Fast 3D computations of compressible flow discharge in buildings and complex networks

Alexandre Chiapolino[1a], François Fraysse[2a], Richard Saurel[3a,b]

[a]RS2N SAS, Saint-Zacharie, France
[b]Aix Marseille Univ, CNRS, Centrale Marseille, LMA, Marseille, France

Abstract

This paper presents under-resolved simulations of compressible flows in complex structures like buildings. The mean pressure field is the primary focus as it is of high importance in many cases, such as the overpressure generated by the detonation of an explosive charge. In such high-risk situations, the mean pressure in the different building rooms has to be determined as quickly as possible, in order to help first responders evaluating the risk level of entering the building right after an explosion for instance. Conventional computational fluid dynamics methods are obviously able to deliver those requested pressure fields. However, they also require a long and tedious meshing process due to the presence of small openings in the computational domain (doors, windows or stairwells). In turn, these small openings need spatial and temporal resolution having significant consequences on computation time. But when dealing with pressing situations, a long pre-processing is not acceptable. The present paper proposes a two-step solution method. First, a very rough mesh is constructed which does not consider small openings. The mesh is constructed in an easy and particular way such that simplifications can be made when computing the fluid flux through these openings. Second, a specific Riemann solver dealing with geometric discontinuities is developed to estimate the fluid flow. The Riemann solver takes advantage of the simplifications provided by the meshing method, and accounts for both unchoked and choked flows through openings. The overall method then consists of both a specific meshing process that requires very little pre-processing, which considerably reduces the time needed to produce a complete simulation, and a Riemann solver. The proposed method is named MUZO in reference to its MUlti-ZOne flow solver. The MUZO solver is validated against resolved computations and provides accurate simulated mean pressure fields within a realistic building in a few seconds.

Keywords: Riemann problem, discontinuous area change, choked flow, under-resolved computations, 3D unstructured mesh

[1]Corresponding author: alexandre.chiapolino@rs2n.eu
[2]francois.fraysse@rs2n.eu
[3]richard.saurel@univ-amu.fr

1. Introduction

In many situations, determination of the mean (or quasistatic) pressure field is of importance. Relevant examples are pressure fields resulting from explosions in buildings or in complex structures like aircrafts, plane wings, industrial plants to cite a few. In such a pressing and hazardous situation, knowledge of the mean pressure in the different parts of the structure is needed and is to be determined as quickly as possible, to help first responders evaluate the risk of entering the structure after the explosion for instance. In these situations, it is obviously possible to build a 3D mesh and compute an appropriate flow model, single-phase or two-phase, depending on the configuration. However, it is time consuming for two main reasons:

- Mesh definition, especially as geometric details are needed, such as windows, doors, and various openings. Many thin zones may have first-order effects on the overall flow field.

- Numerical computation of partial differential equations on such domains may be very demanding in computation time and resource. This is typically the case when small openings are present. Small numerical elements are consequently locally needed and affect the global time-step.

In most situations, explosion effects decouple in two timescales: shock and blast wave propagation (fast) and flow discharge effects (slow). When both wave propagation and flow discharge effects are strongly coupled, a conventional 3D computation seems to be the only relevant method. However, in most situations, wave propagation rapidly decouples from flow discharge effects. More precisely, computation of blast effects can be accurately achieved with reduced models, based for example on geometrical shock dynamics (Henshaw et al., 1986 [1], Schwendeman, 1993 [2], Whitham, 2011 [3], Ridoux et al., 2018 [4]). Simplified methods based on Kingery-Bulmash data are popular engineering alternatives (Kingery, 1966 [5], Coulter et al., 1988 [6], Karlos, 2016 [7]). Fast and efficient blast-effect computations have been shown in Frank et al. (2007) [8], Lapébie et al. (2016) [9] and Ruscade (2021) [10]. These simplified methods are very fast and mesh free. They couple Kingery-Bulmash data to an algorithm computing the shortest distance (Dijkstra, 1959 [11]) between two points, such as the explosive source and a given wall. The initial shock overpressure is consequently readily determined with the help of the Kingery-Bulmash curve and the shortest distance.

The present contribution considers that the effects of wave propagation occurring at early times are already taken into account through an appropriate above-mentioned method and focuses only on flow discharge effects occurring at longer timescales, having in mind that a fast method is desired, both for the geometry generation step and for the flow computation step. Indeed, the present effort attempts to create a simple, accurate and fast method to address hazardous and pressing situations, like the aforementioned ones, that require knowledge of the mean pressure fields. To do so, the method uses coarse meshes and few computational cells. The size of a computational cell is typically of the order of the size of a room in a building. Moreover, in order to design a simple and fast method for generating a geometry and its corresponding mesh, geometric details such as doors are neither drawn nor meshed. Only the "footprints" of the geometry are needed, from which a 2D planar mesh is generated. The design of the geometry consequently requires little effort and a conforming 3D mesh with as few elements as possible is then constructed by extruding the 2D mesh along the third, vertical, dimension.

However, using coarse meshes without geometric details requires a specific Riemann solver to take into account openings like doors and windows. Such a Riemann solver is addressed in the present contribution. It is used on the surfaces of elements involving an opening that are only marked during the mesh-generation step. The proposed Riemann solver addresses the previously omitted geometric restrictions directly in the solution states and through the flux distribution as well. Nevertheless, care is needed as choking conditions may be present at these surfaces and the corresponding fluxes must be computed accurately, as the pressure distribution is a direct consequence of the balance of the fluxes. Flux computation at these geometric restrictions is reminiscent of the Riemann problem in 1D channels with discontinuous area change (LeFloch and Thanh, 2003 [12], Warnecke and Andrianov, 2004 [13], Kroner and Thanh, 2005 [14], Thanh, 2009 [15], Han et al., 2012 [16]). In the present contribution, the 1D Riemann problem is revisited and analyzed in a simplified situation where right- and left-facing waves are approximated through acoustic relations. This approximation is sufficiently robust and accurate when dealing with waves of weak amplitude. Furthermore it is demonstrated that when the mesh is constructed in a particular manner, such that the cross-sections on both sides of a marked surface are the same, the geometric restriction becomes transparent in the Riemann problem, at least in the computation of the solution states when an unchoked flow is addressed.

The proposed method then consists of both a specific meshing process that requires very little pre-processing, greatly reducing the time needed to produce a complete simulation of the mean pressure-field in a building, and a special treatment of geometric discontinuities through the Riemann solver. The method is named MUZO in reference to its "MUlti-ZOne" flow solver. It is validated against resolved computations. The MUZO method unsurprisingly appears inaccurate to handle the dynamics of waves, due to the excessive cell size and the assumptions made to compute the solution, but appears accurate for the determination of the mean (or quasistatic) pressure field, which is a very important feature with regard to risk assessment. Isentropic and isenthalpic relations, resulting from a quasi-steady assumption, are indeed used to select the flow regime appropriate to the flow conditions, *i.e.* choked or unchoked, and provide the solution state at the opening. As supported by the results, this assumption is appropriate to address flow discharge effects. It is however inappropriate to deal with blast and wave propagation occurring at earlier times, considered through a fast and mesh free computation based on Kingery-Bulmash data [5] as mentioned earlier.

The stationary assumption is often considered in network problems. For instance Keith et al. (2005) [17] estimated sonic gas flow rates in pipelines with reduced adiabatic and isothermal models. In that context, the stationary assumption leads indeed to a 1D partial differential equation (PDE), integrated over the length of a pipeline. In a similar context, Ke and Ti (2000) [18] used steady state analysis of pipeline network and electrical analogy to develop a set of PDEs for transient analysis of isothermal gas flows. A hierarchy of pipeline models is provided in Brouwer et al. (2011) [19]. Those are of a simpler structure compared to the full Euler equations of gas dynamics and are designed to deal with 1D pipeline-like networks. In the present paper, the full Euler equations are considered to deal with 3D flows in complex structures. The proposed method may be considered as a 3D extension of network-like models. The velocity vector is indeed made of 3 components for each point in space. Flows changing directions are then addressed. Singular pressure drops occurring at geometric restrictions (doors, windows) are treated automatically through a specific Riemann solver.

The paper is organized as follows. The conventional method is first presented in Section 2. In that context, the mesh definition involves every geometric details and the 3D flow is computed with the help of the Euler equations without additional geometry-related terms. The solution is provided by the Godunov (1959) [20] method and the HLLC Riemann solver of Toro et al. (1994) [21] is used for flux computation. Such a method is computationally expensive as already mentioned. Then the new method is addressed in Section 3 with a specific Riemann solver that accounts for geometric restrictions. Doing so, mesh generation is significantly easier and faster. Section 4 then deals with such a particular mesh generation and the specific points simplifying the definition of the geometry. Section 5 is devoted to the MUZO Riemann problem with discontinuous area change and related flux computation, taking advantage of the simplifications provided by the proposed meshing method. Section 6 deals with validations against resolved computations and 3D computational examples of a realistic building. Conclusions are given in Section 7. Finally, an appendix is provided and presents the adaption of the Riemann solver to boundary conditions.

2. Conventional computation

Conventional computations are obviously able to deliver the pressure fields in the various rooms of a building. Those are based on a single-phase or two-phase flow model depending on the configuration. In this work, the single-phase compressible Euler equations are used,

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \operatorname{div}\left(\rho \mathbf{u}\right) = 0, \\ \dfrac{\partial \left(\rho \mathbf{u}\right)}{\partial t} + \operatorname{div}\left(\rho \mathbf{u} \otimes \mathbf{u} + p \underline{\underline{\mathbf{I}}}\right) = 0, \\ \dfrac{\partial \left(\rho E\right)}{\partial t} + \operatorname{div}\left(\left(\rho E + p\right) \mathbf{u}\right) = 0. \end{cases} \tag{2.1}$$

The notations are conventional. A frame of reference $\mathbf{X} = (x, y, z)$ is chosen and the time variable is denoted by t. ρ is the density, $\mathbf{u} = (u, v, w)$ is the velocity vector where u, v and w represent respectively the x-component, y-component and z-component of velocity. E is the total energy defined as: $E = e + \frac{1}{2}\left(\mathbf{u} \cdot \mathbf{u}\right)$ where e is the internal energy. div is the *divergence operator*, $\mathbf{u} \otimes \mathbf{u}$ is the tensor product and $\underline{\underline{\mathbf{I}}}$ is the unit tensor. The pressure p is computed with the caloric ideal-gas equation of state,

$$p = \left(\gamma - 1\right) \rho e, \tag{2.2}$$

where γ represents the isentropic exponent, or ratio of the specific heats $\gamma = C_p/C_v$. Under the thermal form, the ideal-gas equation of state reads,

$$p = \rho \left(\gamma - 1\right) C_v T, \tag{2.3}$$

where T is the temperature. The x-split one-dimensional system is hyperbolic with real eigenvalues $\lambda_1 = u - c$, $\lambda_2 = \lambda_3 = \lambda_4 = u$, $\lambda_5 = u + c$ (with c being the sound speed), corresponding to the wave speeds. $\lambda_3 = \lambda_4 = u$ arises from the multiplicity 3 of the eigenvalue u (in the 3D case). These correspond to two *shear waves* across which the respective tangential velocity components v and w change discontinuously. Note that in three dimensions, the λ_4-field is linearly degenerate. The 1 and 5 characteristic fields are genuinely non-linear and are associated with rarefactions or shock waves (just as in the one-dimensional case). In this work, the solution of System (2.1) is approximated with the finite-volume Godunov (1959) [20] method, that requires to solve the associated Riemann problem on every surface of the numerical elements composing the mesh. Higher-order extensions of the Godunov method are not addressed as wave capturing is not considered. Fast under-resolved computations are of interest. The various openings of the building need to be drawn and meshed during the pre-processing step. The corresponding numerical surfaces then fit the shape of the openings such as doors, and the associated Riemann solver accounts naturally for geometrical effects. Consequently, no specific treatment is required for the Riemann solver computing the fluid flux through the surfaces of the elements, with the exception of the boundary surfaces connected to the atmosphere. Boundary conditions are treated with the Riemann solver presented in Appendix B. The fluxes associated with the internal surfaces are computed with the HLLC solver of Toro et al. (1994) [21]. Before addressing the MUZO method, let us present shortly the Godunov method and the HLLC Riemann solver. The corresponding results will be considered as reference solutions and will be useful to assess the accuracy of the MUZO method.

2.1 Godunov method

System (2.1) may be written under the following form,

$$\frac{\partial \mathbf{U}}{\partial t} + \operatorname{div} \mathbf{F} = 0, \tag{2.4}$$

where $\mathbf{U}$ and $\mathbf{F}$ are the vectors of conservative variables and corresponding fluxes. The vector $\mathbf{W}$ of primitive variables is defined as well,

$$\mathbf{W} = \begin{pmatrix} \rho \\ \mathbf{u} \\ p \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\otimes\mathbf{u} + p\underline{\underline{\mathbf{I}}} \\ (\rho E + p)\,\mathbf{u} \end{pmatrix}. \tag{2.5}$$

The Euler equations are solved on every element composing the mesh. In the present work, conventional computations are addressed with 3D unstructured meshes made of tetrahedral elements, as depicted in Figure 1.

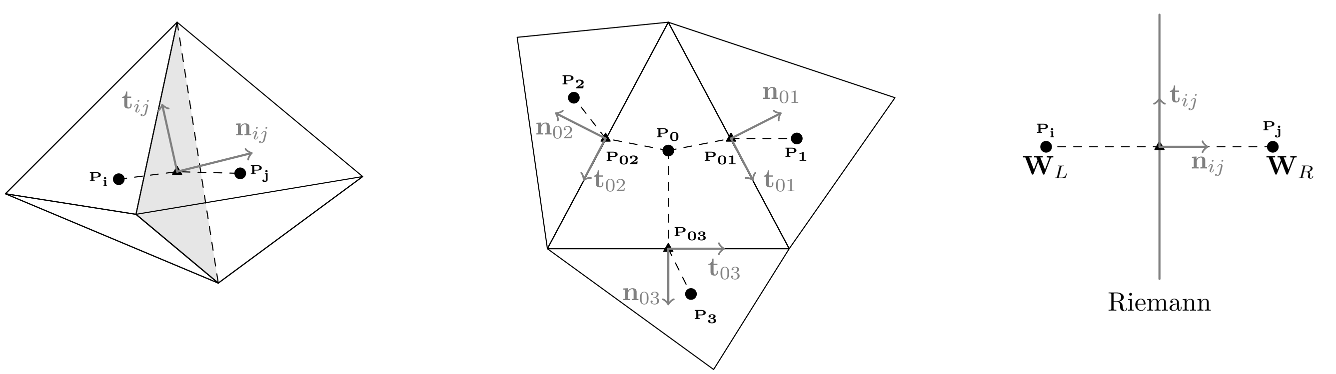

Figure 1: Schematic representation of the Godunov method. The method is cell-centered (finite volumes) and the Riemann problem is solved at each face composing the mesh. On the left, a 3D representation of a tetrahedron element is depicted. In the middle, the representation is reduced to 2D triangular elements, for the sake of clarity. On the right, only the left (L) and right (R) states are presented on both sides of a single face. The Riemann problem is solved at this position with the help of the primitive variables $\mathbf{W}_L$ and $\mathbf{W}_R$. The $\bullet$ symbols represent the centers of the elements. The $\blacktriangle$ symbols represent the centers of the faces. The center of element i is denoted by $\mathbf{P_i}$ and the center of element j is denoted by $\mathbf{P_j}$. The center of face ij separating elements i and j is denoted by $\mathbf{P_{ij}}$. $\mathbf{n}_{ij}$ and $\mathbf{t}_{ij}$ represent respectively the outward normal vector and tangent vector of face ij.

Equation (2.4) is then integrated over time t and the volume Ω_i of numerical element i,

$$\int_t \int_{\Omega_i} \frac{\partial \mathbf{U}}{\partial t}\, dt\, d\Omega_i + \int_t \int_{\Omega_i} \operatorname{div} \mathbf{F}\, dt\, d\Omega_i = 0. \tag{2.6}$$

With the help of the *divergence* theorem, the second volume integral transforms to,

$$\int_{\Omega_i} \operatorname{div} \mathbf{F} \, d\Omega_i = \int_{S_{ij}} \mathbf{F}_{ij} \cdot \mathbf{n}_{ij} \, dS_{ij}, \tag{2.7}$$

where S_{ij} represents the surfaces (faces) separating element i from its neighbors j and $\mathbf{n}_{ij}$ denotes the corresponding outward normal vectors (see Figure 1). Relation (2.6) then becomes,

$$\int_t \int_{\Omega_i} \frac{\partial \mathbf{U}}{\partial t} \, dt \, d\Omega_i + \int_t \int_{S_{ij}} \mathbf{F}_{ij} \cdot \mathbf{n}_{ij} \, dt \, dS_{ij} = 0. \tag{2.8}$$

Assuming constant numerical fluxes during a time step Δt, Eq. (2.8) is approximated as,

$$\left(\mathbf{U}_i^{n+1} - \mathbf{U}_i^n\right) \Omega_i + \Delta t \int_{S_{ij}} \left(\mathbf{F}_{ij}^* \cdot \mathbf{n}_{ij}\right) dS_{ij} = 0, \tag{2.9}$$

where $n+1$ and n denote two consecutive time steps and superscript * denotes the Riemann problem solution. The conservative variables $\mathbf{U}$ are then updated as,

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Omega_i} \int_{S_{ij}} \left(\mathbf{F}_{ij}^* \cdot \mathbf{n}_{ij}\right) dS_{ij} = \mathbf{U}_i^n - \frac{\Delta t}{\Omega_i} \sum_{j=1}^{N_{\text{faces}_i}} \left(\mathbf{F}_{ij}^* \cdot \mathbf{n}_{ij}\right) S_{ij}. \tag{2.10}$$

Relation (2.10) consists of the first-order Godunov scheme. Obviously, higher-order extensions can be considered, see Chiapolino et al. (2017, 2021) [22, 23] for instance in the context of multi-D computations on unstructured meshes. However, such extensions involve unnecessary complexity for the present application. The method is cell-centered (finite volumes) and requires solution of the Riemann problem at each surface (face) ij composing the mesh, in order to provide the numerical fluxes $\mathbf{F}_{ij}^*$. Note that the Godunov scheme is stable under the conventional CFL condition,

$$\Delta t = \text{CFL} \times \min_{ij} \left(\frac{\min\left(\mathbf{r}_{\min,i}, \mathbf{r}_{\min,j}\right)}{S_{\max,ij}^n} \right), \quad \text{with} \quad \mathbf{r}_{\min,i} = \min_k \left(\|\mathbf{P}_i - \mathbf{P}_{ik}\|\right), \quad \text{and} \quad k = \{1, N_{\text{faces}_i}\}, \tag{2.11}$$

and $0 < \text{CFL} < 1$. In Relation (2.11) $S_{\max}^n$ denotes the maximum wave speed throughout the computational domain at time level n. $\mathbf{P}_i$ denotes the center of a cell i and $\mathbf{P}_{ij}$ the center of the face ij separating the elements i and j, see Figure 1.

2.2 HLLC Riemann solver

The HLLC Riemann solver of Toro et al. (1994) [21] is an improvement of the HLL approximation of Harten et al. (1983) [24]. The approximate HLL solver requires estimates for two extreme waves emerging from an initial discontinuity. It results from the integration of the corresponding equations over a two-wave Riemann problem. The HLLC solver is more accurate as the method considers three waves emerging from the initial discontinuity, resulting in better resolution of intermediate waves. The wave structure of the Riemann problem is depicted in Figure 2.

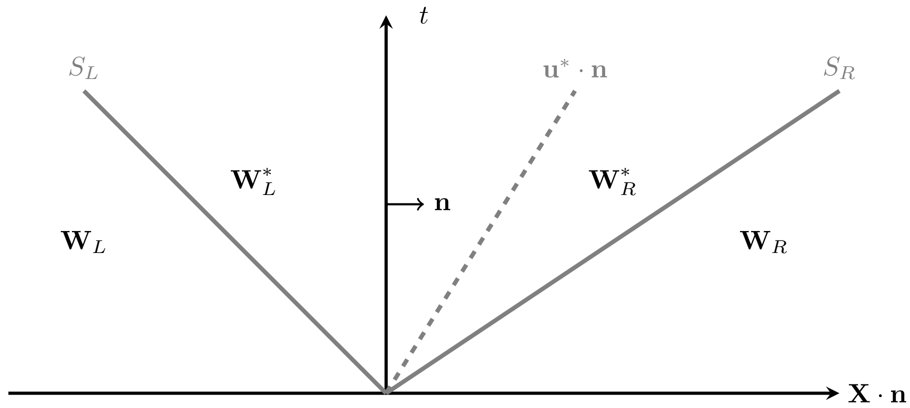

Figure 2: Schematic representation of the wave diagram of the Riemann problem. $\mathbf{u}^* \cdot \mathbf{n}$ represents the contact (in the normal direction) wave velocity (dashed wave) while S_L and S_R represent the left- and right-facing wave speeds (full-line waves). The present waves are a consequence of the eigenvalues of the hyperbolic Euler equations. Across the extreme waves S_L and S_R, the tangential velocity components $v = v^*$ and $w = w^*$ are unchanged, whether those waves are rarefactions or shock waves (Toro, 1997 [25]).

The central idea of the HLL solver is to assume a wave configuration that consists of two waves separating three constant states. The extreme waves denoted S_L and S_R are estimated following Davis (1988) [26],

$$S_L = \min\left(\mathbf{u}_L \cdot \mathbf{n} - c_L, \mathbf{u}_R \cdot \mathbf{n} - c_R\right) \qquad \text{and} \qquad S_R = \max\left(\mathbf{u}_L \cdot \mathbf{n} + c_L, \mathbf{u}_R \cdot \mathbf{n} + c_R\right), \tag{2.12}$$

where indexes L and R denote the left and right states at a given cell boundary. These simple wave speed estimates yield accurate results. Moreover, they are convenient for complicated equations of state and more sophisticated models than the Euler equations. Both HLL and HLLC consider waves as discontinuities. Related jump conditions are the well-known Rankine-Hugoniot conditions:

$$\mathbf{F}_k^* = \mathbf{F}_k + S_k\left(\mathbf{U}_k^* - \mathbf{U}_k\right), \qquad k = L, R, \tag{2.13}$$

where S_k denotes the speed of the considered wave ($k = L, R$). Note that the states involved in Relation (2.13) are spatial integral averages. So strictly speaking these are not the classical Rankine-Hugoniot conditions connecting limiting values left and right of a discontinuity but rather the "Averaged Rankine-Hugoniot" conditions. However the specification "Averaged" will be omitted in the rest of the paper.

In the HLL solver, no distinction is made between states $\mathbf{U}_R^*$ and $\mathbf{U}_L^*$. The solution state in the HLL approximation reads,

$$\mathbf{U}_{\mathrm{HLL}}^* = \frac{\mathbf{F}_R - \mathbf{F}_L + S_L \mathbf{U}_L - S_R \mathbf{U}_R}{S_L - S_R}. \tag{2.14}$$

The resulting HLL Riemann solver forms the basis of very efficient and robust approximate Godunov-type methods. However, as the intermediate wave is omitted, the HLL solver produces more dissipation than the HLLC one. This is not problematic for fast flows as discontinuities are captured and in general maintained sharp enough during sufficiently long time, but becomes problematic for slower flows. Particularly, the HLL solver is unable to maintain contact discontinuities at rest. The HLLC scheme is a modification of the HLL scheme, whereby the missing contact in the Euler equations is restored. The solutions for the two intermediate state vectors $\mathbf{U}_L^*$ and $\mathbf{U}_R^*$ are sought. Similarly to the HLL solver, where the Rankine-Hugoniot relations are used across the two extreme waves, the same jump conditions are used across the intermediate wave yielding contact discontinuity conditions:

$$\begin{cases} p_L^* = p_R^* = p^*, \\ \mathbf{u}_L^* \cdot \mathbf{n} = \mathbf{u}_R^* \cdot \mathbf{n} = \mathbf{u}^* \cdot \mathbf{n} = S_M. \end{cases} \tag{2.15}$$

As the extreme waves S_L and S_R are known from (2.12), algebraic manipulations of the mass and momentum Rankine-Hugoniot relations (2.13) provide the pressure solutions in the left and right perturbed states,

$$p_k^* = p_k + \rho_k\left(S_k - \mathbf{u}_k \cdot \mathbf{n}\right)\left(\mathbf{u}^* \cdot \mathbf{n} - \mathbf{u}_k \cdot \mathbf{n}\right), \qquad k = L, R. \tag{2.16}$$

The equality of the pressures allows determination of the intermediate speed S_M as a function of speeds S_L and S_R, namely,

$$S_M = \mathbf{u}_L^* \cdot \mathbf{n} = \mathbf{u}_R^* \cdot \mathbf{n} = \mathbf{u}^* \cdot \mathbf{n} = \frac{p_R - p_L + \rho_L \mathbf{u}_L \cdot \mathbf{n}\,(S_L - \mathbf{u}_L \cdot \mathbf{n}) - \rho_R \mathbf{u}_R \cdot \mathbf{n}\,(S_R - \mathbf{u}_R \cdot \mathbf{n})}{\rho_L\,(S_L - \mathbf{u}_L \cdot \mathbf{n}) - \rho_R\,(S_R - \mathbf{u}_R \cdot \mathbf{n})}. \tag{2.17}$$

The two intermediate solution states are computed with the help of the Rankine-Hugoniot relations (2.13) and the corresponding values p_L^* and p_R^*. The solutions can be written as,

$$\mathbf{U}_{k,\mathrm{HLLC}}^* = \begin{pmatrix} \frac{\rho_k (S_k - \mathbf{u}_k \cdot \mathbf{n})}{S_k - S_M}, \\ \frac{\rho_k (S_k - \mathbf{u}_k \cdot \mathbf{n})}{S_k - S_M} \mathbf{u}_k^*, \\ \frac{\rho_k (S_k - \mathbf{u}_k \cdot \mathbf{n})}{S_k - S_M} \left(E_k + (S_M - \mathbf{u}_k \cdot \mathbf{n}) \left(S_M + \frac{p_k}{\rho_k (S_k - \mathbf{u}_k \cdot \mathbf{n})} \right) \right) \end{pmatrix}, \tag{2.18}$$

where k denotes either the left (L) or right state (R). In this last expression, the solution velocity vectors $\mathbf{u}_k^*$ read,

$$\mathbf{u}_k^* = (\mathbf{u}_k^* \cdot \mathbf{n})\,\mathbf{n} + (\mathbf{u}_k^* \cdot \mathbf{t})\,\mathbf{t}. \tag{2.19}$$

However, the projected solution speed reads,

$$S_M = \mathbf{u}_k^* \cdot \mathbf{n} = \mathbf{u}^* \cdot \mathbf{n}, \tag{2.20}$$

and the tangential solution speed is unaffected,

$$\mathbf{u}_k^* \cdot \mathbf{t} = \mathbf{u}_k \cdot \mathbf{t}. \tag{2.21}$$

The introduction of (2.20) and (2.21) into (2.19) consequently yields,

$$\mathbf{u}_k^* = S_M \mathbf{n} + (\mathbf{u}_k \cdot \mathbf{t})\,\mathbf{t}. \tag{2.22}$$

Besides, the velocity vector in the left or right state ($k = L, R$) may also be written as,

$$\mathbf{u}_k = (\mathbf{u}_k \cdot \mathbf{n})\,\mathbf{n} + (\mathbf{u}_k \cdot \mathbf{t})\,\mathbf{t}, \tag{2.23}$$

leading to,

$$(\mathbf{u}_k \cdot \mathbf{t})\,\mathbf{t} = \mathbf{u}_k - (\mathbf{u}_k \cdot \mathbf{n})\,\mathbf{n}. \tag{2.24}$$

Thereby, the combination of Eqs. (2.22) and (2.24) results in,

$$\mathbf{u}_k^* = \mathbf{u}_k + (S_M - \mathbf{u}_k \cdot \mathbf{n})\,\mathbf{n}, \tag{2.25}$$

where $\mathbf{u}_k \cdot \mathbf{n}$ is the normal speed of the left ($k = L$) or right ($k = R$) state projected onto the face surface of the Riemann problem. The solution fluxes are finally provided by the HLLC approximation and are computed as follows (Le Martelot et al., 2014 [27]),

$$\mathbf{F}_{\mathrm{HLLC}}^* = \frac{\mathbf{F}_L + \mathbf{F}_R - \mathrm{sign}\,(S_L)\, S_L\,(\mathbf{U}_L^* - \mathbf{U}_L) - \mathrm{sign}\,(S_M)\, S_M\,(\mathbf{U}_R^* - \mathbf{U}_L^*) - \mathrm{sign}\,(S_R)\, S_R\,(\mathbf{U}_R - \mathbf{U}_R^*)}{2}. \tag{2.26}$$

In this last relation, the left and right flux vector, $\mathbf{F}_L$ and $\mathbf{F}_R$, read:

$$\mathbf{F}_k = \begin{pmatrix} \rho_k \mathbf{u}_k \cdot \mathbf{n} \\ \rho_k\,(\mathbf{u}_k \cdot \mathbf{n})\,\mathbf{u}_k + p_k \mathbf{n} \\ (\rho_k E_k + p_k)\,\mathbf{u}_k \cdot \mathbf{n} \end{pmatrix}. \tag{2.27}$$

where k denotes either the left (L) or right state (R). The total energy is determined with the help of an equation of state as $E_k = e_k\,(\rho_k, p_k) + \frac{1}{2}\mathbf{u}_k \cdot \mathbf{u}_k$.

