

COMPUTATIONAL SIMULATION OF IN-CYLINDER FLOWS IN INTERNAL COMBUSTION ENGINES BY MEANS OF THE COUPLING OF ZERO-/ONE-DIMENSIONAL AND MULTIDIMENSIONAL CODES

Ezequiel J. López^a and Norberto M. Nigro^b

^a*Dpto. de Mecánica Aplicada, Facultad de Ingeniería, Universidad Nacional del Comahue, CONICET, Buenos Aires 1400, 8300 Neuquén, Argentina, ezequiel.lopez@fain.uncoma.edu.ar*

^b*Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC), INTEC-CONICET, Universidad Nacional del Litoral, Güemes 3450, 3000 Santa Fe, Argentina, nnigro@santafe-conicet.gov.ar, <http://www.cimec.org.ar>*

Keywords: In-cylinder flows, Internal combustion engine simulation, Code coupling, Geometrical multiscale model.

Abstract. The computational simulation applied to internal combustion engines is a powerful tool useful for both designing and improving performance. Multidimensional CFD (Computational Fluid Dynamics) codes allow a detailed computation of the flow in the various components of an internal combustion engine, being able to assess the impact of the geometry and the operating conditions. However, due to the high computational cost, it is unviable to simulate all the engine components simultaneously. On the other hand, the imposition of the boundary conditions in the multidimensional (multi-D) model is not a simple task, because an engine is a dynamic system. In other words, it is necessary to account for the influence of the rest of the system on the part being solved with the multi-D model. Currently, a typical approach is to simulate a specific part of an engine (as, for instance, the combustion chamber, the intake and exhaust ports, etc.) with a CFD multi-D code and the rest of the machine with a 0D/1D (zero-dimensional/one-dimensional) engine simulator. Thus, the 0D/1D code provides appropriate boundary conditions for the multidimensional computation. This approach is known as Geometrical Multiscale method and allows a substantial reduction of the numerical complexity. The use of this method leads to the need to couple properly dimensionally heterogeneous models, both in the mathematical formulation as in the computational implementation. This paper presents some preliminary results of the coupling of CFD 0D/1D and multi-D codes for the realization of in-cylinder flow computations in internal combustion engines. Generally, there are several orders of magnitude of difference between the computational cost of 0D/1D and multi-D models, the latter being the most costly. The implementation proposed in this paper to solve the coupling between codes is based on the difference in computational costs and in the possibility to modify the source codes. Multi-D models are solved using a stabilized finite element method with an implicit scheme for temporal integration, and the 0D/1D code uses an explicit finite volume method. A subcycling-time-stepping strategy is applied in order to synchronize the simulation in time. Results corresponding to a virtual flowmetry performed on a motored opposed-piston internal combustion engine are presented.

1 INTRODUCTION

Some complex systems that are of interest in engineering and scientific research require for their modeling to consider all its parts or components. While it would make sense only the detailed resolution of some of these parts, generally it is not possible to ignore the influence of the rest of the system on them in order to obtain reliable results. Such systems may include the human cardiovascular system, internal combustion engines, flow in pipe networks, etc. Simplified models for this kind of systems have been proposed in order to get a tractable computational model. These models are typically based on systems of ordinary differential equations (0D –zero-dimensional– models) and/or one-dimensional differential problems (1D –one-dimensional– models). Due to their simplicity these models are able to simulate the entire system, although the level of detail that can be obtained is relatively low. Multi-dimensional (multi-D) models allow a detailed computation of flow variables, but their computational cost makes them impractical to solve all components of a complex system. A typical approach to circumvent this drawback is to use the Geometrical Multiscale method, which is massively applied in the simulation of the human circulatory system (Formaggia et al., 1999, 2001; Moura, 2007). This method consist in apply a multi-D approximation to some component of the system and 0D/1D models for the remaining parts. The simplified 0D/1D models could be viewed as generators of boundary conditions for the multi-D model.

In particular, we are interested in the simulation of internal combustion engines. In these problems, the flow of the working fluid is compressible, turbulent, and inherently unsteady. This fluid is chemically reactive and multiphase in the case of engines feeded via fuel injection. Very few strategies have been proposed for the resolution of 1D/multi-D coupling in internal combustion engine simulation (Onorati et al., 2000; Bella et al., 2006; Montenegro et al., 2007), where the coupled codes apply a Finite Volume Method for the discretization of the governing equations and the coupling strategy is based on the use of ghost cells. In this work, we apply the Finite Element Method (FEM) for the resolution of the multi-D model. A great amount of work was done to solve dimensionally heterogeneous models for incompressible flows in a FEM context. See, for instance, the works by Formaggia et al. (2002), Blanco et al. (2007), Formaggia et al. (2008) and Leiva et al. (2009). Although the strategies proposed in these works could be adapted for their application to the resolution of compressible flows, in the present work we use absorbing boundary conditions in order to solve the domain coupling. At the computational implementation level, the coupling is done in a loose manner between two independent codes.

The paper is organized as follows. Firstly, we present the governing equations for the multi-D model and the numerical techniques applied. Secondly, zero- and one-dimensional models are summarized. Then, we present the coupling strategy and the proposed algorithm. Finally, some results of the simulation of a motored opposed-piston engine are included. The work ends with some conclusions.

2 THE MULTI-D MODEL

2.1 Governing equations

Let $\Omega_t \subset \mathbb{R}^{n_d}$ and $(0, t_f)$ be the spatial and temporal domains, respectively, where n_d is the number of space dimensions, and let Γ_t denote the boundary of Ω_t . The spatial and temporal coordinates are denoted by \mathbf{x} and t , respectively. In order to account for the domain deformation, the Arbitrary Lagrangian Eulerian (ALE) description is applied. In the ALE approach, two configurations of the system are considered: an instantaneous configuration $\Omega_t(\mathbf{x})$ and a

reference configuration $\Omega_0(\boldsymbol{\xi})$. Therefore, a mapping function between $\Omega_t(\mathbf{x})$ and $\Omega_0(\boldsymbol{\xi})$ is defined as $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$. Using the ALE strategy proposed by Donea et al. (1982), the Navier-Stokes equations governing the fluid flow in conservation form are

$$\begin{aligned} \frac{1}{J} \frac{\partial(J\rho)}{\partial t} + \nabla \cdot [\rho(\mathbf{u} - \mathbf{w})] &= 0 \quad \text{on } \Omega_t \times (0, t_f) \\ \frac{1}{J} \frac{\partial(J\rho\mathbf{u})}{\partial t} + \nabla \cdot [\rho\mathbf{u}(\mathbf{u} - \mathbf{w})] + \nabla p - \nabla \cdot \mathbf{T} &= \mathbf{0} \quad \text{on } \Omega_t \times (0, t_f) \\ \frac{1}{J} \frac{\partial(J\rho E)}{\partial t} + \nabla \cdot [\rho E(\mathbf{u} - \mathbf{w})] + \nabla \cdot (p\mathbf{u}) - \nabla \cdot (\mathbf{T}\mathbf{u}) + \nabla \cdot \mathbf{q} &= 0 \quad \text{on } \Omega_t \times (0, t_f) \end{aligned} \quad (1)$$

where ρ , \mathbf{u} , p , \mathbf{T} , E and \mathbf{q} are the density, fluid velocity, pressure, viscous stress tensor, total energy per unit mass, and heat flux vector, respectively; $J = \det \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right)$ and $\mathbf{w} = \frac{d\mathbf{x}}{dt} \Big|_{\boldsymbol{\xi}}$.

In addition, a perfect gas constitutive relation and a Newtonian fluid defined by the two viscosity coefficients λ and μ are assumed. Thus, the viscous stress tensor is defined as

$$\mathbf{T} = \mu((\nabla\mathbf{u}) + (\nabla\mathbf{u})^T) + \lambda(\nabla \cdot \mathbf{u})\mathbf{I} \quad (2)$$

\mathbf{I} being the second order identity tensor and superscript T denoting transpose. Furthermore, it is assumed that $3\lambda + 2\mu = 0$. Pressure is related to the other variables via the equation of state. For ideal gases, this equation has the form

$$p = (\gamma - 1)\rho e \quad (3)$$

where γ is the ratio of specific heats of the fluid at constant pressure (c_p) and at constant volume (c_v), and e is the internal energy per unit mass which is related to the total energy per unit mass and kinetic energy as

$$e = E - \frac{1}{2}\|\mathbf{u}\|^2 \quad (4)$$

where $\|\cdot\|$ is the standard euclidean norm for vectors. The heat flux vector is defined as

$$\mathbf{q} = -\kappa\nabla T \quad (5)$$

where κ is the heat conductivity and T is the temperature.

The governing equations (1) can be written in compact form as (Hirsch, 1990)

$$\frac{1}{J} \frac{\partial(J\mathbf{U})}{\partial t} + \frac{\partial}{\partial x_i} (\mathbf{F}_i^a - w_i\mathbf{U}) = \frac{\partial \mathbf{F}_i^d}{\partial x_i} \quad \text{on } \Omega_t \times (0, t_f), \quad i = 1, \dots, n_d \quad (6)$$

where $\mathbf{U} = [\rho, \rho\mathbf{u}, \rho E]^T$ is the vector of conservative variables, \mathbf{F}^a and \mathbf{F}^d are the advective and viscous flux vectors respectively, defined as

$$\mathbf{F}_i^a = \begin{bmatrix} \rho u_i \\ \rho u_1 u_i + \delta_{i1} p \\ \vdots \\ \rho u_{n_d} u_i + \delta_{in_d} p \\ (\rho E + p) u_i \end{bmatrix}, \quad \mathbf{F}_i^d = \begin{bmatrix} 0 \\ T_{i1} \\ \vdots \\ T_{in_d} \\ T_{ik} u_k - q_i \end{bmatrix} \quad (7)$$

Here, u_i and q_i are the components of the velocity and heat flux vectors, respectively, T_{ik} are the components of the viscous stress tensor, and δ_{ij} is the Kronecker delta.

