by the adjoint, *i.e.,* non-zero adjoint values. We notice that shock waves which will directly impact are clearly detected by the adjoint, but also shocks waves reflected by the left building which will be redirected towards surface S. At the middle of the simulation, the adjoint neglects waves that are traveling in the direction opposite to S and also waves that will not impact surface S before final time T since they won't have an influence on the cost functional. While getting closer to final time T (bottom right picture), the adjoint only focuses on the last waves that will impact surface S and ignores the rest of the flow.

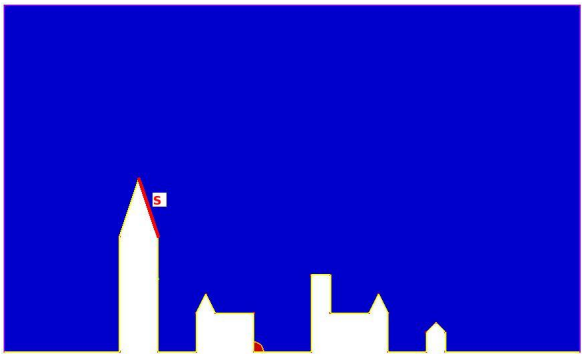


Fig. 1. Initial blast solution and location of the target surface.

3 Optimal unsteady adjoint-based metric

3.1 Error analysis applied to unsteady Euler model

We replace in Estimation (7) operators Ψ and Ψ_h by their expressions given by Relations (8) and (9). In [17], it was observed that even for shocked flows, it is interesting to neglect the numerical viscosity term. We follow again this option. We also discard the error committed when imposing the initial condition. And finally, after integrating by parts, the previous error estimate leads to:

$$\delta j \approx \int_0^T \int_\Omega W^* \left(W - \Pi_h W\right)_t \mathrm{d}\Omega\,\mathrm{d}t \;-\; \int_0^T \int_\Omega \nabla W^* \left(\mathcal{F}(W) - \Pi_h \mathcal{F}(W)\right) \mathrm{d}\Omega\,\mathrm{d}t$$
$$- \int_0^T \int_\Gamma W^* \left(\bar{\mathcal{F}}(W) - \Pi_h \bar{\mathcal{F}}(W)\right).\mathbf{n}\,\mathrm{d}\Gamma\,\mathrm{d}t\,. \tag{13}$$

with $\bar{\mathcal{F}} = \hat{\mathcal{F}} - \mathcal{F}$. We observe that this estimate of δj is expressed in terms of interpolation errors of the Euler fluxes and of the time derivative weighted by continuous functions W^* and ∇W^*. The integrands in Error Estimation (13) contain positive and negative parts which can compensate for some particular meshes. In our strategy, we prefer to not rely on these parasitic effects and to slightly over-estimate the error. To this end, all integrands are bounded by their absolute values. Moreover, we observe that the third term introduce a dependency of the error with respect to the boundary surface mesh. In the present paper, we discard this term and refer to [17] for a discussion of the influence of it. At the end, we get:

$$(g, W_h - W) \leq \int_0^T \int_\Omega |W^*|\,|\left(W - \Pi_h W\right)_t|\,\mathrm{d}\Omega\,\mathrm{d}t + \int_0^T \int_\Omega |\nabla W^*|\,|\mathcal{F}(W) - \Pi_h \mathcal{F}(W)|\,\mathrm{d}\Omega\,\mathrm{d}t\,. \tag{14}$$

3.2 Continuous mesh model

We propose to work in the continuous mesh framework, introduced in [15, 16]. It allows us to define proper differentiable optimization, *i.e.,* to use a calculus of variations which cannot apply on the class of discrete meshes. This framework lies in the class of metric-based methods.

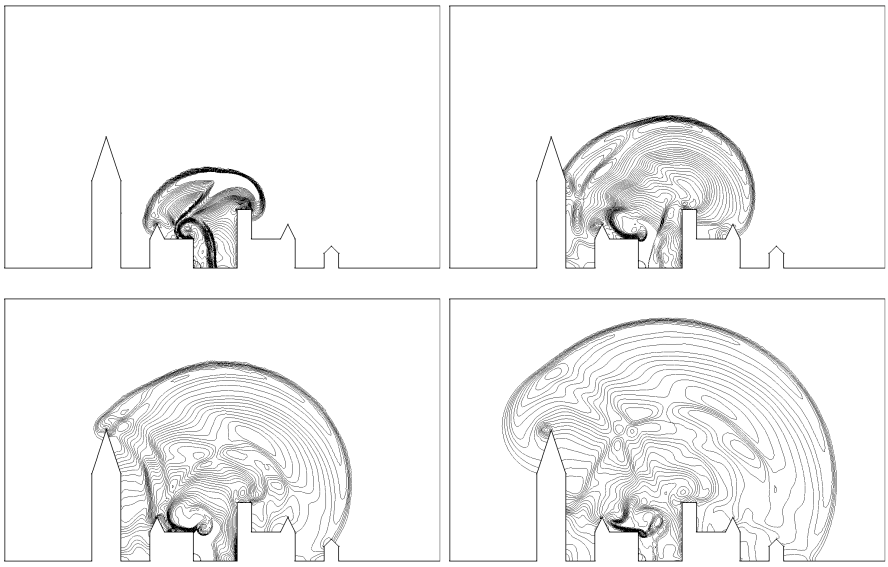


Fig. 2. 2D city blast solution state evolution. From left to right and top to bottom, snapshot of the density isolines at a-dimensional time 1.2, 2.25, 3.3 and 4.35.

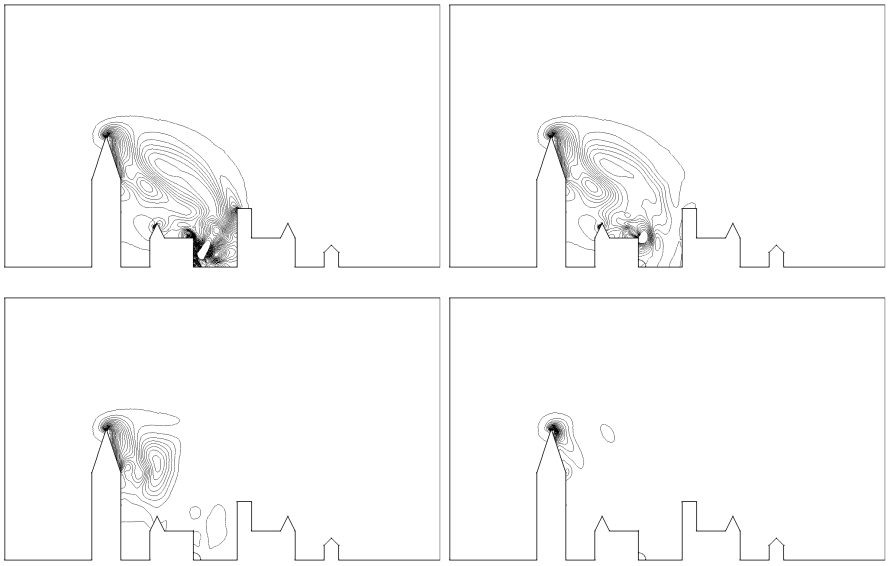


Fig. 3. 2D city blast adjoint state evolution. From left to right and top to bottom, snapshot of the density isolines at a-dimensional time 0.15, 1.2, 2.25 and 3.3.