The previous Godunov scheme (2.10) and HLLC Riemann solver (2.26) are used to update the solution at the center of every element composing the mesh. The method is referred to as the conventional method in the rest of the paper as it is used with meshes involving the various openings of a building. Because those geometric details are drawn and meshed during the pre-processing step, the corresponding surfaces fit the shapes of the openings and the Riemann solver accounts naturally for geometrical effects (with the exception of the boundary condition, see Appendix B). The purpose of the present paper is to determine the mean pressure in the multiple rooms of a building, as quickly as possible. The conventional method can certainly be used with a very coarse mesh, reducing drastically computation time. Results provided in Section 6 show that the mean pressures computed with the help of a coarse mesh are in a quite good agreement with those computed via a fine mesh. Nevertheless, regardless of the mesh quality, the doors, windows and other openings must be drawn and meshed during the pre-processing step. Such a meshing process may be tedious and very time consuming, especially when dealing with real and complex structures. When dealing with pressing situations, a long and tedious pre-processing is not acceptable. This drawback of the conventional method motivates the proposed MUZO method.

3. Riemann problem with discontinuous area change

The MUZO method differs from conventional computations as the geometric restrictions are not drawn nor meshed during the geometry-definition and mesh-generation pre-processing stage. As mentioned earlier, the present effort attempts to create a simple, accurate and fast method to address pressing situations. The surfaces of elements involving an opening are then only marked during the mesh-generation step, greatly reducing the time needed to construct the geometry and its corresponding mesh. The geometrical effects then need to be specifically considered in the solution states and through the flux distribution of the Riemann solver, as the pressure distribution is a direct consequence of the balance of the fluxes. The unmarked faces (or surfaces), where no geometric restriction is present, are treated with the conventional HLLC solver presented in Section 2.2.

The present section addresses the MUZO Riemann solver, needed to compute the fluid flux at the marked surfaces of the mesh. For the sake of simplicity let us consider a given cell boundary separating two volumes, as shown in Figure 3.

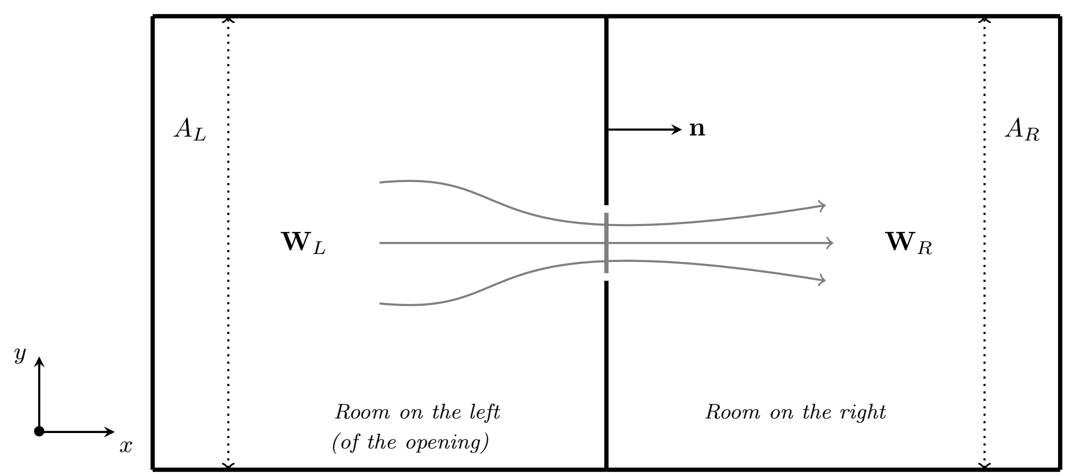

Figure 3: Gas flowing from the room on the left of the opening to the room on the right. These two rooms represent two computational cells. In the present example, the two rooms have the same cross-section $A_L = A_R$. The opening is here only present for the purpose of illustration. This geometric detail is not meshed. The separating face is only marked with a flag recognized by the fluid flow code (see Section 4). A specific Riemann solver is used on such marked faces and addresses the geometric restriction directly in the state solutions and through the flux distribution as well.

The opening seen in Figure 3 is neither meshed nor drawn. It only appears for illustration purposes. The rooms on the left and on the right of the geometric discontinuity represent two computational cells. The separating face is only marked with a flag recognized by the fluid flow code. Consequently, multidimensional effects occurring through the marked face (opening) are not resolved spatially. The main difficulty of the approach dwells at this level, where the dimensional reduction is considered through appropriate quasi-steady relations and assumptions.

Indeed as will be seen in Section 3.1 the flow is assumed quasi-steady at the opening cross-section, resulting in simplified relations. This assumption is in general not valid, especially for highly unsteady regimes like it is the case when a shock wave travels towards the opening. Nevertheless, as mentioned in the Introduction, wave propagation occurs at earlier times and rapidly decouples from flow discharge effects occurring at longer timescales. The present contribution considers that the effects of wave propagation are already taken into account through an appropriate method, based for example on Kingery-Bulmash data [5], and focuses on flow discharge effects only. The quasi-steady assumption becomes consequently appropriate and reasonable in the present context as supported by the numerical results (Section 6).

In the rest of the paper, the dimensional reduction will be denoted as the “throat”, in reference to flows occurring in nozzles. For the sake of clarity the following calculations and analyzes are presented in 1D. As the marked surface involves a throat, the Euler equations for ducts of variable cross-sections are considered to compute the fluid flux through such a specific surface,

$$\begin{cases} \dfrac{\partial A}{\partial t} = 0, \\ \dfrac{\partial (\rho A)}{\partial t} + \dfrac{\partial (\rho u A)}{\partial x} = 0, \\ \dfrac{\partial (\rho u A)}{\partial t} + \dfrac{\partial \left(\left(\rho u^2 + p \right) A \right)}{\partial x} = p \dfrac{\partial A}{\partial x}, \\ \dfrac{\partial (\rho E A)}{\partial t} + \dfrac{\partial \left(\left(\rho E + p \right) u A \right)}{\partial x} = 0. \end{cases} \tag{3.1}$$

The notations remain the same as in Equations (2.1). System (3.1) is also hyperbolic with wave speeds $\lambda_1 = u - c$, $\lambda_2 = u$, $\lambda_3 = u + c$. However, in comparison with the x-split 1D equations of (2.1), an additional stationary wave speed $\lambda_4 = 0$ appears and is a consequence of the cross-section A. System (3.1) consequently involves the Riemann problem schematized in Figure 4 (in the subsonic case), and representing the flow evolving between the two rooms.

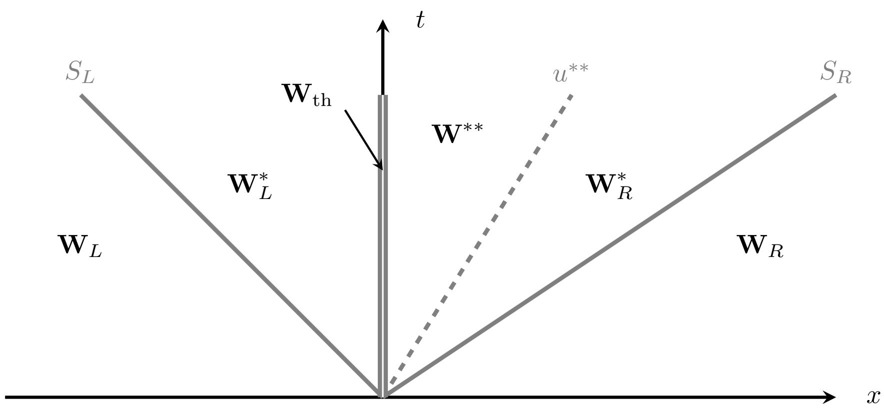

Figure 4: Schematic representation of the wave diagram of the 1D subsonic Riemann problem between the two rooms separated by a geometric discontinuity (double line). This geometric reduction is assimilated to a throat. Solution of the Riemann problem $\mathbf{W}_{\text{th}}$ is required at this location. u^{**} represents the contact wave velocity (dashed wave) while S_L and S_R represent the left- and right-facing wave speeds (full-line waves). The present waves are a consequence of the eigenvalues of the hyperbolic 1D Euler equations for ducts of variable cross-sections.

In Figure 4, $\mathbf{W}$ represents the vector of primitive variables, $\mathbf{W} = (\rho, u, p, A)^T$. The cross-section A varies only between the states $\mathbf{W}_L^*$ and $\mathbf{W}^{**}$ of this figure where four waves are depicted, namely the two left- and right-facing acoustic waves S_L and S_R, the contact wave u^{**} and a stationary wave resulting from the geometric discontinuity. The present waves are a consequence of the eigenvalues of the hyperbolic 1D Euler equations (3.1) for ducts of variable cross-sections. The flow model (3.1) admits the following additional entropy (denoted s) equation:

$$\frac{\partial (\rho s A)}{\partial t} + \frac{\partial (\rho s u A)}{\partial x} = 0. \tag{3.2}$$

This equation is particularly important in the present context. The analysis is restricted to an unchoked subsonic flow with $u^{**} > 0$ which corresponds precisely to the one depicted in Figure 4. Indeed, we will see that a significant simplification appears when the two rooms have the same cross-section $A_L = A_R$.

3.1 Resolution of the 4-wave Riemann problem

In the present section, the flow at the throat is unchoked. In such a situation, the flow does not change regime. That is to say if the flow upstream is subsonic, it remains subsonic at the throat. To simplify the calculations two main assumptions are made:

– The left- and right-facing waves are considered under acoustic approximations. It means that the Rankine-Hugoniot relations as well as the Riemann invariants are replaced by characteristic equations with a constant acoustic impedance.

– The isentropes, or more precisely Laplace's law, is approximated by the sound speed definition. It corresponds to a linear approximation of the isentropes.

These two approximations are conventional and given for example in Toro (1997) [25] in the frame of the approximate acoustic Riemann solver. They are valid for waves of weak amplitude, which is appropriate to the present context of subsonic evolutions. The set of relations to consider is summarized hereafter. For a left-facing wave, the approximations are:

$$p_L + Z_L u_L = p_L^* + Z_L u_L^*, \tag{3.3}$$

$$\rho_L^* = \rho_L + \frac{p_L^* - p_L}{c_L^2}. \tag{3.4}$$

For a right-facing wave, the approximations are:

$$p_R - Z_R u_R = p_R^* - Z_R u_R^*, \tag{3.5}$$

$$\rho_R^* = \rho_R + \frac{p_R^* - p_R}{c_R^2}. \tag{3.6}$$

In these relations $Z = \rho c$ represents the acoustic impedance, where $c^2 = \gamma \frac{p}{\rho}$ represents the sound speed, according to the ideal-gas equation of state. The contact wave is governed by the interface conditions:

$$u_R^* = u^{**}, \tag{3.7}$$

$$p_R^* = p^{**}. \tag{3.8}$$

The flow is assumed subsonic everywhere and in particular between the states $\mathbf{W}_L^*$ and $\mathbf{W}^{**}$. Moreover, the flow is assumed stationary between these two states. Under this latter condition, the integration of the equations of System (3.1) and Equation (3.2) between the states $\mathbf{W}_L^*$ and $\mathbf{W}^{**}$ results in the conservation of mass flow rate, the conservation of specific total enthalpy $\left(H = e + \frac{p}{\rho} + \frac{1}{2}u^2\right)$ and the conservation of specific entropy. The first two relations read:

$$\rho_L^* u_L^* A_L = \rho^{**} u^{**} A_R, \tag{3.9}$$

$$\frac{\gamma p_L^*}{(\gamma - 1)\,\rho_L^*} + \frac{1}{2} u_L^{*2} = \frac{\gamma p^{**}}{(\gamma - 1)\,\rho^{**}} + \frac{1}{2} u^{**2}, \tag{3.10}$$

where the ideal-gas equation of state (2.2) has been introduced. As mentioned earlier, the conservation of specific entropy is here approximated by the sound speed definition, corresponding to a linear approximation of the isentropes:

$$\rho^{**} = \rho_L^* + \frac{p^{**} - p_L^*}{c_L^{*2}}. \tag{3.11}$$

Such an approximation has been used in various contexts to address isentropes in the frame of Riemann solvers, see for instance [28], [29], [30], [31].

Each state, $\mathbf{W}_L^*$, $\mathbf{W}^{**}$ and $\mathbf{W}_R^*$ contains 3 unknowns, corresponding to a total number of 9 unknowns. The algebraic system (3.3) - (3.11) involves 9 equations. Consequently, the system is closed. Its resolution is done as follows:

- Arbitrary guess of p^{**} is set, implying $p_R^* = p^{**}$.
- With the help of (3.5) and (3.6) the following variables are deduced:

$$u_R^* = u^{**} = u_R + \frac{p^{**} - p_R}{Z_R}, \tag{3.12}$$

$$\rho_R^* = \rho_R + \frac{p^{**} - p_R}{c_R^2}. \tag{3.13}$$

At this level, the full state $\mathbf{W}_R^*$ is determined.

- In the state $\mathbf{W}^{**}$, p^{**} has been set and (3.12) provides u^{**}. It remains to determine ρ^{**}. Combining (3.11) and (3.4) yields:

$$\rho^{**} = \rho_L + \frac{p_L^* - p_L}{c_L^2} + \frac{p^{**} - p_L^*}{c_L^{*2}}. \tag{3.14}$$

  An extra simplification can be made, assuming $c_L^{*2} = c_L^2$. In the present subsonic conditions, it appears reasonable. The previous relation thus transforms to,

$$\rho^{**} = \rho_L + \frac{p^{**} - p_L}{c_L^2}. \tag{3.15}$$

  At this level, the full state $\mathbf{W}^{**}$ is determined.

- System (3.9) - (3.11) is considered for the determination of $\mathbf{W}_L^*$. Combining these relations, the following one is obtained,

$$\left(\gamma c_L^2 - \frac{\gamma p^{**}}{\rho^{**}} - \frac{1}{2}(\gamma - 1) u^{**2}\right) \rho_L^{*2} + \gamma \left(p^{**} - c_L^2 \rho^{**}\right) \rho_L^* + \frac{(\gamma - 1)}{2} \left(\frac{A_R}{A_L}\right)^2 \rho^{**2} u^{**2} = 0. \tag{3.16}$$

  The positive root of this quadratic equation is retained, giving ρ_L^*. With the help of (3.4) the pressure p_L^* is determined:

$$p_L^* = p^{**} - c_L^2 \left(\rho^{**} - \rho_L^*\right). \tag{3.17}$$

  Then, using (3.9) the velocity is determined,

$$u_L^* = \frac{\rho^{**} u^{**} A_R}{\rho_L^* A_L}. \tag{3.18}$$

  The state $\mathbf{W}_L^*$ is now fully determined.

However, the solution state $\mathbf{W}_L^*$ is not necessarily compatible with Relation (3.3). If the function,

$$f\left(p^{**}\right) = p_L + Z_L u_L - \left(p_L^* + Z_L u_L^*\right), \tag{3.19}$$

has not reached a certain tolerance, the initial pressure p^{**} must be changed until this condition is reached. The Newton-Raphson iterative procedure can be used in this respect. It is certainly possible to optimize and generalize this algorithm both for $u^{**} > 0$ and for $u^{**} < 0$. However, it does not seem important in the present context, because of the limiting case situation $\frac{A_R}{A_L} = 1$ that follows.

3.2 Limiting case $\frac{A_R}{A_L} = 1$ for subsonic flow

This limiting case corresponds for instance to the situation depicted in Figure 3 where the two rooms have the same cross-section. In this particular case, Relation (3.16) becomes:

$$\left(\gamma c_L^2 - \frac{\gamma p^{**}}{\rho^{**}} - \frac{1}{2}(\gamma - 1) u^{**2}\right) \rho_L^{*2} + \gamma \left(p^{**} - c_L^2 \rho^{**}\right) \rho_L^* + \frac{(\gamma - 1)}{2} \rho^{**2} u^{**2} = 0, \tag{3.20}$$

i.e.,

$$\gamma \rho_L^* \left(c_L^2 - \frac{p^{**}}{\rho^{**}}\right) \left(\rho_L^* - \rho^{**}\right) + \frac{(\gamma - 1)}{2} u^{**2} \left(\rho^{**2} - \rho_L^{*2}\right) = 0. \tag{3.21}$$

After simplifications it implies,

$$\rho_L^* = \rho^{**}. \tag{3.22}$$

Then, considering mass conservation (3.9) $m = \rho^{**} u^{**} A_R$, it results:

$$u_L^* = u^{**}. \tag{3.23}$$

It then follows that,

$$p_L^* = p^{**}, \tag{3.24}$$

from total enthalpy invariance (3.10).

This result is not surprising, but it is very important. As the flow is isentropic and preserves the mass and energy fluxes, the varying cross-section has no effect, at least in the computation of the states of the Riemann problem. This is true in the present case of a subsonic flow everywhere, even at the throat. A relevant simplification appears. As the geometric discontinuity is transparent ($\mathbf{W}_L^* = \mathbf{W}^{**}$) there is no need to consider 4 waves and 3 states in the Riemann problem. In this particular case it reduces to 3 waves and 2 states, as done usually with the Euler equations without cross-section variation. In this frame, the acoustic solver is no longer used and is replaced by the more general and more robust HLLC solver of Toro et al. (1994) [21] with Davis' wave speed estimates (5.2) for S_L and S_R (Figure 4). The previous simplification ($\mathbf{W}_L^* = \mathbf{W}^{**}$) is valid whatever the sign of u^{**} is, provided that the flow is not choked at the throat. Consequently, only situations involving $\frac{A_L}{A_R} = 1$ on a marked cell face presenting a geometric restriction (door, window, etc.) will be considered in the frame of the present method. A particular but easy meshing process is presented in the following, such that this simplification can be made during the resolution of the MUZO Riemann problem.

4. Mesh definition and generation

4.1 Introduction

The present section presents the MUZO meshing method. It consists of a simple and fast method for generating a computational geometry and its corresponding mesh. As a fast method is desired, both on the pre-processing stage and on the solver side, coarse meshes are addressed and the size of a computational cell is typically of the order of the size of a room in a building. For the sake of simplicity, geometric details, like doors or windows, are neither drawn nor meshed. As will be seen in the following, only the "footprints" of the geometry are needed. A simple linear extrusion of the 2D planar mesh then provides a conforming 3D mesh. During the present mesh-generation step, surfaces involving doors and windows are only marked with a flag recognized in a later stage by the fluid flow solver. The geometric restrictions are then considered in a specific Riemann solver, through both its solution states and its flux distribution as well. Section 5 is devoted to this specific Riemann solver.

As seen in Section 3, an important simplification arises from the analysis of the Riemann problem. Such a simplification appears when the cross-sections on both sides of a face separating two numerical elements are the same: $A_L = A_R$. In this particular situation the geometric restriction, such as a door, becomes transparent in the Riemann problem, at least in the computation of the solution states when an unchoked flow is addressed. The present section consequently addresses the construction of a conformal mesh satisfying $A_L = A_R$ for the marked faces involving a geometric restriction. The other surfaces do not need any particular attention. In such cases, the fluxes are provided by a conventional Riemann solver. The HLLC solver of Toro et al. (1994) [21] is used in the present paper (see Section 2.2). Figure 5 depicts two possible strategies to construct a mesh from given input nodes.

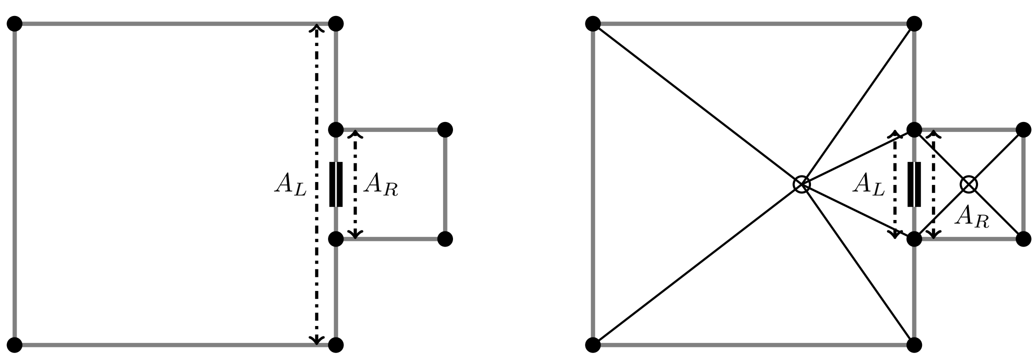

Figure 5: Schematic representation of two strategies to construct a planar mesh for two rooms. On the left side a non-conformal mesh composed of two quadrilaterals is used implying $A_L \neq A_R$. On the right side a conformal constrained Delaunay mesh is used yielding $A_L = A_R$. In these figures, the black filled points • represent input nodes. The thick hatched lines represent a door between the two rooms. The black empty circles ∘ are additional mesh nodes in the case of a conformal constrained Delaunay mesh.

The first strategy represented on the left side of Figure 5 consists of considering each room as a single discrete element, two quadrilaterals in this simple example. This approach is appealing in the aim of fast computations, due to the minimal element count, but has two strong limitations:

– As the mesh is not conformal, left and right cross-sections are different: $\frac{A_L}{A_R} \neq 1$. As a consequence, the above-mentioned simplification of the Riemann solver cannot be used. As seen in Section 3, this simplification is important and yields a Riemann problem involving only 3 waves. In the general case where $\frac{A_L}{A_R} \neq 1$, the Riemann problem involves 4 waves leading consequently to a more complex Riemann solver, affecting potentially both robustness and computational efficiency of the numerical integration.

– Extension to complex 3D configurations is not straightforward, and requires a non-conformal mixed-type element mesh whenever the "footprints" of the rooms cannot be accurately described by a single quadrilateral element.

The strategy adopted in this work is depicted on the right side of Figure 5. A conformal constrained Delaunay-type mesh is built from given input nodes which ensures $\frac{A_L}{A_R} = 1$ by construction at all room partitions. As detailed in Section 3, this condition drastically simplifies the computation of fluxes across doors or windows due to the fact that the geometric restriction becomes transparent in the Riemann problem, when an unchoked flow is addressed. It may appear at first glance that the simplicity gained on the solver side by considering the limiting case $\frac{A_L}{A_R} = 1$ yields an intricate pre-processing stage with the need of an unstructured meshing strategy. Indeed, the aim is to build a fast numerical framework, both on the pre-processing stage and on the solver side. However, as detailed hereafter, some simplifications can be made on the mesh construction and the flexibility to handle complex 3D configurations obtained by following this strategy is huge.

4.2 MUZO 3D mesh construction

The objective is to construct, with as little effort and time as possible, a conforming mesh with few elements. The meshing tool used in this work is GMSH (Geuzaine and Remacle, 2009 [32]). The strategy adopted here is to construct only the footprints of the geometry (nodes, lines and surfaces), generate a 2D mesh and then extrude along the third dimension. In this way, each room is discretized with prismatic elements as can be seen in Figure 6.

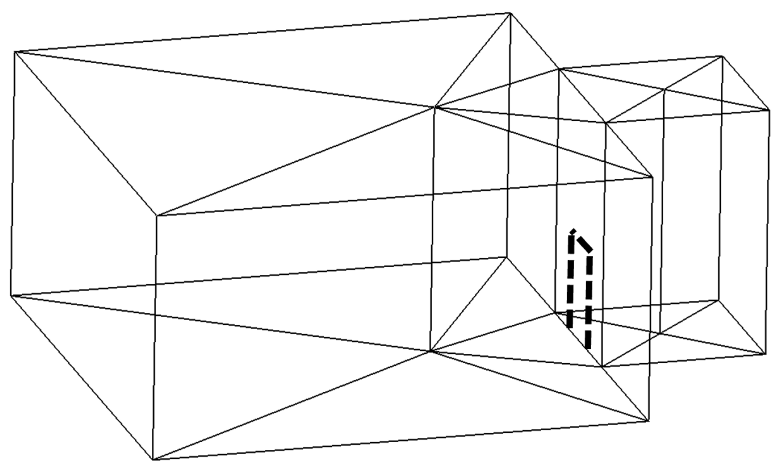

Figure 6: Two-room prismatic mesh. The fluxes across the separating face (where a door is supposed to be present and represented in thick dashed lines) are computed with the algorithm detailed in Section 5 and the fluxes associated with the internal faces are computed with the HLLC Riemann solver (see Section 2.2).

The separating face is then marked with a flag recognized in a later stage by the fluid flow solver, in the same way boundary conditions are handled in unstructured finite volume codes. The fluid flow software used in this work is the multiphysics DalphaDt code. DalphaDt is a multi-purpose code handling unstructured meshes composed of arbitrary elements using a cell-centered finite volume method. One consequence of a 2D mesh extrusion is the need of a special treatment to model a geometric restriction between two floors. Indeed, doors and windows belong to a unique quadrilateral face describing a wall. The situation is different with floors for which a constrained Delaunay algorithm usually generates several triangles to mesh them. This configuration is depicted in Figure 7. We recall that a dimensional reduction will be denoted as the "throat", in reference to flows occurring in nozzles.

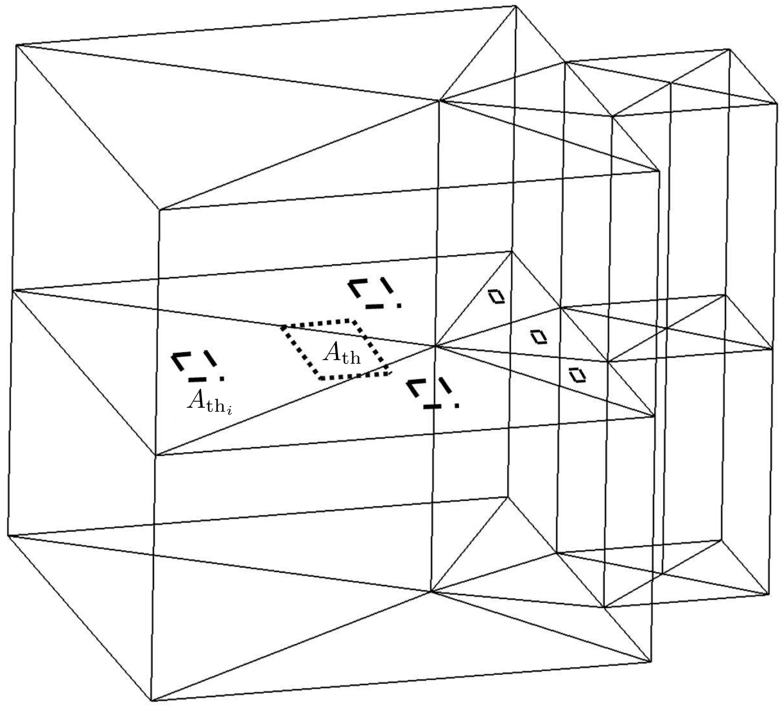

Figure 7: Schematic representation of a separation between two floors. The initial throat area A_{th} (represented with dots), which models for instance a stair, is distributed among all the triangular faces pertaining to the floor. Each face then considers a portion A_{th_i} of the throat area A_{th}.

Consider a stair between two floors modeled as a throat of area A_{th}. The ground floor is discretized in N triangles of area S_i, $i \in [1, N]$. The throat area A_{th} is distributed among all the triangular faces pertaining to the ground floor by maintaining a constant aspect ratio:

$$A_{\mathrm{th}_i} = A_{\mathrm{th}} \frac{S_i}{\sum_{i=1}^{N} S_i}. \tag{4.1}$$

In the same spirit, if several doors or windows belong to the same wall, the throat areas are merged into a single throat area. These approximations allow a fast geometry and mesh construction without sacrificing geometric details as can be seen for illustration purposes in Figure 8.