In the quasi-linear form, equation (6) is written as (Hirsch, 1990)

$$\frac{\partial \mathbf{U}}{\partial t} + (\mathbf{A}_i - w_i \mathbf{I}) \frac{\partial \mathbf{U}}{\partial x_i} = \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} \right) \quad \text{on } \Omega_t \times (0, t_f), \quad i, j = 1, \dots, n_d \quad (8)$$

where

$$\mathbf{A}_i = \frac{\partial \mathbf{F}_i^a}{\partial \mathbf{U}} \quad (9)$$

is the advective jacobian matrix, and \mathbf{K}_{ij} is the diffusivity matrix satisfying

$$\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} = \mathbf{F}_i^d \quad (10)$$

The system of governing equations (6) (or (8)) is completed by the initial and boundary conditions. Regarding the boundary conditions, we assume that the whole boundary Γ_t admits the decomposition $\Gamma_t = \overline{\Gamma_g \cup \Gamma_h \cup \Gamma_f}$ such that $\overline{\Gamma_g \cap \Gamma_h \cap \Gamma_f} = \emptyset$, where Γ_g , Γ_h and Γ_f represent, respectively, the portions of Γ_t on which Dirichlet, Neumann and inlet/outlet type conditions are imposed. We make the distinction between inlet/outlet and Γ_g boundaries since, in this work, absorbing boundary conditions are imposed on the former. The treatment of absorbing boundary conditions is presented in section §2.3 below. Dirichlet and Neumann boundary conditions are written in the general form

$$\begin{cases} \mathbf{U} = \mathbf{g} & \text{on } \Gamma_g \\ \mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} n_i = \mathbf{f} & \text{on } \Gamma_h \end{cases} \quad (11)$$

where n_i are the components of the unit outward normal vector to Γ_h denoted by \mathbf{n} .

In the sequel, with the aim to simplify the notation, we will drop the sub-index t from the symbol representing the problem domain Ω_t .

2.2 Finite element discretization

For the discretization of the equation system (8), the Finite Element Method (FEM) stabilized by means of the Streamline Upwind/Petrov-Galerkin (SUPG) strategy and with the addition of a shock capturing operator is used. Consider a finite element discretization of the domain Ω into n_{el} sub-domains Ω^e , $e = 1, 2, \dots, n_{el}$. Based on this discretization, the finite element function spaces for the trial solutions and for the weighting functions, \mathcal{S}^h and \mathcal{V}^h respectively, can be defined (see equation (13)).

Then, the finite element formulation of problem (8) using SUPG is written as follows:
Find $\mathbf{U}^h \in \mathcal{S}^h$ such that $\forall \mathbf{W}^h \in \mathcal{V}^h$

$$\begin{aligned} & \int_{\Omega} \mathbf{W}^h \cdot \left[\frac{\partial \mathbf{U}^h}{\partial t} + (\mathbf{A}_i^h - w_i \mathbf{I}) \frac{\partial \mathbf{U}^h}{\partial x_i} \right] d\Omega + \int_{\Omega} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} d\Omega \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \boldsymbol{\tau} (\mathbf{A}_k^h - w_k \mathbf{I})^T \frac{\partial \mathbf{W}^h}{\partial x_k} \cdot \left[\frac{\partial \mathbf{U}^h}{\partial t} + (\mathbf{A}_i^h - w_i \mathbf{I}) \frac{\partial \mathbf{U}^h}{\partial x_i} - \frac{\partial}{\partial x_i} \left(\mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} \right) \right] d\Omega^e \quad (12) \\ & + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \delta_{sc} \frac{\partial \mathbf{W}^h}{\partial x_i} \cdot \frac{\partial \mathbf{U}^h}{\partial x_i} d\Omega^e = \int_{\Gamma_h} \mathbf{W}^h \cdot \mathbf{f} d\Gamma \end{aligned}$$

where

$$\begin{aligned} \mathcal{S}^h &= \{\mathbf{U}^h | \mathbf{U}^h \in [\mathbf{H}^{1h}(\Omega)]^{n_{\text{dof}}}, \mathbf{U}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\text{dof}}}, \mathbf{U}^h = \mathbf{g} \text{ on } \Gamma_g\} \\ \mathcal{V}^h &= \{\mathbf{W}^h | \mathbf{W}^h \in [\mathbf{H}^{1h}(\Omega)]^{n_{\text{dof}}}, \mathbf{W}^h|_{\Omega^e} \in [P^1(\Omega^e)]^{n_{\text{dof}}}, \mathbf{W}^h = \mathbf{0} \text{ on } \Gamma_g\} \end{aligned} \quad (13)$$

$\mathbf{H}^{1h}(\Omega)$ being the finite dimensional Sobolev functional space over Ω .

The first summation of element level integrals in equation (12) are added to the variational formulation to stabilize the computations against numerical instabilities. In the advection-dominated range, these terms prevent the node-to-node oscillations of the flow variables, where τ is known as the intrinsic time tensor. The second summation of element level integrals in equation (12) are the shock capturing terms that stabilize the computations in the presence of sharp gradients, δ_{sc} being the coefficient of shock capturing. Matrix τ is defined following the proposal by Aliabadi et al. (1993), but adapted to the formulation with ALE and modifying the ‘inviscid’ part in order to solve all speed flows, in the following way

$$\tau = \max[\mathbf{0}, \tau_a - \tau_d - \tau_\delta] \quad (14)$$

where

$$\begin{aligned} \tau_a &= \left[\left(\frac{\frac{1}{2}[\|\mathbf{u} - \mathbf{w}\|(1 + zM_r^2) + c']}{h/2} \right)^2 + \left(\frac{2}{\Delta t} \right)^2 \right]^{-1/2} \mathbf{I} \\ \tau_d &= \frac{\sum_{j=1}^{n_d} \beta_j^2 \text{diag}(\mathbf{K}_{jj})}{(c + \|\mathbf{u} - \mathbf{w}\|)^2} \mathbf{I} \\ \tau_\delta &= \frac{\delta_{sc}}{(c + \|\mathbf{u} - \mathbf{w}\|)^2} \mathbf{I} \end{aligned} \quad (15)$$

Here, $c = \sqrt{\gamma RT}$ is the sonic speed, R being the particular gas constant; h is the element size computed as the element length in the direction of the streamline; $\beta = \nabla\|\mathbf{U}\|^2/\|\nabla\|\mathbf{U}\|^2\|$ and Δt is the time step. The expression of τ_a in (15) was proposed in López et al. (2010) (see also López et al. (2008)), in which

$$\begin{aligned} c' &= \sqrt{\|\mathbf{u} - \mathbf{w}\|^2 (1 + zM_r^2)^2 + 4zc^2M_r^2 \left(1 - \frac{\|\mathbf{u} - \mathbf{w}\|^2}{c^2} \right)} \\ M_r &= \min(1, \max(\sqrt{M^2 + CFL_c^{-2}}, M_\epsilon)) \\ z &= \max(1, z_{\text{vis}}) \\ z_{\text{vis}} &= \frac{Re_h^{-1}(Re_h^{-1} - 1)}{M_r^2[Re_h^{-1} - 1 + c^2/(\mathbf{u} \cdot \mathbf{s})^2]} \end{aligned} \quad (16)$$

where $M = \|\mathbf{u}\|/c$ is the Mach number, M_ϵ is a cut-off of the Mach number in the vicinity of stagnation points defined by the user, $CFL_c = c\Delta t/h$, $Re_h = \rho\|\mathbf{u}\|h/\mu$ is the cell Reynolds number based on the characteristic element length h , and \mathbf{s} is the unit vector aligned with the flow velocity.

Regarding the shock capturing term, the isotropic operator proposed by Tezduyar and Senga (2004) is applied.

2.3 Inlet/outlet boundary conditions

Conditions for inlet and outlet boundaries are applied following the approach proposed by Storti et al. (2008). Considering a point on an inlet/outlet boundary, it is possible to do

a simplified 1D analysis in the normal direction to the local boundary. The projection matrices onto the right/left-going characteristics modes are defined as

$$\mathbf{\Pi}_n^\pm = \mathbf{S}_n \mathbf{\Pi}_{V_n}^\pm \mathbf{S}_n^{-1} \quad (17)$$

where \mathbf{S}_n is the matrix of eigenvectors diagonalizing the projected system, $\mathbf{\Lambda}_n = \text{diag}[(\lambda_n)_j]$ being their respective eigenvalues; and

$$(\mathbf{\Pi}_{V_n}^-)_{jk} = \begin{cases} 1 & \text{if } j = k \text{ and } (\lambda_n)_j < 0 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

$$\mathbf{\Pi}_{V_n}^- + \mathbf{\Pi}_{V_n}^+ = \mathbf{I}$$

Then, the boundary condition is applied as a constraint to the system of governing equations as follows

$$\mathbf{\Pi}_n^-(\hat{\mathbf{U}})(\mathbf{U} - \hat{\mathbf{U}}) = \mathbf{0} \quad \text{on } \Gamma_f \quad (19)$$

where $\hat{\mathbf{U}}$ is defined depending on whether the boundary is either an inlet or an outlet. Note that in equation (19) the projection matrix, which is a non linear function of the fluid state, is evaluated at the state $\hat{\mathbf{U}}$. This is true if it is assumed that the flow is composed of small perturbations around the state $\hat{\mathbf{U}}$. However, as long as the fluid state departs from the $\hat{\mathbf{U}}$ value, the condition becomes less and less absorbing.

