

Universitat Politècnica de Catalunya

# Internship Report

by

Xavier Corbella Coll

January 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Description of the Internship</b>	<b>2</b>
2.1	COMSOL Multiphysics® . . . . .	2
2.2	Internship activities . . . . .	3
2.2.1	Application Libraries . . . . .	3
2.2.2	Development . . . . .	11
2.2.2.1	Stabilized Convection-Diffusion Equation Interface . . . . .	11
2.2.2.2	Contributions to the Fluid Flow interfaces . . . . .	13
<b>A</b>	<b>Documentation of the Supersonic Air-to-Air Ejector model</b>	<b>16</b>
<b>B</b>	<b>Documentation of the Shock Tube application</b>	<b>41</b>
	<b>Bibliography</b>	<b>51</b>

# Chapter 1

## Introduction

This report is a short description of my six months internship for the Master of Science in Numerical Methods applied to Engineering. The internship was carried out within the company COMSOL AB in 2016 in Stockholm, Sweden.

The activities I have pursued at COMSOL are related to the software COMSOL Multiphysics<sup>®</sup>, which uses advanced numerical methods to model and simulate physics-based problems and can solve various physics and engineering applications, specially coupled phenomena, or multiphysics. The tasks I have performed are:

- Models and applications: Implementation of new models and applications that serve as tutorials / demos of the software, and modification / improvements to old models and apps.
- Development: Small modifications to the Fluid Flow interfaces and the implementation of a new interface called the Stabilized Convection-Diffusion Equation interface.

This report includes a description of the company, the software, and the tasks that I have performed during this internship.

## Chapter 2

# Description of the Internship

### 2.1 COMSOL Multiphysics<sup>®</sup>

COMSOL Multiphysics<sup>®</sup> is a general-purpose software platform, based on advanced numerical methods, for modeling and simulating physics-based problems. It is a cross-platform (Windows, Mac, Linux) project that can solve various physics and engineering applications, especially coupled phenomena, or multiphysics. The main product is Comsol Desktop<sup>®</sup>, which provides a modeling environment for creating, analyzing, and visualizing multiphysics models and applications (for more information, see [1] and [2]). The Comsol Desktop<sup>®</sup> includes:

- Model Builder: It is the tool used to define the model, its components, the solver, analyze the results, and create reports. This is done building a model tree which reflects the underlying data structure and the model object, and stores the state of the model including settings for geometry, mesh, physics, boundary conditions, studies, solvers, postprocessing, and visualizations.
- Applications Builder: It allows to create specialized applications, based on a physics model, with a user interface that avoids the details of the simulation model from the perspective of the end user (see [3] and [4]).
- Physics Builder: It is a tool used to create custom made physics interfaces accessible from the COMSOL Desktop<sup>®</sup> with the same look-and-feel as the built-in physics interfaces (see [5]).
- Application Libraries: The application libraries include COMSOL models and applications that serve as tutorials and examples. Documentation files describing every application / model and how to build them are included.

Several products are available in COMSOL Multiphysics<sup>®</sup> (see [6]) to solve electrical, structural, acoustics, fluid flow, heat transfer, and chemical problems. Different Multiphysics interfaces are also available to solve coupled problems. In addition to the conventional physics-based interfaces, user-designed coupled systems of partial differential equations can also be solved, either defining them directly or introducing their weak form. COMSOL Multiphysics<sup>®</sup> also includes material libraries, an optimization module, CAD import and LiveLinks to link it with other software such as Matlab<sup>®</sup>, Microsoft Excel<sup>®</sup>, SOLIDWORKS<sup>®</sup> or Catia<sup>®</sup> V5.

## 2.2 Internship activities

Initially, the purpose of my internship in COMSOL Multiphysics<sup>®</sup> was to work with the models and apps in the Application Libraries, improving / fixing them and creating new ones. However, I ended up also working in development: I developed a new PDE interface that solves a convection-diffusion-adsorption equation via the Finite Element Method including stabilization, and worked on different improvements to the Fluid Flow interfaces.

### 2.2.1 Application Libraries

During my internship I have created several models and applications to be added to the Application Libraries. I discarded some of them because I considered them not good enough or because they did not provide anything new. Three of the models I created have already been published and included in the software:

- Expansion fan

This 2D example models an expansion fan where the flow is inviscid, and the results are compared with inviscid compressible flow theory. Adaptive meshing is used to resolve the expansion region more finely than the rest of the modeling domain (figure 2.1). Results (figure 2.2) correlate well with those conducted using inviscid compressible flow theory (see for example [7]).

The documentation of the model is available in reference [8].

- 3D Supersonic Flow in a Channel with a Bump

This example models 3D supersonic flow, including the effects of a shock, in a straight channel with a small obstacle at one of the walls (see figure 2.3).

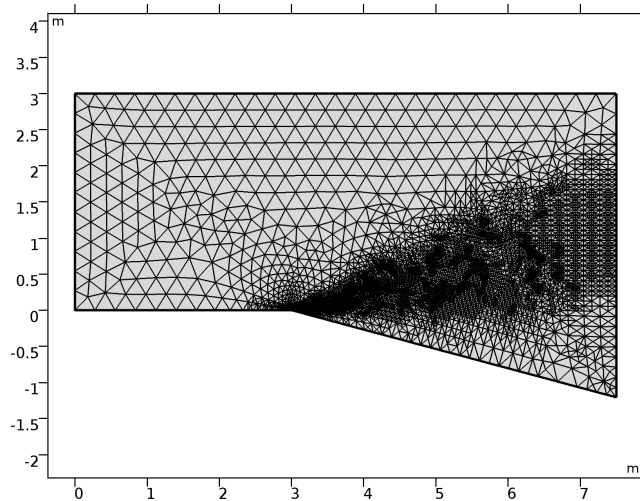


FIGURE 2.1: Mesh obtained in the Expansion Fan model after adaptivity.

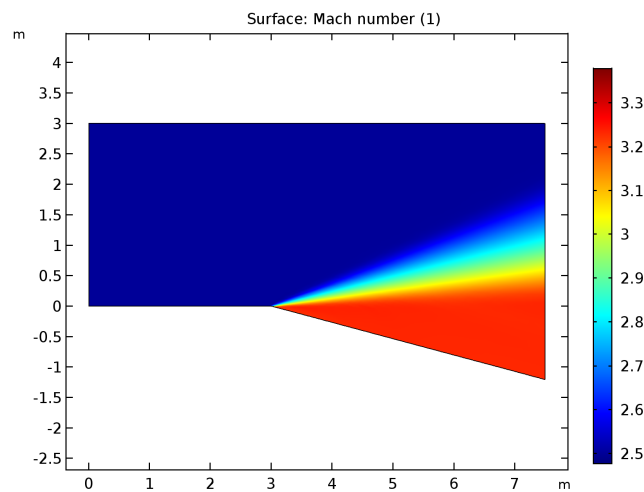


FIGURE 2.2: Mach number distribution in the Expansion Fan model.

As the flow hits the obstacle, shock waves are diffracted from the obstacle and walls of the channel. The propagating shock waves form a pattern in the velocity profile and in the density distribution.

The model makes use of the adaptive mesh refinement feature ( see figure 2.4 ). This feature is important as the shock waves should be well resolved by the mesh, but predetermination of their position is difficult.

The results ( see figure 2.5 ) are in good agreement with those published in literature (see [9]). The documentation of this model is available in [10].

- Supersonic Air-to-Air Ejector

This example models the compressible turbulent flow through a supersonic ejector. Ejectors are simple mechanical components used for a wide range of applications,

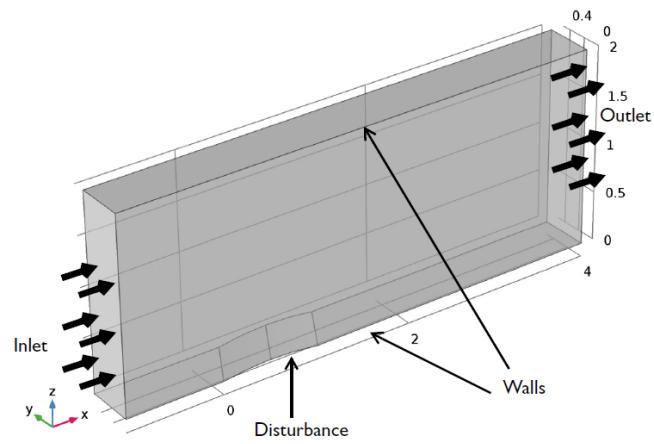


FIGURE 2.3: Geometry and boundary conditions of the 3D Supersonic Flow in a Channel with a Bump model.

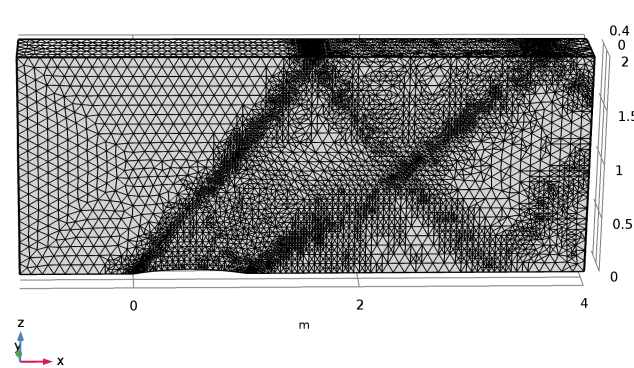


FIGURE 2.4: Mesh obtained after adaptivity in the the 3D Supersonic Flow in a Channel with a Bump model.

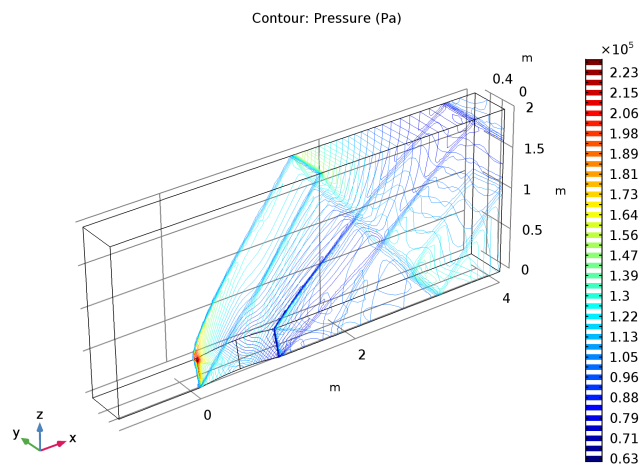


FIGURE 2.5: Contours of pressure showing the position of the shocks in the 3D Supersonic Flow in a Channel with a Bump model.