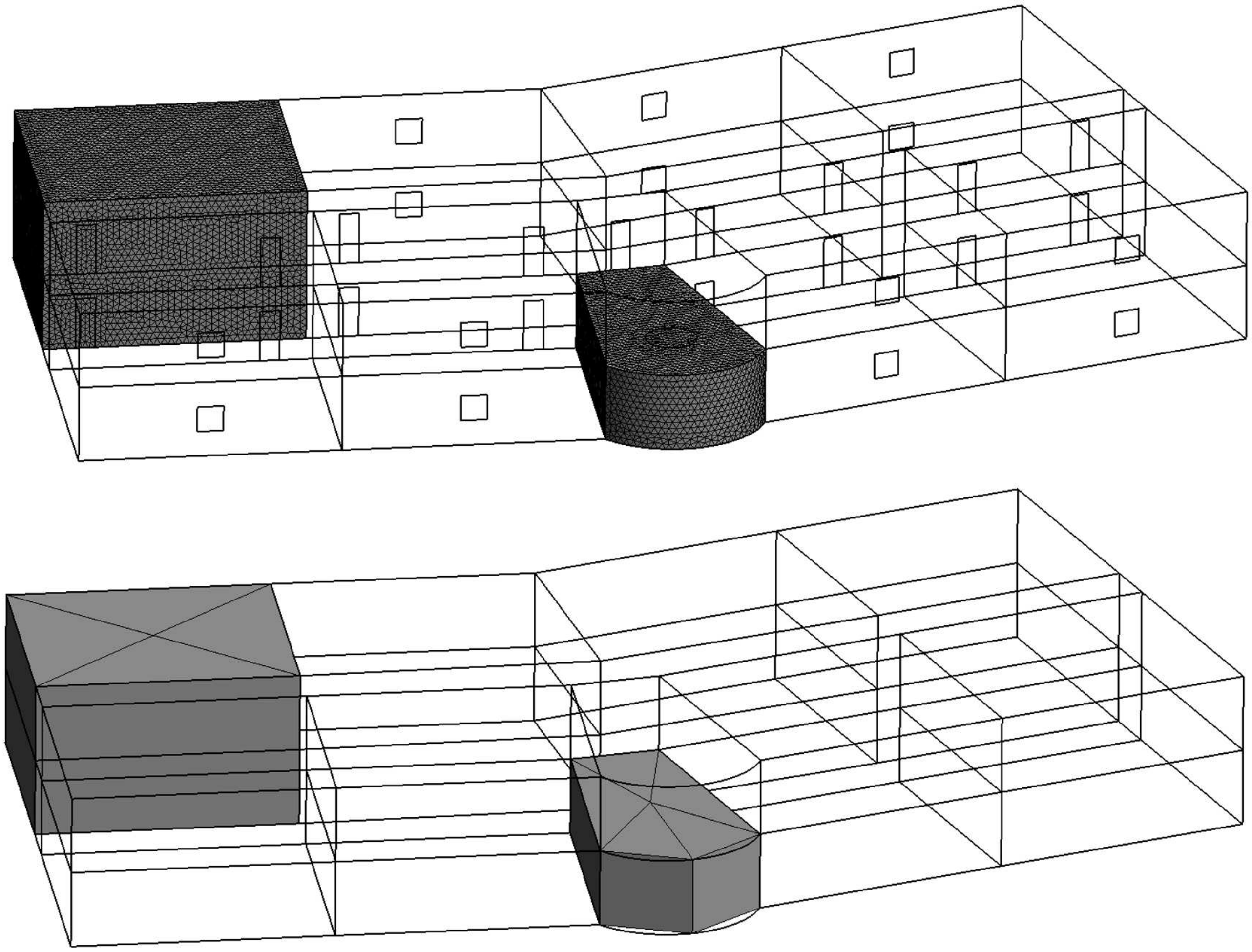

Figure 8: Moderately complex building with various rooms, a ground floor and an upper floor. On top, a partial view of the conventional 3D tetrahedral mesh where doors, windows and stairs are drawn. On bottom, the simplified prismatic MUZO mesh where the faces with doors/stairs are treated with the MUZO Riemann solver developed in Section 5.

The proposed method ensures $\frac{A_L}{A_R} = 1$ by construction at all room partitions. This property will be used in the following section where a specific Riemann solver is developed to address the geometric details (doors, windows, etc.) that have been omitted in the present geometry-definition and mesh-generation pre-processing step. It is worth mentioning that, as very coarse meshes are used, a loss of accuracy appears where curved geometries are considered. This is for example illustrated in Figure 8, where the large numerical cells do not fit the curved portion of the building. One way to remedy to this drawback is to use high-order meshes, see for instance Dobrzynski and Jannoun (2017) [33]. Such an extension will be examined in the future.

5. MUZO Riemann solver and related flux computation

The MUZO Riemann solver addresses the limiting case $\frac{A_L}{A_R} = 1$. This property is easily satisfied from the MUZO meshing process, as seen in Section 4. In this particular case, an important simplification appears. The geometric discontinuity becomes transparent ($\mathbf{W}_L^* = \mathbf{W}^{**}$) and the Riemann problem involves only 3 waves instead of 4 in the general case, as detailed in Section 3.2. In the present section, unchoked and choked conditions at the throat are analyzed separately and two specific procedures are developed. A method allowing to determine the appropriate flow regime, in accordance with the flow conditions, is afterwards presented. The section ends by summarizing the global method determining the solution state of the Riemann problem, and by introducing the specific flux computation. The MUZO Riemann solver is presented in the general multi-D case. The analysis begins with the determination of the flow direction which is a necessary step in order to select the right wave pattern.

5.1 Determination of the flow direction

The flow direction is approximated through the solution of the Riemann problem for the Euler equations at the throat, neglecting the effect of the cross-section variation. Indeed, it is assumed that the presence of the throat does not affect the sign of the velocity. The Riemann problem is solved under the HLL approximation (Harten et al., 1983 [24]):

$$\mathbf{U}^*_{\text{HLL}} = \frac{\mathbf{F}_R - \mathbf{F}_L + S_L \mathbf{U}_L - S_R \mathbf{U}_R}{S_L - S_R}, \tag{5.1}$$

with the notations introduced in Section 2.1 and 2.2. The two extreme wave speeds S_L and S_R are provided with the help of Davis' (1988) [26] estimates (5.2),

$$S_L = \min\left(\mathbf{u}_L \cdot \mathbf{n} - c_L, \mathbf{u}_R \cdot \mathbf{n} - c_R\right) \quad \text{and} \quad S_R = \max\left(\mathbf{u}_L \cdot \mathbf{n} + c_L, \mathbf{u}_R \cdot \mathbf{n} + c_R\right). \tag{5.2}$$

The flow direction is obtained from the momentum components of vector $\mathbf{U}^*_{\text{HLL}}$ as $\mathbf{U}^{*\text{mom}}_{\text{HLL}} \cdot \mathbf{n} = \rho \mathbf{u}^* \cdot \mathbf{n} = \rho S_M$. This flow direction is important for the determination of the critical state, associated with the sonic condition at the throat as will be seen later. First let us consider unchoked flow conditions at the throat.

5.2 Subsonic flow

The present section follows Section 3 where the Riemann problem is presented in a subsonic situation. The subsonic wave diagram is depicted in Figure 4 where the geometric discontinuity is now transparent: $\mathbf{W}^*_L = \mathbf{W}^{**}$. Because the throat is transparent, the HLLC solver of Toro et al. (1994) [21] is used (see Section 2.2) and provides the state $\mathbf{W}^*_L$ of Figure 4. From the state $\mathbf{W}^*_L$, the state $\mathbf{W}_{\text{th}}$ at the throat is then computed. Its determination is based on the same algebraic system as (3.9) - (3.11), except that the conservation of specific entropy is now expressed through Laplace's law,

$$\frac{p^{**}}{\rho^{**\gamma}} = \frac{p^*_L}{\rho_L^{*\gamma}}, \tag{5.2.3}$$

to increase the accuracy of the solution. Knowledge of the state $\mathbf{W}_{\text{th}}$ at the throat is important for two reasons:

- To check the validity of the solution. Indeed, the solution from the HLLC solver is valid only when the opening between the two rooms is transparent. This is correct only if the state at the throat is unchoked. It is then necessary to check whether this state is choked or not. This point is addressed in Section 5.5.
- To determine the effective fluxes that cross the cell boundary through the opening. These fluxes are significantly different from those associated with the state $\mathbf{W}^*_L$. Determination of the solution state $\mathbf{W}_{\text{th}}$ at the throat is addressed in the following. In the present section, the flow is assumed subsonic everywhere, even at the throat.

The method is presented according to the flow situation depicted in Figure 4. The solution speed $\mathbf{U}^{*\text{mom}}_{\text{HLL}} \cdot \mathbf{n}$ is determined by the HLL solution (5.1) and the HLLC solver provides the solution state $\mathbf{W}^*_L$. Naturally the method presented hereafter treats the opposite situation ($S_M < 0$) similarly. In that case the HLLC solver provides the solution state $\mathbf{W}^*_R$. Between a given state $\mathbf{W}^*_L = (\rho^*_L, \mathbf{u}^*_L, p^*_L, A^*_L = A_L)^T$ and the state $\mathbf{W}_{\text{th}}$ at the throat, the corresponding equations read:

$$\rho^*_L A^*_L \mathbf{u}^*_L \cdot \mathbf{n} = \rho_{\text{th}} A_{\text{th}} \mathbf{u}_{\text{th}} \cdot \mathbf{n}, \tag{5.2.4}$$

$$\frac{p^*_L}{\rho_L^{*\gamma}} = \frac{p_{\text{th}}}{\rho_{\text{th}}^{\gamma}}, \tag{5.2.5}$$

$$\frac{\gamma p^*_L}{(\gamma - 1)\rho^*_L} + \frac{1}{2}\left(\mathbf{u}_L \cdot \mathbf{n}\right)^2 = \frac{\gamma p_{\text{th}}}{(\gamma - 1)\rho_{\text{th}}} + \frac{1}{2}\left(\mathbf{u}_{\text{th}} \cdot \mathbf{n}\right)^2. \tag{5.2.6}$$

With the help of the ideal-gas sound speed definition $c^2 = \frac{\gamma p}{\rho}$, the specific total enthalpy relation (5.2.6) yields, after some algebraic manipulations:

$$1 + \frac{\gamma - 1}{2} M_L^{*2} = \frac{c^2_{\text{th}}}{c_L^{*2}} \left(1 + \frac{\gamma - 1}{2} M^2_{\text{th}}\right), \tag{5.2.7}$$

where $M = (\mathbf{u} \cdot \mathbf{n})/c$ is the Mach number. Similar manipulations on Laplace's law (5.2.5) lead to,

$$\frac{\rho_{\text{th}}}{\rho_L^*} = \left(\frac{c_{\text{th}}}{c_L^*}\right)^{\frac{2}{\gamma-1}}. \tag{5.2.8}$$

Mass conservation (5.2.4) is now written under the following form,

$$\frac{\rho_{\text{th}}}{\rho_L^*} = \frac{A_L^* M_L^* c_L^*}{A_{\text{th}} M_{\text{th}} c_{\text{th}}}. \tag{5.2.9}$$

The equality of Relations (5.2.8) and (5.2.9) yields,

$$\left(\frac{c_{\text{th}}}{c_L^*}\right)^2 = \left(\frac{A_L^* M_L^*}{A_{\text{th}} M_{\text{th}}}\right)^{\frac{2(\gamma-1)}{\gamma+1}}. \tag{5.2.10}$$

This last relation is now inserted into Relation (5.2.7), leading to:

$$\frac{A_L^*}{A_{\text{th}}} = \frac{M_{\text{th}}}{M_L^*}\left(\frac{1+\frac{(\gamma-1)}{2}M_L^{*2}}{1+\frac{(\gamma-1)}{2}M_{\text{th}}^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}}. \tag{5.2.11}$$

The Mach number M_{th} at the throat is determined from Relation (5.2.11) using a root-finding method. As the flow is assumed subsonic, the M_{th} solution is necessarily bounded between $[M_L^*, 1[$ or alternatively $]-1, M_R^*]$ if the flow is reversed. The bisection method is then preferred over a Newton-Raphson method. The sound speed at the throat c_{th} is then deduced from Relation (5.2.10). The corresponding projected speed $\mathbf{u}_{\text{th}} \cdot \mathbf{n}$ is readily obtained from the Mach number definition $M_{\text{th}} = (\mathbf{u}_{\text{th}} \cdot \mathbf{n})/c_{\text{th}}$. The density ρ_{th} is afterwards determined from Relation (5.2.8), and the pressure p_{th} is computed from Laplace's law (5.2.5). The velocity vector at the throat $\mathbf{u}_{\text{th}}$ is now addressed. This latter reads,

$$\mathbf{u}_{\text{th}} = (\mathbf{u}_{\text{th}} \cdot \mathbf{n})\,\mathbf{n} + (\mathbf{u}_{\text{th}} \cdot \mathbf{t})\,\mathbf{t}. \tag{5.2.12}$$

However, the tangential solution speed is unaffected,

$$\mathbf{u}_{\text{th}} \cdot \mathbf{t} = \mathbf{u}_L^* \cdot \mathbf{t}. \tag{5.2.13}$$

Relation (5.2.12) then transforms to,

$$\mathbf{u}_{\text{th}} = (\mathbf{u}_{\text{th}} \cdot \mathbf{n})\,\mathbf{n} + (\mathbf{u}_L^* \cdot \mathbf{t})\,\mathbf{t}. \tag{5.2.14}$$

Besides the velocity vector in the solution state $\mathbf{W}_L^*$ may also be written as,

$$\mathbf{u}_L^* = (\mathbf{u}_L^* \cdot \mathbf{n})\,\mathbf{n} + (\mathbf{u}_L^* \cdot \mathbf{t})\,\mathbf{t}, \tag{5.2.15}$$

leading to,

$$(\mathbf{u}_L^* \cdot \mathbf{t})\,\mathbf{t} = \mathbf{u}_L^* - (\mathbf{u}_L^* \cdot \mathbf{n})\,\mathbf{n}. \tag{5.2.16}$$

The introduction of (5.2.16) into (5.2.14) consequently yields,

$$\mathbf{u}_{\text{th}} = \mathbf{u}_L^* + (\mathbf{u}_{\text{th}} \cdot \mathbf{n} - \mathbf{u}_L^* \cdot \mathbf{n})\,\mathbf{n}. \tag{5.2.17}$$

The whole state $\mathbf{W}_{\text{th}}$ at the throat is then fully determined for a subsonic flow at the throat. Note that when $A_L = A_{\text{th}}$, Relation (5.2.11) involves $M_{\text{th}} = M_L^*$. In such a case, the whole HLLC solution is recovered $\mathbf{W}_{\text{th}} = \mathbf{W}_L^*$, through Relations (5.2.10), (5.2.9) and (5.2.5), and the stationary wave occurring at the throat (Figure 4) disappears.

The following section is devoted to a choked flow, a specific situation where the sonic condition is met at the throat. Such a sonic situation only applies when an opening is present in the Riemann problem. The following section does not apply in the limiting situation where no dimensional reduction occurs, *i.e.* $A_L = A_{\text{th}}$. In that event, the HLLC solver shall be used directly.

5.3 Sonic flow

The present section deals with choked conditions at the throat. As the flow is choked, a specific resolution is needed. Indeed, as the flow Mach number reaches unity at the throat, pressure disturbances can no longer be communicated upstream. Consequently, the upstream side is isolated from the downstream side at the throat. A specific Riemann problem must then be solved. This specific Riemann problem only accounts for one side of the throat. More precisely, the resolution of the full Riemann problem is not addressed as only the flux solution at the cell boundary is needed, including an opening (door or window) as shown in Figure 3, where the flow is now choked.

However, as displayed in Figure 3, a part of the marked surface acts as a reflective wall. When the flow at the throat is sonic, the speed of the acoustic wave S_L becomes zero and a reflected wave $S_{\text{reflected}}$ traveling towards the left affects significantly the solution. To illustrate the situation, let us imagine a subsonic flow in a state $\mathbf{W}$ and a wall with a small hole. In the hole cross-section, the flow is sonic. Critical conditions are then reached. But as the main part of the cross-section is closed by the wall, a reflected wave propagates into state $\mathbf{W}$. The amplitude of the reflected wave is such that the modified state ($\mathbf{W}^*$) is associated with a new critical state such that the area at the throat becomes strictly equal to the new critical area: $A_{\text{cr}}^{\text{new}} = A_{\text{th}}$. We will come back to the determination of the critical state later (Section 5.5.1), when determining the flow regime, *i.e.* choked or unchoked.

When sonic conditions are met, the upstream side is isolated from the downstream side at the throat. The situation depicted in Figure 4 then transforms to the situation depicted in Figure 9, representing the present sonic half Riemann problem.

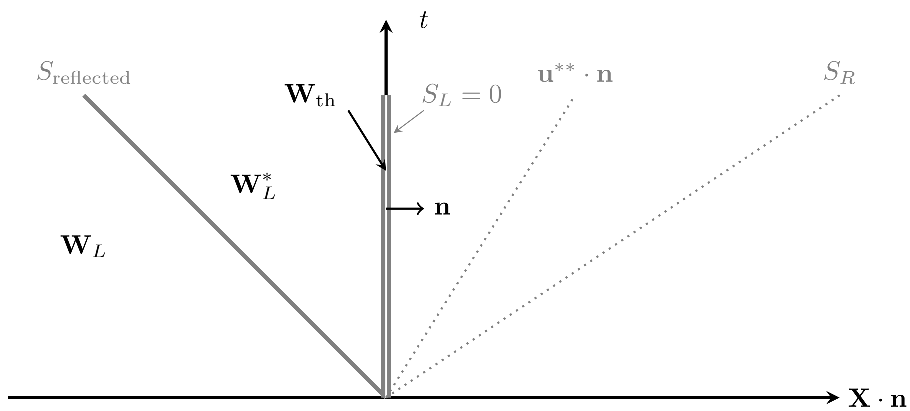

Figure 9: Schematic representation of the wave diagram of the sonic Riemann problem between the two rooms separated by a geometric discontinuity (double line). This geometric reduction is considered as a throat. Solution of the Riemann problem $\mathbf{W}_{\text{th}}$ is required at this location. In the present situation, the flow is sonic at the throat. The speed of the acoustic wave S_L is zero. Choked conditions then appear and isolate the upstream side from the downstream side at the throat. Consequently, the waves $\mathbf{u}^{**} \cdot \mathbf{n}$ and S_R on the upstream side (dotted lines) have no involvement in the solution state $\mathbf{W}_{\text{th}}$ at the throat. Those waves are here only present for the purpose of illustration. As the surface on which the Riemann problem is solved is a wall containing an opening, a reflected wave $S_{\text{reflected}}$ appears and affects the solution.

As previously, let us present the method according to the flow situation depicted in Figure 9, where a reflected wave $S_{\text{reflected}}$ is now considered and travels towards the left. The solution speed is then positive, according to the HLL solution as discussed in Section 5.1 ($\mathbf{U}_{\text{HLL}}^{*\text{mom}} \cdot \mathbf{n} > 0$), and the two solution states to be determined in the present specific Riemann problem are the state $\mathbf{W}_L^*$ behind the reflected wave and the state $\mathbf{W}_{\text{th}}$ at the throat. Naturally the method presented hereafter treats the opposite situation ($\mathbf{U}_{\text{HLL}}^{*\text{mom}} \cdot \mathbf{n} < 0$) similarly. In that case the two solution states to be determined are $\mathbf{W}_R^*$ and $\mathbf{W}_{\text{th}}$. However, we will now include the subscript "sonic" to specify that the following method applies only in the specific situation where sonic conditions are met at the throat. We will then denote the corresponding solution states by $\mathbf{W}_{L,\text{sonic}}^*$ and $\mathbf{W}_{\text{th,sonic}}$. It must be stressed that $\mathbf{W}_{L,\text{sonic}}^*$ represents the solution state that leads, through an isentropic process, to sonic conditions at the throat. $\mathbf{W}_{L,\text{sonic}}^*$ does not involve a Mach number equal to unity. The sonic condition $M_{\text{th,sonic}} = 1$ only applies at the throat.

For the sake of simplicity, the reflected wave from the perforated wall is considered through the acoustic approximations:

$$p_{L,\text{sonic}}^* = p_L + Z_L \left(\mathbf{u}_L \cdot \mathbf{n} - \mathbf{u}_{L,\text{sonic}}^* \cdot \mathbf{n} \right), \tag{5.3.1}$$

$$\rho_{L,\text{sonic}}^* = \rho_L + \frac{p_{L,\text{sonic}}^* - p_L}{c_L^2}. \tag{5.3.2}$$

The characteristic relation (5.3.1) assumes a constant acoustic impedance $Z_L = \rho_L c_L$ across the reflected wave. Such an assumption is sufficiently robust and accurate when dealing with waves of weak amplitude and is considered for the present analysis. The sign "+" applies for a left-facing reflected wave. When the flow is reversed ($\mathbf{U}^*_{\text{mom}} \cdot \mathbf{n} < 0$), the sign "$-$" applies,

$$p^*_{R,\text{sonic}} = p_R - Z_R \left(\mathbf{u}_R \cdot \mathbf{n} - \mathbf{u}^*_{R,\text{sonic}} \cdot \mathbf{n}\right). \tag{5.3.3}$$

Relation (5.3.2) approximates an isentrope with the help of the sound speed definition and is a linearized version of Laplace's law. Note that the present approximations are free of reflected wave speed ($S_{\text{reflected}}$) computation. Then, between the state $\mathbf{W}^*_{L,\text{sonic}}$ and the state $\mathbf{W}_{\text{th,sonic}}$ at the sonic throat, the same system as previously (5.2.4) - (5.2.6) holds. However the sonic condition applies in addition. The corresponding system consequently reads,

$$\rho^*_{L,\text{sonic}} A^*_L \mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n} = \rho_{\text{th,sonic}} A_{\text{th}} \mathbf{u}_{\text{th,sonic}} \cdot \mathbf{n}, \tag{5.3.4}$$

$$\frac{p^*_{L,\text{sonic}}}{\rho^{*\gamma}_{L,\text{sonic}}} = \frac{p_{\text{th,sonic}}}{\rho^{\gamma}_{\text{th,sonic}}}, \tag{5.3.5}$$

$$\frac{\gamma p^*_{L,\text{sonic}}}{(\gamma-1)\rho^*_{L,\text{sonic}}} + \frac{1}{2}\left(\mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n}\right)^2 = \frac{\gamma p_{\text{th,sonic}}}{(\gamma-1)\rho_{\text{th,sonic}}} + \frac{1}{2}\left(\mathbf{u}_{\text{th,sonic}} \cdot \mathbf{n}\right)^2, \tag{5.3.6}$$

$$\left(\mathbf{u}_{\text{th,sonic}} \cdot \mathbf{n}\right)^2 = c^2_{\text{th,sonic}}. \tag{5.3.7}$$

System (5.3.4) - (5.3.7) involves the unknowns $p^*_{L,\text{sonic}}$, $\mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n}$, $\rho^*_{L,\text{sonic}}$, $\rho_{\text{th,sonic}}$, $\mathbf{u}_{\text{th,sonic}} \cdot \mathbf{n}$ and $p_{\text{th,sonic}}$. The geometric areas $A^*_L = A_L$ and A_{th} are perfectly known at this level. System (5.3.4) - (5.3.7) is closed with the help of Relations (5.3.1) - (5.3.2). As seen in the previous section, the combination of the mass equation (5.3.4), Laplace's law (5.3.5) and the total enthalpy equation (5.3.6) yields Relation (5.2.11), linking the Mach numbers to the geometric areas. The present section deals with the sonic condition at the throat where $M_{\text{th,sonic}} = 1$ appears. In this condition, Relation (5.2.11) reduces to,

$$\frac{A_L}{A_{\text{th}}} = \frac{1}{M^*_{L,\text{sonic}}}\left(\frac{2}{\gamma+1}\left(1 + \frac{\gamma-1}{2} M^{*2}_{L,\text{sonic}}\right)\right)^{\frac{\gamma+1}{2(\gamma-1)}}. \tag{5.3.8}$$

Recall that $M^*_{L,\text{sonic}}$ represents the Mach number in the state $\mathbf{W}^*_{L,\text{sonic}}$ when this state is meant to bring sonic conditions at the throat. $M^*_{L,\text{sonic}}$ is then different from 1. Note that the previous relation implies that the throat cross-section is the critical area, $A_{\text{cr}} = A_{\text{th}}$. We will come back to the critical condition in Section 5.5.1. Note also that sign "$-$" appears in Eq. (5.3.8) when the flow is reversed because $M_{\text{th}} = -1$ in Relation (5.2.11). The Mach number $M^*_{L,\text{sonic}}$ is determined from Relation (5.3.8) using an iterative method. In the present work, we only consider the situation where the flow in the state $\mathbf{W}^*_{L,\text{sonic}}$ behind the reflected wave is subsonic and becomes sonic at the throat through an isentropic transformation. The reverse situation with $\mathbf{W}^*_{R,\text{sonic}}$ is treated similarly. Consequently, the Mach number $M^*_{L,\text{sonic}}$ behind the reflected wave is necessarily bounded between $[0, 1[$ or alternatively $M^*_{R,\text{sonic}} \in]-1, 0]$ if the flow is reversed. The bisection method is then used. Using the Mach number definition,

$$\mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n} = M^*_{L,\text{sonic}} c^*_{L,\text{sonic}}\left(p^*_{L,\text{sonic}}, \rho^*_{L,\text{sonic}}\right), \tag{5.3.9}$$

and upon insertion of (5.3.1) - (5.3.2), and the ideal-gas equation of state $c = \sqrt{\gamma \frac{p}{\rho}}$, it becomes,

$$\mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n} = M^*_{L,\text{sonic}} \sqrt{\gamma \frac{p_L + Z_L\left(\mathbf{u}_L \cdot \mathbf{n} - \mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n}\right)}{\rho_L\left(1 + \frac{\mathbf{u}_L \cdot \mathbf{n} - \mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n}}{c_L}\right)}}. \tag{5.3.10}$$

Relation (5.3.10) yields a nonlinear function, requiring another iterative process to compute $\mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n}$. In the present work the Newton-Raphson method is used with $\mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n} = M^*_{L,\text{sonic}} c_L$ as the initial guess. Note that when the flow is reversed ($\mathbf{U}^{*\text{mom}}_{\text{HLL}} \cdot \mathbf{n} < 0$), sign "$-$" applies,

$$\mathbf{u}^*_{R,\text{sonic}} \cdot \mathbf{n} = M^*_{R,\text{sonic}} \sqrt{\gamma \frac{p_R - Z_R\left(\mathbf{u}_R \cdot \mathbf{n} - \mathbf{u}^*_{R,\text{sonic}} \cdot \mathbf{n}\right)}{\rho_R\left(1 - \frac{\mathbf{u}_R \cdot \mathbf{n} - \mathbf{u}^*_{R,\text{sonic}} \cdot \mathbf{n}}{c_R}\right)}}. \tag{5.3.11}$$

Once the velocity $\mathbf{u}^*_{L,\text{sonic}} \cdot \mathbf{n}$ is determined, the pressure $p^*_{L,\text{sonic}}$ and the density $\rho^*_{L,\text{sonic}}$ are computed by Relations (5.3.1) - (5.3.2). The flow variables $\mathbf{W}_{\text{th,sonic}}$ at the throat are then deduced from Relations (5.3.4) - (5.3.7) leading to,

$$\left(\mathbf{u}_{\text{th,sonic}}\cdot\mathbf{n}\right)^2 = c^2_{\text{th,sonic}} = \frac{2}{\gamma+1}\left(\frac{\gamma p^*_{L,\text{sonic}}}{\rho^*_{L,\text{sonic}}} + \frac{1}{2}\left(\gamma-1\right)\left(\mathbf{u}^*_{L,\text{sonic}}\cdot\mathbf{n}\right)^2\right), \tag{5.3.12}$$

$$\rho_{\text{th,sonic}} = \rho^*_{L,\text{sonic}}\left(\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M^{*2}_{L,\text{sonic}}\right)\right)^{\frac{1}{\gamma-1}}, \tag{5.3.13}$$

$$p_{\text{th,sonic}} = \rho_{\text{th,sonic}}\frac{2}{\gamma+1}\left(\frac{p^*_{L,\text{sonic}}}{\rho^*_{L,\text{sonic}}} + \frac{1}{2}\frac{\gamma-1}{\gamma}\left(\mathbf{u}^*_{L,\text{sonic}}\cdot\mathbf{n}\right)^2\right). \tag{5.3.14}$$

The velocity vector at the throat $\mathbf{u}_{\text{th,sonic}}$ is obtained in a similar way as in Section 5.2, by projecting back the normal velocity to the Cartesian coordinate system considering constant tangential components. The fluxes for the mass, momentum and energy equations are computed with this set of variables, as detailed later.