At the discrete level, conditions (19) are imposed via Lagrange multipliers as proposed by [Storti et al. \(2008\)](#). Let i a node lying on the inlet (or outlet) boundary. Then, the equations for this node are modified in the following way

$$\begin{cases} \mathbf{\Pi}_n^-(\hat{\mathbf{U}})(\mathbf{U}_i - \hat{\mathbf{U}}) + \mathbf{\Pi}_n^+(\hat{\mathbf{U}})\mathbf{U}_{\text{lm}} = \mathbf{0} \\ \mathbf{R}_i + \mathbf{\Pi}_n^-(\hat{\mathbf{U}})\mathbf{U}_{\text{lm}} = \mathbf{0} \end{cases} \quad (20)$$

where \mathbf{U}_{lm} is the vector of Lagrange multipliers and \mathbf{R}_i is the FEM residue for node i . At inlet regions $\hat{\mathbf{U}} = \mathbf{U}_{\text{ref}}$, with \mathbf{U}_{ref} a reference state. At outlet regions, [Storti et al. \(2008\)](#) propose to take $\hat{\mathbf{U}}$ as the state of the fluid in the previous time step if the external conditions are unknown. They named this strategy ULSAR (Use Last State As Reference) and show that Riemann invariants are preserved in the limit $\Delta t \rightarrow 0$ and $h \rightarrow 0$, if such invariants exist.

Some internal combustion engines utilize ports for the gas-exchange process, such as two-stroke and rotary engines (Wankel ([Ansdale, 1968](#)), MRCVC ([Toth, 2004](#)), etc.). Generally, the ports are placed on fixed walls of the engine (the cylinder or the housing) and, thus, have a relative motion with respect to the flow domain. For example, figure 1 shows a scheme of a two-stroke engine with intake and exhaust ports located on the cylinder wall. In this case, an observer placed on the centroid of the flow domain sees the ports moving away in the bottom direction.

A port could be modeled as a ‘window’ in relative motion with respect to the boundary domain. This window changes its passage area as the boundary moves, from the open position to the closed one, and vice-versa. Due to the nodal displacement produced by the deformation of the flow domain, mesh nodes lying on a boundary with a window could change their position between the wall and the port (an inlet/outlet for the flow problem). Therefore, the boundary condition applied on each of such nodes must be changed appropriately in order to account for the node position. The strategy proposed consists in switching from an absorbing boundary condition (equation (20)) when the node is placed on the port region to a wall boundary condition when the node moves on the solid wall. The wall boundary condition is applied by means

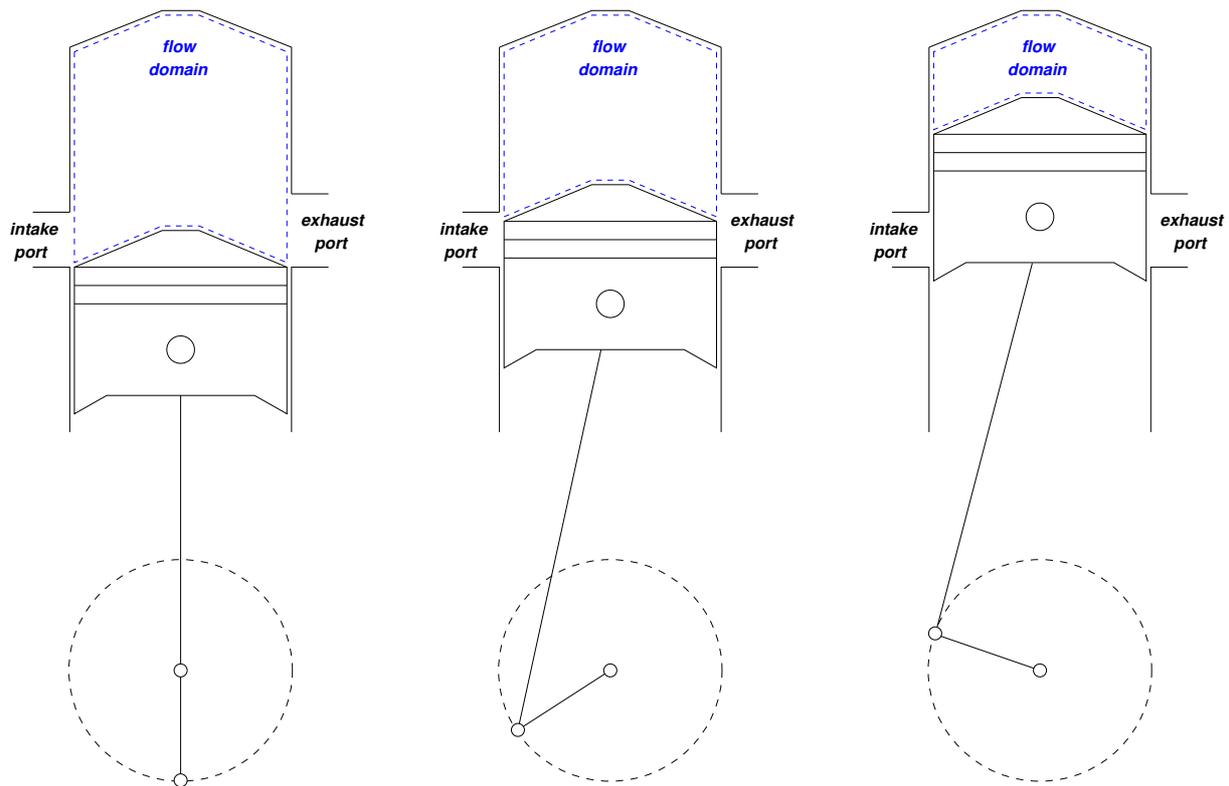


Figure 1: Two-stroke engine scheme.

of constraints using Lagrange multipliers in order to keep constant the total number of degrees of freedom. For instance, in a 3D problem using a no-slip boundary condition and considering the velocity of the solid wall equals to zero, the system of equations to solve for the node i is written as

$$\begin{aligned} \mathbf{M}\mathbf{U}_i + (\mathbf{I} - \mathbf{M})\mathbf{U}_{lm} &= \mathbf{0} \\ \mathbf{R}_i + \mathbf{M}\mathbf{U}_{lm} &= \mathbf{0} \end{aligned} \quad (21)$$

where $\mathbf{M} = \text{diag}[0, 1, 1, 1, 0]$.

3 THE 0D/1D MODEL

In this section we describe the equations and numerical schemes applied to the simulation of components with 0D/1D models. Models included in this paper only consists in those used in the problems presented in the results section. A complete description of models available in our 0D/1D code can be found in [López and Nigro \(2010\)](#).

3.1 Mathematical models

3.1.1 Pipe model

Pipes and manifolds are modeled applying the unsteady one-dimensional gas flow equations. In order to include effects like variable cross-section, viscous friction, and wall heat transfer, some source terms are added to the inviscid gas dynamic model represented by the system of

Euler equations. The resultant system of equations can be written as (Ramos, 1989)

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} &= -\frac{1}{F} \frac{dF}{dx} \rho u \\ \frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} &= -\frac{1}{F} \frac{dF}{dx} \rho u^2 - \rho G \\ \frac{\partial(\rho E)}{\partial t} + \frac{\partial[(\rho E + p)u]}{\partial x} &= -\frac{1}{F} \frac{dF}{dx} u(\rho E + p) + \dot{q} \pi \frac{D}{F}\end{aligned}\quad (22)$$

where F is the pipe cross-section area; $G = f \frac{u|u|}{2} \pi \frac{D}{F}$ is the specific friction force, with the friction coefficient given by $f = 2\tau_w/\rho u^2$, τ_w being the viscous shear stress at the pipe wall and D the equivalent diameter of the pipe; and \dot{q} is the heat transfer per unit mass of fluid per unit time. Again, the equation of state used here corresponds to the ideal gas model.

3.1.2 Cylinder model

The cylinder is modeled using a single-zone model, in which the charge is assumed to be a homogeneous mixture of ideal gases at all times. The equations of the model are the conservation of mass and the first law of thermodynamics

$$\begin{aligned}\frac{dm}{dt} &= \sum_j \dot{m}_j \\ \frac{d}{dt}(me) &= -p\dot{V} + \dot{Q}_{\text{ch}} - \dot{Q}_{\text{ht}} + \sum_j h_j \dot{m}_j\end{aligned}\quad (23)$$

where m is the mass contained into the cylinder; \dot{m}_j is the instantaneous mass flow rate through the j -th inlet/outlet (for instance, the mass flow rate through intake and exhaust valves, mass flow rate due to fuel addition, leakages, etc.); e is the specific internal energy of the mixture; V is the cylinder volume; \dot{Q}_{ch} represents the heat release due to combustion; \dot{Q}_{ht} is the heat transfer rate; and $h_j \dot{m}_j$ represents the enthalpy fluxes through the j -th inlet/outlet.

The model is closed specifying the geometry of the combustion chamber, the heat release rate, the heat transfer rate through the cylinder walls, and the mass flow rate of air and fuel. The sub-models used here are presented below.

Heat transfer model The instantaneous heat transfer rate that appears in equation (23) is calculated applying Nusselt-Reynolds-Prandtl (Nu-Re-Pr) numbers correlations as, for example, the one developed by [Woschni \(1967\)](#) or by [Annand \(1963\)](#). All of them allows to compute a film transfer coefficient h_c with expressions like

$$\text{Nu} = \frac{h_c L}{\kappa} = C \text{Re}^\alpha \text{Pr}^\beta \quad (24)$$

where L is a characteristic length; and C , α and β are constants.