A continuous mesh $\mathbf{M}$ of computational domain Ω is identified to a Riemannian metric field $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. For all $\mathbf{x}$ of Ω, $\mathcal{M}(\mathbf{x})$ is a symmetric tensor having $(\lambda_i(\mathbf{x}))_{i=1,3}$ as eigenvalues along the principal directions $\mathcal{R}(\mathbf{x}) = (\mathbf{v}_i(\mathbf{x}))_{i=1,3}$. Sizes along these directions are denoted $(h_i(\mathbf{x}))_{i=1,3} = (\lambda_i^{-\frac{1}{2}}(\mathbf{x}))_{i=1,3}$ and the three anisotropic quotients r_i are defined by: $r_i = h_i^3 (h_1 h_2 h_3)^{-1}$. The node density d is equal to: $d = (h_1 h_2 h_3)^{-1} = (\lambda_1\lambda_2\lambda_3)^{\frac{1}{2}} = \sqrt{\det(\mathcal{M})}$. By integrating the node density, we define the complexity $\mathcal{C}$ of a continuous mesh which is the continuous counterpart of the total number of vertices:

$$\mathcal{C}(\mathbf{M}) = \int_\Omega d(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}.$$

Given a continuous mesh $\mathbf{M}$, we shall say, following [15, 16], that a discrete mesh $\mathcal{H}$ of the same domain Ω is a **unit mesh with respect to M**, if each tetrahedron $K \in \mathcal{H}$, defined by its list of edges $(\mathbf{e}_i)_{i=1\dots6}$, verifies:

$$\forall i \in [1,6], \quad \ell_{\mathcal{M}}(\mathbf{e}_i) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right] \quad \text{and} \quad Q_{\mathcal{M}}(K) \in [\alpha, 1] \ \text{ with } \ \alpha > 0\,,$$

where the length and quality in the metric are defined similarly to [15, 16].

We want to emphasize that the set of all the discrete meshes that are unit meshes with respect to a unique $\mathbf{M}$ contains an infinite number of meshes. Given a smooth function u, to each unit mesh $\mathcal{H}$ corresponds a local interpolation error $|u - \Pi u|$. In [15, 16], it is shown that all these interpolation errors are well represented by the so-called continuous interpolation error related to $\mathbf{M}$, which is expressed locally in terms of the Hessian H_u of u as follows:

$$(u - \pi_{\mathcal{M}} u)(\mathbf{x}, t) = \frac{1}{10}\mathrm{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\,|H_u(\mathbf{x}, t)|\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x}))\,. \tag{15}$$

3.3 Continuous error model

Working in this framework enables us to write Estimate (14) in a continuous form:

$$\begin{aligned} |(g, W_h - W)| \approx \mathbf{E}(\mathbf{M}) = &\int_0^T \int_\Omega |W^*|\ |\big(W - \pi_{\mathcal{M}} W\big)_t|\,\mathrm{d}\Omega\,\mathrm{d}t \\ &+ \int_0^T \int_\Omega |\nabla W^*|\ |\mathcal{F}(W) - \pi_{\mathcal{M}}\mathcal{F}(W)|\,\mathrm{d}\Omega\,\mathrm{d}t\,, \end{aligned} \tag{16}$$

where $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is a continuous mesh and $\pi_{\mathcal{M}}$ is the continuous linear interpolate. Then, introducing the continuous interpolation error, we can write the simplified error model as follows:

$$\mathbf{E}(\mathbf{M}) \ = \ \int_0^T \int_\Omega \mathrm{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\,\mathbf{H}(\mathbf{x},t)\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\right)\,\mathrm{d}\Omega\,\mathrm{d}t$$

$$\text{with} \quad \mathbf{H}(\mathbf{x},t) = \sum_{j=1}^{5} \left([\Delta t]_j(\mathbf{x},t) + [\Delta x]_j(\mathbf{x},t) + [\Delta y]_j(\mathbf{x},t) + [\Delta z]_j(\mathbf{x},t)\right), \tag{17}$$

$$\text{in which} \qquad [\Delta t]_j = \left|W_j^*\right| \cdot \left|H(W_{j,t})\right|, \qquad [\Delta x]_j = \left|\frac{\partial W_j^*}{\partial x}\right| \cdot \left|H(\mathcal{F}_1(W_j))\right|,$$

$$[\Delta y]_j = \left|\frac{\partial W_j^*}{\partial y}\right| \cdot \left|H(\mathcal{F}_2(W_j))\right|, \qquad [\Delta z]_j = \left|\frac{\partial W_j^*}{\partial z}\right| \cdot \left|H(\mathcal{F}_3(W_j))\right|.$$

Here, W_j^* denotes the j^{th} component of the adjoint vector W^*, $H(\mathcal{F}_i(W_j))$ the Hessian of the j^{th} component of the vector $\mathcal{F}_i(W)$, and $H(W_{j,t})$ the Hessian of the j^{th} component of the time derivative of W. The mesh optimization problem writes:

$$\text{Find } \mathbf{M}_{opt} = \text{Argmin}_{\mathbf{M}}\, \mathbf{E}(\mathbf{M}) \quad \text{under the constraint} \quad \mathcal{C}_{st}(\mathbf{M}) = N_{st}, \tag{18}$$

where N_{st} is a specified total number of nodes. Since we consider an unsteady problem, the space-time (st) complexity used to compute the solution takes into account the time discretization:

$$\mathcal{C}_{st}(\mathbf{M}) = \int_0^T \tau(t)^{-1} \left(\int_\Omega d_{\mathcal{M}}(\mathbf{x},t)\mathrm{d}\mathbf{x} \right) \mathrm{d}t \tag{19}$$

where $\tau(t)$ is the time step used at time t of interval $[0,T]$.