5.4 Supersonic flow

In Section 5.2, the flow is supposed to be subsonic everywhere, even at the throat. The corresponding Riemann problem is presented in Figure 4. As the flow is not choked, the HLLC solver directly provides the solution state before the throat, as discussed in Section 3.2. The set of isentropic and isenthalpic relations then yields the solution state at the throat. In Section 5.3, choked conditions are considered at the throat. As the main part of the marked surface is closed by a wall (Figure 3), a reflected wave $S_{\text{reflected}}$ is considered in addition to the acoustic waves S_L and S_R and the contact wave $\mathbf{u}^{**}\cdot\mathbf{n}$. However, as the flow is sonic at the throat, the speed of the acoustic wave S_L becomes zero. The reflected wave $S_{\text{reflected}}$ is consequently the only wave traveling towards the left (alternatively towards the right if the flow is reversed). The flow behind the reflected wave is supposed to be subsonic. Figure 9 displays the corresponding Riemann problem. Due to the geometric restriction (the throat), the flow accelerates through an isentropic process and reaches the sonic state. Recall that only the flux solution at the cell boundary is needed. The resolution of the full Riemann problem is not necessary.

The supersonic situation is now discussed. When a supersonic flow appears with the conventional method, the stationary wave depicted in Figure 4 is not present, as the Euler equations without cross-section variation are solved, and the two acoustic waves S_L and S_R as well as the contact wave $\mathbf{u}^{**}\cdot\mathbf{n}$ travel all towards the right (alternatively towards the left if the flow is reversed). The situation is different with the MUZO Riemann problem because of the presence of the stationary wave related to the geometric discontinuity (the throat). The reflected wave $S_{\text{reflected}}$ is also to be considered to address the supersonic case. The amplitude of the reflected wave is such that the flow immediately behind is subsonic and sonic at the throat. To illustrate the situation, let us one more time imagine a wall with a small hole. A supersonic flow reaches the wall. A reflected wave is created and consists of a moving shock. Across the reflected wave, the flow becomes subsonic. As the resulting subsonic flow travels towards the geometric restriction (the throat), it is accelerated through an isentropic process. However, the reflected shock wave adjusts its amplitude so that the flow at the throat is choked. As only a subsonic flow accelerating and reaching the sonic condition at the throat is treated, the Riemann problem corresponds to the one presented in Section 5.3 where choked conditions are addressed. A supersonic flow is consequently treated as a sonic case due to the reflected wave. Such a situation occurs when the Mach number in the unperturbed state is greater than unit, *i.e.* $M_L > 1$ (alternatively $M_R < -1$ if the flow is reversed).

5.5 Flow regime

Two specific procedures have been developed previously. The first deals with a subsonic flow at the throat and the second deals with choked conditions. When the upstream unperturbed state reveals a supersonic flow, the throat is necessarily choked. Otherwise, the flow at the throat can be either subsonic or sonic. The present section addresses the selection of the appropriate flow regime according to the flow conditions. The method begins by assuming a subsonic flow everywhere, even at the throat. This assumption is then assessed by comparing the subsonic solution to critical conditions. The subsonic Riemann problem depicted in Figure 4 is considered and used one more time to present the proposed method. The flow direction is assumed “positive”, *i.e.* $\mathbf{u}^{**}\cdot\mathbf{n} > 0$, and the solution state $\mathbf{W}^*_L$ is used. Naturally, the method similarly applies for the reverse situation, *i.e.* $\mathbf{u}^{**}\cdot\mathbf{n} < 0$.

As seen in Section 5.2, assuming a subsonic flow at the throat yields a simple, robust and direct computation of the solution state $\mathbf{W}^*_L$ with the help of the HLLC solver. Recall that we only deal with situations involving $A_L = A_R$. Computation of the solution state $\mathbf{W}_{\text{th}}$ at the throat or the solution states $\mathbf{W}^*_{L,\text{sonic}}$, $\mathbf{W}_{\text{th,sonic}}$ in sonic conditions is not direct and requires iterative resolutions (Sections 5.2 and 5.3). The subsonic solution state $\mathbf{W}^*_L$, resulting from the

HLLC solver, will then be used to determine the flow regime at the throat, *i.e.* subsonic or sonic, and consequently to use the Riemann solver appropriate to the situation.

5.5.1 Critical state

The analysis of the 1D equations of compressible fluid mechanics, in stationary and isentropic situations, reveals two conditions for the flow to be choked at the throat:

$$\begin{cases} A_{\text{th}} \leq A_{\text{cr}}, \\ R_p \leq R_{p,\text{cr}}, \end{cases} \quad \longrightarrow \quad \text{sonic.} \tag{5.5.1}$$

The first condition involves the geometric throat cross-section A_{th} that becomes equal or less than the critical area A_{cr} when sonic conditions are met at the throat. The second condition involves the ratio between the static and stagnation pressures at the throat R_p that must also be equal or less than the critical ratio $R_{p,\text{cr}}$. In the present context, the two properties: stationary and isentropic, appear across the stationary wave, that is to say between the states $\mathbf{W}_L^*$ and $\mathbf{W}^{**}$ of Figure 4. The proposed method begins by assuming a subsonic flow. The assessment of the subsonic assumption must then be carried out between the subsonic solution state $\mathbf{W}_L^*$ and the solution state $\mathbf{W}_{\text{th}}$ at the throat. Relevance of the subsonic assumption is examined through the inequalities:

$$\begin{cases} A_{\text{th}} > A_{\text{cr}}, \\ R_p > R_{p,\text{cr}}, \end{cases} \quad \longrightarrow \quad \text{subsonic.} \tag{5.5.2}$$

Assuming a subsonic flow yields a direct computation of the solution state $\mathbf{W}_L^*$ via the HLLC solver. If such a solution satisfies both inequalities of (5.5.2) then the subsonic assumption is relevant and the solution state $\mathbf{W}_{\text{th}}$ is computed with the unchoked Riemann solver presented in Section 5.2. However, if the inequalities of (5.5.2) are not fulfilled, the subsonic solution $\mathbf{W}_L^*$ must be left out as choked conditions appear at the throat. The specific Riemann solver presented in Section 5.3 is then used and provides the actual solution state $\mathbf{W}_{L,\text{sonic}}^*$ as well as the solution state $\mathbf{W}_{\text{th,sonic}}$ involving sonic conditions at the throat. The method then requires knowledge of A_{th}, A_{cr}, R_p and $R_{p,\text{cr}}$ from the subsonic solution state $\mathbf{W}_L^*$ to determine the flow regime. Let us start by analyzing the critical area.

Critical area

The critical area A_{cr} represents the fictitious minimum throat cross-section that would be necessary to isentropically accelerate the flow to a Mach number of 1. Its expression results from the conservation of mass, specific entropy, and specific total enthalpy between the states $\mathbf{W}_L^*$ and $\mathbf{W}_{\text{th}}$ in addition to the sonic condition $\mathbf{u}_{\text{th}} \cdot \mathbf{n} = c_{\text{th}}$ at the throat. The combination of those last points yields Eq. (5.3.8), developed during the analysis of the choked situation in Section 5.3, and reformulated hereafter as:

$$A_{\text{cr}} = A_L M_L^* \left(\frac{1 + \frac{(\gamma-1)}{2} M_L^{*2}}{1 + \frac{(\gamma-1)}{2}} \right)^{-\frac{\gamma+1}{2(\gamma-1)}}. \tag{5.5.3}$$

The critical area A_{cr} is then known from Eq. (5.5.3) and the subsonic solution state $\mathbf{W}_L^*$ obtained from the HLLC solver. It provides the fictitious throat area related to the current state $\mathbf{W}_L^*$, for choked flow conditions to appear. As long as $A_{\text{th}} > A_{\text{cr}}$, the flow does not change regime. However, the previous inequality is not the only condition to be satisfied for the subsonic assumption to be relevant. The ratio R_p between the static and stagnation pressures must also be compared to the critical one $R_{p,\text{cr}}$.

Critical pressure ratio

The critical pressure ratio $R_{p,\text{cr}}$ represents the fictitious minimum ratio between the static and stagnation pressures at the throat required for the flow to become choked. It reads:

$$R_{p,\text{cr}} = \frac{p_{\text{th,sonic}}}{p_{0,\text{sonic}}}, \tag{5.5.4}$$

where the subscript "sonic" has been one more time added to specify that $p_{\text{th,sonic}}$ and $p_{0,\text{sonic}}$ represent respectively the static pressure at the throat and the stagnation pressure when sonic conditions are met at the throat. In the following the specification "sonic" will be used every time sonic conditions are involved. The stagnation pressure (indexed $_0$) describes the fictitious pressure of a fluid adiabatically brought to rest ($\mathbf{u} = \mathbf{0}$). Its expression results from the invariance of specific entropy and specific total enthalpy $\left(H = e + \frac{p}{\rho} + \frac{1}{2}\mathbf{u} \cdot \mathbf{u}\right)$ between the fictitious stagnation state ($_0$) and the states $\mathbf{W}^*_{L,\text{sonic}}$ and $\mathbf{W}_{\text{th,sonic}}$ (Figure 9). The analysis shows that the stagnation pressure does not vary throughout an isentropic flow. With the help of the ideal-gas equation of state (2.2), the sound speed $c^2 = \frac{\gamma p}{\rho}$ and the isentropic relation (Laplace's law (5.2.3)) written under the form $\frac{p_{0,\text{sonic}}}{\rho^\gamma_{0,\text{sonic}}} = \frac{p_{L,\text{sonic}}}{\rho^\gamma_{L,\text{sonic}}} = \frac{p_{\text{th,sonic}}}{\rho^\gamma_{\text{th,sonic}}}$, the stagnation pressure reads:

$$p_{0,\text{sonic}} = p_{\text{th,sonic}} \left(1 + \frac{\gamma - 1}{2} \underbrace{M^2_{\text{th,sonic}}}_{1}\right)^{\frac{\gamma}{\gamma-1}} = p^*_{L,\text{sonic}} \left(1 + \frac{\gamma - 1}{2} M^{*2}_{L,\text{sonic}}\right)^{\frac{\gamma}{\gamma-1}}. \tag{5.5.5}$$

Again, it is important to stress that $p^*_{L,\text{sonic}}$ and $M^*_{L,\text{sonic}}$ represent the pressure and Mach number in the state $\mathbf{W}^*_{L,\text{sonic}}$ when this state is meant to bring sonic conditions at the throat. $M^*_{L,\text{sonic}}$ is then different from 1. As the sonic condition is considered: $M_{\text{th,sonic}} = 1$, and Relation (5.5.4) transforms to:

$$R_{p,\text{cr}} = \left(1 + \frac{\gamma - 1}{2}\right)^{-\frac{\gamma}{\gamma-1}}. \tag{5.5.6}$$

Note that for air with $\gamma = 1.4$, the well-known result $R_{p,\text{cr}} \simeq 0.52828$ appears. The critical pressure ratio $R_{p,\text{cr}}$ is then expressed at the throat through Relation (5.5.6). The pressure ratio R_p is then to be expressed at the throat as well and compared to $R_{p,\text{cr}}$. Similarly the pressure ratio reads:

$$R_p = \frac{p_{\text{th}}}{p_0} = \frac{p_{\text{th}}}{p_{\text{th}} \left(1 + \frac{\gamma-1}{2} M^2_{\text{th}}\right)^{\frac{\gamma}{\gamma-1}}} = \left(1 + \frac{\gamma - 1}{2} M^2_{\text{th}}\right)^{-\frac{\gamma}{\gamma-1}}. \tag{5.5.7}$$

Under the form (5.5.7), the pressure ratio requires knowledge of the solution state $\mathbf{W}_{\text{th}}$ at the throat, under the assumption of a subsonic flow. However, at this level, only the subsonic solution state $\mathbf{W}^*_L$ resulting from the HLLC solver is directly known. From this state, the state $\mathbf{W}_{\text{th}}$ can be determined at the throat, but an iterative resolution is required, see Section 5.2. Nevertheless, solution existence may fail for $\mathbf{W}_{\text{th}}$. This typically happens when the flow regime is sonic at the throat. When such a situation appears, $\mathbf{W}_{\text{th}}$ as well as R_p are unavailable. The absence of a mathematical solution suggests a sonic regime. However, as will be seen in the following, it is possible to reach this conclusion by reformulating the inequality $R_p > R_{p,\text{cr}}$ in the state $\mathbf{W}^*_L$.

Indeed, after some algebraic manipulations, the combination of Relations (5.5.2), (5.5.6) and (5.5.7) results in:

$$R_p > R_{p,\text{cr}} \Leftrightarrow M_{\text{th}} < 1 \qquad \longrightarrow \qquad \text{subsonic}. \tag{5.5.8}$$

This result appears obvious but will be useful in the following. It shows that comparing the pressure ratio R_p to the critical one $R_{p,\text{cr}}$ is equivalent to comparing the Mach number at the throat, obtained under the subsonic assumption, to unity. The state $\mathbf{W}_{\text{th}}$ is still unknown at this point. However, Relation (5.5.8) can be reformulated in the state $\mathbf{W}^*_L$. Indeed, let us come back to Relation (5.2.11), describing the conservation of mass, specific entropy, and specific total enthalpy between the states $\mathbf{W}^*_L$ and $\mathbf{W}_{\text{th}}$,

$$\frac{A_L}{A_{\text{th}}} = \frac{M_{\text{th}}}{M^*_L} \left(\frac{1 + \frac{(\gamma-1)}{2} M^{*2}_L}{1 + \frac{(\gamma-1)}{2} M^2_{\text{th}}}\right)^{\frac{\gamma+1}{2(\gamma-1)}}. \tag{5.5.9}$$

This relation is valid for an isentropic flow. When sonic conditions are considered, Relation (5.5.9) reduces to (5.3.8). Relation (5.5.9) links the Mach number at the throat M_{th} in the state $\mathbf{W}_{\text{th}}$ to the Mach number M^*_L in the state $\mathbf{W}^*_L$, and depends only on the geometric areas A_L/A_{th} and the isentropic exponent γ, those being necessarily positive. Between those two states, the Mach number varies due to the isentropic acceleration induced by the throat area. The transition from M_{th} to M^*_L (or the opposite) requires an iterative method. However, the flow direction is unchanged so the transition from M_{th} to M^*_L does not change the sign of inequality (5.5.8). Relation (5.5.8) can then be reformulated in the state $\mathbf{W}^*_L$ through Relation (5.5.9):

$$\begin{array}{ll} M_{\text{th}} & < 1, \\ \updownarrow (5.5.9) & \updownarrow (5.5.9) \\ M^*_L & < M^*_{L,\text{sonic}}. \end{array} \tag{5.5.10}$$

One must bear in mind that $M^*_{L,\text{sonic}}$ represents the Mach number in the state $\mathbf{W}^*_{L,\text{sonic}}$ when this state is meant to bring the sonic condition at the throat. $M^*_{L,\text{sonic}}$ is then different from 1. Its value is obtained by solving Equation

(5.3.8) with the help of a root-finding procedure. In the present work the bisection method is used as the subsonic solution is necessarily bounded between $[0, 1[$ (alternatively $]-1, 0]$ if the flow is reversed).

Concluding remarks

The previous analysis shows that comparing the pressure ratio R_p to the critical one $R_{p,\mathrm{cr}}$ is equivalent to comparing the Mach number at the throat, obtained under the subsonic assumption, to unity. Moreover, it is also identical to comparing the Mach numbers in the state $\mathbf{W}_L^*$:

$$R_p > R_{p,\mathrm{cr}} \Leftrightarrow M_{\mathrm{th}} < 1 \Leftrightarrow M_L^* < M_{L,\mathrm{sonic}}^* = M_{\max} \qquad \longrightarrow \qquad \text{subsonic.} \tag{5.5.11}$$

The last form is more convenient in the present context as it does not involve the solution state $\mathbf{W}_{\mathrm{th}}$ at the throat, that may be unavailable. $M_L^* < M_{L,\mathrm{sonic}}^* = M_{\max}$ compares the Mach number M_L^*, resulting from the HLLC solver and the subsonic assumption, to the Mach number $M_{L,\mathrm{sonic}}^*$ in the state $\mathbf{W}_{L,\mathrm{sonic}}^*$ that would be required to bring sonic conditions at the throat. It represents the maximum Mach number allowed $M_{L,\mathrm{sonic}}^* = M_{\max}$ in the state $\mathbf{W}_L^*$ for the flow to be subsonic. Beyond $M_{\max}$ the flow at the throat is necessarily sonic. An iterative method is nonetheless needed to find $M_{L,\mathrm{sonic}}^*$ through Relation (5.3.8). However, a simple analysis of function (5.3.8) shows that there always exists a solution in the interval $M_{L,\mathrm{sonic}}^* \in [0, 1]$.

Finally, let us comment on the mass flow rate m_{th} transiting through the throat cross-section A_{th}. When the Mach number M_{th} reaches unity at the throat, the flow is choked and the maximum mass flow rate $m_{\mathrm{th,sonic}} = m_{\max}$ is reached. After some algebraic manipulations, the critical mass flow rate for an ideal gas reads:

$$m_{\mathrm{th,sonic}} = m_{\max} = \sqrt{\frac{\gamma}{RT_0}}\, p_0 \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}} A_{\mathrm{th}}, \tag{5.5.12}$$

where T_0 is the stagnation temperature and $R = \hat{R}/\hat{W}$ with $\hat{R} \simeq 8.314$ J/mol/K describing the universal gas constant and $\hat{W}$ the molar mass. The critical mass flow rate (5.5.12) suggested by the 1D theory of a compressible flow in a stationary and isentropic situation (Chapman and Walker, 1971 [34]) is recovered and depends on the thermodynamic conditions and on the fluid properties ($\gamma = c_p/c_v$ and $R = \hat{R}/\hat{W}$). The global solution then depends on the equation of state through γ and R. At every point where the sonic state appears, Relation (5.5.12) giving the critical mass flow rate controls the mass flux transiting through the sonic area.

5.6 Complete method for flux computations

Solution states determination for flux computation

The overall method is summarized hereafter, for the computation of the solution state at the throat area.

a) Direction of the flow The HLL solver is used between the two fluid states associated with the two rooms to estimate the flow velocity direction at the throat. Relation (5.1) is used and the projected velocity $\mathbf{U}_{\mathrm{HLL}}^{*\mathrm{mom}} \cdot \mathbf{n} = \rho\mathbf{u} \cdot \mathbf{n}$ extracted from the state vector $\mathbf{U}_{\mathrm{HLL}}^*$ provides the sign of the normal velocity. The Mach number in the unperturbed state is known. When $M_L > 1$ (alternatively $M_R < -1$ if the flow is reversed), a supersonic flow goes in the throat direction. However, due to the reflected wave (see Section 5.4), the flow at the throat is choked. The sonic Riemann solver of Section 5.3 (summarized by Case 2) is then used. Otherwise, the flow at the throat can be either subsonic or sonic.

b) Subsonic assumption and critical state A subsonic regime is supposed at the throat. The situation is depicted in Figure 4 with a subsonic Riemann problem. However, as only situations involving $A_L = A_R$ (see Section 4) are considered, the HLLC solver of Toro et al. (1994) [21] (see Section 2.2) is used directly and provides the solution states $\mathbf{W}_L^*$ and $\mathbf{W}_R^*$. The Mach numbers M_L^* and M_R^* are consequently known. According to the flow direction, one of these two states is retained for the computation of the fictitious critical area A_{cr}. More precisely, it is determined with the help of Relation (5.5.3). $M_{L,\mathrm{sonic}}^*$ is also computed by solving Relation (5.3.8) with the help of an iterative method. It represents the maximum Mach number $M_{L,\mathrm{sonic}}^* = M_{\max}$ allowed in the state $\mathbf{W}_L^*$ (alternatively in the state $\mathbf{W}_R^*$) for the flow to be subsonic at the throat. Beyond $M_{\max}$ the flow at the throat is necessarily sonic. Two situations may then occur.

i. Case 1: The throat area is larger than the critical area and the Mach number is less than $M_{\max}$, *i.e.* $A_{\mathrm{th}} > A_{\mathrm{cr}}$ and $M_L^* < M_{L,\mathrm{sonic}}^* = M_{\max}$

The solution resulting from the assumption of a subsonic flow at the throat is in agreement with the criteria describing the subsonic conditions. The solution state, $\mathbf{W}_L^*$ or $\mathbf{W}_R^*$, resulting from the HLLC solver is consequently valid. From this solution state, the solution $\mathbf{W}_{\mathrm{th}}$ at the throat is computed through the set of isentropic and isenthalpic relations, *i.e.* (5.2.11) requiring an iterative method and (5.2.10), (5.2.9), (5.2.5). The solution state $\mathbf{W}_{\text{th}}$ being known, the criteria $R_p > R_{p,\text{cr}}$ and $M_{\text{th}} < 1$, previously replaced by $M_L^* < M_{L,\text{sonic}}^* = M_{\max}$ are verified and ensure that the computed solution agrees with the whole set of subsonic conditions.

ii. Case 2: At least one of the inequalities $A_{\textbf{th}} > A_{\textbf{cr}}$ and $M_L^* < M_{L,\textbf{sonic}}^* = M_{\textbf{max}}$ is not fulfilled
The subsonic HLLC solution is left out as it is in disagreement with the above criteria. Those indicate that the flow is choked at the throat. The subsonic Riemann problem of Figure 4 is replaced by the specific sonic Riemann problem depicted in Figure 9 where a reflected wave $S_{\text{reflected}}$ is considered. The actual Mach number in the solution state $\mathbf{W}_{L,\text{sonic}}^*$ or $\mathbf{W}_{R,\text{sonic}}^*$ is computed by solving Relation (5.3.8). The velocity in the same state is then obtained by solving Relation (5.3.10). Iterative procedures are necessary. The rest of the solution state is determined with Relations (5.3.1) and (5.3.2). Then the solution state $\mathbf{W}_{\text{th,sonic}}$ at the throat is determined with Relations (5.3.12) - (5.3.14).

Flux computation and solution update

We now have in hand the solution vector at the throat $\mathbf{W}_{\text{th}}$. The flux at the throat $\mathbf{F}_{\text{th}}$ is computed as:

$$\mathbf{F}_{\text{th}} = \begin{pmatrix} \rho_{\text{th}} \mathbf{u}_{\text{th}} \cdot \mathbf{n} \\ \rho_{\text{th}} \left(\mathbf{u}_{\text{th}} \cdot \mathbf{n}\right) \mathbf{u}_{\text{th}} + p_{\text{th}} \mathbf{n} \\ \left(\rho_{\text{th}} E_{\text{th}} + p_{\text{th}}\right) \mathbf{u}_{\text{th}} \cdot \mathbf{n} \end{pmatrix}, \tag{5.6.1}$$

with $E_{\text{th}} = e_{\text{th}}\left(\rho_{\text{th}}, p_{\text{th}}\right) + \frac{1}{2}\mathbf{u}_{\text{th}} \cdot \mathbf{u}_{\text{th}}$, and $e_{\text{th}}\left(\rho_{\text{th}}, p_{\text{th}}\right)$ provided by the ideal-gas equation of state (2.2). The solution, in terms of conservative variables $\mathbf{U} = \left(\rho, \rho\mathbf{u}, \rho E\right)^T$, is updated with the help of the Godunov (1959) [20] first-order scheme, as seen in Section 2.1:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Omega_i} \sum_{j=1}^{N_{\text{faces}_i}} \mathbf{\Phi}_{ij}^*, \tag{5.6.2}$$

where $n+1$ and n denote two consecutive time steps and superscript * denotes the Riemann problem solution. Index i represents the current numerical cell, and index j the direct neighbors of cell i. Ω_i is the volume of cell i and ij denotes the face separating the cells i and j. Obviously, higher-order extensions can be considered. However, such extensions involve unnecessary complexity and are not considered in the present work. Recall that a simple, fast, robust and accurate method dealing with very coarse 3D meshes is desired. Recall also that the Godunov scheme is stable under the conventional CFL condition affecting the global time step Δt as,

$$\Delta t = \text{CFL} \times \min_{ij} \left(\frac{\min\left(\mathbf{r}_{\min,i}, \mathbf{r}_{\min,j}\right)}{S_{\max,ij}^n} \right), \quad \text{with} \quad \mathbf{r}_{\min,i} = \min_k \left(\left\|\mathbf{P}_i - \mathbf{P}_{ik}\right\|\right), \quad \text{and} \quad k = \{1, N_{\text{faces}_i}\} \tag{5.6.3}$$

and $0 < \text{CFL} < 1$. In Relation (5.6.3) $S_{\max}^n$ denotes the maximum wave speed throughout the computational domain at time level n. $\mathbf{P}_i$ denotes the center of a cell i and $\mathbf{P}_{ij}$ the center of the face ij separating the elements i and j, see Figure 1.

$\mathbf{\Phi}_{ij}$ regroups the flux $\mathbf{F}_{\text{th}}$ at the cell face ij across the fluid section A_{fluid} and the wall contribution, as given by (5.6.4). Indeed, recall that in the present situation, a cell face represents an inner wall containing an opening (a throat) as depicted in Figure 3. The area of the reflective wall also contributes to the flux distribution. Moreover, an inner wall is treated as a zero-thickness volume to further reduce meshing labor. A wall-tangent face thus requires pressure distribution on both sides, *i.e.* on the left (L) and right (R) sides of the wall. Indeed, the integration of the conservation laws (3.1) on a control volume containing the perforated wall involves the following effective fluxes:

$$\mathbf{\Phi}_{ij,L}^* = A_{\text{fluid}} \mathbf{F}_{\text{th}} + A_{\text{wall}} \begin{pmatrix} 0 \\ p_{ij,L}^* \mathbf{n} \\ 0 \end{pmatrix}, \quad \mathbf{\Phi}_{ij,R}^* = A_{\text{fluid}} \mathbf{F}_{\text{th}} + A_{\text{wall}} \begin{pmatrix} 0 \\ p_{ij,R}^* \mathbf{n} \\ 0 \end{pmatrix}, \tag{5.6.4}$$

where $\mathbf{F}_{\text{th}}$ contains the solution of the specific Riemann problem, either subsonic or sonic (summarized above) when a throat is present. Naturally, the fluxes $\mathbf{F}_{\text{th}}$ across the throat area are common to both sides of a face ij. Those are computed with the help of the solution state at the throat, $\mathbf{F}_{\text{th}} = \mathbf{F}_{\text{th}}\left(\mathbf{W}_{\text{th}}\right)$ (Eq. (5.6.1)). A_{fluid} represents the cross-section through which the fluid flows. When an opening is present, it corresponds to the throat area $A_{\text{fluid}} = A_{\text{th}}$.

Otherwise, it is simply the cross-section of the cell face: $A_{\text{fluid}} = S_{ij}$. Finally, A_{wall} represents the area of the wall, *i.e.* the cross-section of the cell face S_{ij} without the fluid area: $A_{\text{wall}} = S_{ij} - A_{\text{fluid}}$.

In relation (5.6.4), $p^*_{ij,L/R}$ represents the pressure solution to the Riemann problem for the reflective impermeable part of the inner wall (Figure 3). Two Riemann problems are considered to compute respectively $p^*_{ij,L}$ on the left side and $p^*_{ij,R}$ on the right side of the wall. Pressure solutions are provided by the HLLC solver with reflective conditions. Such conditions are modeled by creating a fictitious state (fict), see Toro (1997) [25] for example. On the left side of a face ij, the conditions for the Riemann problem are,

$$\rho_{ij,R,\text{fict}} = \rho_{ij,L}, \qquad p_{ij,R,\text{fict}} = p_{ij,L}, \qquad \mathbf{u}_{ij,R,\text{fict}} \cdot \mathbf{n} = -\mathbf{u}_{ij,L} \cdot \mathbf{n} + 2\mathbf{u}_{\text{wall}} \cdot \mathbf{n}, \tag{5.6.5}$$

where $\mathbf{u}_{\text{wall}}$ is the velocity of the inner wall, *i.e.* $\mathbf{u}_{\text{wall}} = \mathbf{0}$ in the present context. The opposite treatment is used for the right side of face ij. The conditions for the Riemann problem are,

$$\rho_{ij,L,\text{fict}} = \rho_{ij,R}, \qquad p_{ij,L,\text{fict}} = p_{ij,R}, \qquad \mathbf{u}_{ij,L,\text{fict}} \cdot \mathbf{n} = -\mathbf{u}_{ij,R} \cdot \mathbf{n} + 2\mathbf{u}_{\text{wall}} \cdot \mathbf{n}. \tag{5.6.6}$$

The HLLC pressure solution reads:

$$p^*_k = p_k + \rho_k \left(S_k - \mathbf{u}_k \cdot \mathbf{n}\right) \left(\mathbf{u}^* \cdot \mathbf{n} - \mathbf{u}_k \cdot \mathbf{n}\right), \tag{5.6.7}$$

with index k denoting either the left (L) or right (R) state of the Riemann problem. In this last relation $\mathbf{u}^* \cdot \mathbf{n}$ is the projected speed solution. With the previous boundary conditions, the solution for the speed becomes $\mathbf{u}^* \cdot \mathbf{n} = \mathbf{u}_{\text{wall}} \cdot \mathbf{n} = 0$.