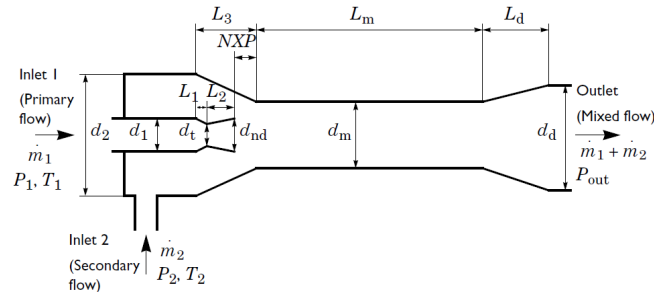


FIGURE 2.6: Geometry of the Supersonic Air-to-Air Ejector model.

including industrial refrigeration, vacuum generation, gas recirculation, and thrust augmentation in aircraft propulsion systems.

Ejectors induce a secondary flow by momentum and energy transfer from a high-velocity primary jet. The high-energy fluid (primary flow) passes through a convergent-divergent nozzle and reaches supersonic conditions. After exiting the nozzle, it interacts with the secondary flow and is accelerated through an entrainment-induced effect. The mixing between both flows takes place along a constant-area duct called the mixing chamber where complex interactions between the mixing layer and shocks can be observed. A diffuser is usually placed before the outlet to recover pressure and bring the flow back to stagnation.

The ejector modeled in this example works with air in both the primary and secondary streams. The geometry (see figure 2.6) and boundary conditions are based on references [11] and [12]. A peculiarity of this example is that the deceleration from supersonic to subsonic flow does not take place through a single normal shock, but through a complex succession of shocks called a shock train or a pseudo-shock wave phenomenon (see [13] and figures 2.7 and 2.8).

This model can be found in [14]. I added its documentation to Appendix A.

I also created three applications which have not been published, but could be published in the future:

- Turbulent Flow through Smooth Pipes

This application has been designed to model turbulent flow through smooth pipes and to study the friction factor, velocity profile and pressure drop along the pipe. The flow is solved using the Algebraic  $yPlus$  turbulence model. The input parameters are the pipe radius and length, fluid properties and the Reynolds number of the flow. The application uses these parameters to solve the problem and provides plots of the axial velocity profile and the distribution of pressure, and numerical values for the pressure drop, maximum and average velocities and dimensionless



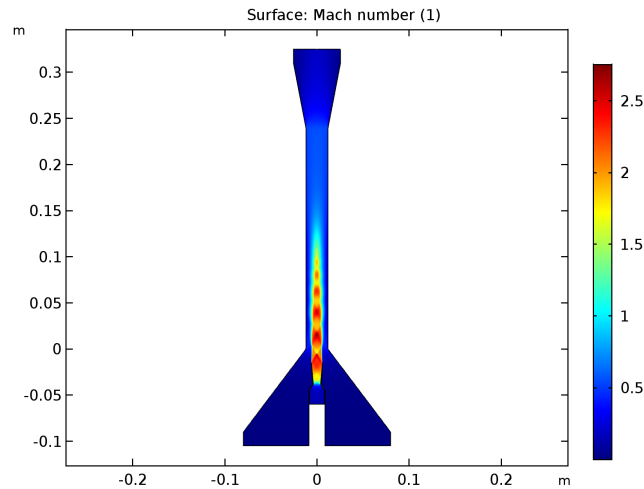


FIGURE 2.7: Mach number distribution in the Supersonic Air-to-Air Ejector model.

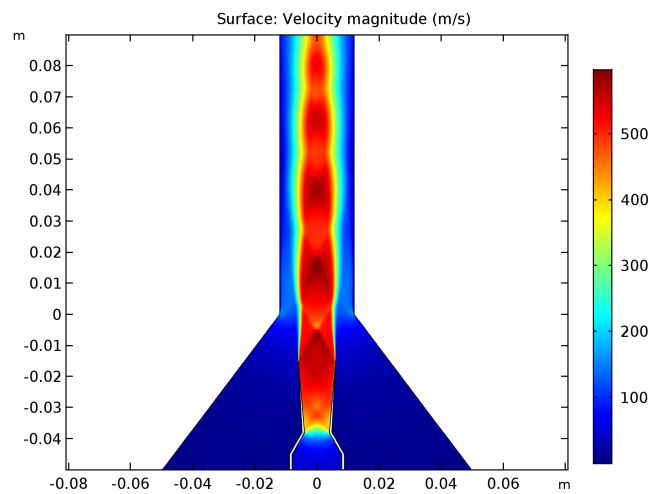


FIGURE 2.8: Distribution of velocity in the nozzle and the mixing chamber of the Supersonic Air-to-Air Ejector model.

friction factor, and creates a report. The GUI of the application is shown in figure 2.9.

The results obtained correlate well with Prandtl's universal law of friction for smooth pipes (see for example [15]) for Reynolds number  $10^4 < Re < 10^6$ , which is the range where this law is known to correlate well with experiments (see figure 2.10).

- Shock Tube

In this application, the compressible 1D Euler equations are solved with nodal discontinuous Lagrange shape functions to compute the flow in a shock tube. The purpose of this application is to simulate the flow in the tube and estimate the distributions of pressure, density, velocity, temperature, speed of sound, Mach number,

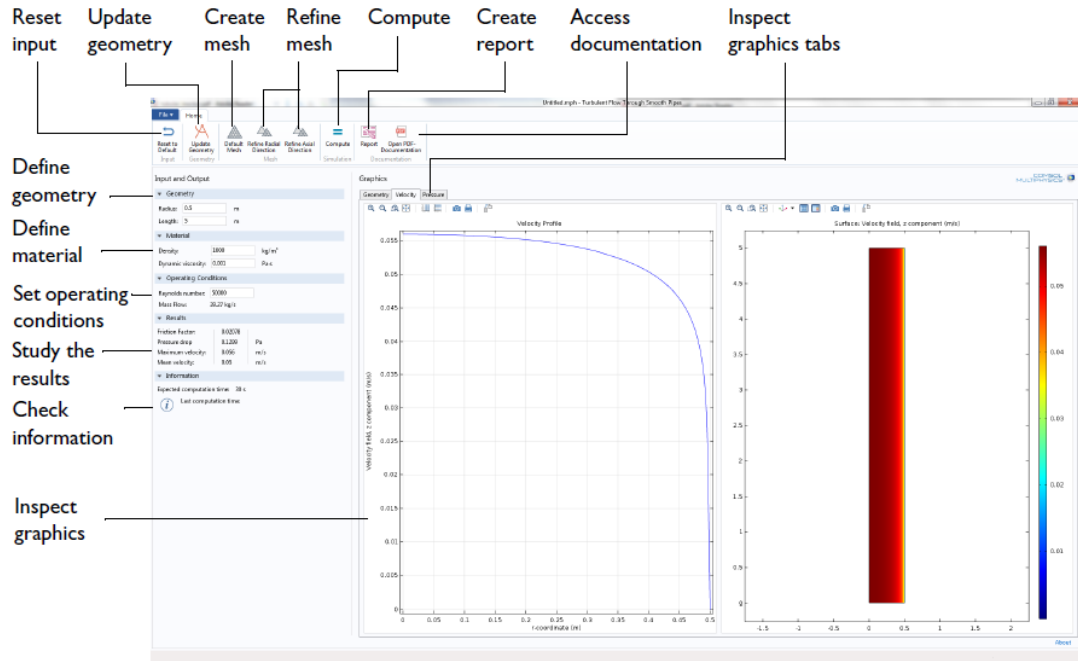


FIGURE 2.9: GUI of the Turbulent Flow through Smooth Pipes application.

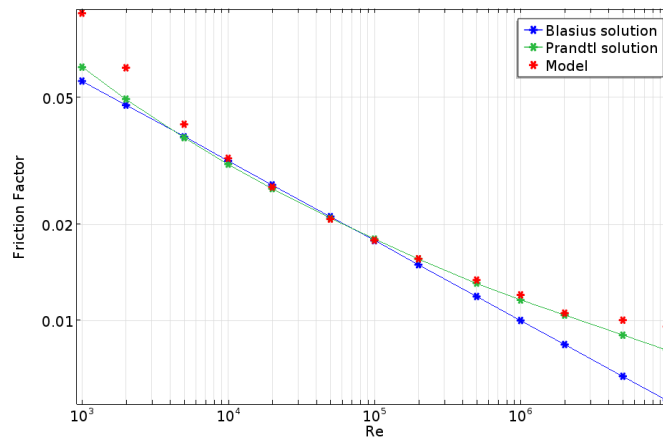


FIGURE 2.10: Dimensionless friction factor.

and mass flow along the tube, as well as the position of the waves at any given time.

This application uses the Wave Form PDE interface, which solves wave equations formulated with first order derivatives in time and space using explicit Runge-Kutta time stepping in combination with a Discontinuous Galerkin method in space. In order to solve the compressible Euler equations, I implemented the Osher-Solomon numerical flux [16].

This application can be used to solve the Sod's shock tube problem [17] (see figure 2.11).

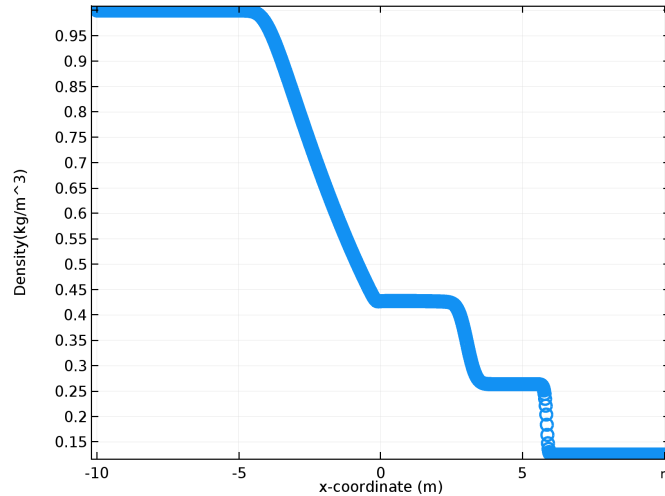


FIGURE 2.11: Distribution of density for the shock tube problem at  $t = 0.01005s$ .