3.4 Spatial minimization for a fixed t

Let us assume that at time t, we seek for the optimal continuous mesh $\mathbf{M}_{go}(t)$ which minimizes the instantaneous error, *i.e.,* the spatial error for a fixed time t:

$$\tilde{\mathbf{E}}(\mathbf{M}(t)) = \int_\Omega \text{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\, \mathbf{H}(\mathbf{x},t)\, \mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t) \right) \mathrm{d}\mathbf{x}$$

under the constraint that the number of vertices is prescribed to

$$\mathcal{C}(\mathbf{M}(t)) = \int_\Omega d_{\mathcal{M}(t)}(\mathbf{x},t)\, \mathrm{d}\mathbf{x} = N(t). \tag{20}$$

Similarly to [17], solving the optimality conditions provides the *optimal goal-oriented instantaneous continuous mesh* $\mathbf{M}_{go}(t) = (\mathcal{M}_{go}(\mathbf{x},t))_{\mathbf{x}\in\Omega}$ at time t defined by:

$$\mathcal{M}_{go}(\mathbf{x},t) = N(t)^{\frac{2}{3}}\, \mathcal{M}_{go,1}(\mathbf{x},t)\,, \tag{21}$$

where $\mathcal{M}_{go,1}$ is the optimum for $\mathcal{C}(\mathbf{M}(t)) = 1$:

$$\mathcal{M}_{go,1}(\mathbf{x},t) = \mathcal{K}(t)^{-\frac{2}{5}}\, (\det \mathbf{H}(\mathbf{x},t))^{-\frac{1}{5}}\, \mathbf{H}(\mathbf{x},t). \tag{22}$$

with $\mathcal{K}(t) = \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{1}{5}} \mathrm{d}\mathbf{x} \right)^{\frac{5}{3}}$. The corresponding optimal instantaneous error at time t writes:

$$\tilde{\mathbf{E}}(\mathbf{M}_{go}(t)) = 3\, N(t)^{-\frac{2}{3}} \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{1}{5}} \mathrm{d}\mathbf{x} \right)^{\frac{5}{3}} = 3\, N(t)^{-\frac{2}{3}}\, \mathcal{K}(t)\,.$$

3.5 Temporal minimization

To complete the resolution of optimization Problem (18), we perform a temporal minimization in order to get the optimal space-time continuous mesh. In other words, we need to find the optimal time law $t \to N(t)$ for the instantaneous mesh size. Here, we only consider the simpler case where the time step τ is specified by the user as a function of time $t \to \tau(t)$. A similar analysis can be done to deal with the case of an explicit time advancing solver subject to Courant time step condition, but such an analysis is out of the scope of this proceeding.

Let us consider the case where the time step τ is specified by a function of time $t \to \tau(t)$. After the spatial optimization, the space-time error writes:

$$\mathbf{E}(\mathbf{M}_{go}) = \int_0^T \tilde{\mathbf{E}}(\mathbf{M}_{go}(t))\,\mathrm{d}t = 3\int_0^T N(t)^{-\frac{2}{3}}\,\mathcal{K}(t)\,\mathrm{d}t \tag{23}$$

and we aim at minimizing it under the following space-time complexity constraint:

$$\int_0^T N(t)\tau(t)^{-1}\,\mathrm{d}t = N_{st}. \tag{24}$$

In other words, we concentrate on seeking for *the optimal distribution of $N(t)$ when the space-time total number of nodes N_{st} is prescribed*. Let us apply the one-to-one change of variables:

$$\tilde{N}(t) = N(t)\tau(t)^{-1} \quad \text{and} \quad \tilde{\mathcal{K}}(t) = \tau(t)^{-\frac{2}{3}}\,\mathcal{K}(t)\,.$$

Then, our temporal optimization problem becomes:

$$\min_{\mathbf{M}}\ \mathbf{E}(\mathbf{M}) = \int_0^T \tilde{N}(t)^{-\frac{2}{3}}\,\tilde{\mathcal{K}}(t)\,\mathrm{d}t \quad \text{under constraint} \quad \int_0^T \tilde{N}(t)\,\mathrm{d}t = N_{st}\,.$$

The solution of this problem is given by:

$$\tilde{N}_{opt}(t)^{-\frac{5}{3}}\,\tilde{\mathcal{K}}(t) = const \quad \Rightarrow \quad N_{opt}(t) = C(N_{st})\ (\tau(t)\,\mathcal{K}(t))^{\frac{3}{5}}$$

Here, constant $C(N_{st})$ can be obtained by introducing the above expression in space-time complexity Constraint (24), leading to:

$$C(N_{st}) = \left(\int_0^T \tau(t)^{-\frac{2}{5}}\,\mathcal{K}(t)^{\frac{3}{5}}\mathrm{d}t\right)^{-1} N_{st}\,,$$

which completes the description of the optimal space-time metric for a prescribed time step. Using Relation (21), the analytic expression of the optimal space-time goal-oriented metric $\mathbf{M}_{go}$ writes:

$$\mathcal{M}_{go}(\mathbf{x},t) = N_{st}^{\frac{2}{3}}\left(\int_0^T \tau(t)^{-\frac{2}{5}}\mathcal{K}(t)^{\frac{3}{5}}\mathrm{d}t\right)^{-\frac{2}{3}} \tau(t)^{\frac{2}{5}}(\det\mathbf{H}(\mathbf{x},t))^{-\frac{1}{5}}\mathbf{H}(\mathbf{x},t)\,. \tag{25}$$

We get the following optimal error:

$$\mathbf{E}(\mathbf{M}_{go}) = 3\,N_{st}^{-\frac{2}{3}}\ \left(\int_0^T \tau(t)^{-\frac{2}{5}}\mathcal{K}(t)^{\frac{3}{5}}\mathrm{d}t\right)^{\frac{5}{3}}\,. \tag{26}$$

3.6 Space-time minimization for time sub-intervals

The previous analysis provides the optimal size of the adapted meshes for each time level. Hence, this analysis requires the mesh to be adapted at each flow solver time step. But, in practice this approach involves a very large number of remeshing which is CPU consuming and spoils solution accuracy due to many solution transfers from one mesh to a new one. In consequence, a new adaptive strategy has been proposed in [1, 4] where the number of remeshing is controlled (thus drastically reduced) by

generating adapted meshes for several solver time steps. The idea is to split the simulation time interval into n_{adap} *sub-intervals* $[t_i, t_{i+1}]$ for $i = 1, .., n_{adap}$. Each spatial mesh $\mathbf{M}^i$ is then kept constant during each sub-interval $[t_i, t_{i+1}]$. We could consider this partition as a *time discretization of the mesh adaptation problem.*