6. Validations and illustrations

6.1 Simplified building

Computed results with the MUZO method using very coarse meshes are compared to those of conventional 3D computations, involving both fine and coarse meshes. Recall that blast wave effects occurring at early times are supposed to be determined beforehand through an appropriate method based for example on Kingery-Bullmash data [5] (Section 1: Introduction). The present test cases are meant to compare the 3D conventional computation to the MUZO under-resolved 3D computation that focuses only on flow discharge effects occurring at longer timescales. The Godunov first-order scheme (2.10), (5.6.2) is used with CFL = 0.5 for the under-resolved and conventional computations of all test cases. A simplified building made of only two rooms is first considered with various pressure conditions and variable throat areas. Figure 10 displays the first configuration and associated initial conditions. Conventional and MUZO computations are initialized in the same way. All cells belonging to a given zone are initialized with a given pressure, temperature (or density) and velocity vector.

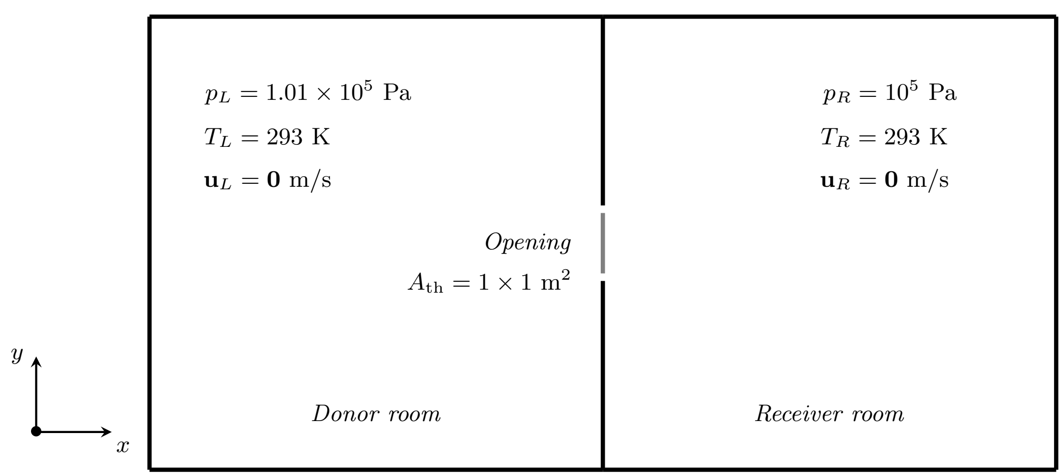

Figure 10: Schematic representation of a simple building made of two rooms only. The two rooms are of volume $V = 4 \times 4 \times 4$ m^3. An opening separates the two rooms. Its area is $A_{\text{th}} = 1 \times 1$ m^2. Air ($\gamma = 1.4$, $C_v = 719$ J/kg/K) is initially at rest $\mathbf{u}_L = \mathbf{u}_R = \mathbf{0}$ and at temperature $T_L = T_R = 293$ K. The initial pressure in the room on the left (donor room) is $p_L = 1.01 \times 10^5$ Pa. In the room on the right (receiver room), the initial pressure is $p_R = 10^5$ Pa. Shock-tube type conditions are then set. The boundary surfaces are treated as reflective walls.

As two different initial pressures are set in the two rooms, the initial pressure profile in the whole building is discontinuous and of Heaviside type. Moreover, as an opening separates the two rooms, the cross-section profile in the whole building is also discontinuous. At the opening, the geometric non-conservative term $p\frac{\partial A}{\partial x}$ in System (3.1) consists of the product of a Heaviside and a Dirac function.

Results provided by the MUZO computations are compared to those provided by the conventional 3D computations. These latter ones involve both fine and coarse meshes including every geometric details such as doors and windows. In the present section, about $60,000-70,000$ elements are considered for the fine meshes of the conventional computations. The coarse meshes of those latter ones involve as few elements as possible. However, the amount of elements depends on the size of the various openings that need to be fully meshed with a certain degree of quality for the simulations to be successful. As seen in Section 4, the MUZO under-resolved computations involve very few numerical elements and a special treatment to address the various openings. The meshes dealing with the present test case (Figure 10) are provided in Figure 11 .

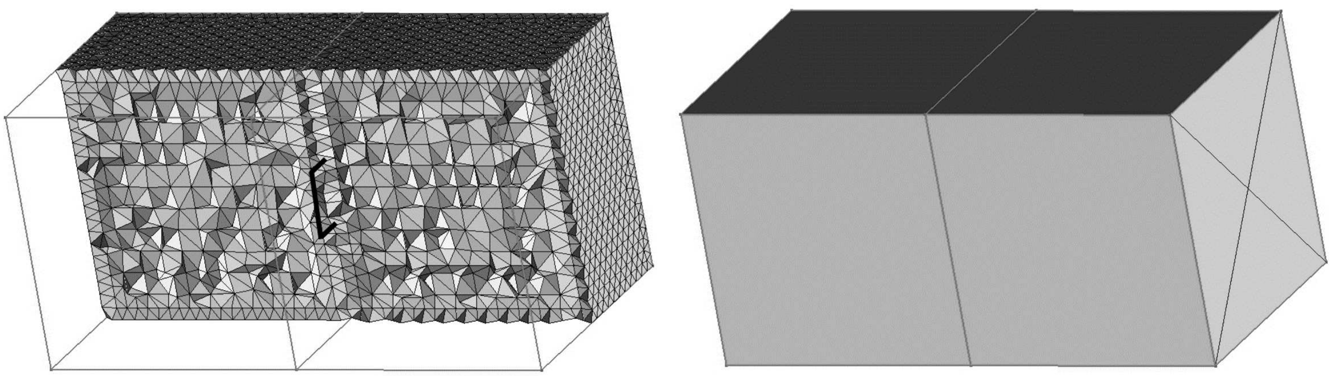

Figure 11: 3D meshes used for the test case depicted in Figure 10. On the left, the fine mesh for the conventional 3D computation is partly shown. It consists of 61.270 tetrahedral elements. The coarse mesh used for the conventional computation consists of 72 tetrahedral elements. The opening is fully drawn and meshed. It is represented by the dark lines that were purposely thickened for the sake of clarity. On the right, the mesh used for the MUZO computation is shown. Only 4 prismatic elements per room are used. The mesh is made from a linear extrusion needing only the "footprints" of the building.

The results are provided in terms of mean pressure and mean density in the two rooms, for all computations. These mean values are post-processed from the numerical solution of the governing equations. The mean density is numerically approximated as,

$$\overline{\rho} = \frac{1}{V}\int_V \rho dV \simeq \frac{1}{V}\sum_{i=1}^{N}\rho_i\Omega_i, \tag{6.1.1}$$

where V represents the volume of the corresponding room and Ω_i the volume of every element i (out of N) composing the room. The mean pressure is determined with the help of the mean density $\overline{\rho}$, the mean momentum $\overline{\rho\mathbf{u}}$, and the mean total energy $\overline{\rho E}$,

$$\overline{\rho\mathbf{u}} = \frac{1}{V}\int_V (\rho\mathbf{u})\, dV \simeq \frac{1}{V}\sum_{i=1}^{N}(\rho\mathbf{u})_i\,\Omega_i, \tag{6.1.2}$$

$$\overline{\rho E} = \frac{1}{V}\int_V (\rho E)\, dV \simeq \frac{1}{V}\sum_{i=1}^{N}(\rho E)_i\,\Omega_i. \tag{6.1.3}$$

The mean internal energy $\overline{e}$ is then computed as,

$$\overline{e} = \overline{E} - \frac{1}{2}\left(\overline{\mathbf{u}}\cdot\overline{\mathbf{u}}\right), \text{ with } \overline{\mathbf{u}} = \frac{\overline{\rho\mathbf{u}}}{\overline{\rho}}, \tag{6.1.4}$$

and the equation of state $\overline{p}\left(\overline{\rho},\overline{e}\right)$ (Eq. (2.2)) finally provides the mean pressure in the corresponding room. Results for the test case depicted in Figure 10 are shown in Figure 12.

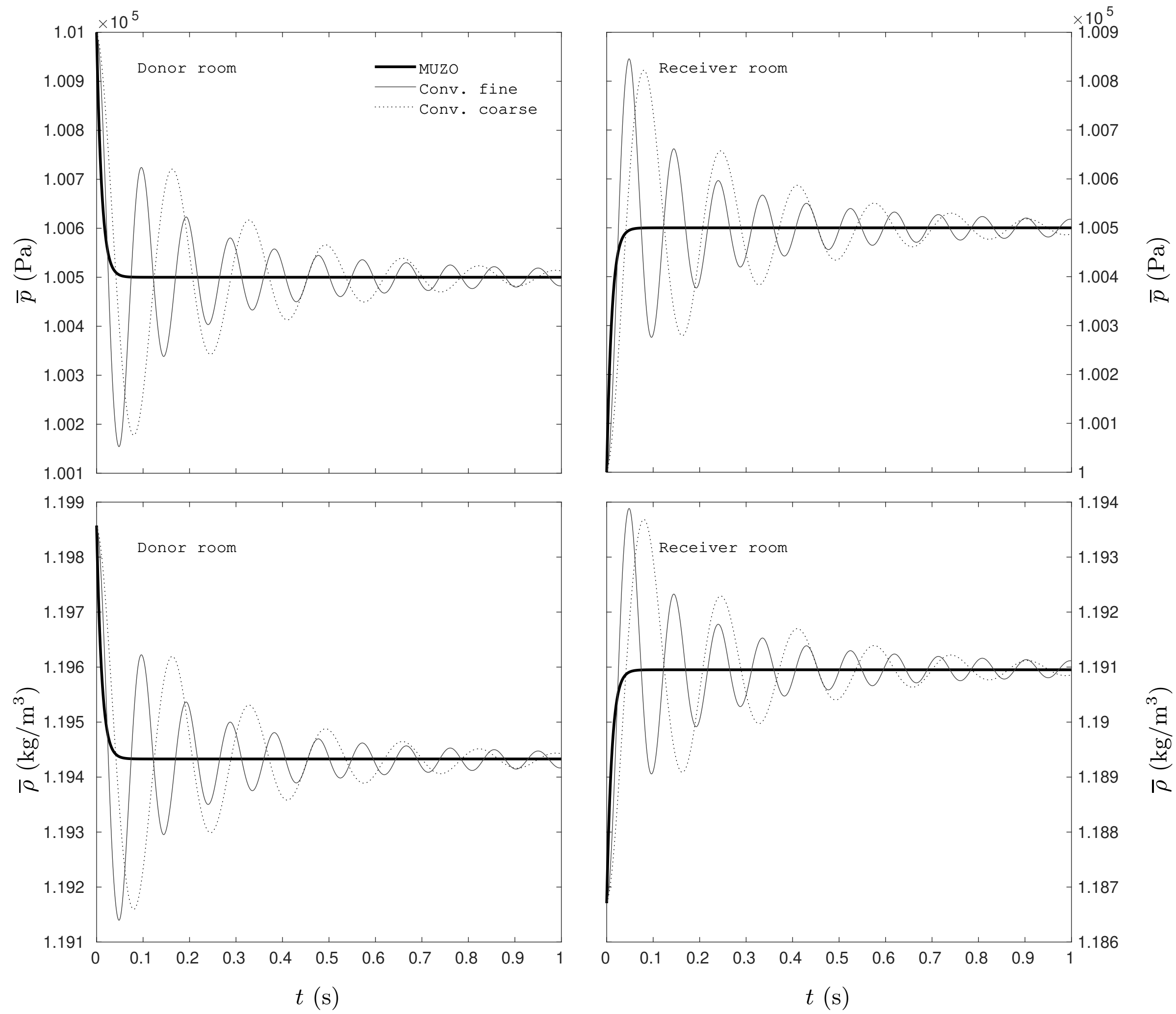

Figure 12: Test 1: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration is depicted in Figure 10. It consists of a simple building made of two rooms separated by an opening. The rooms are of volume $V = 4 \times 4 \times 4$ m^3. The opening is of area $A_{\mathrm{th}} = 1 \times 1$ m^2. The conventional 3D computations (denoted "Conv.") are performed on meshes composed of 61,270 (fine) and 72 (coarse) tetrahedral elements. The MUZO under-resolved computation is performed on a mesh made of 8 prismatic elements. The initial high pressure is 1.01×10^5 Pa.

The initial pressure in the donor room has been taken weak (1.01×10^5 Pa). In such conditions, only the subsonic part of the present Riemann solver is called. The mean density and mean pressure fields are well-determined with the MUZO under-resolved computation that required less than 1 second with a sequential implementation. At long timescales, the quasi-steady assumption made at the opening cross-section (Section 3) appears appropriate. Flow discharge effects are indeed well-reproduced. However, the MUZO method necessarily lacks accuracy regarding wave propagation as it is not designed to capture wave dynamics. The conventional computation using the fine mesh naturally shows more details of the wave dynamic solution. However, it required 2 hours and 29 minutes with a sequential implementation as well. This time is reduced to less than 1 second when the coarse mesh is used and the results are in a quite good agreement with those obtained with the fine mesh. The wave dynamic is captured as well. However, much more effort and consequently time regarding mesh generation is needed because the opening must be drawn and meshed. Such a long and tedious pre-processing is not acceptable when dealing with realistic buildings and pressing situations. The potential of the MUZO under-resolved method dwells on its ability to provide the mean density and pressure fields in a very short time. This peculiarity is possible thanks to a very easy and fast pre-processing stage and an efficient Riemann solver, adapted to the specific MUZO mesh. In the following, the capabilities of the method are tested further with various academic configurations.

The test is repeated with a high pressure of $p_L = 10^6$ Pa. The pressure being higher, a sonic situation occurs in the early stages of the solution (until $t \simeq 0.12$ s). The corresponding results are provided in Figure 13.

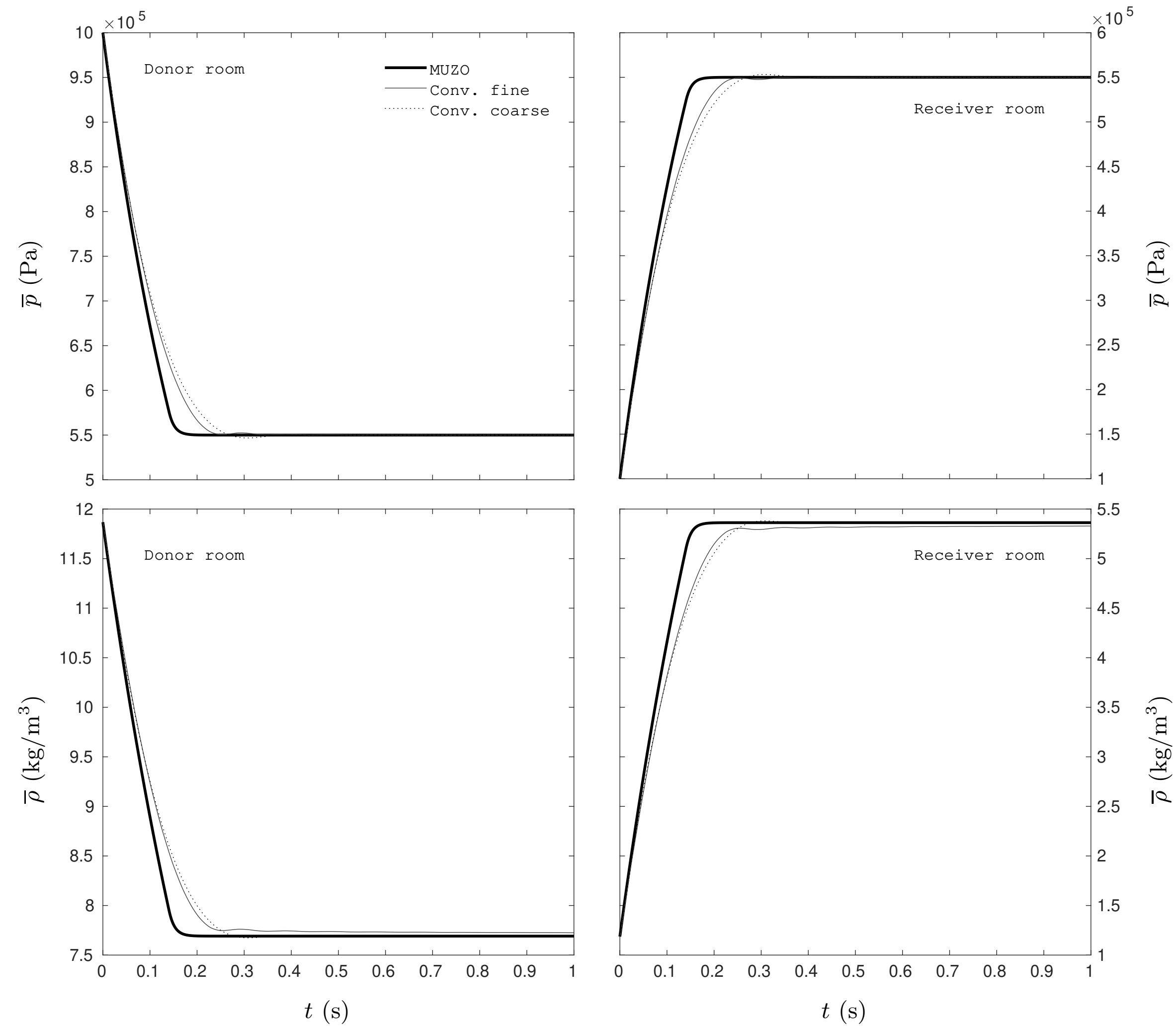

Figure 13: Test 2: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration of Figure 12 is repeated with a high pressure of $p_L = 10^6$ Pa yielding choking conditions at early times (until $t \simeq 0.12$ s).

Before the pressure reaches its equilibrium value (quasistatic pressure), a sonic flow occurs through the opening. The results provided by the MUZO under-resolved computation are again in good agreement with those obtained by the conventional computations, both in terms of pressure relaxation time, and mean (quasistatic) pressure field. The density profile agrees with the solutions of the conventional computations as well.

In the following, the initial high pressure is unchanged but the initial speed in both rooms is set to $u_L = u_R = 350$ m/s in the x-axis direction. A supersonic flow is then created (by the conventional computation) in the early stages of the solution (until the pressure in the donor room stops rising). A good agreement is observed between the MUZO solution and the results provided by the conventional computations as seen in Figure 14.

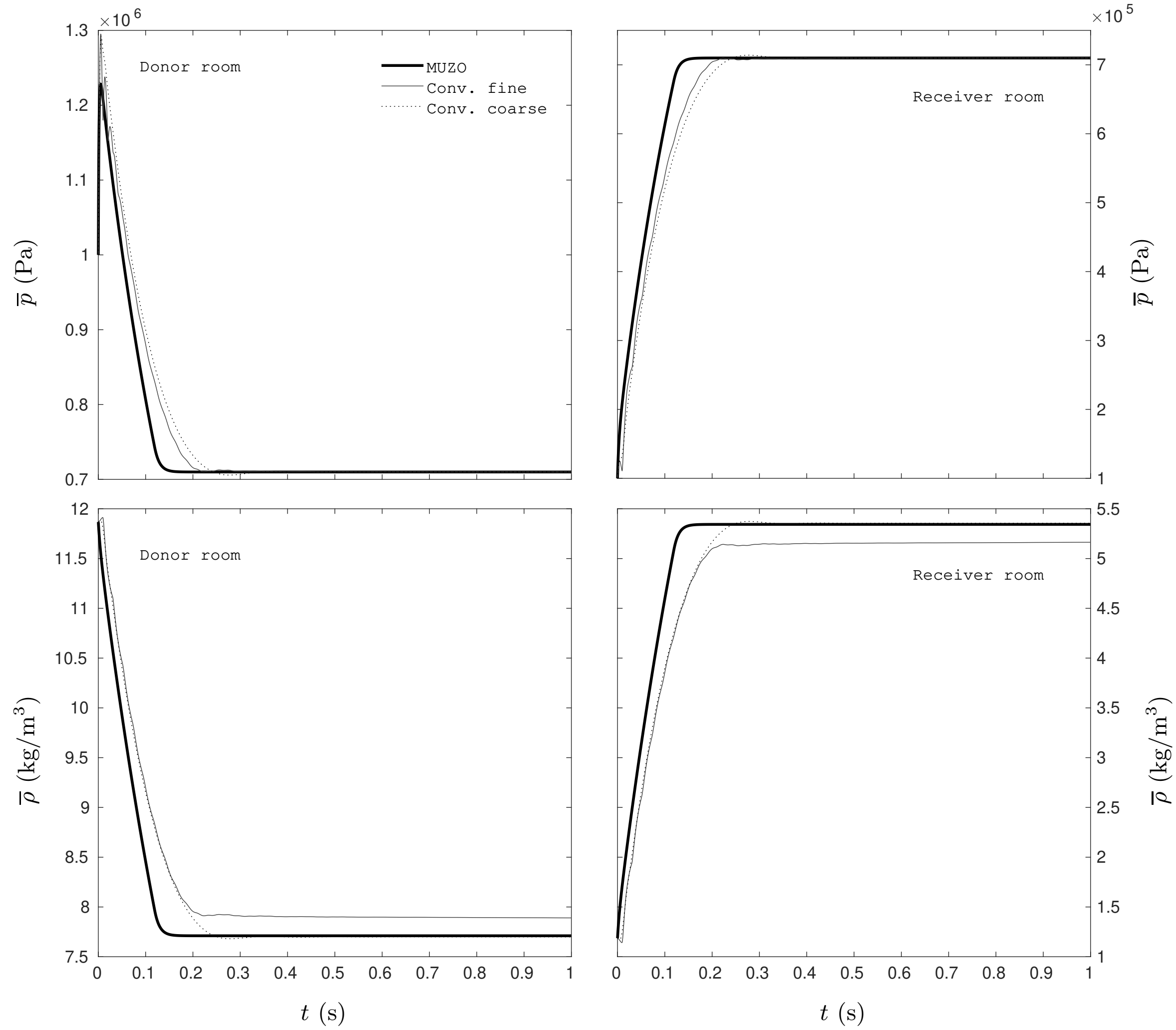

Figure 14: Test 3: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration of Figure 13 is repeated with an initial speed of $u_L = u_R = 350$ m/s in the x-axis direction yielding supersonic conditions at early times (until the pressure in the donor room stops rising). The supersonic solution is only treated by the conventional computations through the HLLC solver. Nevertheless, the MUZO numerical treatment appears sufficiently accurate.

To test the method further, the previous configuration is now slightly modified. Figure 15 displays the configuration and initial conditions.

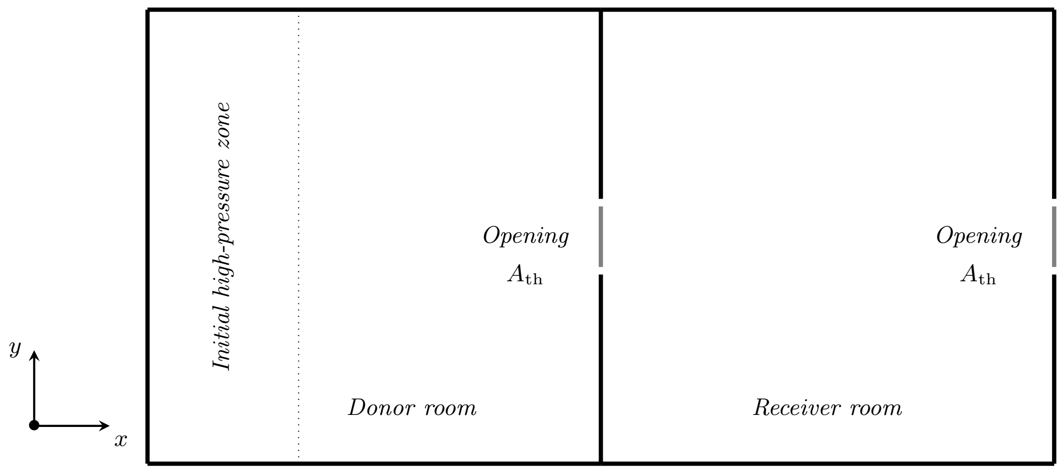

Figure 15: Schematic representation of a simple building made of only two rooms. The two rooms are of volume $V = 1 \times 1 \times 1$ m^3. A high pressure is set in a portion of the donor room. This zone is of dimensions $0.2 \times 1 \times 1$. An opening separates the two rooms, and a second opening separates the receiver room from the exterior. Air is initially at rest. The boundary surfaces are treated as reflective walls, except for the opening connected to the atmosphere. Appendix A and Appendix B provide details for the treatment of the boundary conditions.

The high pressure is now initially set only in a portion of the donor room (representing in a simplified way gases at an elevated pressure resulting from an explosion). Besides a second opening is considered. This additional opening separates the building from the exterior. In the following test, the high pressure is increased to 10^7 Pa, and the area of the opening separating the two rooms is lowered to $A_{\rm th} = 1 \times 0.01$ m^2, creating arduous sonic and subsonic conditions as time evolves. Because of the very small area of the opening, the conventional computation could not be performed with a homogeneous coarse mesh. Such a small opening requires indeed small numerical elements and consequently a certain local mesh quality. Consequently, a heterogeneous mesh made of small elements near the opening and larger elements away from it was used. The mesh consists of 397 tetrahedral elements in total. The computation time is then affected as seen in Table 1, as well as the time needed to create the mesh, that requires special care in such a situation. This difficulty is not encountered with the MUZO method that becomes particularly convenient when dealing with very small openings. Results are provided in Figure 16 in terms of mean pressure and mean density both in the donor and receiver rooms. First, the opening separating the building from the atmosphere is considered closed and reflective wall conditions are used. It will be considered open (connected to the atmosphere) for the next test configurations.

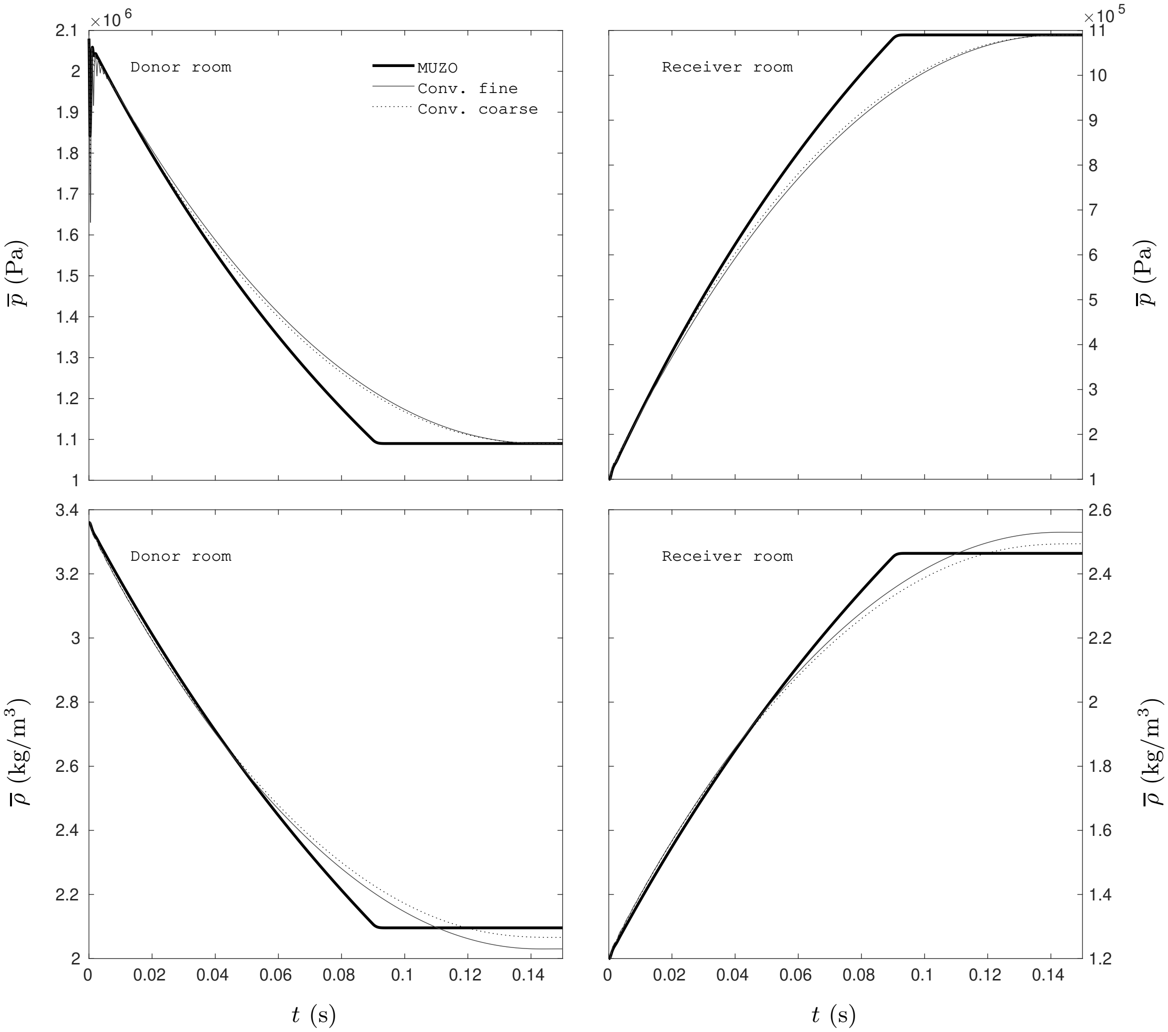

Figure 16: Test 4: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration is depicted in Figure 15. It consists of a simple building made of two rooms separated by an opening. The rooms are of volume $V = 1 \times 1 \times 1$ m^3. The opening is of area $A_{\text{th}} = 1 \times 0.01$ m^2. The high pressure is initially set only in a portion of the donor room. The initial conditions are $p_{\text{HP}} = 10^7$ Pa, $\rho_{\text{HP}} = 12$ kg/m^3 and $\mathbf{u}_{\text{HP}} = \mathbf{0}$. In the rest of the building, the initial conditions are $p_{\text{LP}} = 10^5$ Pa, $\rho_{\text{LP}} = 1.2$ kg/m^3 and $\mathbf{u}_{\text{LP}} = \mathbf{0}$. The conventional 3D computations (denoted "Conv.") are performed on meshes composed of 66, 565 (fine) and 397 (coarse) tetrahedral elements. The MUZO under-resolved computation is performed on a mesh made of 12 prismatic elements.