Then, the heat transfer rate to the walls is

$$\dot{Q}_{\text{ht}} = Ah_c(T - T_{\text{wall}}) \quad (25)$$

where T is the temperature of gas into the cylinder, and T_{wall} is the temperature of the cylinder wall.

3.1.3 Valve model

In order to calculate the flow rates through the intake and exhaust valves, we use an analogy with the steady flow through convergent nozzles proposed by Benson (1982). The model assumes the passage area through the valve as the nozzle throat (whose state is represented by the subscript T in the equations below), the nozzle connecting the cylinder (subscript C in the equations) and the end of the pipe (subscript P in the equations). Depending on the direction of the flow velocity with respect to the pipe end, the problem could be an *inlet* (from cylinder to pipe) or an *outlet* (from pipe to cylinder). In addition, the flow at the throat could be sonic or subsonic. The equations of the model are the following

- Subsonic inlet:

$$\begin{aligned} \left(\frac{dp}{dt}\right)_P \pm \rho_P c_P \left(\frac{du}{dt}\right)_P &= (\text{RHS}_2^\pm)_P \\ \rho_T u_T \psi &= \rho_P u_P \\ c_C^2 &= c_P^2 + \delta u_P^2 \\ \frac{p_C}{p_T} &= \left(\frac{\rho_C}{\rho_T}\right)^\gamma \\ c_C^2 &= c_T^2 + \delta u_T^2 \\ p_T &= p_P \end{aligned} \tag{26}$$

where $\psi = F_T/F_P$, $\delta = (\gamma - 1)/2$, and

$$\text{RHS}_2^\pm = (\gamma - 1) \left(\dot{q} \pi \frac{D}{F} + \rho u G \right) \mp \rho c G - \frac{\rho u c^2}{F} \frac{dF}{dx} \tag{27}$$

In system (26), the first equation accounts for the compatibility along the incoming Mach line λ^\pm ; the second equation is the mass conservation between T and P ; the third and fifth equations represent the energy conservation between C and P , and C and T , respectively; the fourth equation represents an isentropic evolution between the cylinder and the nozzle throat; and the last equation is the condition on the pressure at the nozzle exit.

- Subsonic outlet:

$$\begin{aligned} \left(\frac{dp}{dt}\right)_P \pm \rho_P c_P \left(\frac{du}{dt}\right)_P &= (\text{RHS}_2^\pm)_P \\ \rho_T u_T \psi &= \rho_P u_P \\ \left(\frac{Dp}{Dt}\right)_P - c_P^2 \left(\frac{D\rho}{Dt}\right)_P &= (\text{RHS}_1)_P \\ \frac{p_P}{p_T} &= \left(\frac{\rho_P}{\rho_T}\right)^\gamma \\ c_P^2 + \delta u_P^2 &= c_T^2 + \delta u_T^2 \\ p_T &= p_C \end{aligned} \tag{28}$$

with

$$\text{RHS}_1 = (\gamma - 1) \left(\dot{q} \pi \frac{D}{F} + \rho u G \right) \tag{29}$$

and D/Dt denoting the material derivative. From the first equation to the last one in the system (28), they represent, respectively, the compatibility along the incoming Mach line,

the mass conservation between T and P , the compatibility along the incoming path line λ^0 , the isentropic evolution between P and T , the energy conservation between P and T , and the condition on the pressure at the nozzle exit.

- Sonic inlet: in this case, the system of equations is the same as (26) with the last equation replaced by the condition $c_T = u_T$.
- Sonic outlet: in this case, the system of equations is the same as (28) with the last equation replaced by $c_T = u_T$.

3.1.4 Pipe junction model

The pipe junction model applied here was proposed by Corberan (1992). If the junction is composed by r incoming pipes and s outgoing pipes, the model is expressed as

- Mass conservation

$$\sum_{j=1}^N \dot{m}_j = 0, \quad \text{with} \quad \dot{m}_j = \rho_j F_j u_j n_j$$

where $N = r + s$ is the total number of pipes at the junction, F_j is the cross-section area of the j -th pipe and n_j its exterior normal.

- Energy conservation

$$\sum_{j=1}^N \dot{h}_j = 0, \quad \text{with} \quad \dot{h}_j = \frac{\dot{m}_j}{\gamma - 1} (c_j^2 + \delta u_j^2)$$

- Compatibility equation along incoming Mach lines λ_j^\pm

$$\left(\frac{dp}{dt} \right)_j \pm \rho_j c_j \left(\frac{du}{dt} \right)_j = (\text{RHS}_2^\pm)_j, \quad j = 1, \dots, N$$

- Compatibility equation along incoming path lines λ_j^0

$$\left(\frac{Dp}{Dt} \right)_j - c_j^2 \left(\frac{D\rho}{Dt} \right)_j = (\text{RHS}_1)_j, \quad j \in r$$

- Equality of pressure at all branches in the junction

$$p_i = p_j, \quad \forall i \neq j$$

- Equality of enthalpy at all outgoing branches in the junction

$$c_i^2 + \delta u_i^2 = c_j^2 + \delta u_j^2, \quad \forall i, j \in s, \quad i \neq j$$

3.2 Numerical implementation

System (22) is discretized in space using a TVD (Total Variation Diminishing) Finite Volume Method (FVM). In particular, we transform the system (22) in three decoupled scalar equations and then we apply the method by Harten (1983) to each one. Finally, the solution is transformed back to the original basis. Time derivatives are discretized applying the forward Euler difference scheme.

Non-linear systems, such as valve and junction models, are solved using the Newton-Raphson method enhanced with a line search strategy.

4 COUPLING STRATEGY

We consider now a domain splitted into two or more sub-domains, where in some sub-domains the multi-D model presented in section §2 is applied and in the remaining sub-domains the 0D/1D model developed in section §3 is used. For the sake of simplicity in the analysis that follows, we consider only two sub-domains as shown in figure 2. The sub-domains interchange mass, momentum and energy through the *coupling interface* Γ_c . Thus, from the point of view of each sub-problem, the coupling interface is an inlet/outlet boundary and could be solved using absorbing boundary conditions. In this case, the reference state for computing the absorbing boundary condition is provided for the corresponding state at Γ_c .

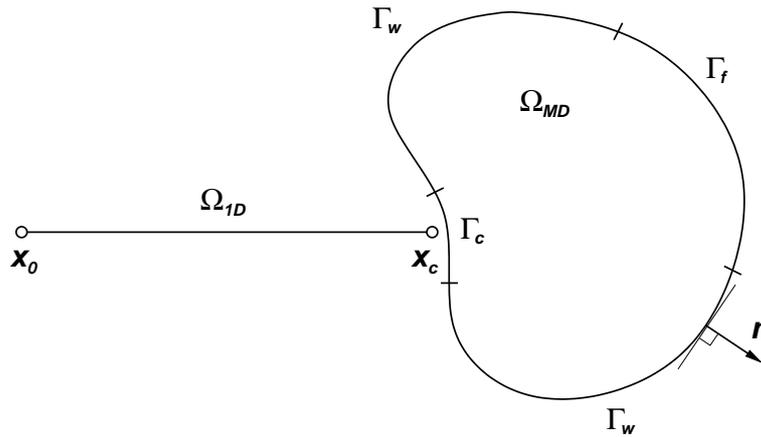


Figure 2: Computational domain for the multi-D/1D model.

Let $\Omega_{1D} = (x_0, x_c)$ the spatial 1D domain, on which the system (22) governs the fluid flow ¹. The problem is completely defined once the initial and boundary conditions are provided. At both end points, we apply absorbing boundary conditions. The reference state at point x_0 is defined by either atmospheric conditions or by the resulting state of the left sub-domain (0D or 1D). The condition at x_c (the coupling ‘interface’ for Ω_{1D}) could be imposed as

$$\Pi_{\bar{U}_{p1D}}[\mathbf{U}_{p1D}(x_c, t) - \bar{\mathbf{U}}_{pMD \rightarrow 1D}(t)] = \mathbf{0}, \quad t \geq 0 \tag{30}$$

where $\bar{\mathbf{U}}_{pMD \rightarrow 1D}(t) = [\bar{\rho}_{MD}(t), \bar{u}_{MD}(t), \bar{p}_{MD}(t)]^T$ is a reference state arising from the *condensation* of the variables at the coupling interface of the multi-D domain and $\mathbf{U}_p = [\rho, \mathbf{u}, p]^T$ is the primitive variables vector. For scalar components of the state vector ($\bar{\rho}_{MD}$ and \bar{p}_{MD}) this

¹For internal combustion engine problems, this domain is part of a network of 1D domains linked between them by 0D models.

condensation is simply the mean value of the variable on the coupling surface

$$\bar{\rho}_{MD}(t) = \frac{1}{\text{meas}(\Gamma_c)} \int_{\Gamma_c} \rho_{MD}(\mathbf{x}, t) d\sigma, \quad \bar{p}_{MD}(t) = \frac{1}{\text{meas}(\Gamma_c)} \int_{\Gamma_c} p_{MD}(\mathbf{x}, t) d\sigma \quad (31)$$

where $\text{meas}(\Gamma_c) = \int_{\Gamma_c} d\sigma$. For vectorial components (the fluid velocity),

$$\bar{\mathbf{u}}_{MD}(t) = \frac{1}{\text{meas}(\Gamma_c)} \int_{\Gamma_c} \mathbf{u}_{MD}(\mathbf{x}, t) \cdot \mathbf{n} d\sigma \quad (32)$$

Multi-D problem is defined by equations (6), (11) and (19); and a proper condition on Γ_c that must ensures the well-posedness of the problem. Then, we would need for all \mathbf{x} on Γ_c n_- boundary conditions, n_- being the rank of $\Pi_{U_{pn}}^-$. If we impose a condition of type

$$\Pi_{U_{pn}}^- \left[\int_{\Gamma_c} \mathbf{U}_{pMD} d\sigma - \text{meas}(\Gamma_c) \bar{\mathbf{U}}_{p1D \rightarrow MD}(t) \right] = \mathbf{0}, \quad (33)$$

where $\bar{\mathbf{U}}_{p1D \rightarrow MD}(t)$ is computed appropriately from the vector state of the 1D problem at x_c , the initial-boundary value problem (6), (11), (19), (33) is not well-posed since its solution is not unique. Indeed, we are prescribing on Γ_c just n_- scalar conditions rather than n_- at every point $\mathbf{x} \in \Gamma_c$, as it should be. Boundary conditions like (33) for multi-D domain problems are referred as *defective* (Formaggia et al., 2002; Leiva and Buscaglia, 2006).