More information about this app can be found in its documentation, see Appendix B.

- Vibrating beam in fluid flow

This application solves for a fluid-structure interaction problem consisting of laminar incompressible flow around an elastic object which results in self-induced oscillations of the structure. The configuration of the problem is based on the Benchmark proposed by Turek and Horn [18], which is based on the flow around a cylinder setting developed in [19]. An incompressible laminar flow enters the channel with a fully developed velocity profile and interacts with the structure consisting of a rigid circular domain and an elastic rectangular beam.

The problem can be solved for different material properties, inlet velocities and sizes of the elastic beam. The user can solve either for the cases proposed by Turek and Horn (three fluid-structure interaction cases and three CFD cases in which the beam consists of a rigid material) or use user-defined input parameters.

The graphical user interface of this app is depicted in figures 2.12 and 2.13.

Apart from developing new models and apps, I have been working on old models and apps. See for example: Solar Panel in Periodic Flow [20], Modular Mixer [21], NACA Airfoil Optimization [22], Water Purification Reactor [23], Micromixer [24], Vacuum Drying [25] and Transonic Flow in a Sajben Diffuser [26].

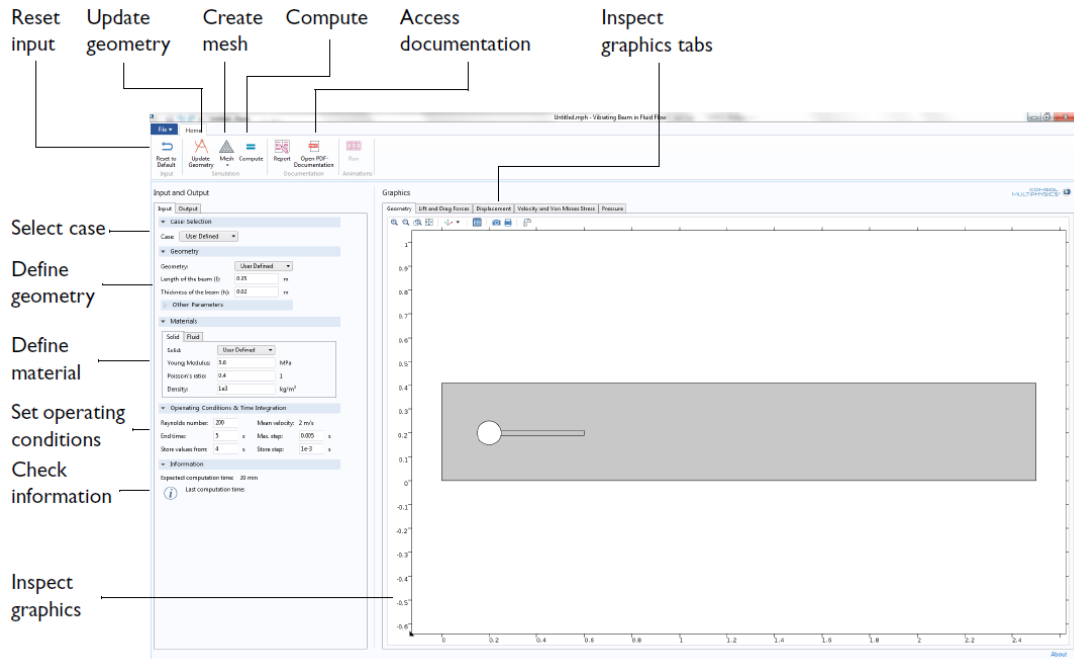


FIGURE 2.12: GUI of the Vibrating Beam in Fluid Flow application when the input mode is active.

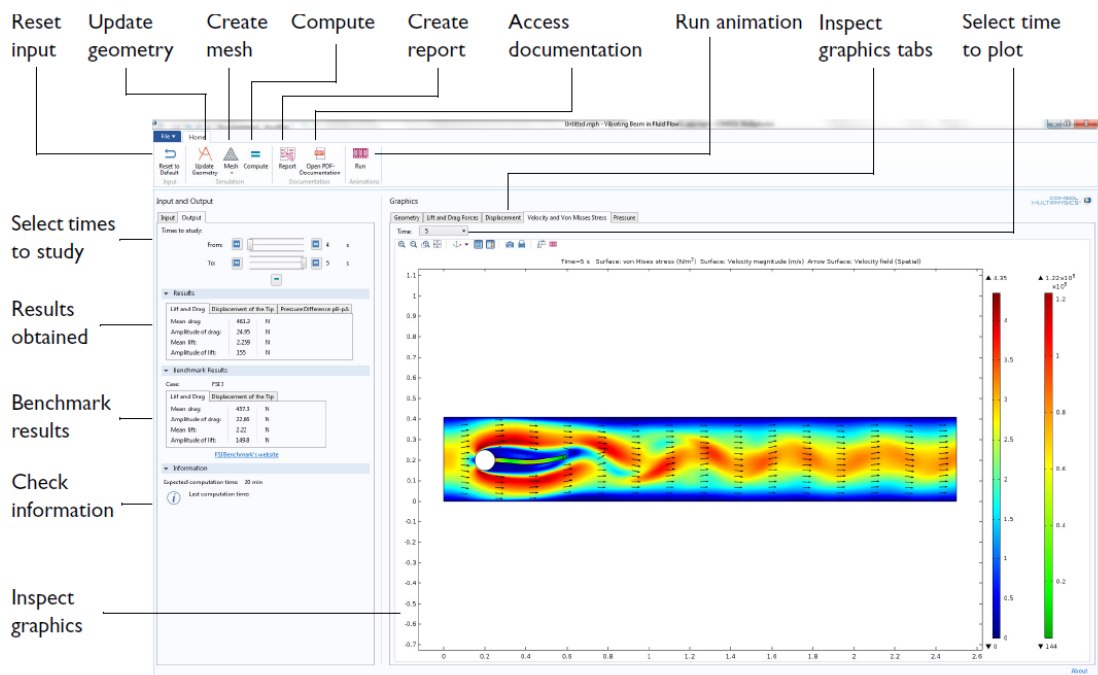


FIGURE 2.13: GUI of the Vibrating Beam in Fluid Flow application when the output mode is active.

## 2.2.2 Development

During my internship, I also worked in development. I implemented a new interface, the Stabilized Convection-Diffusion Equation interface, and added minor modifications to the Fluid flow interfaces.

### 2.2.2.1 Stabilized Convection-Diffusion Equation Interface

The Stabilized Convection-Diffusion Equation is a new interface that solves the following scalar transport equation

$$d_a \frac{\partial u}{\partial t} + \nabla \cdot (-\mathbf{c} \nabla u + \alpha u) + \beta \cdot \nabla u + a u = f$$

using the Finite Element method and including stabilization: Streamline, Crosswind and Isotropic diffusion can be added. Streamline and Crosswind diffusion are selected by default.

The geometry of the problem can be either 1D, 2D, 3D, 1D axisymmetric or 2D axisymmetric. Transient and steady problems are allowed.

The user can define the coefficients in the equation

- Damping or mass coefficient  $d_a$ .
- Diffusion coefficient (second order tensor)  $\mathbf{c}$ .
- Conservative flux convection coefficient (vector)  $\alpha$ .
- Convection coefficient (vector)  $\beta$ .
- Absorption coefficient  $a$ .
- Source term  $f$ .

which can be constants or any valid mathematical expression. Nonlinear coefficients are allowed.

The user can also select the units of the dependent variable  $u$  and the source term. When these units are changed, the units of all the parameters, coefficients and variables used in the interface are updated. The symbol of the dependent variable can also be modified.

The interface includes the following features:

- Initial values: Node used to define the initial values (for transient problems) or the initial solution of the non-linear solver (for steady problems).
- Dirichlet Boundary condition: Constrains the value of the dependent variable.
- No flux boundary condition

$$-\mathbf{n} \cdot (-\mathbf{c}\nabla u + \alpha u + \beta u) = 0$$

- Flux/source boundary condition:

$$-\mathbf{n} \cdot (-\mathbf{c}\nabla u + \alpha u + \beta u) = g - qu$$

Where  $g$  is a flux/source term and  $q$  is an absorption/impedance term. The user can select which fluxes to impose: Diffusive flux ( $\mathbf{n} \cdot (\mathbf{c}\nabla u)$ ), Conservative convective flux ( $-\mathbf{n} \cdot (\alpha u)$ ) or /and Nonconservative convective flux ( $-\mathbf{n} \cdot (\beta u)$ ). If no flux is selected, this feature acts as a "Supersonic Outlet" and the fluxes are computed from inside the domain.

- Symmetry boundary condition:

$$-\mathbf{n} \cdot (-\mathbf{c}\nabla u) = 0$$

- Periodic boundary condition:

Two boundaries must be selected ( source and destination boundaries ). The following options are available:

- Continuity:

$$u_{dst} = u_{src} - \Delta u$$

$$-\mathbf{n}_{dst} \cdot (-\mathbf{c}\nabla u)_{dst} = \mathbf{n}_{src} \cdot (-\mathbf{c}\nabla u)_{src}$$

where  $\Delta u$  is an offset.

- Antiperiodicity:

$$u_{dst} = -u_{src}$$

$$-\mathbf{n}_{dst} \cdot (-\mathbf{c}\nabla u)_{dst} = \mathbf{n}_{src} \cdot (-\mathbf{c}\nabla u)_{src}$$

The periodic boundary condition uses the Discontinuous Galerkin method.

- Continuity:

$$u_{dst} = u_{src}$$

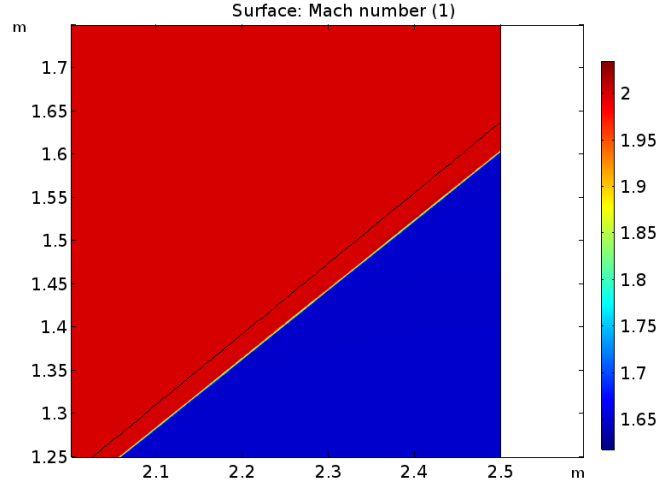


FIGURE 2.14: Position of the oblique shock obtained in COMSOL. The black line is the position of the shock according to the analytical solution of the problem.

$$-\mathbf{n}_{dst} \cdot (-\mathbf{c}\nabla u)_{dst} = \mathbf{n}_{src} \cdot (-\mathbf{c}\nabla u)_{src}$$

This condition is only applicable on interior boundaries.