A good agreement between the MUZO and conventional computations appears, both in terms of pressure relaxation time and mean pressure field. The density solution also agrees with the solution provided by the conventional computations.

The previous test is now repeated with the second opening separating the receiver room from the exterior, now connected to the atmosphere. Its area is the same as the area of the opening separating the two rooms, $A_{\text{th}} = 1 \times 0.01$ m^2. The initial conditions remain the same. The atmospheric conditions are $p_{\text{atm}} = 10^5$ Pa and $T_{\text{atm}} = 289.75$ K. Appropriate treatment of the boundary condition is reported in Appendix A and Appendix B. Results are shown in Figure 17. As previously, because the areas of the openings are very small, the conventional computation could not be performed with a homogeneous coarse mesh and particular care is requested during the pre-processing step. This difficulty highlights one more time the capabilities of the MUZO method that does not need to mesh the various openings, making the method consequently very attractive when dealing with pressing situations. A heterogeneous mesh made of small elements near the openings and larger elements away from them was then used. The mesh consists of 831 tetrahedral elements in total. More elements than before are needed because of the presence of the second opening, connected to the atmosphere. The computation time is consequently affected as reported in Table 1.

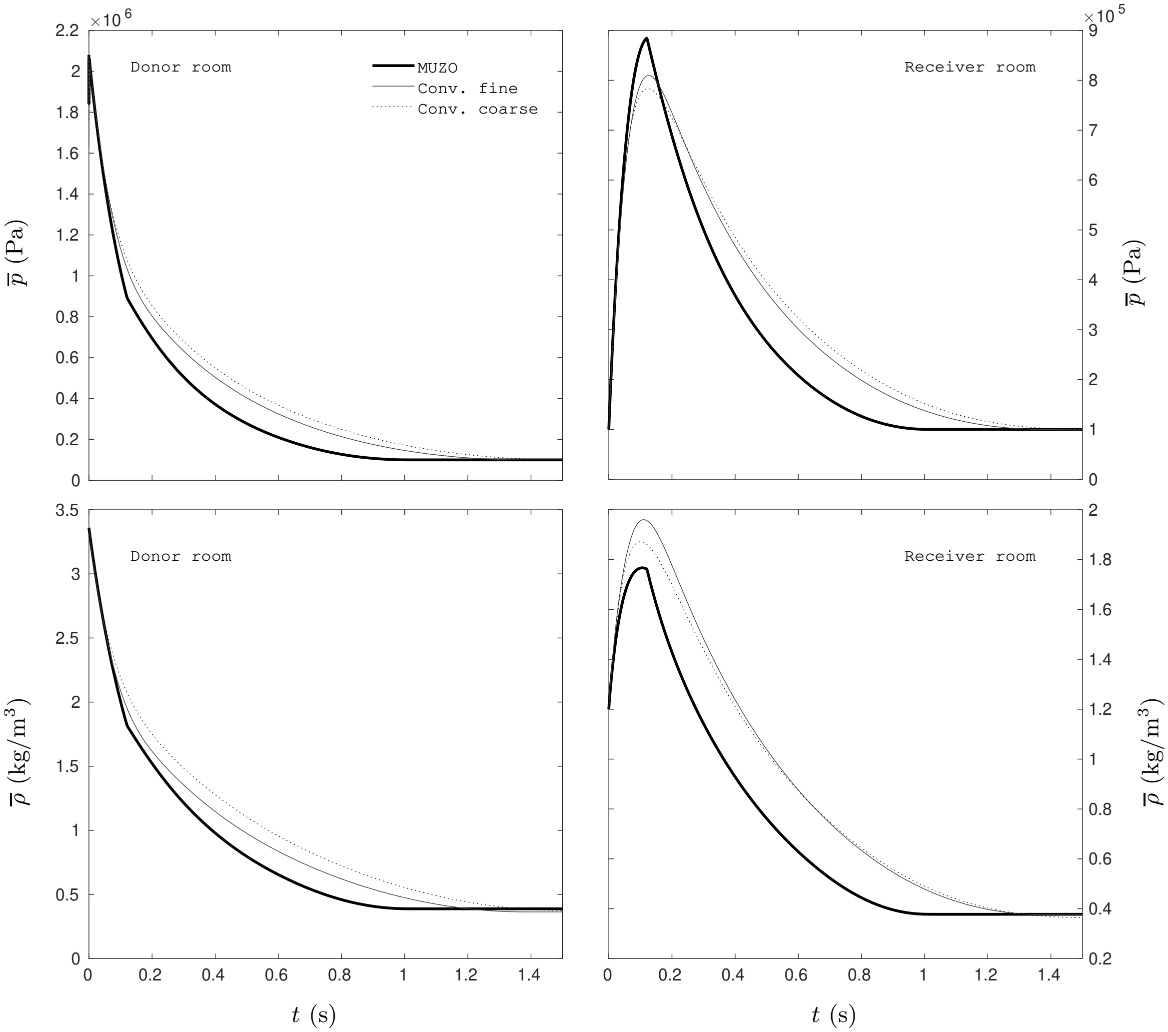

Figure 17: Test 5: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration of Figure 16 is repeated with a second opening of area $A_{\text{th}} = 1 \times 0.01$ m^2. This opening separates the receiver room from the exterior where the atmospheric conditions dwell, $p_{\text{atm}} = 10^5$ Pa and $T_{\text{atm}} = 289.75$ K. The conventional 3D computations (denoted "Conv.") are performed on meshes composed of $71,309$ (fine) and 831 (coarse) tetrahedral elements. The MUZO under-resolved computation is performed on a mesh made of 12 prismatic elements.

A good agreement between the MUZO and conventional computations appears one more time. Relaxation to the atmospheric conditions, induced by the window, is clearly seen and the solution agrees with the results of the conventional computations.

In the following test, the areas of both openings are increased to $A_{\text{th}} = 1 \times 0.1$ m^2. Results are shown in Figure 18. The areas of the openings being 10 times greater than previously, less elements are necessary for the conventional computation using the coarse mesh. The numerical elements need to fit the areas of the two openings though. A heterogeneous coarse mesh totaling 362 tetrahedral elements was then used.

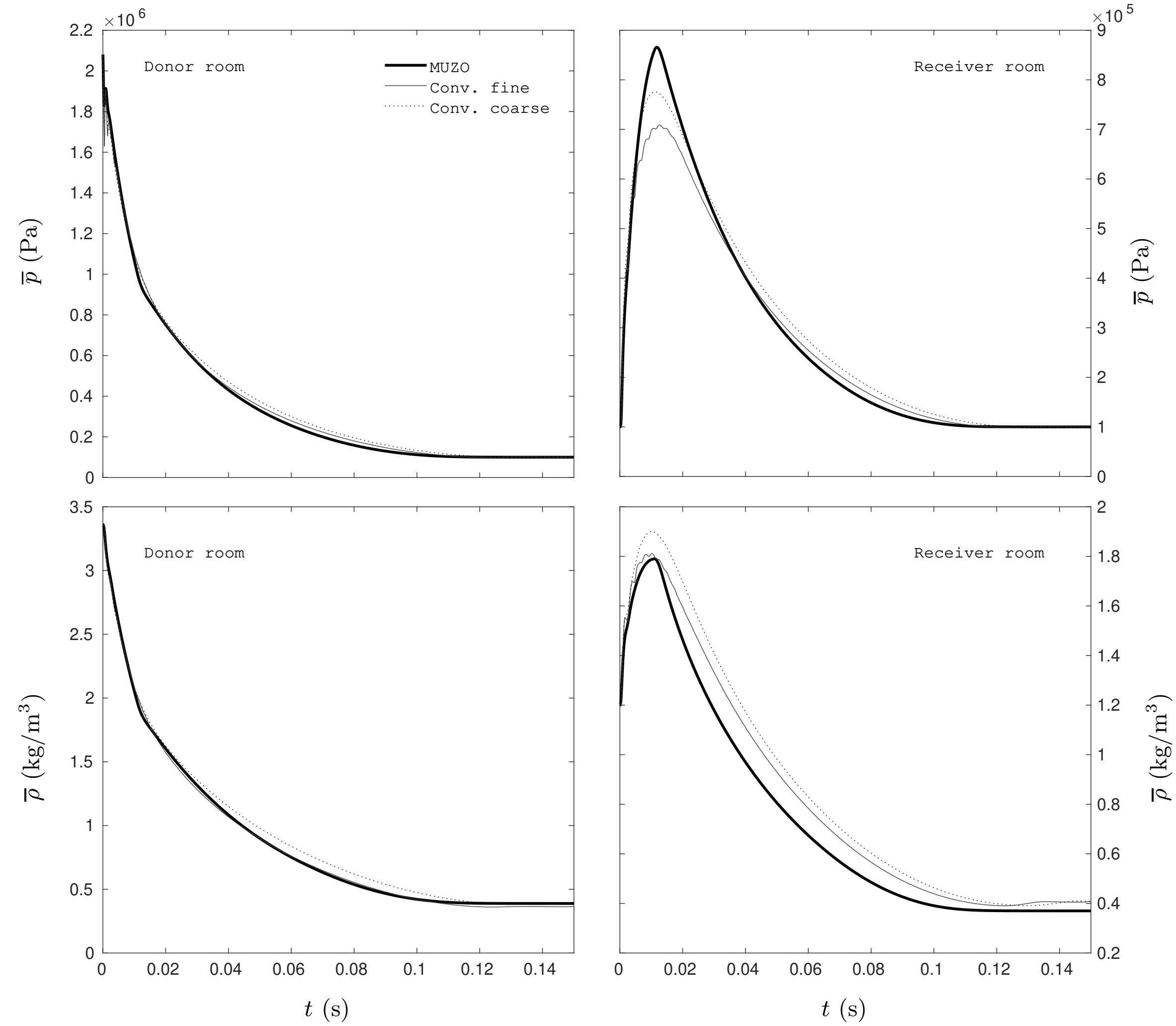

Figure 18: Test 6: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration of Figure 17 is repeated with an area of $A_{\rm th} = 1 \times 0.1$ m^2 for the two openings. The conventional 3D computations (denoted "Conv.") are performed on meshes composed of $60,602$ (fine) and 362 (coarse) tetrahedral elements. The MUZO under-resolved computation is performed on a mesh made of 12 prismatic elements.

Again, a good agreement in terms of relaxation time and mean pressure and mean density fields, between the MUZO and conventional computations, is observed.

In the following test, the high-pressure in the donor room is lowered to $p_{\rm HP} = 1.01 \times 10^5$ Pa. Results are shown in Figure 19.

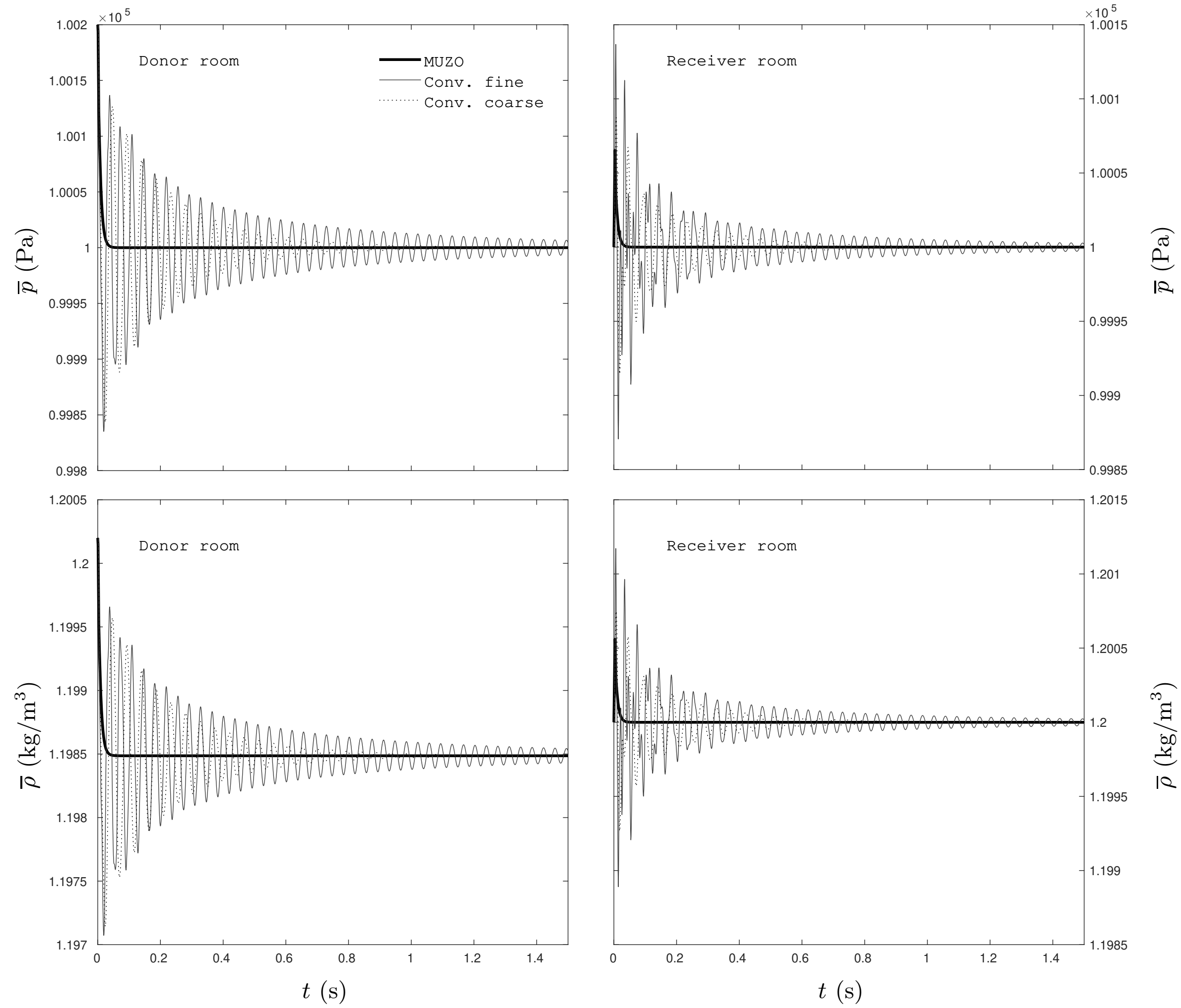

Figure 19: Test 7: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration of Figure 18 is repeated with an initial high pressure of $p_{\mathrm{HP}} = 1.01 \times 10^5$ Pa and an initial high density of $\rho_{\mathrm{HP}} = 1.201$ kg/m^3.

Similarly to Figure 12, where the high pressure is quite low, the conventional computations show more details of the wave dynamic solution than the MUZO under-resolved computation. This lack of accuracy (regarding wave dynamics) is not surprising as the present method averages the solution along the full cross-section of the marked faces (doors, windows for example), and supposes a quasi-steady flow at the opening cross-section (Section 3). This assumption is indeed appropriate at long timescales but is inappropriate to capture shock waves at early times. However, the mean density and mean (quasistatic) pressure fields, the primary focus of the method, are once more accurately determined with the MUZO computation.

Various levels of pressure and area have been considered to address a simple building made of two rooms. The present treatment of the Riemann problem involving cross-section variations yields results in good agreement with those provided by a conventional computation, both in terms of pressure relaxation time and quasistatic pressure field. Depending on the geometry, excellent results can be provided in a short CPU time by a conventional computation using a coarse mesh. However such a method requires effort during the pre-processing step because the various openings must be drawn and meshed. Furthermore, depending on the size of the openings, caution must be used as well to create a mesh of sufficient quality. The time needed to produce a complete simulation becomes consequently significant and not acceptable when dealing with pressing situations involving realistic buildings. A practical example is provided in the following. The MUZO method uses a very straightforward and fast meshing procedure. The gain in terms of time and labor to produce a complete simulation is consequently tremendous. As very few numerical elements are used, the CPU time needed to compute the flow solution is also very short. The reported computation times (involving only the flow computation part of the simulation) are provided in Table 1.

Test	Conventional computation Fine mesh CPU time (sequential)	Conventional computation Coarse mesh CPU time (sequential)	MUZO under-resolved computation CPU time (sequential)
1 (Figure 12)	2 hours and 29 minutes	<1 second	<1 second
2 (Figure 13)	2 hours and 42 minutes	<1 second	<1 second
3 (Figure 14)	3 hours and 42 minutes	<1 second	<1 second
4 (Figure 16)	20 hours and 23 minutes	42 seconds	<1 second
5 (Figure 17)	134 hours and 25 minutes	27 minutes	7 seconds
6 (Figure 18)	5 hours and 29 minutes	18 seconds	<1 second
7 (Figure 19)	24 hours and 16 minutes	79 seconds	2 seconds

Table 1: CPU times (involving only the flow computation part of the simulation) reported for the various tests addressing simple geometries. The computations are performed with a sequential implementation.

6.2 Realistic building

To illustrate the capabilities of the MUZO method, the 3D building of Figure 8 is now addressed, in the direction of flow computations in realistic buildings. The conventional computation uses a 3D fine mesh made of about 1 million tetrahedral elements, and a coarse mesh made of 4625 tetrahedral elements. The MUZO under-resolved computation uses a 3D mesh made of 116 prismatic elements. The Godunov first-order scheme (2.10), (5.6.2) is used with CFL $= 0.5$ for all computations (under-resolved and conventional). The building involves 35 openings composed of 18 doors of area 0.75×2 m^2, 16 windows of area 1×1 m^2 and 1 stairwell of area 3 m^2. The full 3D geometry is depicted in Figure 20.

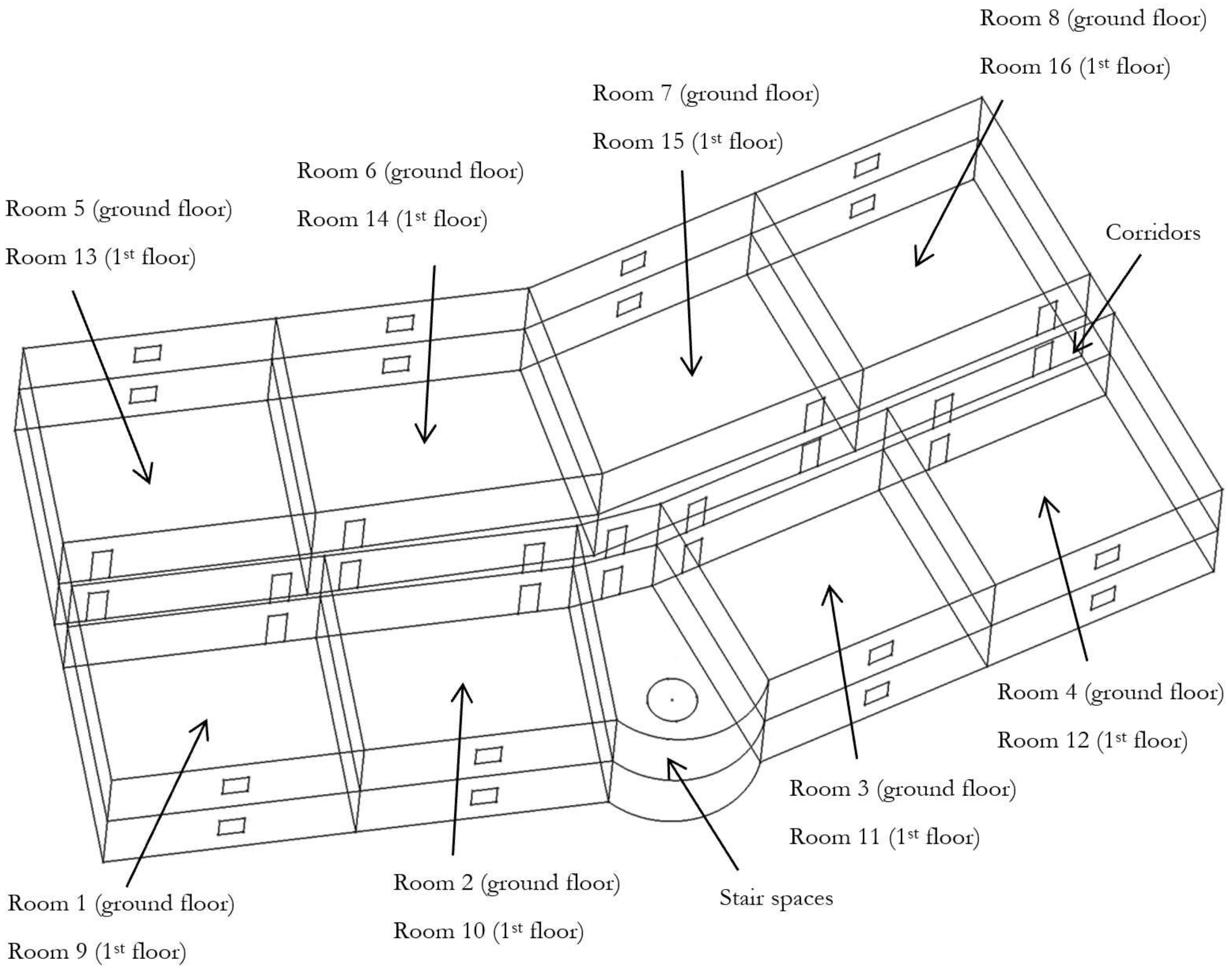

Figure 20: Full 3D geometry of a realistic building made of 16 rooms, 2 stair spaces and 2 corridors. The building involves 35 openings composed of 18 doors of area 1.5 m^2, 16 windows of area 1 m^2 and 1 stairwell of area 3 m^2. The stair space on the ground floor (hall) is initially set with a high pressure (HP) of $p_{\mathrm{HP}} = 10^7$ Pa. The pressure in the rest of the building is $p = 10^5$ Pa. Air is initially at rest $\mathbf{u} = \mathbf{0}$ and at temperature $T = 293$ K in the whole building. Atmospheric conditions are $p_{\mathrm{atm}} = 10^5$ Pa and $T_{\mathrm{atm}} = 289.75$ K. The building is meshed with about 1 million tetrahedral elements. Computation on the full geometry is to be compared to the MUZO under-resolved computation where the openings are not meshed and few elements are used (see Figure 8), and to the conventional computation using a coarse mesh.

Results in terms of mean pressure are provided in the following figures. Mean densities are not presented for the sake of space. Two sets of computation are carried out. The first one considers all the windows closed. The corresponding surfaces are then treated as reflective walls. The second considers all the windows open and connected to the atmosphere. Note that the purpose of the present tests is to compare the results from the MUZO under-resolved computations to the results provided by the conventional computations. More realistic situations, where inner walls get gradually destroyed under the effect of pressure are part of future investigations.

Rooms on the ground floor are first addressed. Figure 21 shows the results for the rooms on the southern side of the building depicted in Figure 20.

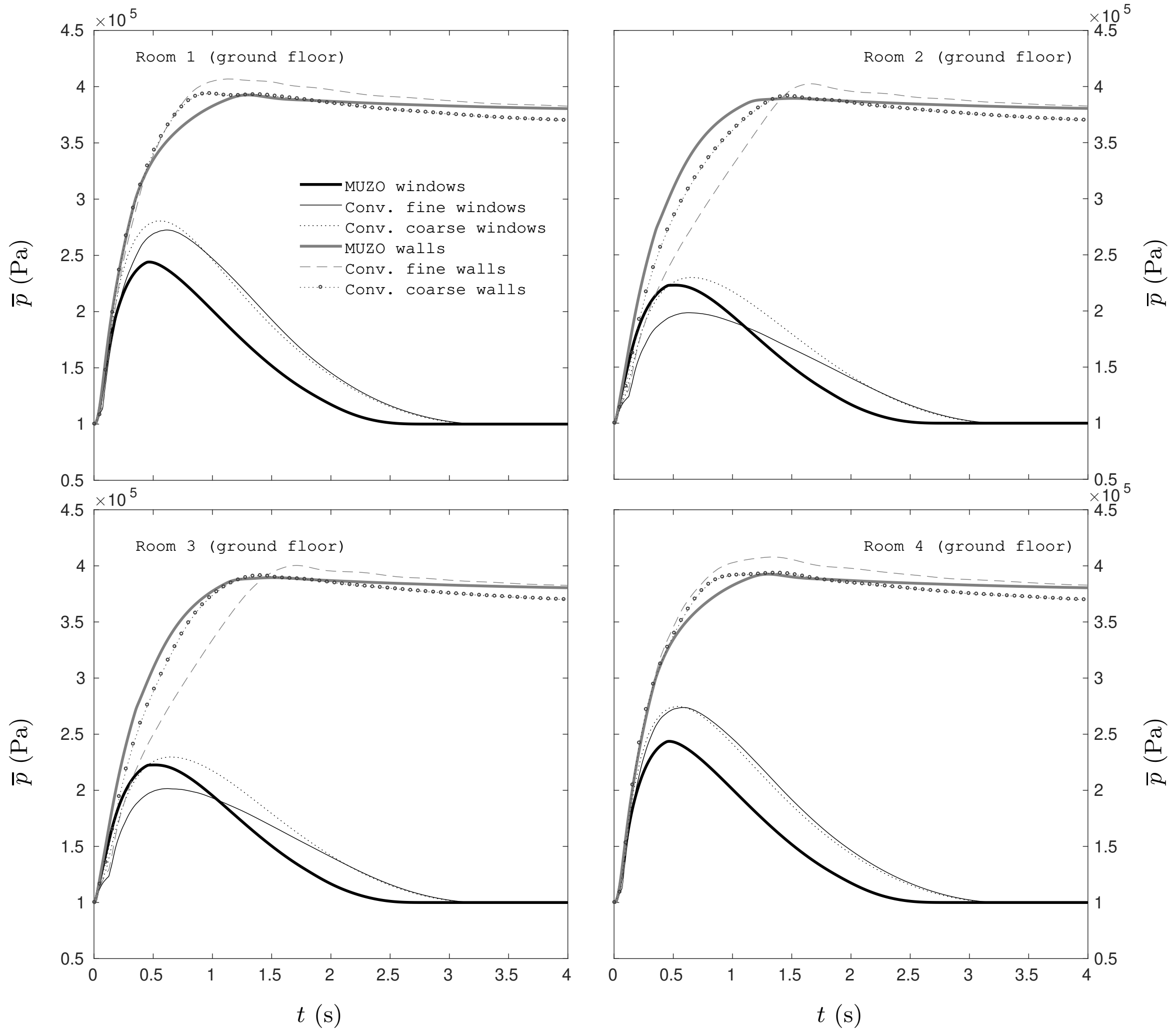

Figure 21: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration is depicted in Figures 8 and 20. It consists of a realistic building made of two floors. Each floor is made of 8 rooms with doors and windows, 1 stair space and 1 corridor. In the hall on the ground floor (stair space, see Figure 20), the pressure $p_{\mathrm{HP}} = 10^7$ Pa is initially set. In the rest of the building, the initial pressure is $p = 10^5$ Pa. Air is initially at rest $\mathbf{u} = \mathbf{0}$ and at temperature $T = 293$ K in the whole building. The atmospheric conditions are $p_{\mathrm{atm}} = 10^5$ Pa and $T_{\mathrm{atm}} = 289.75$ K. The conventional 3D computations (denoted "Conv.") are performed on meshes composed of about 1 million and 4625 tetrahedral elements. The MUZO under-resolved computation is performed on a mesh made of 116 prismatic elements. The mean pressure is plotted both for the closed-window situation (denoted as "walls") and for the open-window situation (denoted as "windows"). The present figure shows the results for the four rooms located on the ground floor, on the southern side of the building (see Figure 20).