In this study we propose to impose at every $\mathbf{x} \in \Gamma_c$ the same reference state $\bar{\mathbf{U}}_{p1D \rightarrow MD}(t)$, *i.e.*

$$\Pi_{U_{pn}}^- [\mathbf{U}_{pMD}(\mathbf{x}, t) - \bar{\mathbf{U}}_{p1D \rightarrow MD}(t)] = \mathbf{0}, \quad \forall \mathbf{x} \in \Gamma_c \quad (34)$$

with $\bar{\mathbf{U}}_{p1D \rightarrow MD}(t) = [\rho_{1D}(x_c, t), -u_{1D}(x_c, t) \mathbf{n}_{n_{1D}}, p_{1D}(x_c, t)]^T$, where n_{1D} is the outward normal to the 1D domain ($n_{1D} = \pm 1$). Now, problem (6), (11), (19), (34) has a unique solution, since (34) could be viewed as a condition of the type imposed on Γ_f in (19).

In some applications, as in the computational simulation of the combustion chamber of internal combustion engines, the multi-D domain changes its topology due to opening/closure of valves and/or ports. For instance, figure 3(a) shows a sketch of the cylinder and the intake port of a two-stroke engine. With the goal to simulate the combustion chamber using a multi-D model and the remaining engine parts with simplified models (0D/1D), sub-figure 3(b) presents a plausible model in which the pipe is modeled as a 1D domain, and the cylinder and port exit are simulated by means of a multi-D domain. This multi-D domain has a deformable region (cylinder) and a fixed one (port exit). One possibility is to use a dynamic mesh in the whole domain and apply a remeshing when the mesh quality reach a minimum admissible. Other possibility is to use a dynamic mesh in the deformable region and a fixed grid in the fixed region, linked between them by a sliding mesh strategy, as depicted in figure 3(b). In this work we propose an approach that consists in modeling with a multi-D domain only the cylinder, while the port exit is modeled as a convergent nozzle (see figure 3(c)). Therefore, the coupling between multi-D and 1D domains is through the convergent nozzle. We assume a quasi-steady flow in the nozzle. In the results section below, we present the simulation of an opposed-piston engine applying this last strategy.

4.1 Coupling algorithm

In this section the temporal algorithm that performs the coupling between the multi-D and the 0D/1D codes is presented. In terms of computational cost, the multi-D model determines

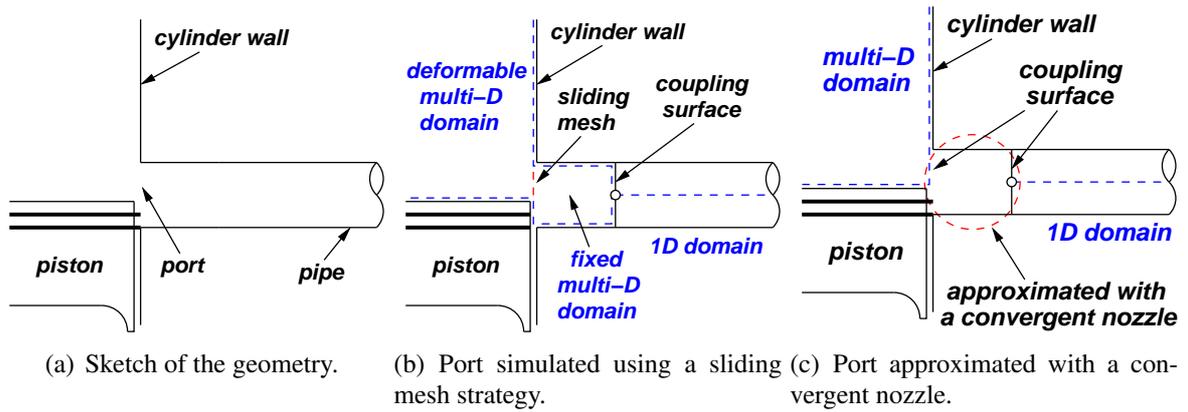


Figure 3: Some approaches for the simulation of ports.

practically the cost of the coupled problem. Therefore, we designed an algorithm taking into account this issue. The time step of the coupled simulation is set by the flow problem in the multi-D domain Δt_{MD} . Since the 0D/1D code uses an explicit scheme for time integration, the maximum time step (Δt_{1D}) allowed for this code will be limited by the CFL (Courant-Friedrichs-Levy) condition. Generally Δt_{1D} is smaller than Δt_{MD} thus, a sub-cycling strategy is needed in order to maintain the run synchronization. Since both codes interact between the reference state at the coupling interface, the multi-D problem only can provide reference states at times t and $t + \Delta t_{MD}$ for the 0D/1D code (and not at every time in the sub-cycling time steps). Then, we assume a linear interpolation between $\bar{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}(t)$ and $\bar{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}(t + \Delta t_{MD})$.

The proposed algorithm has a loose coupling between the multi-D and the 0D/1D code, thus, a ‘stage loop’ was added in order to reach a strong coupling when such a loop converges. The basic algorithm could be stated as

- 1: initialize variables
- 2: **for** $n = 0$ to n_{step} **do** {main time loop}
- 3: $t^n = n\Delta t_{MD}$
- 4: **for** $i = 0$ to n_{stage} **do** {stage loop}
- 5: **for** $k = 0$ to n_{nw} **do** {newton loop of MD code}
- 6: $\mathbf{U}_{MD}^{n+1,i+1,k+1} = \text{CFD-MD}(\mathbf{U}_{MD}^n, \mathbf{U}_{MD}^{n+1,i+1,k}, \bar{\mathbf{U}}_{\text{ref},1D \rightarrow MD}^{n+1,i})$
- 7: **end for**
- 8: compute $\bar{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}^{n+1,i+1}$ from $\mathbf{U}_{MD}^{n+1,i+1}$ and send it to the 0D/1D code
- 9: $m = 0$ {sub-cycling counter of the 0D/1D code}
- 10: $t_{1D}^m = t^n$ {0D/1D time initialization}
- 11: $\bar{\mathbf{U}}_{1D}^0 = \mathbf{U}_{1D}^n$
- 12: **while** $t_{1D}^m < t^n + \Delta t_{MD}$ **do** {0D/1D time loop}
- 13: compute Δt_{1D}^m {time step computation satisfying the CFL condition}
- 14: **if** $t_{1D}^m + \Delta t_{1D}^m > t^n + \Delta t_{MD}$ **then**
- 15: $\Delta t_{1D}^m = t^n + \Delta t_{MD} - t_{1D}^m$
- 16: **end if**
- 17: $t_{1D}^{m+1} = t_{1D}^m + \Delta t_{1D}^m$
- 18: compute $\tilde{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}^{m+1} = \bar{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}^n + \frac{t_{1D}^{m+1} - t^n}{\Delta t_{MD}} (\bar{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}^{n+1,i+1} - \bar{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}^n)$ {linear interpolation of the reference state during the sub-cycling iteration}
- 19: $\tilde{\mathbf{U}}_{1D}^{m+1} = \text{CFD-0D/1D}(\tilde{\mathbf{U}}_{1D}^m, \tilde{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}^{m+1})$
- 20: $m = m + 1$

```

21:   end while
22:    $\mathbf{U}_{1D}^{n+1,i+1} = \tilde{\mathbf{U}}_{1D}^m$ 
23:   compute  $\bar{\mathbf{U}}_{\text{ref},1D \rightarrow MD}^{n+1,i+1}$  from  $\mathbf{U}_{1D}^{n+1,i+1}$  and send it to the MD code
24:   end for
25: end for

```

In the algorithm, n_{step} is the number of time steps in the simulation, n_{nwt} is the number of Newton loops in the nonlinear problem, and n_{stage} is the number of stages in the coupling scheme. $\mathbf{U}_{MD}^{n+1,i+1,k+1} = \text{CFD-MD}(\mathbf{U}_{MD}^n, \mathbf{U}_{MD}^{n+1,i+1,k}, \bar{\mathbf{U}}_{\text{ref},1D \rightarrow MD}^{n+1,i+1})$ is the operator inside the Computational Fluid Dynamics multi-D code that advances the multi-D fluid state using the reference state $\bar{\mathbf{U}}_{\text{ref},1D \rightarrow MD}^{n+1,i}$; whereas $\tilde{\mathbf{U}}_{1D}^{m+1} = \text{CFD-0D/1D}(\tilde{\mathbf{U}}_{1D}^m, \tilde{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}^{m+1})$ is the operator inside the 0D/1D code which use the reference state $\tilde{\mathbf{U}}_{\text{ref},MD \rightarrow 1D}^{m+1}$ in the coupling interface at time t_{1D}^m in the sub-cycled time. Lines 9 to 22 in the algorithm could be encapsulated into an operator of the form $\mathbf{U}_{1D}^{n+1,i+1} = \text{CFD-0D/1D}_2(\mathbf{U}_{1D}^n, \mathbf{U}_{\text{ref},MD \rightarrow 1D}, t^n, \Delta t_{MD})$, where t^n is the initial time, $t^n + \Delta t_{MD}$ is the final time, and $\mathbf{U}_{\text{ref},MD \rightarrow 1D}$ is a function of time (in this case, the prescribed linear interpolation computed at line 18 in the algorithm).