Spatial minimization on a sub-interval. Given the continuous mesh complexity N_i for the single adapted mesh used during time sub-interval $[t_i, t_{i+1}]$, we seek for the optimal continuous mesh $\mathbf{M}^i_{go}$ solution of the following problem:

$$\min_{\mathbf{M}^i} \mathbf{E}^i(\mathbf{M}^i) = \int_\Omega \text{trace}\left((\mathcal{M}^i)^{-\frac{1}{2}}(\mathbf{x})\,\mathbf{H}^i(\mathbf{x})\,(\mathcal{M}^i)^{-\frac{1}{2}}(\mathbf{x})\right)\,\mathrm{d}\mathbf{x} \quad \text{such that} \quad \mathcal{C}(\mathbf{M}^i) = N^i\,,$$

where hessian metric $\mathbf{H}^i$ on the sub-interval can be defined by either using an $\mathbf{L}^1$ or an $\mathbf{L}^\infty$ norm:

$$\mathbf{H}^i_{\mathbf{L}^1}(\mathbf{x}) = \int_{t_i}^{t_{i+1}} \mathbf{H}(\mathbf{x},t)\,\mathrm{d}t \quad \text{or} \quad \mathbf{H}^i_{\mathbf{L}^\infty}(\mathbf{x}) = \Delta t_i \max_{t\in[t_i,t_{i+1}]} \mathbf{H}(\mathbf{x},t)\,,$$

with $\Delta t_i = t_{i+1} - t_i$. Processing as previously, we get the spatial optimality condition:

$$\mathcal{M}^i_{go}(\mathbf{x}) = (N^i)^{\frac{2}{3}}\ \mathcal{M}^i_{go,1}(\mathbf{x}) \quad \text{with } \mathcal{M}^i_{go,1}(\mathbf{x}) = (\mathcal{K}^i)^{-\frac{2}{5}}\,(\det \mathbf{H}^i(\mathbf{x}))^{-\frac{1}{5}}\,\mathbf{H}^i(\mathbf{x}).$$

The corresponding optimal error $\mathbf{E}^i(\mathbf{M}^i_{go})$ writes:

$$\mathbf{E}^i(\mathbf{M}^i_{go}) = 3\,(N^i)^{-\frac{2}{3}}\left(\int_\Omega (\det \mathbf{H}^i(\mathbf{x}))^{\frac{1}{5}}\mathrm{d}\mathbf{x}\right)^{\frac{5}{3}} = 3\,(N^i)^{-\frac{2}{3}}\,\mathcal{K}^i\,.$$

Temporal minimization for specified τ. To complete our analysis, we shall now perform a temporal minimization. After the spatial minimization, the temporal optimization problem reads:

$$\min_{\mathbf{M}} \mathbf{E}(\mathbf{M}) = 3\sum_{i=1}^{n_{adap}} (N^i)^{-\frac{2}{3}}\,\mathcal{K}^i \quad \text{such that} \quad \sum_{i=1}^{n_{adap}} N^i\left(\int_{t_i}^{t_{i+1}} \tau(t)^{-1}\mathrm{d}t\right) = N_{st}\,.$$

We set the one-to-one mapping:

$$\tilde{N}^i = N^i\left(\int_{t_i}^{t_{i+1}} \tau(t)^{-1}\mathrm{d}t\right) \quad \text{and} \quad \tilde{\mathcal{K}}^i = \mathcal{K}^i\left(\int_{t_i}^{t_{i+1}} \tau(t)^{-1}\mathrm{d}t\right)^{\frac{2}{3}}\,,$$

then the optimization problem reduces to:

$$\min_{\mathbf{M}} \sum_{i=1}^{n_{adap}} (\tilde{N}^i)^{-\frac{2}{3}}\,\tilde{\mathcal{K}}^i \quad \text{such that} \quad \sum_{i=1}^{n_{adap}} \tilde{N}^i = N_{st}\,.$$

We deduce the optimal continuous mesh $\mathbf{M}_{go} = \{\mathbf{M}^i_{go}\}_{i=1,..,n_{adap}}$ and error:

$$\mathcal{M}^i_{go}(\mathbf{x}) = N_{st}^{\frac{2}{3}}\left(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{3}{5}}\,\mathcal{T}^i\right)^{-\frac{2}{3}} (\mathcal{T}^i)^{-1}(\det \mathbf{H}^i(\mathbf{x}))^{-\frac{1}{5}}\,\mathbf{H}^i(\mathbf{x})$$

$$\mathbf{E}(\mathbf{M}) = 3\,N_{st}^{-\frac{2}{3}}\left(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{3}{5}}\,\mathcal{T}^i\right)^{\frac{5}{3}}\,,$$

with $(\mathcal{K}^i)^{\frac{3}{5}} = \displaystyle\int_\Omega (\det \mathbf{H}^i(\mathbf{x}))^{\frac{1}{5}}\mathrm{d}\mathbf{x}$ and $\mathcal{T}^i = \left(\displaystyle\int_{t_i}^{t_{i+1}} \tau(t)^{-1}\mathrm{d}t\right)^{\frac{2}{5}}$.

4 From theory to practice

In order to remedy all the problematics relative to mesh adaptation for time-dependent simulations stated in the introduction, an innovative strategy based on a fixed-point algorithm has been initiated in [2] and fully developed in [1]. The fixed-point algorithm aims at avoiding the generation of a new mesh at each solver iteration which would imply serious degradation of the CPU time and of the solution accuracy due to the large number of mesh modifications. It is also an answer to the lag problem occurring when computing the solution at t^n and accordingly adapt the mesh at each time step. Indeed, by doing this, the mesh is always late as compared to the solution as it is not adapted for the displacement of the solution between t^n and t^{n+1}.