- Point/Edge source: Applies a source term at an edge or at a point.

### 2.2.2.2 Contributions to the Fluid Flow interfaces

During my stay in COMSOL, I have also had the opportunity to add small contributions to the code for the Fluid Flow interfaces, including some modifications to the default results plots, coefficients used in the stabilization, boundary conditions and fixing some minor bugs.

One of the projects I have been working on is related to the conservation properties of the Navier-Stokes equations. This project started when I detected that oblique shocks were slightly off compared to the analytic solution ( see for example reference [7] ) due to lack of conservation ( see figures 2.14 and 2.15 ).

COMSOL solves the following non-conservative system of equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.1)$$

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} = \rho \mathbf{F} - \nabla p + \nabla \cdot \tau \quad (2.2)$$

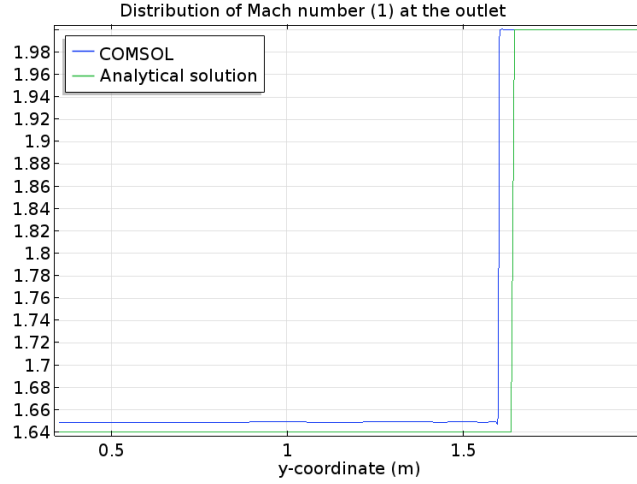


FIGURE 2.15: Distribution of Mach number at the outlet.

$$\rho C_p \frac{\partial T}{\partial t} + \rho C_p \mathbf{u} \cdot \nabla T + \nabla \cdot \mathbf{q} = \alpha_p T \left( \frac{\partial p}{\partial t} + \mathbf{u} \cdot \nabla p \right) + \tau : \nabla \mathbf{u} + Q \quad (2.3)$$

However, the conservative system of equations that should be solved is instead

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{F} - \nabla p + \nabla \cdot \tau \quad (2.4)$$

$$\frac{\partial}{\partial t} \left( \rho \left( E_i + \frac{1}{2} \mathbf{u}^2 \right) \right) + \nabla \cdot \left( \rho \mathbf{u} \left( E_i + \frac{1}{2} \mathbf{u}^2 \right) \right) + \nabla \cdot \mathbf{q} = -\nabla \cdot (p \mathbf{u}) + \nabla \cdot (\tau \mathbf{u}) + \rho \mathbf{u} \cdot \mathbf{F} + Q \quad (2.5)$$

Note that for incompressible flow the energy equation (2.3 and 2.5) is not needed and the continuity equation (2.1) reduces to  $\nabla \cdot \mathbf{u} = 0$ .

The momentum equation solved in COMSOL (2.2) will be equivalent to the conservative momentum equation (2.4) if the continuity equation (2.1) is fulfilled. The energy equation (2.3) will also be conservative and equivalent to (2.5) if both the momentum (2.2) and continuity (2.1) equations are satisfied. However, these assumptions are not fulfilled if the continuity and momentum equations are stabilized, or if we define the spaces of velocity and pressure such that the inf-sup stability condition (see for example [27]) is fulfilled.



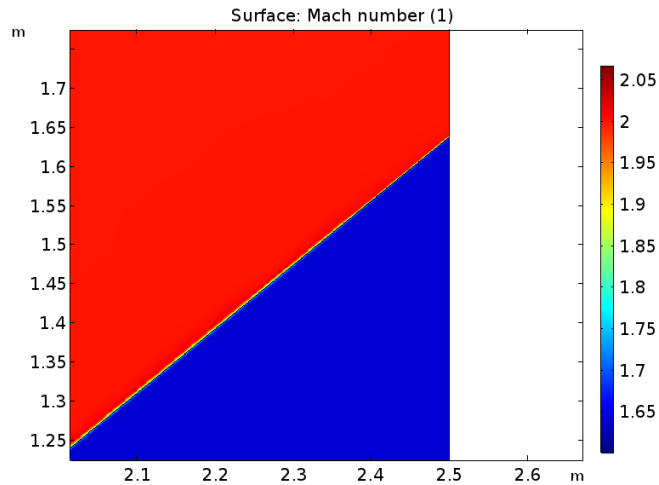


FIGURE 2.16: Position of the oblique shock obtained after adding the proposed corrections. The black line is the position of the shock according to the analytical solution of the problem.

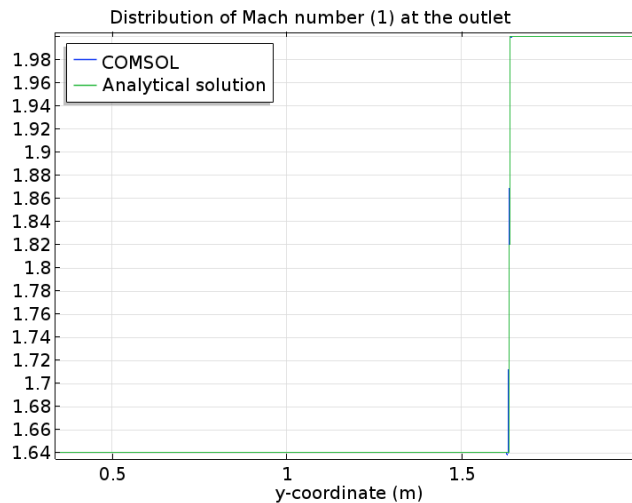


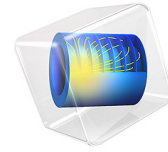
FIGURE 2.17: Distribution of Mach number at the outlet obtained after adding the proposed corrections.

The easiest way to achieve convergence would be to solve equations (2.4) and (2.5) instead of equations (2.2) and (2.3). However, this leads to a system of equations which is more complicated to solve, is less robust and gives poorer results. An alternative option is to add correction terms that compensate the added stabilization, and to in turn stabilize these. One example of this can be found in a paper by T. J. Hughes and G. N. Wells (see [28]).

After adding the terms I propose and stabilizing them, I obtained a solution for oblique shocks which matches the analytical solution (see figures 2.16 and 2.17). Moreover, these conservative corrections are also positive for other problems, including incompressible flow and heat transfer problems.

## Appendix A

# Documentation of the Supersonic Air-to-Air Ejector model



# Supersonic Air-to-Air Ejector

## *Introduction*

---

In this study, the compressible turbulent gas flow in a supersonic air ejector is modeled using the High Mach Number Flow interface in COMSOL Multiphysics. Ejectors are simple mechanical components used to induce a secondary flow by momentum and energy transfer from a high-velocity primary jet. The high-energy fluid (primary flow) passes through a convergent-divergent nozzle and reaches supersonic conditions. After exiting the nozzle, it interacts with the secondary flow and is accelerated through an entrainment-induced effect. The mixing between both flows takes place along a constant-area duct called the mixing chamber where complex interactions between the mixing layer and shocks can be observed. A diffuser is usually placed before the outlet to recover pressure and bring the flow back to stagnation.

Ejectors are used for a wide range of applications, including industrial refrigeration, vacuum generation, gas recirculation, and thrust augmentation in aircraft propulsion systems. A great effort has been made to determine their optimum design and operating conditions, as well as how to describe the flow within them ([Ref. 1](#)).

In this model, an ejector working with air in both the primary and secondary streams is modeled. The geometry and boundary conditions are based on the ones used in [Ref. 2](#), and [Ref. 3](#). The items of interest are the primary and secondary mass flows, the static pressure distribution along the centerline of the ejector, and the resolution of the flow in the mixing region.

## *Model Definition*

---

The geometry of the ejector is depicted in [Figure 1](#), and its dimensions can be found in [Table 1](#). A two-dimensional axisymmetric geometry is used to approximate the 3D geometry of the device and to reduce the size of the problem.

The flow velocity in the ejector is large enough to introduce significant variations in the density and temperature of the fluid, and the flow is governed by the fully compressible Navier-Stokes equations. Moreover, the Mach number is expected to be larger than one in the divergent section of the primary nozzle, as well as in the region of the mixing chamber upstream of the mixing zone. Due to the interaction between the boundary and mixing layers, the deceleration from supersonic to subsonic flow does not take place through a single normal shock, but through a complex succession of shocks called shock train or pseudo-shock wave phenomenon ([Ref. 4](#)). Thus, the mesh will have to be fine

enough to accurately capture this phenomenon.

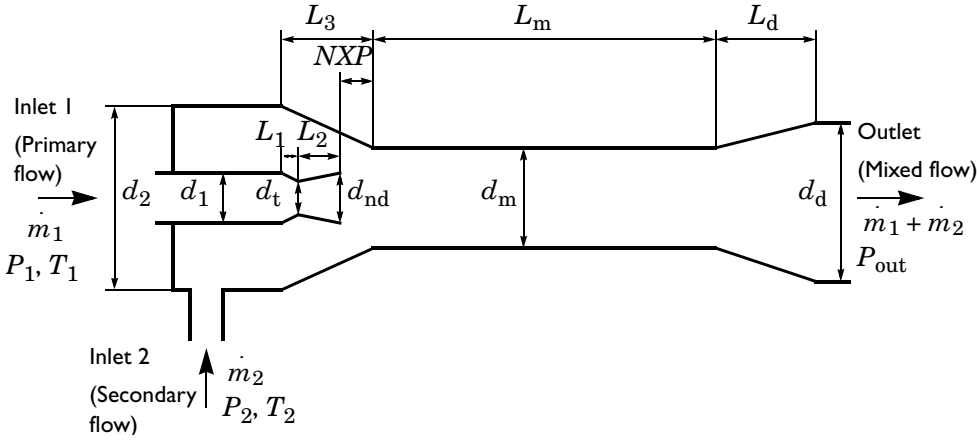


Figure 1: The geometry of the ejector.