Results provided by the MUZO under-resolved computation show a very reasonable agreement with those provided by the conventional computation using the fine mesh (considered as the reference solution), both for the situation where the windows are closed (reflective walls) and for the situation where the windows are open. In this last situation, atmospheric conditions significantly affect the solution. The largest differences appear at the pressure peaks. Let us define the difference (expressed in percentage) related to the pressure peaks between the reference solution and the results provided by the MUZO method,

$$L = 100\left(1 - \frac{p_{\mathrm{max,MUZO}}}{p_{\mathrm{max,Conv.fine}}}\right), \tag{6.2.1}$$

where $p_{\mathrm{max,MUZO}}$ denotes the maximum pressure obtained by the MUZO computation and $p_{\mathrm{max,Conv.fine}}$ the maximum pressure delivered by the conventional computation using the fine mesh, *i.e.* the reference solution. The corresponding differences are reported in Table 2.

Location	Conventional computation Coarse mesh L (%) open $\mid$ closed	MUZO under-resolved computation L (%) open $\mid$ closed
Room 1	$-3.01 \mid +3.18$	$+10.44 \mid +3.49$
Room 2	$-15.9 \mid +2.69$	$-12.36 \mid +3.22$
Room 3	$-14.0 \mid +2.26$	$-10.46 \mid +2.70$
Room 4	$-0.39 \mid +3.53$	$+10.99 \mid +3.75$

Table 2: Differences in the pressure peaks observed in Figure 21 for rooms 1, 2, 3 and 4 located on the ground floor (see Figure 20). The pressure difference L compares the maximum pressure delivered by the MUZO computation or alternatively the conventional computation using the coarse mesh, to the maximum pressure provided by reference solution, *i.e.* the solution of the conventional computation using the fine mesh. For both the open and closed-window computations, the pressure peak corresponds to the maximum pressure of the simulation. The pressure difference L is determined by Eq. (6.2.1) and is expressed in percentage. Results are given for both the open and closed window situations respectively.

As seen in Figure 21 and Table 2, the results provided by the conventional (Conv.) computation using the coarse mesh appear closer to those obtained with the fine mesh than the MUZO results. The differences in terms of pressure peaks are expressed by replacing $p_{\mathrm{max,MUZO}}$ in Eq. (6.2.1) by $p_{\mathrm{max,Conv.coarse}}$ denoting the maximum pressure delivered by the conventional computation using the coarse mesh. Nevertheless, the construction of the geometry required much more time and effort than the MUZO pre-processing step as discussed previously.

Figure 22 shows the results for the rooms on the northern side of the building (see Figure 20).

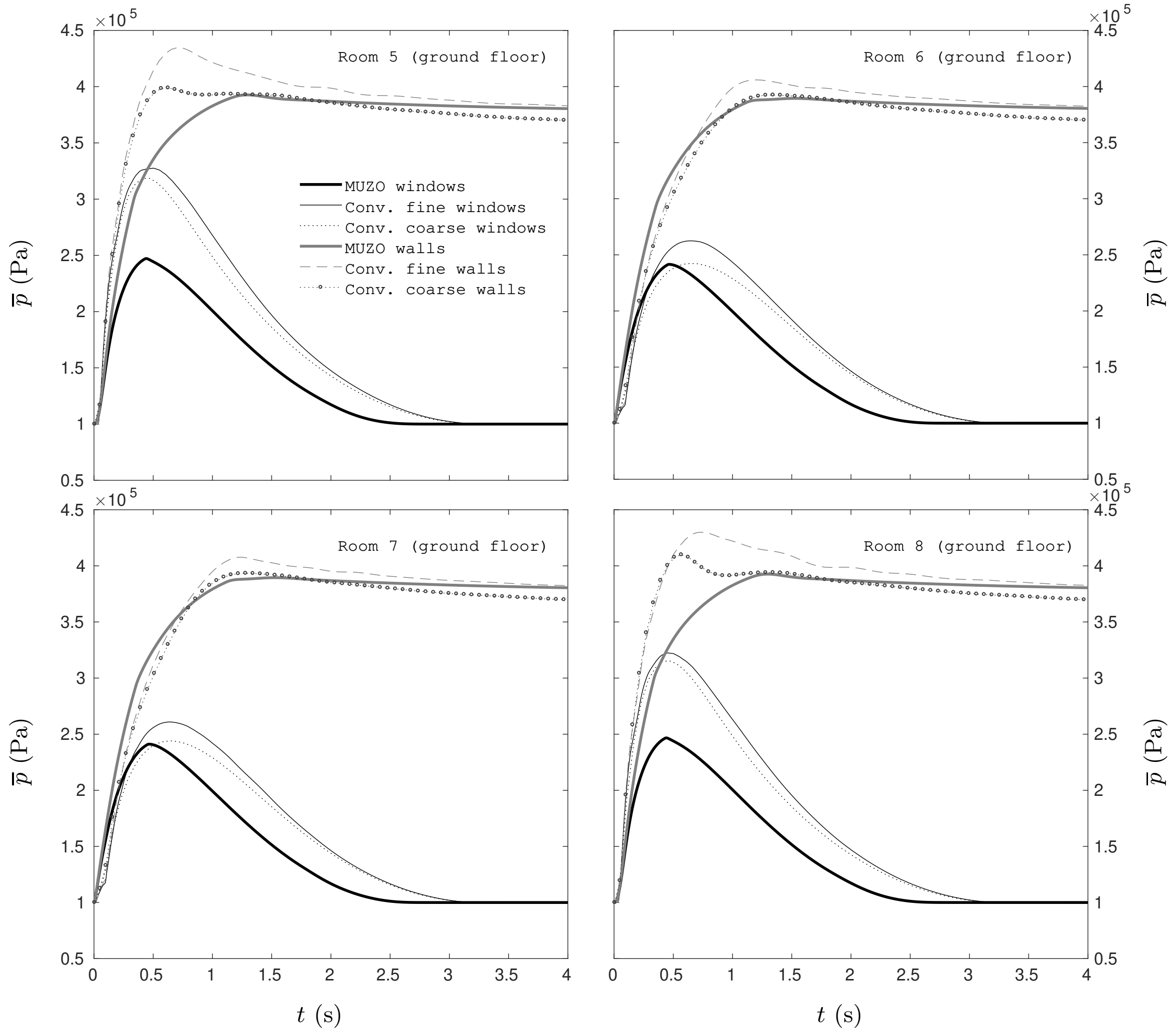

Figure 22: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration consists of the realistic building depicted in Figures 8 and 20. The present figure shows the results for the four rooms located on the ground floor, on the northern side of the building (see Figure 20).

Results provided by the MUZO under-resolved computations appear quite reasonable, although underestimated pressure peaks appear in the plots for rooms 5 and 8. Table 3 reports the differences observed in the pressure peaks.

Location	Conventional computation Coarse mesh L (%) open | closed	MUZO under-resolved computation L (%) open | closed
Room 5	+2.589 | +8.186	+24.52 | +9.645
Room 6	+7.644 | +3.319	+7.980 | +4.060
Room 7	+6.467 | +3.516	+7.577 | +4.462
Room 8	+2.094 | +4.655	+23.48 | +8.672

Table 3: Differences in the pressure peaks observed in Figure 22 for rooms 5, 6, 7 and 8 located on the ground floor (see Figure 20). The pressure difference L compares the maximum pressure delivered by the MUZO computation or alternatively the conventional computation using the coarse mesh, to the maximum pressure provided by reference solution, *i.e.* the solution of the conventional computation using the fine mesh. For both the open and closed-window computations, the pressure peak corresponds to the maximum pressure of the simulation. The pressure difference L is determined by Eq. (6.2.1) and is expressed in percentage. Results are given for both the open and closed window situations respectively.

Rooms on the first floor are now addressed. Figure 23 shows the results for the rooms on the southern side of the building (see Figure 20).

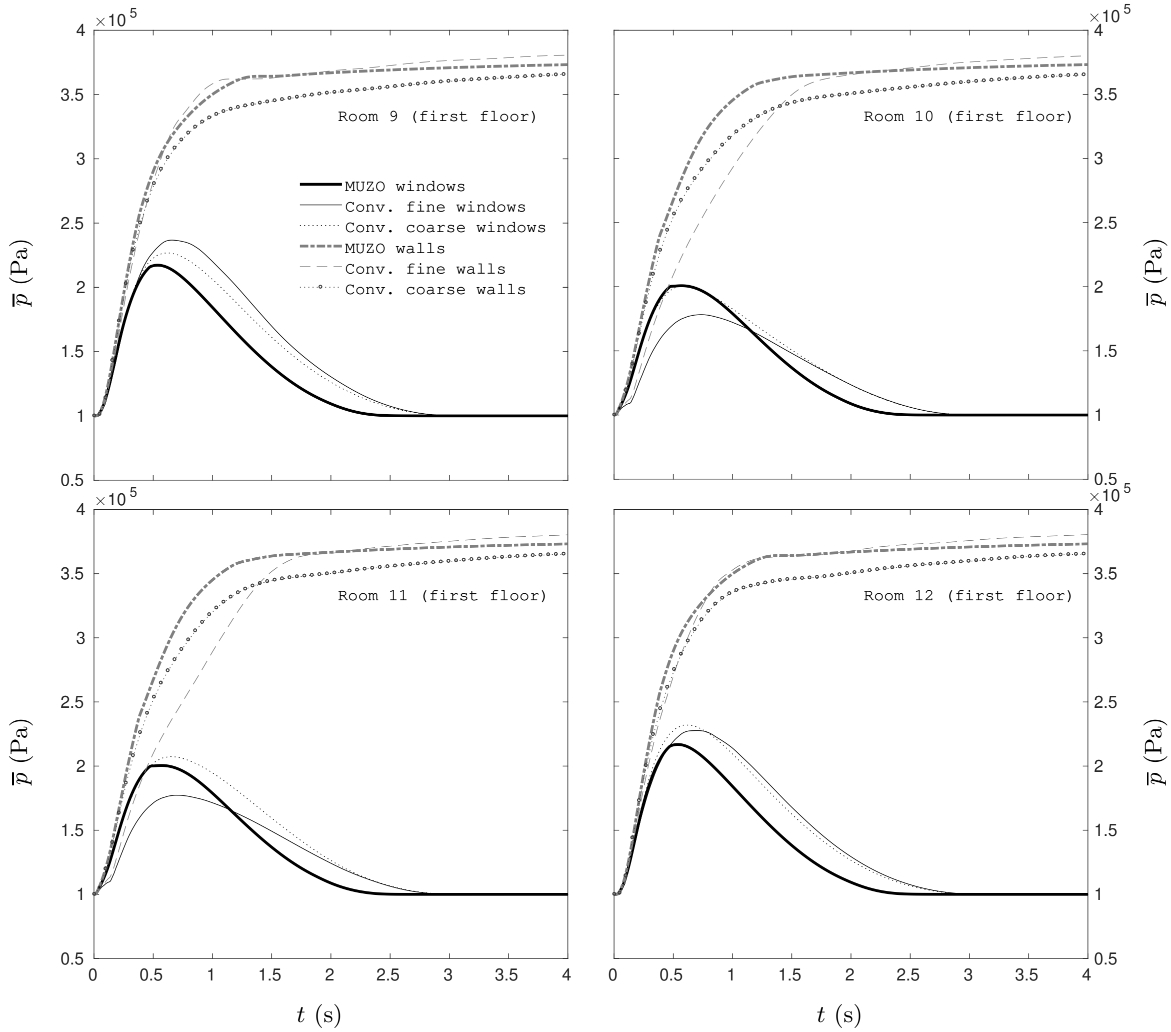

Figure 23: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration consists of the realistic building depicted in Figures 8 and 20. The present figure shows the results for the four rooms located on the first floor, on the southern side of the building (see Figure 20).

Again, a very reasonable agreement is observed between the conventional and the MUZO under-resolved computations, for both the situation where the windows are closed (reflective walls) and the situation where the windows are open (connected to the atmosphere). The differences observed in the pressure peaks are reported in Table 4.

Location	Conventional computation Coarse mesh L (%) open | closed	MUZO under-resolved computation L (%) open | closed
Room 9	$+4.155 \mid +3.931$	$+8.239 \mid +1.934$
Room 10	$-12.59 \mid +3.872$	$-12.64 \mid +1.809$
Room 11	$-17.01 \mid +3.919$	$-13.08 \mid +1.852$
Room 12	$-1.966 \mid +3.959$	$+4.767 \mid +1.900$

Table 4: Differences in the pressure peaks observed in Figure 23 for rooms 9, 10, 11 and 12 located on the first floor (see Figure 20). The pressure difference L compares the maximum pressure delivered by the MUZO computation or alternatively the conventional computation using the coarse mesh, to the maximum pressure provided by reference solution, *i.e.* the solution of the conventional computation using the fine mesh. For the open-window computations, the pressure peak corresponds to the maximum pressure of the simulation. For the closed-window computations, the maximum pressure is reached at the end of the simulation ($t = 4$ s). The pressure difference L is determined by Eq. (6.2.1) and is expressed in percentage. Results are given for both the open and closed window situations respectively.

Figure 24 shows the results for the rooms on the northern side of the building (see Figure 20).

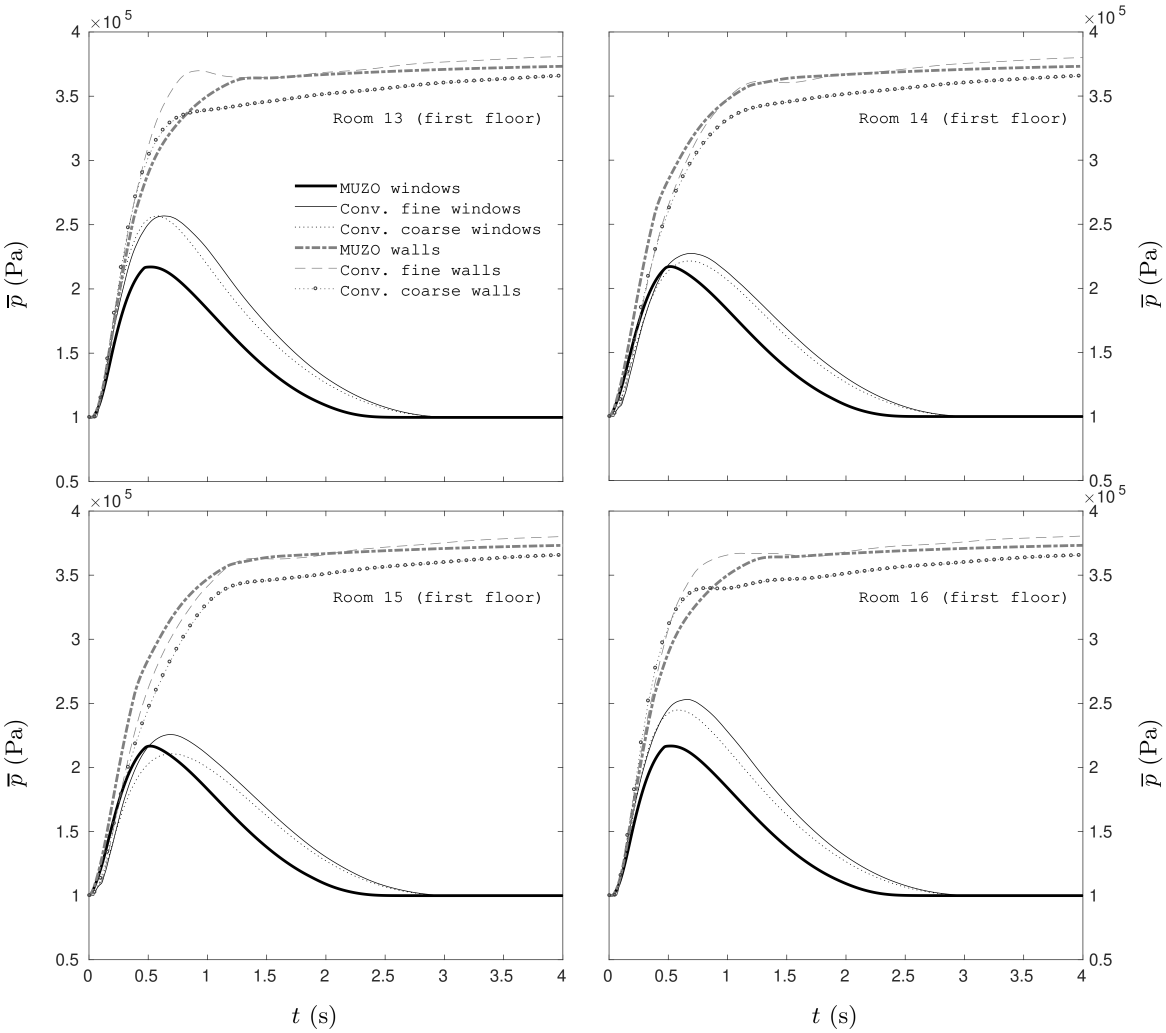

Figure 24: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration consists of the realistic building depicted in Figures 8 and 20. The present figure shows the results for the four rooms located on the first floor, on the northern side of the building (see Figure 20).

One more time, a quite reasonable agreement is observed, although underestimated pressure peaks appear in the plots for rooms 13 and 16, similarly to rooms 5 and 8 (Figure 22). Table 5 reports the differences observed in the pressure peaks.

Location	Conventional computation Coarse mesh L (%) open | closed	MUZO under-resolved computation L (%) open | closed
Room 13	-0.026 | $+3.962$	$+15.46$ | $+1.967$
Room 14	$+2.531$ | $+3.781$	$+4.491$ | $+1.777$
Room 15	$+6.604$ | $+3.859$	$+3.966$ | $+1.816$
Room 16	$+3.156$ | $+3.958$	$+14.34$ | $+1.912$

Table 5: Differences in the pressure peaks observed in Figure 24 for rooms 13, 14, 15 and 16 located on the first floor (see Figure 20). The pressure difference L compares the maximum pressure delivered by the MUZO computation or alternatively the conventional computation using the coarse mesh, to the maximum pressure provided by reference solution, *i.e.* the solution of the conventional computation using the fine mesh. For the open-window computations, the pressure peak corresponds to the maximum pressure of the simulation. For the closed-window computations, the maximum pressure is reached at the end of the simulation ($t = 4$ s). The pressure difference L is determined by Eq. (6.2.1) and is expressed in percentage. Results are given for both the open and closed window situations respectively.

Finally, stair spaces and corridors are addressed. Results are provided in Figure 25.

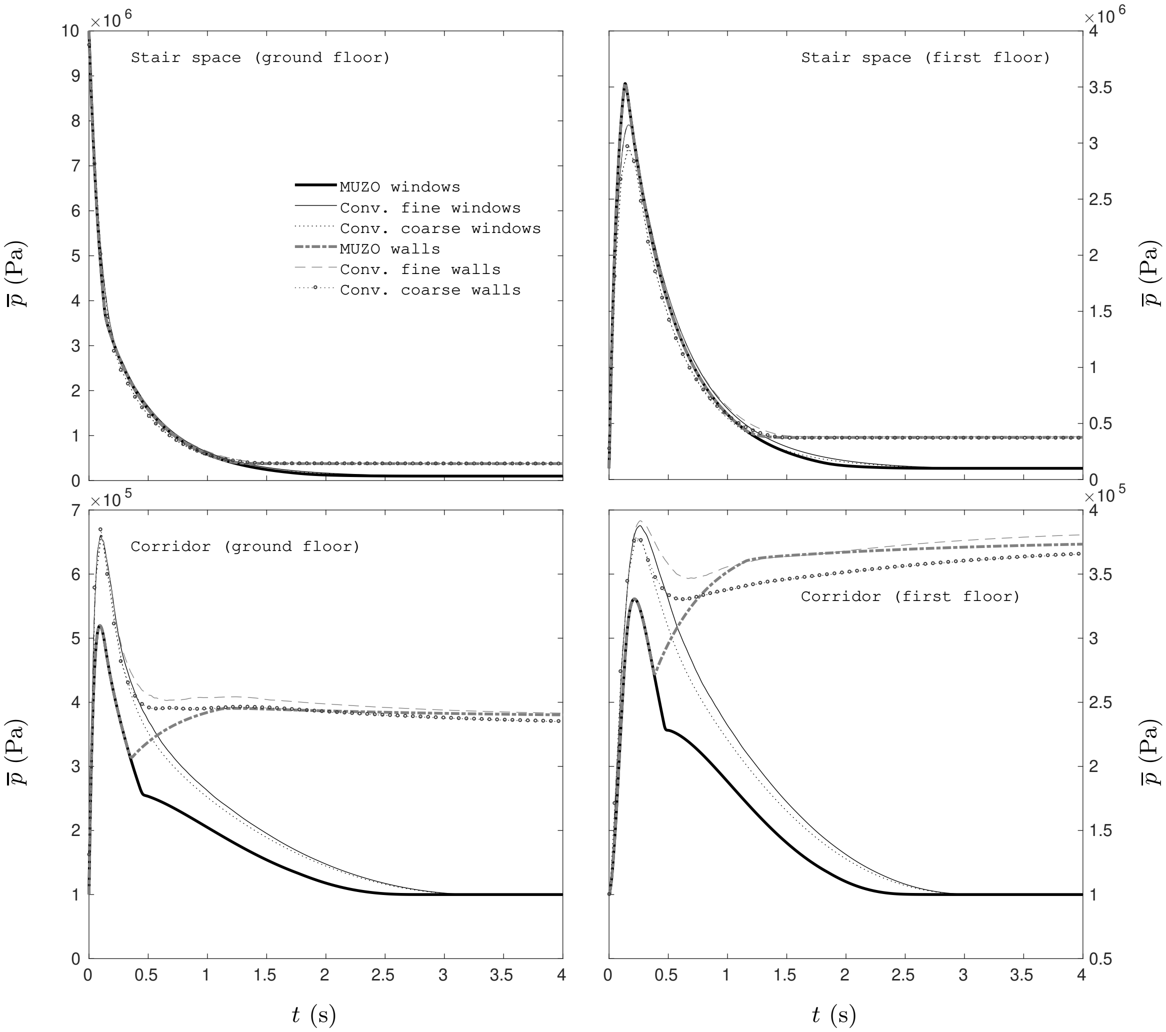

Figure 25: MUZO under-resolved 3D computation versus conventional 3D computations. The test configuration consists of the realistic building depicted in Figures 8 and 20. The present figure shows the results for the stair spaces and the corridors (ground and first floors, see Figure 20). The initial high-pressure zone $p_{\mathrm{HP}} = 10^7$ Pa is in the stair space, on the ground floor.

The high pressure is initially set in the stair space (hall), on the ground floor. Pressure fields in the stair spaces are in good agreement with the pressure profiles provided by the conventional computations. Results in the corridors are quite reasonable as well. The differences observed in the pressure peaks are reported in Table 6. As the lower stair space is initially set with a high pressure of $p_{\mathrm{HP}} = 10^7$ Pa, it is not relevant to evaluate in this zone the pressure differences between the MUZO and reference solutions. As seen in Figure 25, the agreement is excellent both for the open-window situation and for the closed-window one. The agreement is also excellent between the solutions of the conventional computation using the coarse mesh and the reference ones.

Location	Conventional computation Coarse mesh L (%) open | closed	MUZO under-resolved computation L (%) open | closed
Upper stair space	$+6.099 \mid +6.111$	$-11.66 \mid -11.62$
Lower corridor	$-1.558 \mid -1.959$	$+21.26 \mid +21.25$
Upper corridor	$+2.042 \mid +3.198$	$+14.75 \mid +7.916$

Table 6: Differences in the pressure peaks observed in Figure 25 for the lower corridor, the upper corridor and the upper stair space (see Figure 20). The pressure difference L compares the maximum pressure delivered by the MUZO computation or alternatively the conventional computation using the coarse mesh, to the maximum pressure provided by reference solution, *i.e.* the solution of the conventional computation using the fine mesh. The maximum pressure is reached at the peak for both the open-window situation and the closed-window one, with the exception of the upper corridor where the MUZO computation, considering the windows closed, delivers the maximum pressure at $t = 4$ s. The MUZO maximum pressure in the upper corridor is then chosen at $t = 0.2124$ s when the pressure reaches the peak (see Figure 25). However, it is not the maximum pressure of the whole simulation. For the MUZO computation, the maximum pressure is reached at the end of the simulation ($t = 4$ s). The pressure difference L is determined by Eq. (6.2.1) and is expressed in percentage. Results are given for both the open and closed window situations respectively.

The overall results provided by the MUZO under-resolved computations show a very reasonable agreement with the reference solutions, both in terms of pressure relaxation time and mean pressure, although the simplicity and rapidity of the method yield sometimes underestimated pressure peaks. The largest observed difference between the MUZO computation and the conventional computation using the fine mesh is about 25 % and about an average of 10 %. Nevertheless, compared to the conventional method, the construction of the geometry is very simple and fast as only the 2D "footprints" of the building are necessary. A conforming mesh requiring very little pre-processing and as few elements as possible is then constructed by extruding along the third dimension (see Section 4). The geometric openings such as doors, windows, stairwells, are not meshed. They are taken into account directly in the Riemann solver through a simple but specific flux distribution (see Section 5). The MUZO method consequently reduces significantly the time needed to produce a complete simulation. Table 7 presents the CPU times needed to compute the flow solutions.

Test	Conventional computation (fine mesh) CPU time domain decomposition MPI architecture using 63 CPUs	MUZO under-resolved computation CPU time sequential
Realistic building (walls)	4 hours and 46 minutes	8 seconds
Realistic building (windows)	4 hours and 18 minutes	9 seconds

Test	Conventional computation (coarse mesh) CPU time domain decomposition MPI architecture using 35 CPUs	Conventional computation (coarse mesh) CPU time sequential
Realistic building (walls)	37 seconds	13 minutes
Realistic building (windows)	31 seconds	11 minutes

Table 7: CPU times (involving only the flow computation part of the simulation) reported for the computations of the realistic building. The MUZO under-resolved computations are performed with a sequential implementation due to the few numbers of numerical elements. The conventional computations are performed on both fine ($\simeq$ 1 million elements) and coarse (4625 elements) meshes with both parallel and sequential implementations. Parallel computations are run with MPI architecture and 63 CPUs for the fine mesh and 35 CPUs for the rough mesh.

Because of the presence of doors and windows in the geometry, 4625 tetrahedral elements were necessary for the conventional method using a coarse mesh. Compared to the MUZO method (using 116 prismatic cells), more elements were then used and the computation time is quite larger. The conventional computations required about 12 minutes with a sequential implementation. Results appear closer to the reference results provided by the conventional computations using a fine mesh (about 1 million elements). The computation time is reduced to about 35 seconds when a parallel architecture is used with 35 CPUs. Much more computational resource is then required in comparison to the MUZO computations, such an IT-resource dependence being undesirable under pressing circumstances. Furthermore, the construction of the geometry requires much more time and effort than the MUZO pre-processing step as already discussed. The MUZO method then appears very helpful when hazardous and pressing situations are involved and require knowledge of the pressure fields. Another important asset also rises. Indeed, because geometric restrictions are neither drawn nor meshed but are only marked for the specific Riemann solver, it appears straightforward to introduce a time-dependent throat area. Such situations may describe for instance the gradual destruction of a wall under the effect of pressure and will be investigated in future works. The treatment of this situation is not straightforward with the conventional approach where the openings are initially drawn and meshed, yielding consequently a mesh only adapted for the initial time.

7. Conclusion

A fast numerical framework, both on the pre-processing stage and on the solver side, has been developed to compute the mean (or quasistatic) pressure fields in complex structures, like buildings, aircrafts, plane wings or industrial plants to cite a few. Knowledge of those pressure fields is indeed very important in many situations. A relevant example is the overpressure generated by the detonation of an explosive charge in a building. In such a pressing and hazardous situation, knowledge of the mean pressure in the different parts of the structure is needed and is to be determined as quickly as possible, to help first responders evaluate the risk of entering the structure after the explosion for instance. The present work is incentivized by such urgent situations. The proposed method is named MUZO in reference to its "MUlti-ZOne" flow solver.