The basic idea consists in splitting the simulation time frame $[0,\, T]$ into n_{adap} adaptation sub-intervals:

$$[0,\, T] \;=\; [0 = t_0,\, t_1] \cup \ldots \cup [t_i,\, t_{i+1}] \cup \ldots \cup [t_{n_{adap}-1},\, t_{n_{adap}}]\,,$$

and to keep the same adapted mesh for each sub-interval. On each sub-interval, the mesh is adapted to control the solution accuracy from t^n to t^{n+1}. Consequently, the time-dependent simulation is performed with n_{adap} different adapted meshes. This can be seen as a coarse discretization of the time axis where the spatial mesh is constant for each sub-interval when the global space-time mesh is visualized, providing thus a first step in the adaptation of the whole space-time mesh.

4.1 Choice of the goal-oriented metric

The optimal adapted meshes for each sub-interval are generated according to analysis of Section 3.6. In this work, the following particular choice has been made:

- the hessian metric for sub-interval i is based on a control of the temporal error in $\mathbf{L}^\infty$ norm:

$$\mathbf{H}^i_{\mathbf{L}^\infty}(\mathbf{x}) = \Delta t_i \max_{t\in[t_i,t_{i+1}]} \mathbf{H}(\mathbf{x},t) = \Delta t_i\, \mathbf{H}^i_{\max}(\mathbf{x})\,,$$

- function $\tau : t \to \tau(t)$ is constant and equal to 1,
- all sub-intervals have the same time length Δt.

The optimal goal-oriented metric $\mathbf{M}_{go} = \{\mathbf{M}^i_{go}\}_{i=1,..,n_{adap}}$ then simplifies to:

$$\mathcal{M}^i_{go}(\mathbf{x}) = N_{st}^{\frac{2}{3}} \left(\sum_{i=1}^{n_{adap}} \Big(\int_\Omega (\det \mathbf{H}^i_{\max}(\mathbf{x}))^{\frac{1}{5}} \mathrm{d}\mathbf{x} \Big) \right)^{-\frac{2}{3}} (\Delta t)^{\frac{1}{3}} \left(\det \mathbf{H}^i_{\max}(\mathbf{x}) \right)^{-\frac{1}{5}} \mathbf{H}^i_{\max}(\mathbf{x})\,.$$

Remark: We notice that we obtain a similar expression of the optimal metric as the one proposed in [4], but here in the goal-oriented context and by means of a space-time error minimization.

4.2 Global fixed-point mesh adaptation algorithm

To converge the non-linear mesh adaptation problem, *i.e.,* converging the couple mesh-solution, we propose a fixed-point mesh adaptation algorithm. This is also a way to predict the solution evolution and to adapt the mesh accordingly. Nevertheless, to compute all metrics fields $\mathbf{M}^i_{go}$, we have to evaluate the global normalization term which requires the knowledge of all $\mathbf{H}^i_{\max}$. Thus, all the simulation must be performed to be able to evaluate all metrics $\mathbf{M}^i_{go}$. Similarly to [4], a global fixed point strategy covering the whole time-frame $[0, T]$, called *Global adjoint fixed-point mesh adaptation algorithm*, is considered:

```
//--- Fixed-point loop to converge the global space-time mesh adaptation
For j=1,nptfx
  //--- Solve state once to get checkpoints
  For i=1,n_adap
    • S^j_{0,i} = ConservativeSolutionTransfer(H^j_{i-1}, S^j_{i-1}, H^j_i)
    • S^j_i = SolveStateForward(S^j_{0,i}, H^j_i)
  End for
  //--- Solve state and adjoint backward and store samples
  For i=n_adap,1
    • (S*)^j_i = AdjointStateTransfer(H^j_{i+1}, (S*_0)^j_{i+1}, H^j_i)
    • {S^j_i(k), (S*)^j_i(k)}_{k=1,n_k} = SolveStateAndAdjointBackward(S^j_{0,i}, (S*)^j_i, H^j_i)
    • (H_max)^j_i = ComputeGoalOrientedHessianMetric(H^j_i, {S^j_i(k), (S*)^j_i(k)}_{k=1,n_k})
  End for
  • C^j = ComputeSpaceTimeComplexity({(H_max)^j_i}_{i=1,n_adap})
  • {M^j_i}_{i=1,n_adap} = ComputeUnsteadyGoalOrientedMetrics(C^j, {|H_max|^j_i}_{i=1,n_adap})
  • {H^{j+1}_i}_{i=1,n_adap} = GenerateAdaptedMeshes({H^j_i}_{i=1,n_adap}, {M^j_i}_{i=1,n_adap})
End for
```

5 Numerical examples

The procedures described in this paper have been implemented in our Finite Element/Finite Volume CFD code `Wolf` which is thoroughly detailed in [3]. As regards the meshing part, we consider a local remeshing strategy. We use `Yams` [10] for the adaptation in 2D and `Feflo.a` [19] in 3D.

2D Blast wave propagation. We first apply the goal-oriented adaptive strategy to the example presented in Section 2.3. It consists in a 2D blast in a geometry representing a city. We recall that the cost function of interest j is the quadratic deviation of the pressure ambient pressure on target surface S (see Figure 1):

$$j(W) = \int_0^T \int_S \frac{1}{2}(p(t) - p_{air})^2 \mathrm{d}S\mathrm{d}t.$$

The simulation time frame is split into 30 time sub-intervals, *i.e.,* 30 different adapted meshes are used to perform the simulation. Hence, 30 checkpoints are stored for the backward computation of the unsteady adjoint.

The resulting adjoint-based anisotropic adapted meshes are shown in Figure 4. The corresponding density isolines are depicted in Figure 5. These adapted meshes

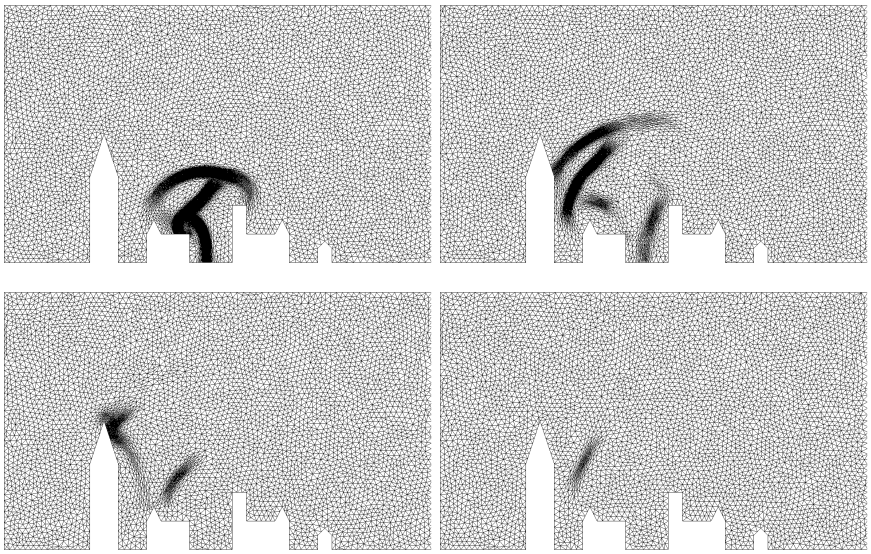


Fig. 4. 2D city blast adjoint-based adapted meshes evolution. From top to bottom and left to right, meshes corresponding to sub-intervals 8, 15, 22 and 29 at a-dimensional time 1.2, 2.25, 3.3 and 4.35.