TABLE 1: DIMENSIONS OF THE EJECTOR IN MM

$d_1$	$d_2$	$d_t$	$d_{nd}$	$d_m$	$d_d$
16	160	8	12	24	51
$L_1$	$L_2$	$L_3$	$L_m$	$L_d$	NXP
7	23	90	240	70	15

The problem will be modeled using the Favre-averaged Navier-Stokes equations and the standard  $k-\epsilon$  turbulence model.

Both primary and secondary flows are air with a specific gas constant of 287 J/(kg·K) and a ratio of specific heats of 1.4. The dynamic viscosity and thermal conductivity of the air are computed using Sutherland's law.

## BOUNDARY CONDITIONS

### Inlet

The flow at the inlets is specified in terms of its total properties:  $T_0 = 300$  K and  $P_0 = 5$  atm for the primary flow and  $T_0 = 300$  K and  $P_0 = 0.55$  atm for the secondary flow.

The inlet conditions are applied using an Inlet feature, where the Flow condition is specified to be Characteristics based. This provides a numerically consistent boundary condition by evaluating the flow characteristics at the inlet.

The velocities at both inlets are unknown. However, they are expected to be very small compared to the velocities inside the nozzle and mixing chamber. The values that must be

prescribed at the inlet are the total values of temperature and pressure, which define the energy of the flow. The Mach number can be set to 0 and will be determined by the characteristics based boundary condition at the inlet. This provides a good initial solution, but the total values of pressure and temperature may differ slightly. The solution can be improved if the problem is solved again setting the Mach number to the values computed by the characteristics based boundary condition at the inlets, which are 0.14 and 0.0065 for the primary and secondary inlets, respectively.

The inlet values for the turbulent kinetic energy,  $k$ , and the turbulent dissipation rate,  $\varepsilon$ , are approximated from the turbulent intensity,  $I_T$ , and turbulence length scale,  $L_T$ . Turbulent intensity is set by default to 0.05 (5%). The length scale can be approximated as 7% of the pipe diameter or hydraulic diameter.

For more background on this boundary condition, see the *CFD Module User's Guide*.

### Outlet

The flow reaches supersonic conditions inside the ejector. However, it is expanded and decelerated along the mixing chamber and the diffuser, reaching subsonic conditions before being discharged into the atmosphere. The outlet is then subsonic with atmospheric static pressure. This is modeled using an Outlet node with the Flow condition set to subsonic.

## Results and Discussion

---

The mass flows obtained are depicted in [Table 2](#). The distributions of Mach number and velocity inside the ejector are depicted in [Figure 2](#) and [Figure 3](#). The primary flow is accelerated in the convergent section of the nozzle, reaching sonic conditions at the throat, and is expanded further in the divergent section. At the outlet of the primary nozzle, the secondary flow acts as an artificial wall for the primary flow. This leads to the formation of virtual nozzle throats, and a succession of expansion and compression waves can be observed in the region upstream of the mixing zone. Then the flow is decelerated along the constant-area duct and brought back to stagnation in the diffuser. The region where both flows mix can be visualized by plotting the turbulent kinetic energy, see [Figure 4](#).

The distribution of pressure along the centerline and walls of the mixing chamber is plotted in [Figure 5](#). At the centerline of the duct, the flow successively changes from supersonic to subsonic flow via multiple shocks. However, this cannot be detected by wall pressure measurements because the surface pressures tend to be smeared out due to the dissipation in the boundary layer (see [Ref. 4](#)). The distribution of temperature is shown in [Figure 6](#). Very low temperatures can be observed inside the device. This must be taken

into account when designing an ejector, specially when working with two-phase flows. The results obtained correlate well with [Ref. 2](#) and [Ref. 3](#).

TABLE 2: MASS FLOWS

$\dot{m}_1$	$\dot{m}_2$	$\dot{m}_{mixed}$
0.058 kg/s	0.029 kg/s	0.087 kg/s

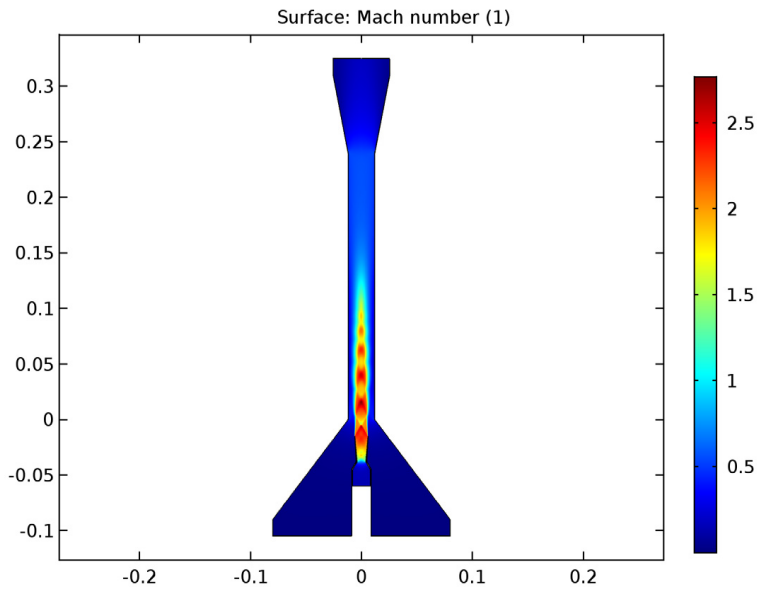


Figure 2: Mach number distribution.

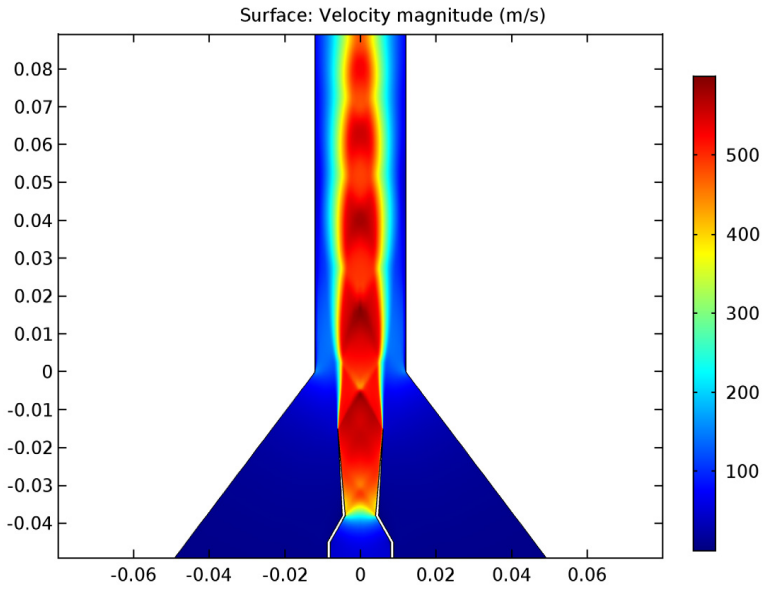


Figure 3: The distribution of velocity in the nozzle and the mixing chamber.

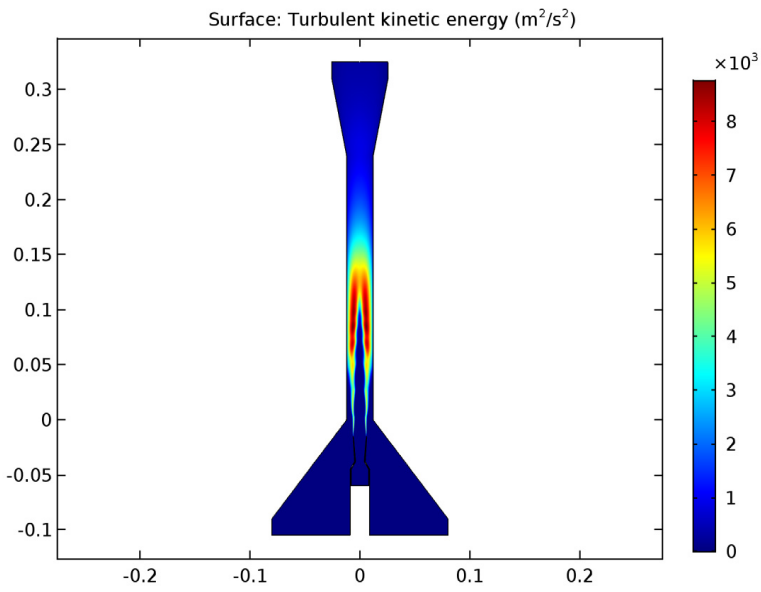


Figure 4: The distribution of turbulent kinetic energy. The mixing region is clearly identified.



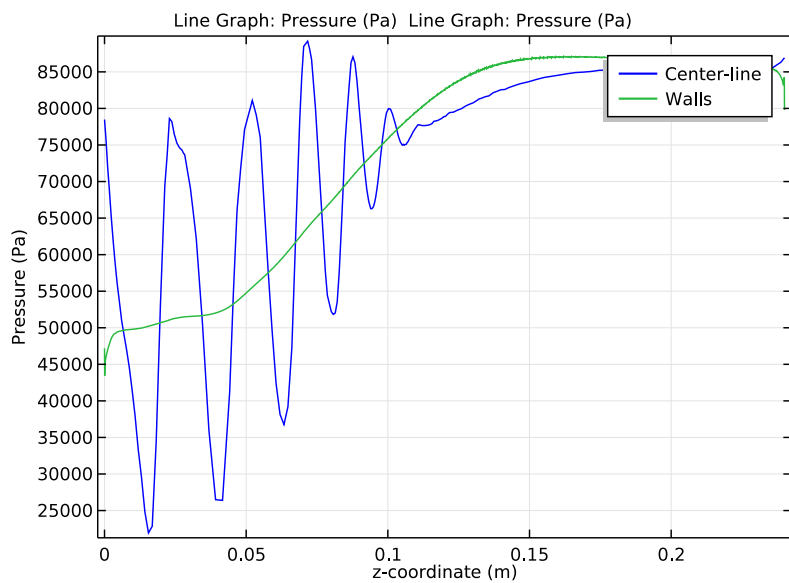


Figure 5: The distribution of pressure along the mixing chamber.