The construction of the geometry and its corresponding mesh is very simple, fast, and flexible. The meshing tool used in this paper is GMSH. However, the present work is not restricted to this software and one's favorite tool may be used. The authors are currently developing a specific meshing tool to further decrease pre-processing tasks. The MUZO strategy consists of constructing only the "footprints" of the building (nodes, lines, and surfaces), generate a coarse but conformal 2D mesh and then extrude along the third dimension. The design of the geometry consequently requires little effort and a conforming 3D mesh with as few elements as possible is then constructed. Moreover, the mesh is constructed in a particular manner such that simplifications can be made when computing the fluid flux. The geometric details, like doors, windows, staircases, are not required during the geometry-and-mesh-construction step. The pre-processing stage is consequently very fast, compared to the long and tedious meshing process involved with the conventional method, where every geometric detail needs to be drawn and meshed.

With the MUZO approach, such geometric openings are treated through a specific Riemann solver that can handle both unchoked and choked situations occurring through openings. The present Riemann solver is simple and robust. It is based on the following observation. In the limiting case where the cross-sections on both sides of the geometric discontinuity are the same $A_L = A_R$, the geometric discontinuity becomes transparent (for an unchoked flow). Such a geometrical property is easily satisfied from the previous MUZO pre-processing step, where a conformal constrained Delaunay-type mesh is built from given input nodes which ensures $A_L = A_R$ by construction at all room partitions. An essential simplification appears as there is no need to consider 4 waves and 3 states in the unchoked Riemann problem. In this particular case it reduces to 3 waves and 2 states, as done usually with the Euler equations without cross-section variation. The simple and robust HLLC solver can then be used and provides the solution state upstream from the geometric opening. Isentropic and isenthalpic relations, resulting from a quasi-steady assumption, are used afterwards to select the flow regime appropriate to the flow conditions, *i.e.* subsonic or sonic, and provide the solution state at the opening (the throat). Finally, a specific but simple flux distribution is performed.

The MUZO method has been tested on both simple 3D geometries, with various levels of pressure and opening area, and a realistic building. The present test cases focus only on flow discharge effects occurring at long timescales. Blast wave effects occurring at early times are supposed to be determined beforehand through an appropriate method based for example on Kingery-Bullmash data [5]. Results provided by the MUZO computations on simple 3D geometries show a very good agreement with results from the conventional computations. Such a good agreement is obtained both in terms of pressure relaxation time, a direct consequence of the computation of the fluxes, and in terms of mean (or quasistatic) pressure fields. When realistic complex structures are addressed, the quasistatic pressure and relaxation time appear reasonably accurate, making the present method a simple and very fast numerical tool to address flows in complex buildings. It is worth mentioning that using very coarse meshes may yield a loss of accuracy where curved geometries are considered. One way to remedy to this drawback is to use high-order meshes, see for instance Dobrzynski and Jannoun (2017) [33]. This topic is part of future investigations.

Depending on the complexity of the building, and the available computational resource, a conventional computation using a coarse mesh may provide the desired results with a low computation time. However, because geometric details like doors are needed, such a method requires much more effort and time during the pre-processing step. Moreover, small openings involve an additional difficulty. Indeed, those need to be fully meshed with a certain degree of quality for the simulation to be successful. Extra numerical elements are then needed to ensure a local mesh quality. The computation time is then affected as well as the time needed to create the mesh. A special care is indeed required when small openings are present. The time needed to produce a complete simulation becomes significant and not acceptable when dealing with pressing situations. This difficulty is not encountered with the MUZO method that is very convenient in such circumstances as it requires very little pre-processing, minimum computational resource and CPU time.

Another major asset is in favor of the MUZO method. Because geometric restrictions are neither drawn nor meshed but are only marked for the specific Riemann solver, it appears straightforward to introduce a time-dependent throat area, unlike conventional computations. Such situations may describe for instance the gradual destruction of a wall under the effect of pressure and is also part future investigations. Other research directions involve comparing the MUZO method to small and full-scale experiments and extending the method to deal with multiple chemical species and thin solid particles, in order to account for combustion products and dust clouds.

Acknowledgments

This work was partly funded by CEA Gramat. Antoine Osmont, Sébastien Courtiaud and Emmanuel Lapébie are gratefully acknowledged.

Declaration of Competing Interests

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.

Appendix A. Boundary condition for the MUZO method

The present appendix addresses the boundary condition. Such a boundary describes for example a flow occurring through a window, placed on a boundary wall of a building. The MUZO Riemann solver presented in Section 5, taking into account the effects of a dimensional reduction, must be adapted to a boundary cell face. To illustrate the situation, let us once again use the subsonic wave diagram depicted in Figure 4. However, as the cell face separates the numerical domain from the exterior, the waves appearing on the right side of Figure 4 are now multidimensional. Yet, mathematical relations across such waves are unavailable. The states on the right of the opening in Figure 4 are then inaccessible and the Riemann problem situation transforms to the one depicted in Figure A.26, involving only one extreme wave traveling towards the left and the stationary wave.

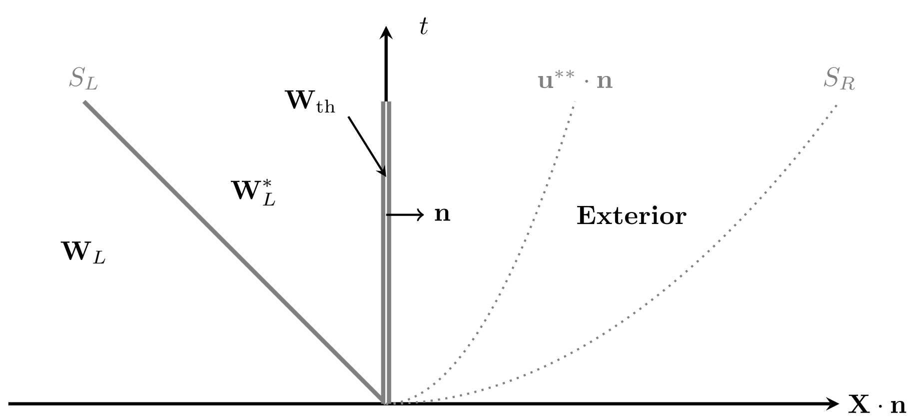

Figure A.26: Schematic representation of the wave diagram of the Riemann problem for a boundary cell face, in the subsonic case. A wall presenting a window (throat) separates a room on the left from the exterior. In the present example, the fluid flows from the left of the opening (numerical domain) to the right (exterior). The geometric discontinuity is indicated with the double line. The waves $\mathbf{u}^{**} \cdot \mathbf{n}$ and S_R travel towards the exterior and are consequently multidimensional (depicted by the dotted curved lines). Mathematical relations across such waves are unavailable. The waves $\mathbf{u}^{**} \cdot \mathbf{n}$ and S_R are here only present for the purpose of illustration. Only one extreme wave propagates into the numerical domain.

Figure A.26 depicts the Riemann problem in a subsonic case. When a sonic case is addressed, the speed of the acoustic wave S_L is zero and a reflected wave is considered and is the only one traveling towards the left. The Riemann problem for a boundary cell face is then specific in the sense that only two waves appear whether or not the flow is choked at the throat (window area). Those waves are the acoustic wave S_L for a subsonic situation or the reflected wave $S_{\text{reflected}}$ for a sonic case, in addition to the stationary wave for both events. The method is presented hereafter based on the flow situation depicted in Figure A.26, involving a fluid flowing from the left of the opening (numerical domain) to the right (exterior). Naturally the method treats the reversed flow situation similarly.

Sonic flow

When the flow is choked, the upstream side (numerical domain) is isolated from the downstream side (exterior) at the throat. The speed of the acoustic wave S_L depicted in Figure A.26 becomes zero and a reflected wave $S_{\text{reflected}}$ is considered as it affects significantly the solution. In such conditions, the Riemann problem is the same as the one presented in Section 5.3 (Figure 9). The corresponding sonic Riemann solver is then directly used and provides the solution state $\mathbf{W}_{\text{th,sonic}}$ at the throat, as well as the solution state $\mathbf{W}^*_{L,\text{sonic}}$, *i.e.* the state behind the reflected wave (or alternatively $\mathbf{W}^*_{R,\text{sonic}}$ if the flow is reversed). Recall that only the flux solution at the cell boundary is needed. The resolution of the full Riemann problem is not necessary. The solution fluxes are computed according to the method presented in Section 5.6.

Subsonic flow

However, when the flow is subsonic, the Riemann solver of Section 5.2 must be adapted. Across the extreme wave S_L, the acoustic approximation is one more time used to simplify the calculations. A relation linking the normal speed $\mathbf{u}^*_L \cdot \mathbf{n}$ to the pressure p^*_L appears:

$$\mathbf{u}^*_L \cdot \mathbf{n} = \frac{p_L - p^*_L + Z_L \mathbf{u}_L \cdot \mathbf{n}}{Z_L}. \tag{A.1}$$

Note that with the present example (Figure A.26), the fluid flows from the left of the opening to the right. The extreme wave S_L then propagates into the unperturbed state $\mathbf{W}_L$. In situations where the flow is reversed, Relation (A.1) transforms to,

$$\mathbf{u}^*_R \cdot \mathbf{n} = -\frac{p_R - p^*_R - Z_R \mathbf{u}_R \cdot \mathbf{n}}{Z_R}, \tag{A.2}$$

as sign "−" shall be used in the acoustic approximation. Moreover, the linearized version of Laplace's law, based on the sound speed definition, is used once again. A relation linking the density ρ^*_L to the pressure p^*_L appears:

$$\rho^*_L = \rho_L + \frac{p^*_L - p_L}{c_L^2}. \tag{A.3}$$

Finally, recall that between the state $\mathbf{W}^*_L$ and the state $\mathbf{W}_{\text{th}}$ at the throat, the flow is assumed stationary (and consequently isentropic and isenthalpic) resulting in the following relations:

$$\rho^*_L A^*_L \mathbf{u}^*_L \cdot \mathbf{n} = \rho_{\text{th}} A_{\text{th}} \mathbf{u}_{\text{th}} \cdot \mathbf{n}, \tag{A.4}$$

$$\frac{p^*_L}{\rho^{*\gamma}_L} = \frac{p_{\text{th}}}{\rho^\gamma_{\text{th}}}, \tag{A.5}$$

$$\frac{\gamma p^*_L}{(\gamma-1)\,\rho^*_L} + \frac{1}{2}\left(\mathbf{u}^*_L \cdot \mathbf{n}\right)^2 = \frac{\gamma p_{\text{th}}}{(\gamma-1)\,\rho_{\text{th}}} + \frac{1}{2}\left(\mathbf{u}_{\text{th}} \cdot \mathbf{n}\right)^2. \tag{A.6}$$

Laplace's law (A.5) and mass conservation (A.4) yield:

$$\rho_{\text{th}} = \rho^*_L \left(\frac{p_{\text{th}}}{p^*_L}\right)^{\frac{1}{\gamma}}, \tag{A.7}$$

$$\mathbf{u}_{\text{th}} \cdot \mathbf{n} = \frac{\rho^*_L\, A_L\, \mathbf{u}^*_L \cdot \mathbf{n}}{\rho_{\text{th}}\, A_{\text{th}}}. \tag{A.8}$$

For a subsonic flow occurring through the throat, it is fairly conceivable to consider that the multidimensional waves, propagating into the exterior, quickly impose the exterior pressure p_{atm}. Such an observation closes the mathematical system as the pressure at the throat is supposed to be equal to the exterior pressure:

$$p_{\mathrm{th}} = p_{\mathrm{atm}}. \tag{A.9}$$

The total specific enthalpy equation (A.6) becomes consequently a function depending only on the pressure p_L^* in the state $\mathbf{W}_L^*$,

$$\frac{\gamma p_L^*}{(\gamma-1)\,\rho_L^*} + \frac{1}{2}\left(\mathbf{u}_L^* \cdot \mathbf{n}\right)^2 = \frac{\gamma p_{\mathrm{th}}}{(\gamma-1)\,\rho_{\mathrm{th}}} + \frac{1}{2}\left(\mathbf{u}_{\mathrm{th}} \cdot \mathbf{n}\right)^2. \tag{A.10}$$

Solution of the boundary Riemann problem, in the subsonic case, is then obtained by solving Relation (A.10) with the help of an iterative method. In the present work, the Newton-Raphson method is used. The initial guess is computed as $p_L^* = \left(p_{\mathrm{atm}} + p_L\right)/2$. The remaining solution variables are determined with the previous relations and the solution fluxes are computed according to the method presented in Section 5.6.

Flow regime

We then have in hand two Riemann solvers, subsonic and sonic, for a boundary cell face. As in Section 5.5, selecting the solver appropriate to the flow conditions is done with the help of the critical state. The proposed method consists of assuming a subsonic (unchoked) flow and assessing the relevance of this assumption by comparing the subsonic solution to critical conditions. The first step is then to solve Relation (A.10). Sometimes, there may be no mathematical solution, indicating that the flow is choked at the throat. In Section 5.5, this situation could be circumvented by reformulating the critical pressure ratio $R_{p,\mathrm{cr}}$ at the throat in the state $\mathbf{W}_L^*$ and with the help of the HLLC solver providing necessarily a mathematical solution state $\mathbf{W}_L^*$, even fictitious. The situation is different for a boundary cell face. Indeed, no HLLC-type solver is available. The flow is then considered as choked if Relation (A.10) presents no solution. Otherwise when a subsonic solution, even fictitious, is available for Relation (A.10) both solution states $\mathbf{W}_L^*$ and $\mathbf{W}_{\mathrm{th}}$ are known. The subsonic assumption is relevant if the following criteria (Section 5.5) are fulfilled,

$$\begin{cases} A_{\mathrm{th}} > A_{\mathrm{cr}}, \\ R_p > R_{p,\mathrm{cr}}, \\ M_{\mathrm{th}} < 1, \\ M_L^* < M_{L,\mathrm{sonic}}^*, \end{cases} \quad \longrightarrow \quad \text{subsonic}, \tag{A.11}$$

with:

$$A_{\mathrm{cr}} = A_L M_L^* \left(\frac{1+\frac{(\gamma-1)}{2}M_L^{*2}}{1+\frac{(\gamma-1)}{2}}\right)^{-\frac{\gamma+1}{2(\gamma-1)}}, \tag{A.12}$$

$$R_p = \left(1+\frac{\gamma-1}{2}M_{\mathrm{th}}^2\right)^{-\frac{\gamma}{\gamma-1}}, \tag{A.13}$$

$$R_{p,\mathrm{cr}} = \left(1+\frac{\gamma-1}{2}\right)^{-\frac{\gamma}{\gamma-1}}. \tag{A.14}$$

$M_{L,\mathrm{sonic}}^* = M_{\max}$ is obtained by solving Relation (5.3.8) with the help of an iterative method. The equation is reminded hereafter:

$$\frac{A_L}{A_{\mathrm{th}}} = \frac{1}{M_{L,\mathrm{sonic}}^*}\left(\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M_{L,\mathrm{sonic}}^{*2}\right)\right)^{\frac{\gamma+1}{2(\gamma-1)}}. \tag{A.15}$$

As the solution $M_{L,\mathrm{sonic}}^*$ is necessarily bounded, (see Section 5.5) between $[0,1[$ or alternatively $M_{R,\mathrm{sonic}}^* \in]-1,0]$ if the flow is reversed, the bisection method is preferred over the Newton-Raphson procedure.

Appendix B. Boundary condition for the conventional method

With the conventional method, the stationary wave (geometric discontinuity) depicted in Figure A.26 is absent. Consequently, there is no involvement of the stationary, isentropic, and isenthalpic relations (A.4), (A.5), (A.6) and the solution fluxes are computed with the help of the solution state $\mathbf{W}_L^*$ as $\mathbf{F}_L^*(\mathbf{W}_L^*)$ (for a subsonic flow), or alternatively $\mathbf{F}_R^*(\mathbf{W}_R^*)$ if the flow is reversed. The absence of the stationary, isentropic, and isenthalpic relations also implies the absence of the critical area A_{cr} (Relation (A.12)) and the Mach number $M^*_{L,\mathrm{sonic}}$ (Relation (A.15)), those being derived from the combination of the stationary, isentropic, and isenthalpic relations. This corresponds to the main difference between the conventional and MUZO methods, in addition to the treatment of the supersonic case. Numerical treatment of the boundary condition with the conventional method is presented according to the flow situation of Figure A.26, in the absence of the stationary wave (geometric discontinuity).

Subsonic and supersonic flow

When $\mathbf{u}_L \cdot \mathbf{n} > c_L$, *i.e.* $S_L > 0$ (according to Davis' estimates (5.2)), the flow is supersonic and the solution state is $\mathbf{W}_L$. Otherwise, the flow is assumed subsonic. As previously, the solution pressure is assumed to be the exterior pressure, $p_L^* = p_{\mathrm{atm}}$. The acoustic approximations (A.1) and (A.3), across the left-facing acoustic wave S_L then provide the solution speed $\mathbf{u}_L^* \cdot \mathbf{n}$ and the solution density ρ_L^*. The Mach number can then be computed with the help of the equation of state, $M_L^* = \mathbf{u}_L^* \cdot \mathbf{n} / c_L^*(p_L^*, \rho_L^*)$. The relevance of the subsonic solution is then assessed by comparing the Mach number to unity. If $M_L^* < 1$ the subsonic solution is valid and the solution fluxes are computed as $\mathbf{F}_L^*(\mathbf{W}_L^*)$. Otherwise, the sonic solution is considered.

Sonic flow

Unlike the MUZO method, no reflected wave appears because the boundary faces fit the window. In the present example, the fluid flows from the left to the right. The wave S_L is then a rarefaction fan that involves a head speed S_{head} and a tail speed S_{tail}, when entirely envisioned. However, when the flow is sonic at the boundary, *i.e.* at the position where the fluxes are to be computed, the speed S of a particular beam of the rarefaction is zero, as schematically depicted in Figure B.27. The transition to the sonic condition then appears inside the rarefaction.

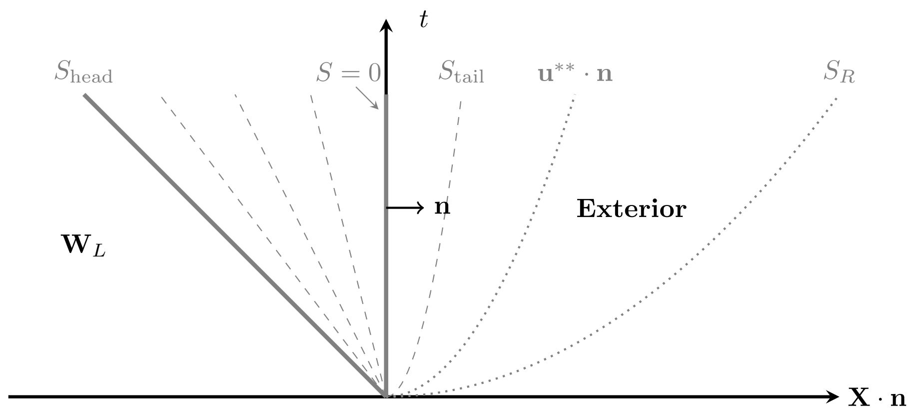

Figure B.27: Schematic representation of the wave diagram of the Riemann problem for a boundary cell face, in the sonic case for the conventional method. As the boundary faces fit the window, no reflected wave is present. The stationary wave (geometric discontinuity) is not present either because the Euler equations are considered without the cross-section term. In the present situation, the flow is positive and sonic at the boundary (axis t). The boundary separates the numerical domain on the left from the exterior on the right. The fluid flows from the left to the right. The left-facing wave is a rarefaction fan that is here entirely envisioned. The head of the rarefaction S_{head} travels towards the numerical domain. Nevertheless, as the flow is sonic, the speed S of a particular beam of the rarefaction is zero, at the position where the fluxes are to be computed. The flow between the unperturbed state $\mathbf{W}_L$ and the rarefaction fan (dashed lines) satisfies the Riemann invariants. However, in this work, those are replaced by the acoustic relations for the sake of simplicity. The waves $\mathbf{u}^{**} \cdot \mathbf{n}$ and S_R travel towards the exterior and are consequently multidimensional (depicted by the dotted curved lines). Mathematical relations across such waves are unavailable.

The flow between the unperturbed state $\mathbf{W}_L$ and the rarefaction fan satisfies the Riemann invariants. However, in this work, those are replaced by the acoustic approximations (A.1) and (A.3) for the sake of simplicity. The speed $\mathbf{u}^*_{\mathrm{sonic}} \cdot \mathbf{n}$ and the density ρ^*_{sonic} depend then only on the solution pressure p^*_{sonic}. Consequently the sound speed depends only on the pressure as well $c^*_{\text{sonic}}\left(p^*_{\text{sonic}}\right)$. The sonic solution at the boundary is then found by solving the equation $\left(\mathbf{u}^*_{\text{sonic}} \cdot \mathbf{n}\right) = c^*_{\text{sonic}}$. An iterative method is necessary. In this work the Newton-Raphson root-finding procedure is used with $p^*_{\text{sonic}} = \left(p_{\text{atm}} + p_L\right)/2$ as the initial guess. Solution fluxes are computed with the help of the $\mathbf{W}^*_{\text{sonic}}$ solution state as $\mathbf{F}^*_{\text{sonic}}\left(\mathbf{W}^*_{\text{sonic}}\right)$.

References

[1] W. Henshaw, N. Smyth, D. Schwendeman, Numerical shock propagation using geometrical shock dynamics, Journal of Fluid Mechanics 171 (1986) 519–545. doi:https://doi.org/10.1017/S0022112086001568.

[2] D. Schwendeman, A new numerical method for shock wave propagation based on geometrical shock dynamics, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 441 (1912) (1993) 331–341. doi:https://doi.org/10.1098/rspa.1993.0064.

[3] G. Whitham, Linear and nonlinear waves, Vol. 42, John Wiley & Sons, 2011.
URL https://books.google.fr/books?id=84Pulkf-Oa8C

[4] J. Ridoux, N. Lardjane, L. Monasse, F. Coulouvrat, Comparison of geometrical shock dynamics and kinematic models for shock-wave propagation, Shock Waves 28 (2) (2018) 401–416. doi:https://doi.org/10.1007/s00193-017-0748-2.

[5] C. Kingery, Air blast parameters versus distance for hemispherical TNT surface bursts, Tech. rep., Army Ballistic Research Lab Aberdeen Proving Ground MD (1966).
URL https://apps.dtic.mil/sti/citations/AD0811673

[6] G. Coulter, G. Bulmash, C. Kingery, Simulation techniques for the prediction of blast from underground munitions storage facilities, Tech. rep., Army Ballistic Research Lab Aberdeen Proving Ground MD (1988).
URL https://apps.dtic.mil/sti/citations/ADA196062

[7] V. Karlos, G. Solomos, M. Larcher, Analysis of the blast wave decay coefficient using the Kingery–Bulmash data, International Journal of Protective Structures 7 (3) (2016) 409–429. doi:https://doi.org/10.1177/2041419616659572.

[8] S. Frank, R. Frank, J. Hurley, Fast-Running Model for Arbitrary Room Air-blast, in: Proceedings of the International Symposium for the Interaction of the Effects of Munitions on Structures (ISIEMS 12.1), Orlando, FL, 2007.

[9] E. Lapébie, R. Soulié, L. Youinou, Flash: Fast Lethality Assessment for Structures and Humans, Tech. rep. (2016).
URL https://mabs.tcnet.ch/data/documents/24-104.pdf

[10] G. Ruscade, Etude de la propagation des ondes de choc dans une double chambre, Ph.D. thesis, INSA Val de Loire, France (2021).

[11] E. Dijkstra, A note on two problems in connexion with graphs, Numerische mathematik 1 (1) (1959) 269–271. doi:https://doi.org/10.1007/BF01386390.

[12] P. LeFloch, M. Thanh, The Riemann problem for fluid flows in a nozzle with discontinuous cross-section, Communications in Mathematical Sciences 1 (4) (2003) 763–797. doi:https://dx.doi.org/10.4310/CMS.2003.v1.n4.a6.

[13] G. Warnecke, N. Andrianov, On the solution to the Riemann problem for the compressible duct flow, SIAM Journal on Applied Mathematics 64 (3) (2004) 878–901. doi:https://doi.org/10.1137/S0036139903424230.

[14] D. Kröner, M. Thanh, Numerical solutions to compressible flows in a nozzle with variable cross-section, SIAM Journal on Numerical Analysis 43 (2) (2005) 796–824. doi:https://doi.org/10.1137/040607460.

[15] M. Thanh, The Riemann problem for a nonisentropic fluid in a nozzle with discontinuous cross-sectional area, SIAM Journal on Applied Mathematics 69 (6) (2009) 1501–1519. doi:https://doi.org/10.1137/080724095.

[16] E. Han, M. Hantke, G. Warnecke, Exact Riemann solutions to compressible Euler equations in ducts with discontinuous cross-section, Journal of Hyperbolic Differential Equations 9 (03) (2012) 403–449. doi:https://doi.org/10.1142/S0219891612500130.

[17] J. Keith, D. Crowl, Estimating sonic gas flow rates in pipelines, Journal of loss prevention in the process industries 18 (2) (2005) 55–62. doi:https://doi.org/10.1016/j.jlp.2004.12.004.

[18] S. Ke, H. Ti, Transient analysis of isothermal gas flow in pipeline network, Chemical Engineering Journal 76 (2) (2000) 169–177. doi:https://doi.org/10.1016/S1385-8947(99)00122-9.

[19] J. Brouwer, I. Gasser, M. Herty, Gas pipeline models revisited: model hierarchies, nonisothermal models, and simulations of networks, Multiscale Modeling & Simulation 9 (2) (2011) 601–623. doi:https://doi.org/10.1137/100813580.

[20] S. Godunov, A finite difference scheme for numerical computation of the discontinuous wave solutions of equations of fluid dynamics, Math. Sb. 47 (1959) 271–306.
URL https://ci.nii.ac.jp/naid/10003581711/en

[21] E. Toro, M. Spruce, W. Spears, Restoration of the contact surface in the HLL-Riemann solver, Shock Waves 4 (1) (1994) 25–34. doi:https://doi.org/10.1007/BF01414629.

[22] A. Chiapolino, R. Saurel, B. Nkonga, Sharpening diffuse interfaces with compressible fluids on unstructured meshes, Journal of Computational Physics 340 (2017) 389–417. doi:https://doi.org/10.1016/j.jcp.2017.03.042.

[23] A. Chiapolino, F. Fraysse, R. Saurel, A Method to Solve Hamilton-Jacobi Type Equation on Unstructured Meshes, Journal of Scientific Computing 88 (1) (2021) 1–43. doi:https://doi.org/10.1007/s10915-021-01517-9.

[24] A. Harten, P. Lax, B. van Leer, On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws, SIAM Review 25 (1) (1983) 35–61. doi:https://doi.org/10.1137/1025002.

[25] E. Toro, Riemann solvers and numerical methods for fluid dynamics: A practical introduction, Springer Science & Business Media, 1997. doi:10.1007/b79761.

[26] S. Davis, Simplified second-order Godunov-type methods, SIAM Journal on Scientific and Statistical Computing 9 (3) (1988) 445–473. doi:https://doi.org/10.1137/0909030.

[27] S. Le Martelot, R. Saurel, B. Nkonga, Towards the direct numerical simulation of nucleate boiling flows, International Journal of Multiphase Flow 66 (2014) 62–78. doi:https://dx.doi.org/10.1016/j.ijmultiphaseflow.2014.06.010.

[28] F. Fraysse, R. Saurel, Automatic Differentiation using Operator Overloading (ADOO) for implicit resolution of hyperbolic single phase and two-phase flow models, Journal of Computational Physics 399 (2019) 108942. doi:https://doi.org/10.1016/j.jcp.2019.108942.

[29] D. Furfaro, R. Saurel, L. David, F. Beauchamp, Towards sodium combustion modelling with liquid water, Journal of Computational Physics 403 (2020) 109060. doi:https://doi.org/10.1016/j.jcp.2019.109060.

[30] Q. Carmouze, R. Saurel, A. Chiapolino, E. Lapébie, Riemann solver with internal reconstruction (RSIR) for compressible single-phase and non-equilibrium two-phase flows, Journal of Computational Physics 408 (2020) 109176. doi:https://doi.org/10.1016/j.jcp.2019.109176.

[31] A. Chiapolino, R. Saurel, E. Toro, Investigation of Riemann solver with internal reconstruction (RSIR) for the Euler equations, Communications in Computational Physics 29 (4) (2021) 1059–1094. doi:https://doi.org/10.4208/cicp.OA-2020-0083.

[32] C. Geuzaine, J. Remacle, Gmsh: A 3D finite element mesh generator with built-in pre-and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (11) (2009) 1309–1331. doi:https://doi.org/10.1002/nme.2579.

[33] C. Dobrzynski, G. El Jannoun, High order mesh untangling for complex curved geometries, Tech. rep., RR-9120, INRIA Bordeaux, équipe CARDAMOM (2017).
URL https://hal.inria.fr/hal-01632388

[34] A. Chapman, W. Walker, Introductory gas dynamics, Holt, Rinehart and Winston, 1971.