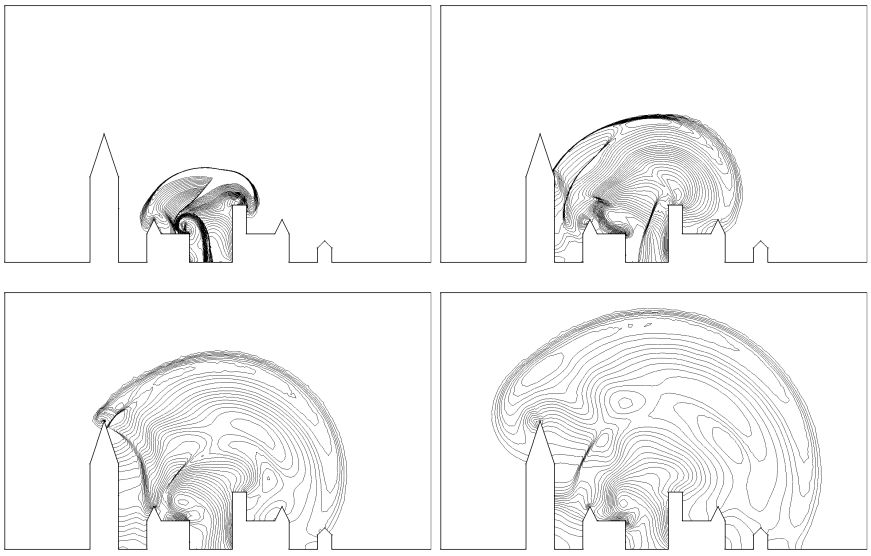


Fig. 5. 2D city blast adaptive solution state evolution. From top to bottom and left to right, density iso-lines corresponding to the end of sub-intervals 8, 15, 22 and 29 at a-dimensional time 1.2, 2.25, 3.3 and 4.35.

indubitably illustrate that, thanks to the unsteady adjoint, the mesh adaptation only focuses on shock waves that impact the observation region and ignores other area of the flow field. Therefore, waves traveling toward the observation are accurately captured whereas the rest of the flow is poorly computed. We also observe that once waves go beyond the target surface, the mesh is no more refined even if they continue traveling throughout the computational domain. Indeed, they do not impact anymore the functional.

It is then quite interesting to compare the Hessian-based approach of [4] with our adjoint-based method. This comparison is shown in Figure 6. It demonstrates how the adjoint defines an optimal distribution of the degrees of freedom for the specific functional, while it is clear that in this context the Hessian-based approach gives a non-optimal result for the evaluation of the functional but capture accurately the whole flow.

In conclusion, if an output functional of interest is provided then the reduction of the simulation number of degrees of freedom can be even more improved by considering a goal-oriented analysis instead of an Hessian-based methodology.

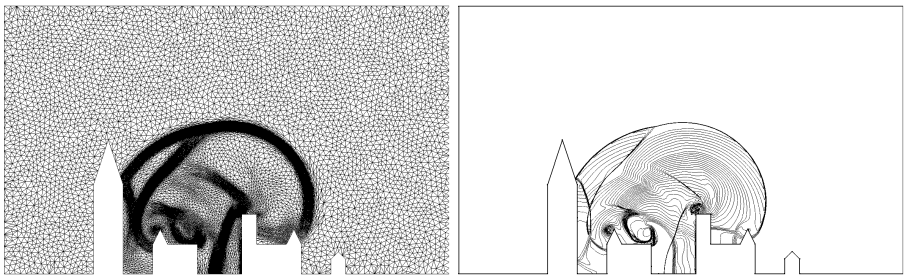


Fig. 6. Adapted mesh and corresponding density iso-lines for sub-interval 15 at a-dimensional time 2.25 obtained with the Hessian-based method of [4].

3D Blast wave propagation. Finally, we consider exactly the same blast test case but in a 3D city geometry. Cost function j is again the quadratic deviation from ambient pressure on target surface Γ which is composed of one building for simulation 1 or two buildings for simulation 2, see Figure 7. The simulation time frame is split into 40 time sub-intervals, *i.e.,* 40 different adapted meshes are used to perform the simulation.

The resulting adjoint-based anisotropic adapted meshes (surface and volume) for both simulations at sub-interval 10, 15 and 20 are shown in Figures 8 and 9. It is very interesting to see that we are not restricted to just one target surface. As previously in 2D, we notice that mesh refinement only focuses on shock waves that will impact the target buildings. Other waves are neglected thus leading to a large reduction of the mesh size. To illustrate this point, we provide meshes size for several sub-intervals for simulation 2: for sub-intervals 1, 5, 10 and 20 the mesh number of vertices is respectively 1 051 805, 678 802, 233 116 and 45 500. The mesh size has been reduced by a factor 20 between the first and the twentieth sub-interval.

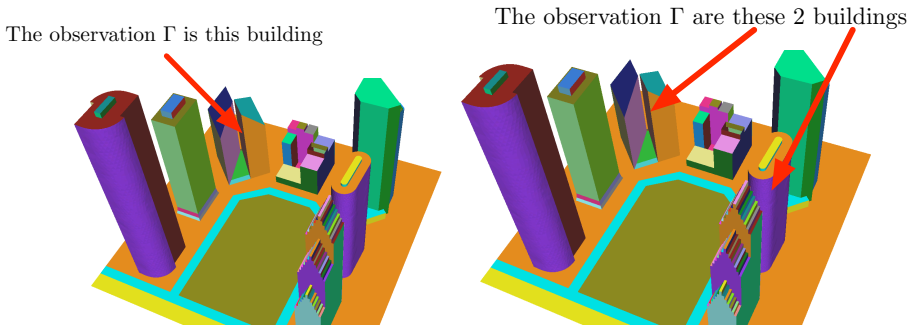


Fig. 7. 3D City test case geometry and location of target surface Γ composed of one building for simulation 1 (left) or two buildings for simulation 2 (right).