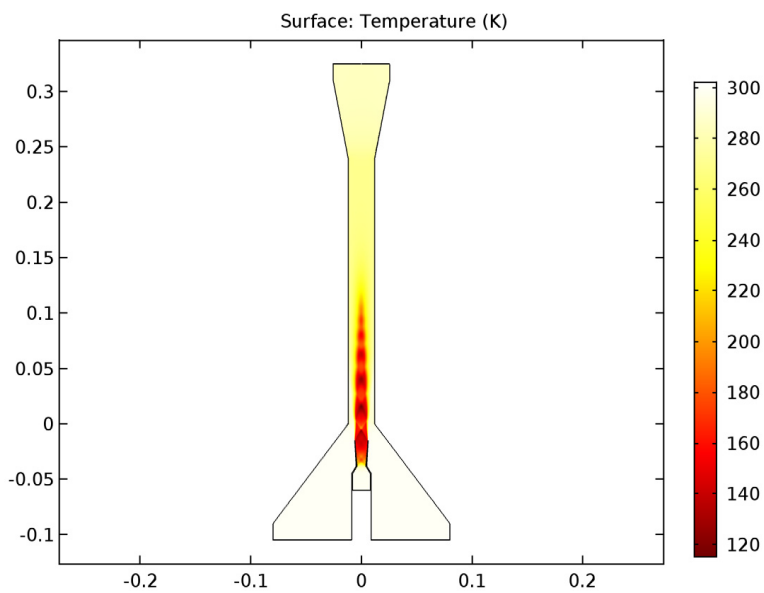


Figure 6: The distribution of temperature. Very low temperatures can be observed at the outlet of the nozzle. The temperature at the outlet of the ejector is lower than at both inlets.

## *Notes About the COMSOL Implementation*

---

The present model is highly nonlinear and sensitive to the solution procedure. The mesh needed to capture the interaction between the shocks and the mixing layer, and to resolve the solution near the walls, is extremely fine. However, convergence may be hard to achieve with such a fine mesh unless a very good initial solution is used. A way to overcome this is to first solve the problem on a coarse mesh and then refine it. The solution on the coarse mesh provides good initial values, but lacks accuracy in three important areas: wall resolution, capture of shocks, and resolution of the mixing layer. The adaptive mesh refinement feature could be used to overcome this. However, in order to fully resolve the mesh, a high element growth rate would be needed, leading to convergence problems. An alternative option is to first refine manually both on the boundary layers and in the nozzle, and then to use the adaptive mesh refinement feature to resolve the mixing layer.

## *References*

---

1. S. He, Y. Li, and R.Z. Wang, “Progress of Mathematical Modeling on Ejectors,” *Renew. Sustain. Energy Rev.*, vol. 13, pp 1760–1780, 2009.
2. Y. Bartosiewicz, Zine Aidoun, P. Desevaux, and Yves Mercadier, “Numerical and Experimental Investigations on Supersonic Ejectors,” *Int. J. of Heat and Fluid Flow*, vol. 26, pp 56–70, 2005.
3. P. Desevaux, A. Bouhangel, and E. Gavignet, “Flow Visualization in Supersonic Ejectors Using Laser Tomography Techniques,” *Int. J. of Refrigeration*, vol. 34, pp 1633–1640, 2010.
4. F. Gnani, H. Zare-Behtash, and K. Kontis, “Pseudo-shock Waves and their Interactions in High-speed Intakes,” *Progress in Aerospace Sciences*, vol. 82, pp 36–56, 2016.

---

**Application Library path:** CFD\_Module/High\_Mach\_Number\_Flow/  
supersonic\_ejector

---

## *Model Instructions*

---

From the **File** menu, choose **New**.

## NEW

In the **New** window, click **Model Wizard**.

## MODEL WIZARD

- 1 In the **Model Wizard** window, click **2D Axisymmetric**.
- 2 In the **Select Physics** tree, select **Fluid Flow>High Mach Number Flow>Turbulent Flow>High Mach Number Flow, k-ε (hmnf)**.
- 3 Click **Add**.
- 4 Click **Study**.
- 5 In the **Select Study** tree, select **Preset Studies>Stationary**.
- 6 Click **Done**.

## GEOMETRY I

The model geometry is available as a parameterized geometry sequence in a separate MPH-file.

- 1 On the **Geometry** toolbar, click **Insert Sequence**.
- 2 Browse to the model's Application Libraries folder and double-click the file `supersonic_ejector_geom_sequence.mph`.
- 3 On the **Geometry** toolbar, click **Build All**.

## GLOBAL DEFINITIONS

### Parameters

- 1 In the **Model Builder** window, under **Global Definitions** click **Parameters**.
- 2 In the **Settings** window for Parameters, locate the **Parameters** section.
- 3 In the table, enter the following settings:

Name	Expression	Value	Description
P1	5[atm]	5.066E5 Pa	Total pressure, primary flow
P2	0.55[atm]	5.573E4 Pa	Total pressure, secondary flow
Pout	1[atm]	1.013E5 Pa	Pressure, outlet
T1	300[K]	300 K	Total temperature, primary flow
T2	T1	300 K	Total temperature, secondary flow

Name	Expression	Value	Description
Rs	287 [J/kg/K]	287 J/(kg·K)	Specific gas constant
gamma	1.41	1.41	Ratio of specific heats

## HIGH MACH NUMBER FLOW, K-ε (HMNF)

### Fluid 1

- 1 In the **Model Builder** window, under **Component 1 (comp1)>High Mach Number Flow, k-ε (hmnf)** click **Fluid 1**.
- 2 In the **Settings** window for Fluid, locate the **Thermodynamics** section.
- 3 From the  $R_g$  list, choose **User defined**. In the associated text field, type Rs.
- 4 From the **Specify Cp or  $\gamma$**  list, choose **Ratio of specific heats**.
- 5 From the  $\gamma$  list, choose **User defined**. In the associated text field, type gamma.

### Initial Values 1

- 1 In the **Model Builder** window, under **Component 1 (comp1)>High Mach Number Flow, k-ε (hmnf)** click **Initial Values 1**.
- 2 In the **Settings** window for Initial Values, locate the **Initial Values** section.
- 3 Specify the **u** vector as

0	r
100	z

- 4 In the  $p$  text field, type 2[atm].

### Inlet 1

- 1 On the **Physics** toolbar, click **Boundaries** and choose **Inlet**.
- 2 In the **Settings** window for Inlet, locate the **Boundary Selection** section.
- 3 From the **Selection** list, choose **Primary inlet**.
- 4 Locate the **Flow Properties** section. From the **Input state** list, choose **Total**.
- 5 In the  $p_{0,tot}$  text field, type P1.
- 6 In the  $T_{0,tot}$  text field, type T1.
- 7 In the  $Ma_0$  text field, type 0.14.
- 8 Locate the **Turbulence Properties** section. In the  $L_T$  text field, type d\_primary\*0.07.

### Inlet 2

- 1 On the **Physics** toolbar, click **Boundaries** and choose **Inlet**.

- 2 In the **Settings** window for Inlet, locate the **Boundary Selection** section.
- 3 From the **Selection** list, choose **Secondary inlet**.
- 4 Locate the **Flow Properties** section. From the **Input state** list, choose **Total**.
- 5 In the  $p_{0,tot}$  text field, type P2.
- 6 In the  $T_{0,tot}$  text field, type T2.
- 7 In the  $Ma_0$  text field, type 0.0065.
- 8 Locate the **Turbulence Properties** section. In the  $L_T$  text field, type  $0.07 * (d_{secondary} - (d_{primary} + thickness))$ .

#### *Outlet 1*

- 1 On the **Physics** toolbar, click **Boundaries** and choose **Outlet**.
- 2 In the **Settings** window for Outlet, locate the **Boundary Selection** section.
- 3 From the **Selection** list, choose **Outlet**.
- 4 Locate the **Flow Condition** section. From the **Flow condition** list, choose **Subsonic**.
- 5 Locate the **Flow Properties** section. From the **Boundary condition** list, choose **Pressure**.
- 6 In the  $p_0$  text field, type Pout.

The next step is to generate a coarse mesh to compute an initial solution.

#### **MESH 1**

- 1 In the **Model Builder** window, under **Component 1 (comp1)** click **Mesh 1**.
- 2 In the **Settings** window for Mesh, locate the **Mesh Settings** section.
- 3 From the **Element size** list, choose **Coarse**.
- 4 Click **Build All**.

#### **COMPONENT 1 (COMP1)**

Generate a second mesh and refine it in the boundary layer and on the nozzle.

#### *Mesh 2*

On the **Mesh** toolbar, click **Add Mesh**.

#### **MESH 2**

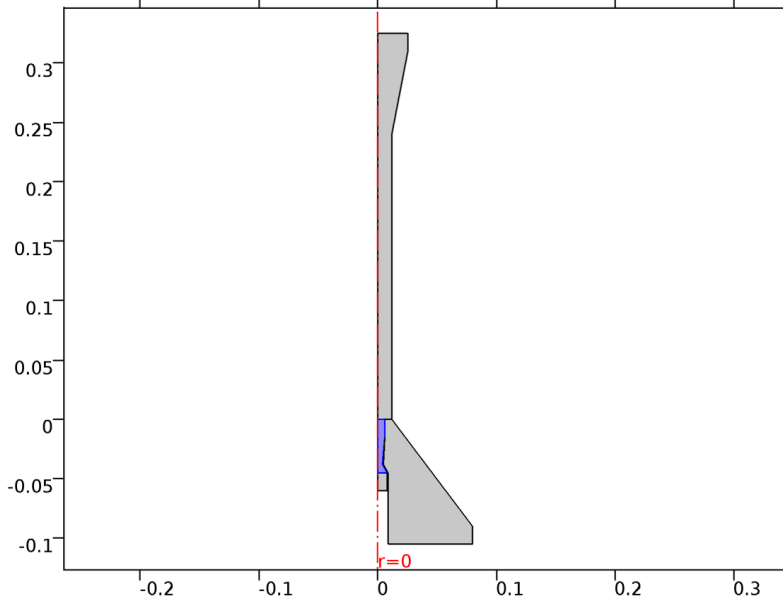
In the **Model Builder** window, under **Component 1 (comp1)**>**Meshes** right-click **Mesh 2** and choose **Size**.

#### *Size 1*

- 1 In the **Settings** window for Size, locate the **Geometric Entity Selection** section.