If the reference state at the coupling interface for the multi-D problem has a small (relative) variation between two consecutive times, the basic algorithm could converge slowly and, thus, incrementing the computational cost. For these cases, we propose to include the resolution of the 0D/1D problem *inside* the Newton loop of the multi-D code; *i.e.*, for each Newton iteration the 0D/1D problem (from t^n to t^{n+1}) is solved. In the structure of the basic algorithm, this modification could be performed taking $n_{\text{nwt}} = 1$ and checking the convergence of the non-linear iteration in the stage loop. Of course, this strategy demands to have access to the source code of the multi-D model, particularly to the non-linear loop. We found a good convergence rate when the cited strategy is applied in problems where the geometry of the coupling surface remains unchanged, with only a few number of non-linear iterations respecting to the basic algorithm. However, the strategy could fail if the area of Γ_c changes in time with a high area ratio

The implementation was performed in the scripting language Python (van Rossum, 1990-2011), using the packages mpi4py (Dalcín, 2009-2011a) and petsc4py (Dalcín, 2009-2011b) for parallel computing managing; and pf4py (Dalcín, 2008) for driving the multi-D code PETS-c-FEM (Storti et al., 1999-2011). The 0D/1D code ICESym (Nigro et al., 2010-2011) is driven by Python scripts and an interface written in the language Cython (Behnel et al., 2008-2011) in order to deal with classes and methods implemented in C++.

5 RESULTS

5.1 Motored opposed-piston engine

The case consists in the resolution of the fluid flow inside the cylinder of a supercharged opposed-piston engine under cold conditions, *i.e.* without firing it. The engine has four cylinders, with two intake volutes and one exhaust volute per cylinder. A scheme of the engine configuration is shown in figure 4, in which the approximation level applied to model each component is indicated. The cylinder bore is 77 mm, the stroke of each piston is 53 mm, the length of the connecting rod is 125 mm and the geometric compression ratio is 11:1. The cylinder has 8 exhaust ports evenly distributed in the circumferential direction and 8 intake ports uniformly separated also. Assuming the reference angle as the EDC (External Dead Center), the timing of the ports are

- Intake Port Opening (IPO) = 289°
- Intake Port Closing (IPC) = 89°
- Exhaust Port Opening (EPO) = 271°
- Exhaust Port Closing (EPC) = 71°

A plant view of the geometry of the two intake volutes and the cylinder is shown in figure 5. The exhaust volute has a similar geometry, but it connects the exhaust ports to an unique exhaust pipe. As discussed in section 4, the volutes are simulated by means of a convergent nozzle (see figure 3). A scheme of the modeling used for the volutes is depicted in figure 6. In the figure, the normal vectors considered for each port are also represented. These vectors are precisely those used to compute the reference state for each port from the state at the volute (see equation (34)).

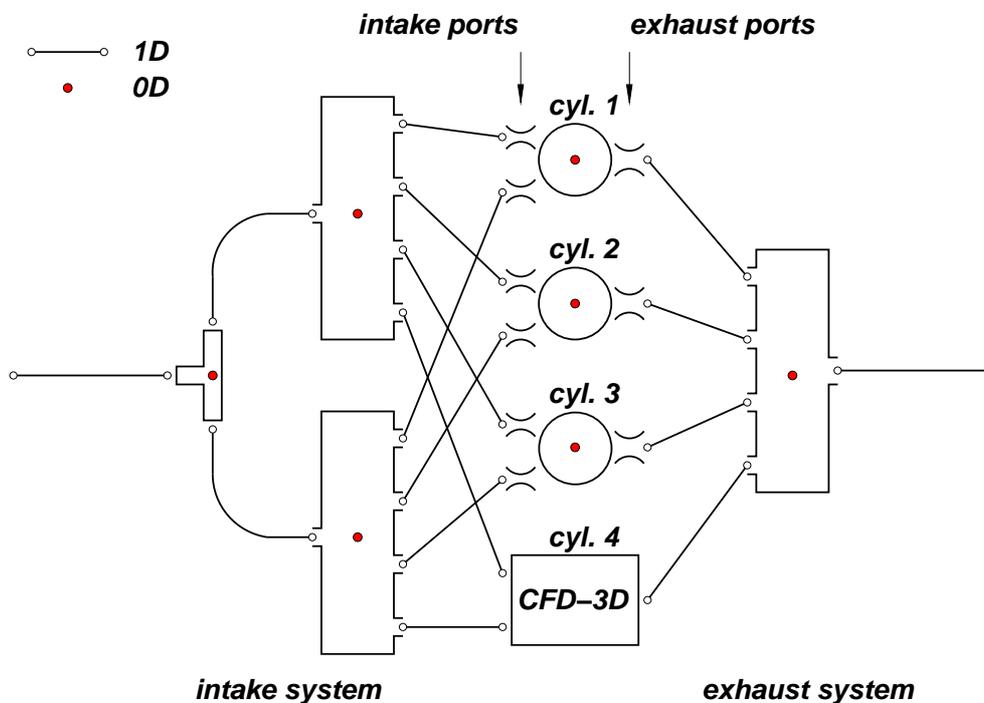


Figure 4: Sketch of the opposed-piston engine configuration.

The 1D domains were discretized with elements of length 5 mm, approximately. The mesh for the 3D model was generated with the pistons at EDC and has 1.87M tetrahedra and 358.8K nodes. The mean element size is $h = 1$ mm. Due to the simplicity of the geometry and the boundary movement, the mesh dynamics is solved using an algebraic law following a linear distribution with respect to the position of pistons at IDC (Internal Dead Center).

No-slip condition is imposed on solid walls. In addition, these walls are assumed to be insulated. Mixed absorbing/wall boundary conditions are used to model the ports, as explained in section §2.3. Turbulence is modeled applying the simplest LES (Large Eddy Simulation) Smagorinsky model (Smagorinsky (1963); Wilcox (2002)), which takes the Smagorinsky coefficient as constant. The engine speed is 3000 rpm. The time step used in the 3D simulation was $\Delta t = 5 \times 10^{-5}$ s. The reference state imposed at the inlet of the intake system was 192 kPa and 70°C , while atmospheric conditions are assumed to be 100 kPa and 25°C .

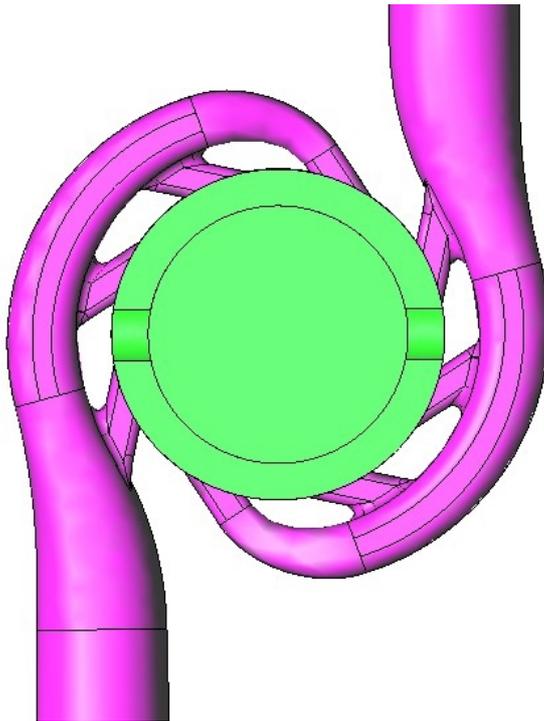


Figure 5: Plan view of the geometry of the cylinder and intake volutes.

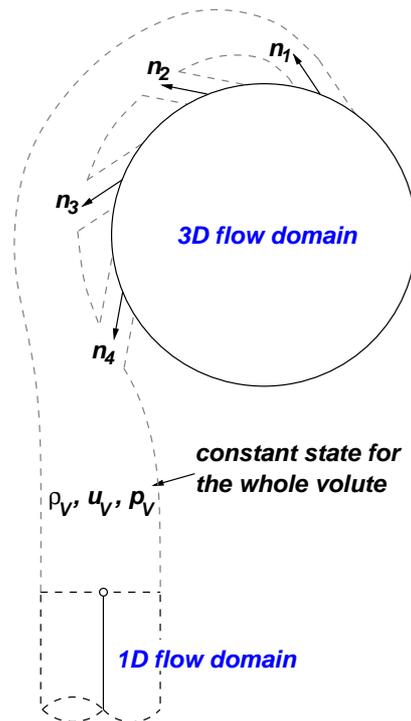


Figure 6: Schematic representation of the modeling of the volutes and coupling between 1D and 3D domains.

Regarding the coupling strategy, we apply in this case the algorithm presented in section §4.1 with $n_{\text{stage}} = 2$. If the passage area of ports is not very small, the number of stages used is sufficient for the convergence of the coupling. However, at port opening and closing there are a very large area ratio between the ducts and ports, which could deteriorate the convergence of the algorithm.