6 CONCLUSIONS

We have designed a new mesh adaptation algorithm which prescribes the spatial mesh of an unsteady simulation as the optimum of a goal-oriented error analysis. This method specifies both mesh density and mesh anisotropy by variational calculus. Accounting for unsteadiness is applied in a time-implicit mesh-solution coupling which needs a non-linear iteration, the fixed point. In contrast to the Hessian-based fixed-point of [1] which iterates on each sub-interval, the new iteration covers the whole time interval, including forward steps for evaluating the state and backward ones for the adjoint. This algorithm was applied to 2D and 3D blast wave Euler test cases. Numerical results demonstrate the favorable behavior expected from an adjoint-based method, which gives an automatic selection of the mesh necessary for the target output.

Several important issues remain to be addressed. Among them, the strategies for choosing the splitting in time sub-intervals and the accurate integration of time errors in the mesh adaptation process with a more general formulation of the mesh optimization problem is examined seriously in [4].

Time discretization error is not considered in this study. Solving this question is not so important for the type of calculation that are shown in this paper, but can be of paramount impact in many other cases, in particular when implicit time advancing is used. The additional effort in time-error reduction has to be integrated in the convergence analysis sketched in this paper, and the authors plan to propose this global analysis in some future.

References

1. F. Alauzet, P.J. Frey, P.L. George, and B. Mohammadi. 3D transient fixed point mesh adaptation for time-dependent problems: Application to CFD simulations. *J. Comp. Phys.*, 222:592–623, 2007.
2. F. Alauzet, P.L. George, B. Mohammadi, P.J. Frey, and H. Borouchaki. Transient fixed point based unstructured mesh adaptation. *Int. J. Numer. Meth. Fluids*, 43(6-7):729–745, 2003.

3. F. Alauzet and A. Loseille. High order sonic boom modeling by adaptive methods. *J. Comp. Phys.*, 229:561–593, 2010.
4. F. Alauzet and G. Olivier. Extension of metric-based anisotropic mesh adaptation to time-dependent problems involving moving geometries. In *49th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2011-0896, Orlando, FL, USA, Jan 2011.
5. M.J. Baines. *Moving finite elements*. Oxford University Press, Inc., New York, NY, 1994.
6. R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.
7. M. Berger and P. Colella P. Local adaptive mesh refinement for shock hydrodynamics. *J. Comp. Phys.*, 82(1):67–84, 1989.
8. P.-H. Cournède, B. Koobus, and A. Dervieux. Positivity statements for a Mixed-Element-Volume scheme on fixed and moving grids. *European Journal of Computational Mechanics*, 15(7-8):767–798, 2006.
9. J. Dompierre, M.G. Vallet, M. Fortin, Y. Bourgault, and W.G. Habashi. Anisotropic mesh adaptation: towards a solver and user independent CFD. In *AIAA 35th Aerospace Sciences Meeting and Exhibit*, AIAA-1997-0861, Reno, NV, USA, Jan 1997.
10. P.J. Frey. `Yams`, a fully automatic adaptive isotropic surface remeshing procedure. RT-0252, INRIA, November 2001.
11. M.B. Giles and E. Suli. *Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality*, pages 145–236. Cambridge University Press, 2002.
12. F. Hecht and B. Mohammadi. Mesh adaptation by metric control for multi-scale phenomena and turbulence. *AIAA Paper*, 97-0859, 1997.
13. R. Löhner. Adaptive remeshing for transient problems. *Comput. Methods Appl. Mech. Engrg.*, 75:195–214, 1989.
14. A. Loseille and F. Alauzet. Optimal 3D highly anisotropic mesh adaptation based on the continuous mesh framework. In *Proceedings of the 18th International Meshing Roundtable*, pages 575–594. Springer, 2009.
15. A. Loseille and F. Alauzet. Continuous mesh framework. Part I: well-posed continuous interpolation error. *SIAM J. Numer. Anal.*, 49(1):38–60, 2011.
16. A. Loseille and F. Alauzet. Continuous mesh framework. Part II: validations and applications. *SIAM J. Numer. Anal.*, 49(1):61–86, 2011.
17. A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. *J. Comp. Phys.*, 229:2866–2897, 2010.
18. A. Loseille, A. Dervieux, P.J. Frey, and F. Alauzet. Achievement of global second-order mesh convergence for discontinuous flows with adapted unstructured meshes. In *37th AIAA Fluid Dynamics Conference and Exhibit*, AIAA-2007-4186, Miami, FL, USA, Jun 2007.
19. A. Loseille and R. Löhner. Adaptive anisotropic simulations in aerodynamics. In *48th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2010-169, Orlando, FL, USA, Jan 2010.
20. R. Verfürth. *A review of A Posteriori Error Estimation and Adaptative Mesh-Refinement techniques*. Wiley Teubner Mathematics, New York, 1996.
21. M. Wintzer, M. Nemec, and M.J. Aftosmis. Adjoint-based adaptive mesh refinement for sonic boom prediction. In *AIAA 26th Applied Aerodynamics Conference*, AIAA-2008-6593, Honolulu, HI, USA, Aug 2008.