4 Select Domain 2 only.



5 Locate the **Element Size** section. From the **Calibrate for** list, choose **Fluid dynamics**.

6 From the **Predefined** list, choose **Extremely fine**.

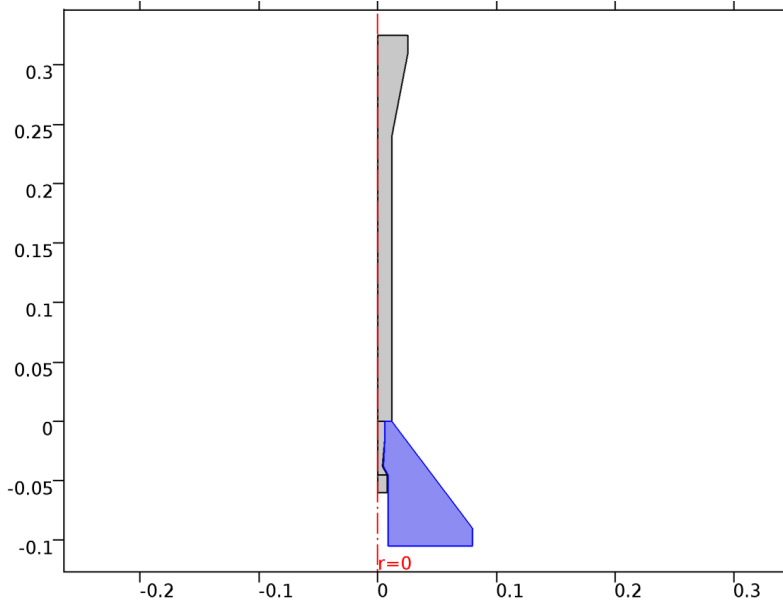
*Size 4*

1 Right-click **Mesh 2** and choose **Size**.

2 In the **Settings** window for Size, locate the **Geometric Entity Selection** section.

3 From the **Geometric entity level** list, choose **Domain**.

4 Select Domain 3 only.



5 Locate the **Element Size** section. From the **Calibrate for** list, choose **Fluid dynamics**.

6 From the **Predefined** list, choose **Coarser**.

#### *Size*

1 In the **Model Builder** window, under **Component 1 (comp1)>Meshes>Mesh 2** click **Size**.

2 In the **Settings** window for Size, locate the **Element Size** section.

3 From the **Calibrate for** list, choose **Fluid dynamics**.

4 From the **Predefined** list, choose **Fine**.

#### *Corner Refinement 1*

1 In the **Model Builder** window, right-click **Mesh 2** and choose **Corner Refinement**.

2 In the **Settings** window for Corner Refinement, locate the **Boundary Selection** section.

3 Select the **Active** toggle button.

4 From the **Selection** list, choose **Walls**.

#### *Free Triangular 1*

Right-click **Mesh 2** and choose **Free Triangular**.

#### *Boundary Layer Properties*

1 Right-click **Mesh 2** and choose **Boundary Layers**.



- 2 In the **Settings** window for Boundary Layer Properties, locate the **Boundary Selection** section.
- 3 From the **Selection** list, choose **Walls**.
- 4 In the list, choose **4**, **5**, and **8**.
- 5 Click **Remove from Selection**.
- 6 Locate the **Boundary Layer Properties** section. In the **Number of boundary layers** text field, type 5.
- 7 From the **Thickness of first layer** list, choose **Manual**.
- 8 In the **Thickness** text field, type 5e-5.

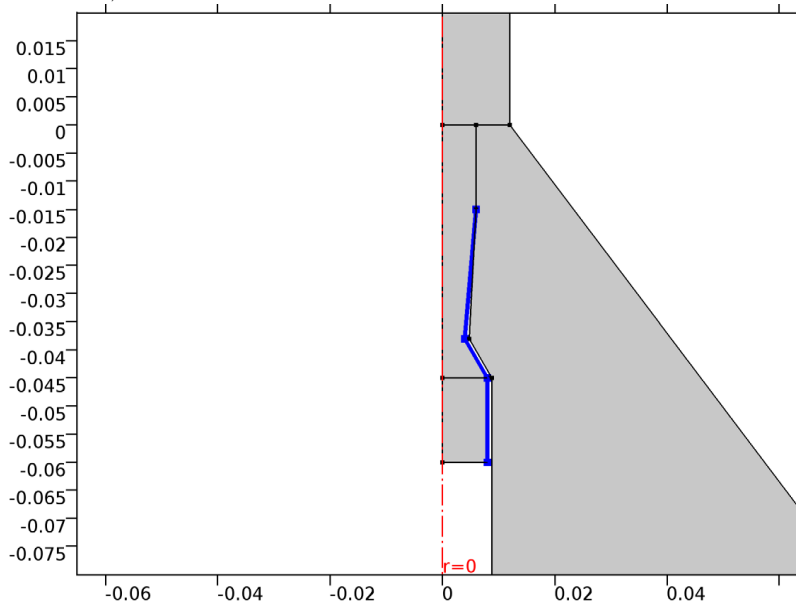
#### *Boundary Layers 1*

In the **Model Builder** window, under **Component 1 (comp1)>Meshes>Mesh 2** right-click **Boundary Layers 1** and choose **Boundary Layer Properties**.

#### *Boundary Layer Properties 1*

- 1 In the **Settings** window for Boundary Layer Properties, locate the **Boundary Layer Properties** section.
- 2 In the **Number of boundary layers** text field, type 10.
- 3 From the **Thickness of first layer** list, choose **Manual**.
- 4 In the **Thickness** text field, type 1e-5.

5 Select Boundaries 4, 5, and 8 only (select the walls of both the primary inlet and the nozzle).



The last step is to convert the boundary layer elements into triangles. This is necessary since the adaptive mesh refinement feature is only implemented for triangular meshes.

*Convert 1*

- 1 In the **Model Builder** window, right-click **Mesh 2** and choose **More Operations>Convert**.
- 2 In the **Settings** window for Convert, locate the **Element Split Method** section.
- 3 From the **Element split method** list, choose **Insert center points**.
- 4 Click **Build All**.

**STUDY 1**

*Step 1: Stationary*

- 1 In the **Settings** window for Stationary, click to expand the **Mesh selection** section.
- 2 Locate the **Mesh Selection** section. In the table, enter the following settings:

Geometry	Mesh
Geometry 1	mesh 1

Edit the study step to solve the model in two steps in order to increase the robustness of the solution procedure. First, solve for  $P_2=P_{out}$  (no adverse gradient of pressure), and then solve the problem again adding the adverse gradient of pressure ( $P_2$  smaller than

$P_{out}$ ). Continuation is used, implying that the previous solution will be used as initial condition for the next step.

- 3 Click to expand the **Study extensions** section. Locate the **Study Extensions** section. Select the **Auxiliary sweep** check box.
- 4 Click **Add**.
- 5 In the table, enter the following settings:

Parameter name	Parameter value list	Parameter unit
P2	Pout 0.55[atm]	

Solve the problem with a finer mesh, and use the adaptive mesh refinement feature to resolve the mixing layer. The solution of study step 1 is used as the initial guess.

### Stationary 2

On the **Study** toolbar, click **Study Steps** and choose **Stationary>Stationary**.

### Step 2: Stationary 2

- 1 In the **Settings** window for Stationary, click to expand the **Mesh selection** section.
- 2 Locate the **Mesh Selection** section. In the table, enter the following settings:

Geometry	Mesh
Geometry 1	mesh2

- 3 Click to expand the **Study extensions** section. Locate the **Study Extensions** section. Select the **Adaptive mesh refinement** check box.
- 4 On the **Study** toolbar, click **Compute**.

## RESULTS

### Derived Values

Compute the primary mass flow.

### Line Integration 1

- 1 On the **Results** toolbar, click **More Derived Values** and choose **Integration>Line Integration**.
- 2 In the **Settings** window for Line Integration, type Primary Mass Flow in the **Label** text field.
- 3 Locate the **Data** section. From the **Data set** list, choose **Study 1/Adaptive Mesh Refinement 1 (sol3)**.

- 4 Locate the **Selection** section. From the **Selection** list, choose **Primary inlet**.
- 5 Locate the **Expressions** section. In the table, enter the following settings:

Expression	Unit	Description
hmnf.rho*w	kg/(m*s)	

- 6 Locate the **Integration Settings** section. Select the **Compute surface integral** check box.
- 7 Click **Evaluate**.

*Derived Values*

Compute the secondary mass flow.

*Line Integration 2*

- 1 On the **Results** toolbar, click **More Derived Values** and choose **Integration>Line Integration**.
- 2 In the **Settings** window for Line Integration, type Secondary mass flow in the **Label** text field.
- 3 Locate the **Data** section. From the **Data set** list, choose **Study I/Adaptive Mesh Refinement I (sol3)**.
- 4 Locate the **Selection** section. From the **Selection** list, choose **Secondary inlet**.
- 5 Locate the **Expressions** section. In the table, enter the following settings:

Expression	Unit	Description
hmnf.rho*w	kg/(m*s)	

- 6 Locate the **Integration Settings** section. Select the **Compute surface integral** check box.
- 7 Click **Evaluate**.

*Derived Values*

Compute the mixed mass flow.

*Line Integration 3*

- 1 On the **Results** toolbar, click **More Derived Values** and choose **Integration>Line Integration**.
- 2 In the **Settings** window for Line Integration, type Mixed mass flow in the **Label** text field.
- 3 Locate the **Data** section. From the **Data set** list, choose **Study I/Adaptive Mesh Refinement I (sol3)**.
- 4 Locate the **Selection** section. From the **Selection** list, choose **Outlet**.

5 Locate the **Expressions** section. In the table, enter the following settings:

Expression	Unit	Description
hmnf.rho*w	kg/(m*s)	

6 Locate the **Integration Settings** section. Select the **Compute surface integral** check box.

7 Click **Evaluate**.

#### *Derived Values*

Check if the average of the Mach number at the primary inlet is similar to the value defined in **Inlet 1**.

#### *Line Average 1*

1 On the **Results** toolbar, click **More Derived Values** and choose **Average>Line Average**.

2 In the **Settings** window for Line Average, type Mach number, primary inlet in the **Label** text field.

3 Locate the **Data** section. From the **Data set** list, choose **Study 1/Adaptive Mesh Refinement 1 (sol3)**.

4 Locate the **Selection** section. From the **Selection** list, choose **Primary inlet**.

5 Locate the **Expressions** section. In the table, enter the following settings:

Expression	Unit	Description
hmnf.Ma	1	Mach number

6 Click **Evaluate**.

#### *Derived Values*

Check if the average of the Mach number at the secondary inlet is similar to the value defined in **Inlet 2**.