The results correspond to the last cycle simulated of a total of five. In figures 7 to 12 the velocity over a plane parallel to the crown of the nearest piston to the intake ports (the ‘intake piston’) are shown. The plane is placed to a distance of 10 mm from the piston crown and the figures correspond to several crank angle degrees (CAD, θ) along the intake period. Similarly, figures 13 to 18 present the velocity field over a plane parallel to the crown of the ‘exhaust piston’ (the nearest piston to the exhaust ports) for some angles during the exhaust phase. Again, this plane is 10 mm from the piston crown.

6 CONCLUSIONS

We presented a staged algorithm for the resolution of the coupling between 0D/1D and multi-D models for compressible flow problems. The coupling strategy is based on the use of absorbing boundary conditions for both the 0D/1D and multi-D models. The proposed algorithm was successfully applied in the simulation of a motored opposed-piston engine. This case is more demanding for the coupling algorithm since we couple a constant diameter pipe with a variable passage area (intake and exhaust ports) directly. Using a sliding mesh strategy for the resolution of the flow through ports, we plan to avoid this type of coupling between components with variable cross-section area.

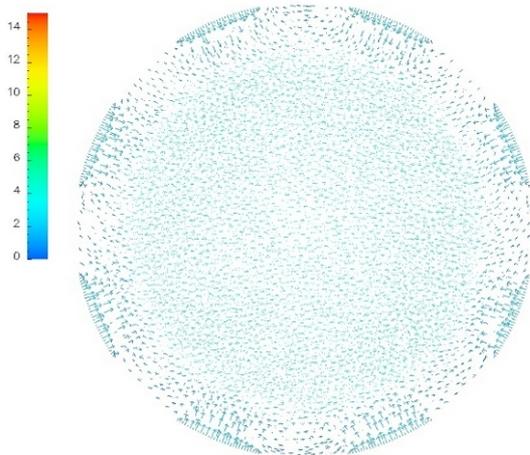


Figure 7: Velocity field ([m/s]) over the plane parallel to the intake piston crown for $\theta = -39.77$ CAD.

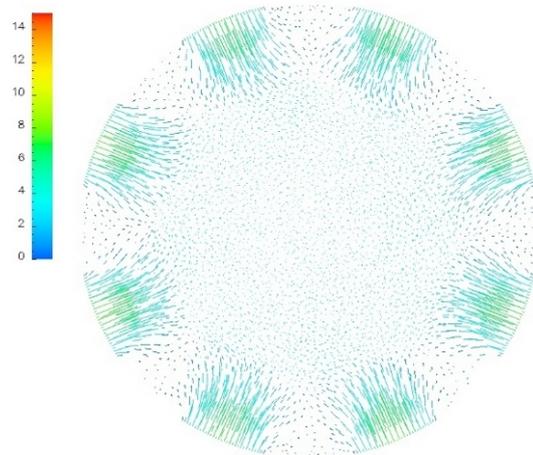


Figure 8: Velocity field ([m/s]) over the plane parallel to the intake piston crown for $\theta = -19.97$ CAD.

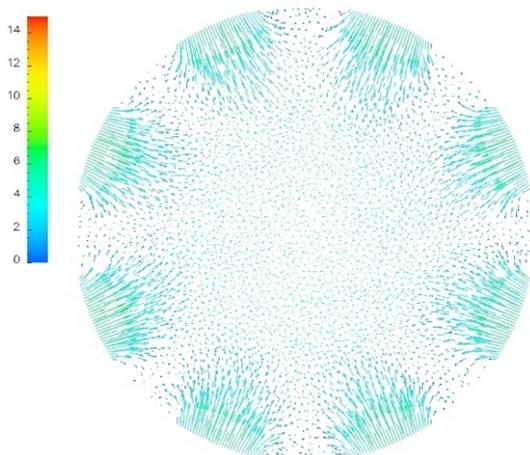


Figure 9: Velocity field ([m/s]) over the plane parallel to the intake piston crown for $\theta = -0.17$ CAD.

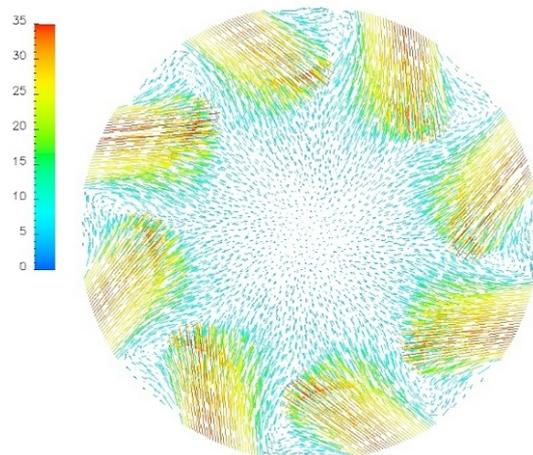


Figure 10: Velocity field ([m/s]) over the plane parallel to the intake piston crown for $\theta = 19.63$ CAD.

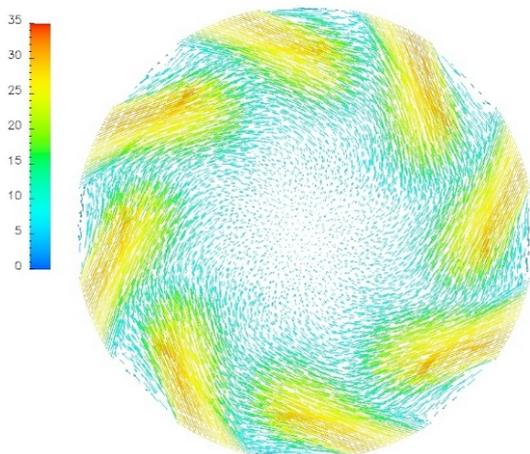


Figure 11: Velocity field ([m/s]) over the plane parallel to the intake piston crown for $\theta = 40.33$ CAD.

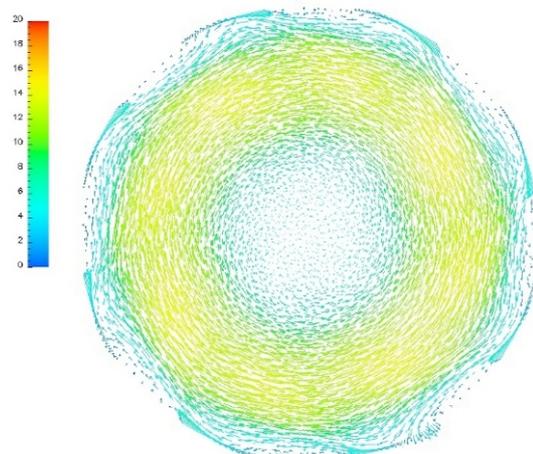


Figure 12: Velocity field ([m/s]) over the plane parallel to the intake piston crown for $\theta = 60.13$ CAD.

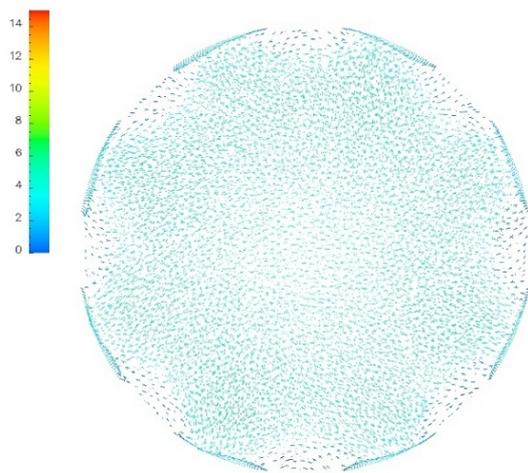


Figure 13: Velocity field ([m/s]) over the plane parallel to the exhaust piston crown for $\theta = -39.77$ CAD.

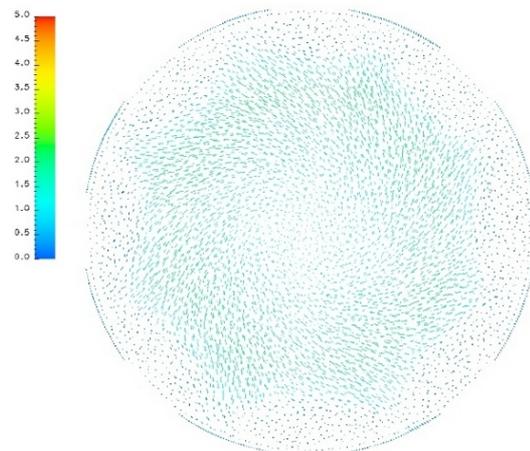


Figure 14: Velocity field ([m/s]) over the plane parallel to the exhaust piston crown for $\theta = -19.97$ CAD.

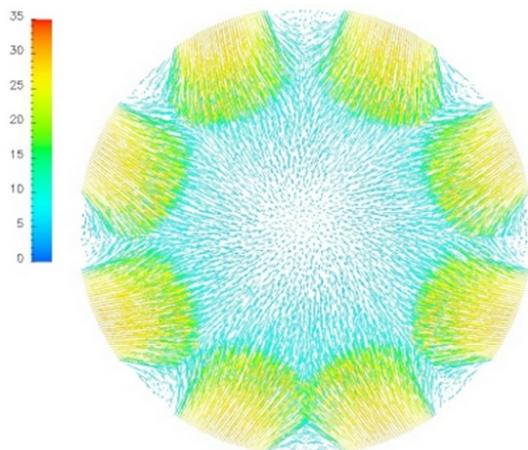


Figure 15: Velocity field ([m/s]) over the plane parallel to the exhaust piston crown for $\theta = -0.17$ CAD.