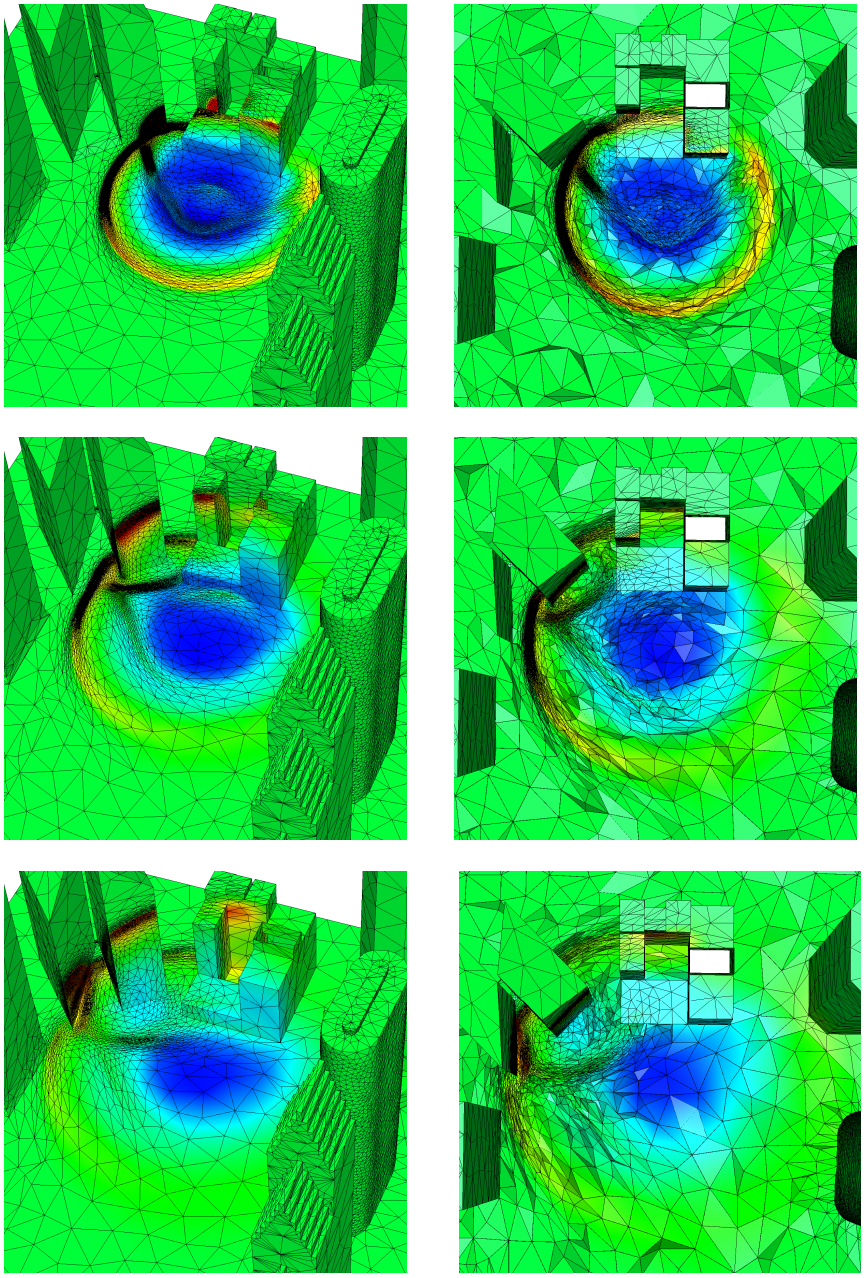


Fig. 8. 3D Blast wave propagation: simulation1. Adjoint-based anisotropic adapted surface (left) and volume (right) meshes at sub-interval 10, 15 and 20 and corresponding solution density at a-dimensioned time 5, 7.5 and 10.

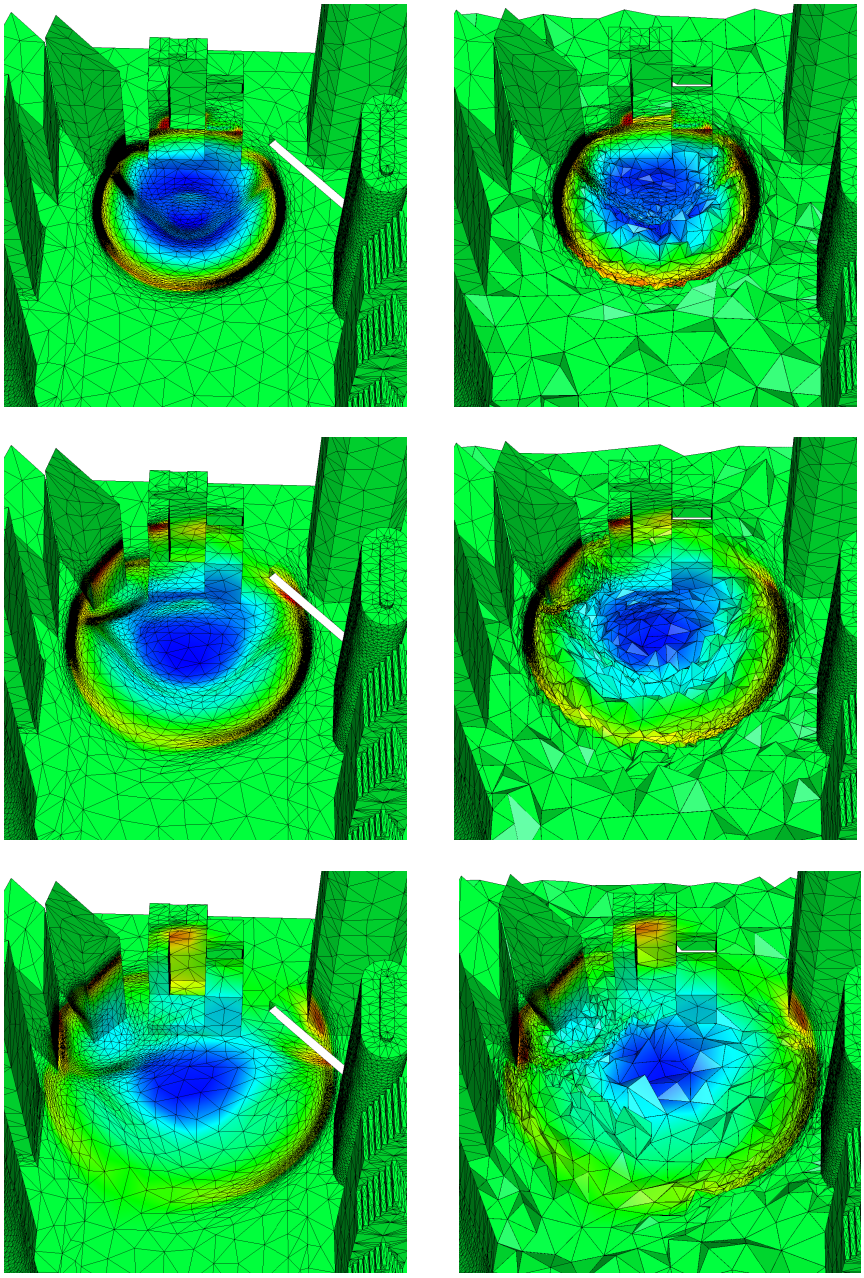


Fig. 9. 3D Blast wave propagation: simulation2. Adjoint-based anisotropic adapted surface (left) and volume (right) meshes at sub-interval 10, 15 and 20 and corresponding solution density at a-dimensioned time 5, 7.5 and 10.