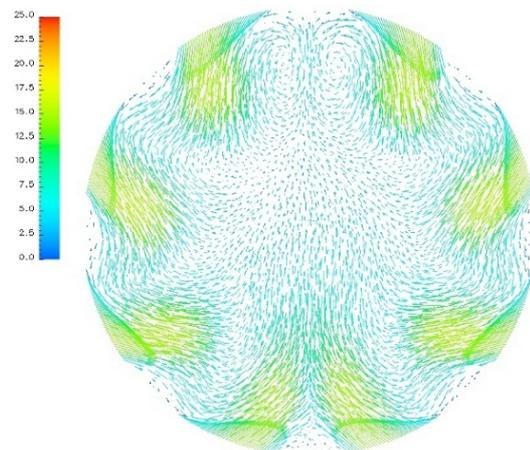


Figure 16: Velocity field ([m/s]) over the plane parallel to the exhaust piston crown for $\theta = 19.63$ CAD.

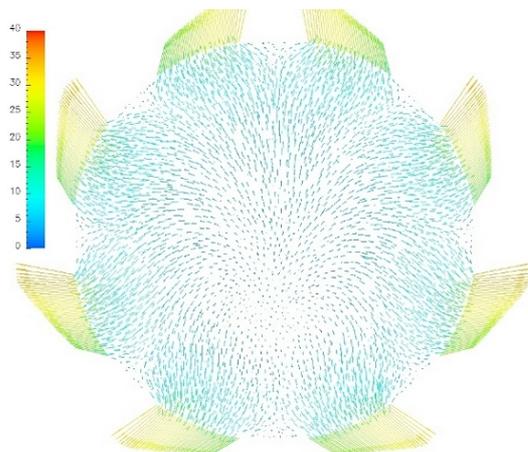


Figure 17: Velocity field ([m/s]) over the plane parallel to the exhaust piston crown for $\theta = 40.33$ CAD.

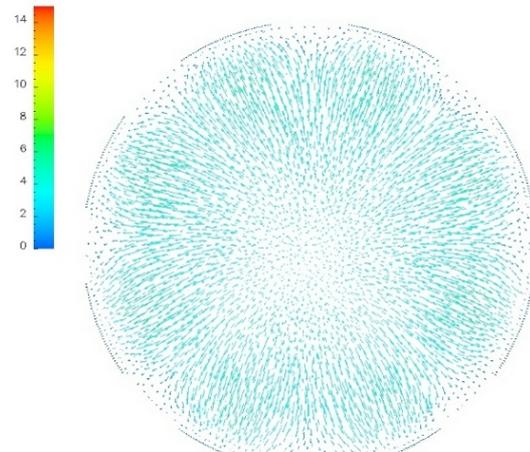


Figure 18: Velocity field ([m/s]) over the plane parallel to the exhaust piston crown for $\theta = 60.13$ CAD.

ACKNOWLEDGMENTS

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina), Universidad Nacional del Comahue (UNCo, Argentina), Universidad Nacional del Litoral (UNL, Argentina, grants CAI+D 65/334/2009, III-4-2/2009) and Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT, Argentina, grants PICT 1141/2007, 2492/2010), and was partially performed with the Free Software Foundation GNU-Project resources as GNU/Linux OS, GCC compilers, and GNU/Octave, as well as other Open Source resources as Python, Cython, Perl, PETSc, MPICH, Open-DX and L^AT_EX, among many others.

REFERENCES

- Aliabadi S., Ray S., and Tezduyar T. SUPG finite element computation of viscous compressible flows based on the conservation and entropy variables formulations. *Computational Mechanics*, 11:300–312, 1993.
- Annam W. Heat transfer in the cylinders of reciprocating internal combustion engines. *Proceedings of the Institution of Mechanical Engineers*, 177:973–980, 1963.
- Ansdale R. *The Wankel RC Engine Design and Performance*. Iliffe Books, London, 1968.
- Behnel S., Bradshaw R., and Sverre Seljebotn D. Cython: C-Extensions for Python. <http://www.cython.org/>, 2008-2011.
- Bella G., Bozza F., De Maio A., Del Citto F., and Fillippone S. An enhanced parallel version of Kiva-3V, coupled with a 1D CFD code, and its use in general purpose engine applications. *Lecture Notes in Computer Science*, 4208:11–20, 2006.
- Benson R. *The Thermodynamics and Gas Dynamics of Internal Combustion Engines*, volume I. Clarendon Press, Oxford, 1982.
- Blanco P., Feijóo R., and Urquiza S. A unified variational approach for coupling 3D–1D models and its blood flow applications. *Computer Methods in Applied Mechanics and Engineering*, 196:4391–4410, 2007.
- Corberan J. A new constant pressure model for N-branch junctions. *Proceedings of the Institution of Mechanical Engineers. Part D, Journal of automobile engineering*, 206:117–123, 1992.
- Dalcín L. *Techniques for high performance distributed computing in computational fluid mechanics*. Ph.D. thesis, CIMEC-INTEC-UNL, 2008.
- Dalcín L. MPI for Python - Python bindings for MPI. <http://code.google.com/p/mmpi4py/>, 2009-2011a.
- Dalcín L. PETSc for Python - Python bindings for PETSc libraries. <http://code.google.com/p/petsc4py/>, 2009-2011b.
- Donea J., Giuliani S., and Halleux J. An arbitrary, Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *SIAM Journal on Scientific Computing*, 33:689–700, 1982.
- Formaggia L., Gerbeau J.F., Nobile F., and Quarteroni A. On the coupling of 3D and 1D Navier-Stokes equations for flow problems in compliant vessels. *Computer Methods in Applied Mechanics and Engineering*, 191:267–287, 2001.
- Formaggia L., Gerbeau J.F., Nobile F., and Quarteroni A. Numerical treatment of defective boundary conditions for the navier-stokes equations. *SIAM Journal on Numerical Analysis*, 40(1):376–401, 2002.
- Formaggia L., Nobile F., Quarteroni A., and Veneziani A. Multiscale modelling of the circu-

- latory system: A preliminary analysis. *Computing and Visualization in Science*, 2:75–83, 1999.
- Formaggia L., Veneziani A., and Vergara C. A new approach to numerical solution of defective boundary value problems in incompressible fluid dynamics. *SIAM Journal on Numerical Analysis*, 46(6):2769–2794, 2008.
- Harten A. A high resolution scheme for the computation of weak solutions of hyperbolic conservation laws. *Journal of Computational Physics*, 49:357–393, 1983.
- Hirsch C. *Numerical Computation of Internal and External Flows. Volume 2: Computational Methods for Inviscid and Viscous Flows*. John Wiley & Sons, 1990.
- Leiva J., Blanco P., and Buscaglia G. Iterative strong coupling of dimensionally heterogeneous models. *International Journal for Numerical Methods in Engineering*, 81(12):1558–1580, 2009.
- Leiva J. and Buscaglia G. Estrategias de acoplamiento entre códigos 0D/1D y códigos CFD 3D. *Mecánica Computacional, Volumen XXV*, pages 53–82, 2006.
- López E. and Nigro N. Validation of a 0D/1D computational code for the design of several kind of internal combustion engines. *Latin-American Applied Research*, 40(2):175–184, 2010.
- López E., Nigro N., Sarraf S., and Márquez Damián S. Stabilized finite element method based on local preconditioning for unsteady compressible flows in deformable domains with emphasis on the low mach number limit application. *International Journal for Numerical Methods in Fluids*, 2010. (submitted).
- López E., Nigro N., and Storti M. Estrategia de acondicionamiento local para flujos compresibles a bajos números de Mach. *Mecánica Computacional, Volumen XXVII*, 2008.
- Montenegro G., Onorati A., Piscaglia F., and D’Errico G. Integrated 1D-multiD fluid dynamic models for the simulation of I.C.E. intake and exhaust systems. SAE International Congress and Exposition. Detroit, Mich., USA, 2007. SAE Paper N° 2007-01-0495.
- Moura A. *The Geometrical Multiscale modelling of the cardiovascular system: coupling 3D FSI and 1D models*. Ph.D. thesis, Politecnico di Milano, 2007.
- Nigro N., López E., and Gimenez J. ICESym: An Internal Combustion Engine Simulator. <http://code.google.com/p/icesym/>, 2010-2011.
- Onorati A., Ferrari G., and D’Errico G. The coupling of 1D and 2D fluid dynamic models for the prediction of unsteady flows in I.C. engine duct-systems. IMechE International Conference on Computational and Experimental Methods in Reciprocating Engines. London, U.K., 2000.
- Ramos J. *Internal Combustion Engine Modeling*. Hemisphere Publishing Corporation, 1989.
- Smagorinsky J. General circulation experiments with the primitive equation: I the basic experiment. *Monthly Weather Review*, 91:216–211, 1963.
- Storti M., Nigro N., Paz R., and Dalcín L. Dynamic boundary conditions in computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering*, 197:1219–1232, 2008.
- Storti M., Nigro N., Paz R., Dalcín L., and López E. PETSc-FEM: A General Purpose, Parallel, Multi-Physics FEM Program. <http://venus.ceride.gov.ar/petscfem/>, 1999-2011.
- Tezduyar T. and Senga M. Determination of the shock-capturing parameters in SUPG formulation of compressible flows. In T.U. Press and Springer-Verlag, editors, *Computational Mechanics WCCM IV*. Beijing, China, 2004.
- Toth J. *Motor Rotativo de Combustión a Volumen Constante (MRCVC)*, 2004. Patent Res. N° AR004806B1, Rec. N° P 19960105411.
- van Rossum G. Python programming language. <http://www.python.org/>, 1990-2011.
- Wilcox D. *Turbulence Modeling for CFD*. D C W Industries, 2nd edition, 2002.

Woschni G. A universally applicable equation for the instantaneous heat transfer coefficient in the internal combustion engine. SAE International Congress and Exposition. Detroit, Mich., USA, 1967. SAE Paper N° 670931.