#### *Line Average 2*

1 On the **Results** toolbar, click **More Derived Values** and choose **Average>Line Average**.

2 In the **Settings** window for Line Average, type Mach number, secondary inlet in the **Label** text field.

3 Locate the **Data** section. From the **Data set** list, choose **Study 1/Adaptive Mesh Refinement 1 (sol3)**.

4 Locate the **Selection** section. From the **Selection** list, choose **Secondary inlet**.

5 Locate the **Expressions** section. In the table, enter the following settings:

Expression	Unit	Description
hmnf.Ma	1	Mach number

6 Click **Evaluate**.

In case that the Mach number at one or both inlets diverge from the imposed values, the problem should be solved again using the new Mach numbers at the inlet.

#### *Mirror 2D 1*

- 1 On the **Results** toolbar, click **More Data Sets** and choose **Mirror 2D**.
- 2 In the **Settings** window for Mirror 2D, locate the **Data** section.
- 3 From the **Data set** list, choose **Study 1/Adaptive Mesh Refinement 1 (sol3)**.
- 4 Click **Plot**.

Plot the Mach number.

#### *2D Plot Group 7*

- 1 On the **Results** toolbar, click **2D Plot Group**.
- 2 In the **Settings** window for 2D Plot Group, type Mach Number in the **Label** text field.
- 3 Locate the **Data** section. From the **Data set** list, choose **Mirror 2D 1**.

#### *Surface 1*

- 1 Right-click **Mach Number** and choose **Surface**.
- 2 In the **Settings** window for Surface, locate the **Expression** section.
- 3 In the **Expression** text field, type hmnf.Ma.
- 4 On the **Mach Number** toolbar, click **Plot**.
- 5 Click the **Zoom Extents** button on the **Graphics** toolbar (see [Figure 2](#)).

Plot the turbulent kinetic energy.

#### *2D Plot Group 8*

- 1 On the **Home** toolbar, click **Add Plot Group** and choose **2D Plot Group**.
- 2 In the **Settings** window for 2D Plot Group, type Turbulent Kinetic Energy in the **Label** text field.
- 3 Locate the **Data** section. From the **Data set** list, choose **Mirror 2D 1**.

#### *Surface 1*

- 1 Right-click **Turbulent Kinetic Energy** and choose **Surface**.
- 2 In the **Settings** window for Surface, locate the **Expression** section.

- 3 In the **Expression** text field, type  $k$ .
- 4 On the **Turbulent Kinetic Energy** toolbar, click **Plot** (see [Figure 4](#)).  
Plot the temperature.

#### *2D Plot Group 9*

- 1 On the **Home** toolbar, click **Add Plot Group** and choose **2D Plot Group**.
- 2 In the **Settings** window for 2D Plot Group, type Temperature, 2D in the **Label** text field.
- 3 Locate the **Data** section. From the **Data set** list, choose **Mirror 2D 1**.

#### *Surface 1*

- 1 Right-click **Temperature, 2D** and choose **Surface**.
- 2 In the **Settings** window for Surface, locate the **Expression** section.
- 3 In the **Expression** text field, type  $T$ .
- 4 Locate the **Coloring and Style** section. From the **Color table** list, choose **ThermalLight**.
- 5 On the **Temperature, 2D** toolbar, click **Plot** (see [Figure 6](#)).

Plot the evolution of pressure along the mixing chamber.

#### *Cut Line 2D 1*

- 1 On the **Results** toolbar, click **Cut Line 2D**.
- 2 In the **Settings** window for Cut Line 2D, type Center-line in the **Label** text field.
- 3 Locate the **Data** section. From the **Data set** list, choose **Study 1/Adaptive Mesh Refinement 1 (sol3)**.
- 4 Locate the **Line Data** section. In row **Point 2**, set  $r$  to 0.
- 5 In row **Point 2**, set  $z$  to  $L_{\text{mixing}}$ .
- 6 Click **Plot**.

#### *Cut Line 2D 2*

- 1 On the **Results** toolbar, click **Cut Line 2D**.
- 2 In the **Settings** window for Cut Line 2D, type Wall in the **Label** text field.
- 3 Locate the **Data** section. From the **Data set** list, choose **Study 1/Adaptive Mesh Refinement 1 (sol3)**.
- 4 Locate the **Line Data** section. In row **Point 1**, set  $r$  to  $d_{\text{mixing}}/2$ .
- 5 In row **Point 2**, set  $r$  to  $d_{\text{mixing}}/2$ .
- 6 In row **Point 2**, set  $z$  to  $L_{\text{mixing}}$ .

7 Click **Plot**.

#### *ID Plot Group 10*

On the **Results** toolbar, click **ID Plot Group**.

#### *Line Graph 1*

- 1 On the **ID Plot Group 10** toolbar, click **Line Graph**.
- 2 In the **Settings** window for Line Graph, locate the **Data** section.
- 3 From the **Data set** list, choose **Center-line**.
- 4 Locate the **y-Axis Data** section. In the **Expression** text field, type p.
- 5 Locate the **x-Axis Data** section. From the **Parameter** list, choose **Expression**.
- 6 In the **Expression** text field, type z.
- 7 Click to expand the **Legends** section. Select the **Show legends** check box.
- 8 From the **Legends** list, choose **Manual**.
- 9 In the table, enter the following settings:

---

<b>Legends</b>
Center-line

---

#### *ID Plot Group 10*

In the **Model Builder** window, under **Results** click **ID Plot Group 10**.

#### *Line Graph 2*

- 1 On the **ID Plot Group 10** toolbar, click **Line Graph**.
- 2 In the **Settings** window for Line Graph, locate the **Data** section.
- 3 From the **Data set** list, choose **Wall**.
- 4 Locate the **y-Axis Data** section. In the **Expression** text field, type p.
- 5 Locate the **x-Axis Data** section. From the **Parameter** list, choose **Expression**.
- 6 In the **Expression** text field, type z.
- 7 Locate the **Legends** section. Select the **Show legends** check box.
- 8 From the **Legends** list, choose **Manual**.
- 9 In the table, enter the following settings:

---

<b>Legends</b>
Walls

---

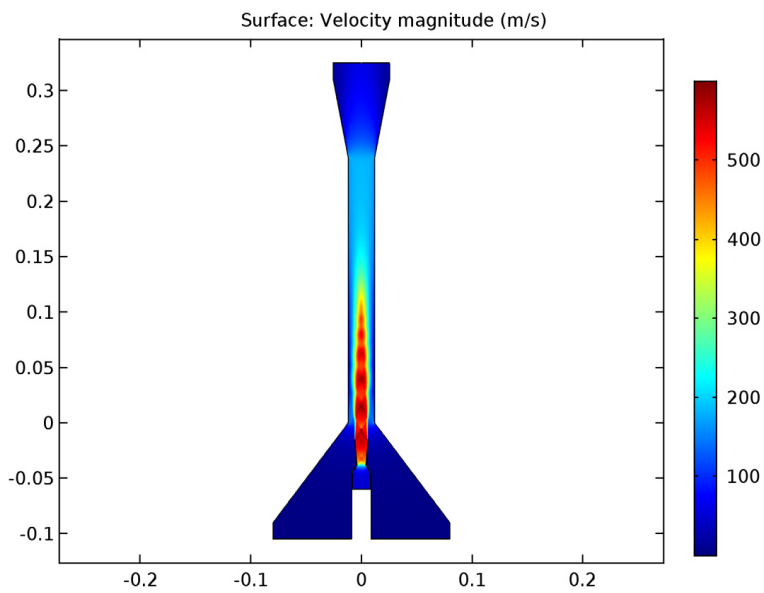


### 1D Plot Group 10

- 1 In the **Model Builder** window, under **Results** click **1D Plot Group 10**.
- 2 On the **1D Plot Group 10** toolbar, click **Plot** (see [Figure 5](#)).

### Velocity (hmnf)

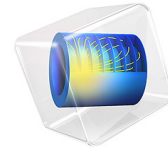
- 1 In the **Model Builder** window, under **Results** click **Velocity (hmnf)**.
- 2 In the **Settings** window for 2D Plot Group, locate the **Data** section.
- 3 From the **Data set** list, choose **Mirror 2D 1**.
- 4 On the **Velocity (hmnf)** toolbar, click **Plot**.





## Appendix B

# Documentation of the Shock Tube application



# Shock Tube

## *About the Application*

---

An important application of unsteady wave motion is a shock tube. Shock tubes are widely used in the study of compressible flow phenomena, high-temperature gases, gas-phase combustion reactions, testing of supersonic bodies and other applications (Ref. 1). They consist of a tube closed at both ends and a diaphragm separating two regions: A high-pressure gas on the left (driver section) and a low-pressure gas on the right (driven section). Both fluids are at rest before the diaphragm is broken at time  $t = 0$ , producing a normal shock wave propagating into right section, an expansion wave propagating into left section and the displacement to the right of the interface between driven and driver gases, also called contact wave.

The purpose of this application is to simulate the flow within a shock tube and estimate the distributions of pressure, density, velocity, temperature, speed of sound, Mach number and mass flow along the tube, as well as the position of the waves at every time step. The app is limited to inviscid flow, which is considered a good approximation in the modeling of shock tubes, and gravity effects are neglected.

### **USER INTERFACE**

The graphical user interface of the app is depicted in Figure 1. The user can select the following input parameters:

- Dimensions of the tube, including the length and the cross-sectional area.
- The mesh size.
- Material properties: Specific gas constant and ratio of specific heats.
- Initial conditions: Pressure and density at both the driver and the driven sections. The pressure at the driver (left) section must be larger than at the driven section.
- Boundary conditions: Use of wall or open boundaries to simulate a very large tube.
- The end time.

The geometry and mesh are build clicking the **Update Geometry** and **Mesh** buttons. The simulation is computed when clicking the **Compute** button. After solving the problem, the user can see the results depicted in different plots:

- 1D plots: The distributions of the desired variables along the shock tube are depicted for the desired time step, which can be selected using the slider placed in the **Graphics** section and clicking the **Plot** button.
- 2D plots: The evolution of the distribution of the variables is depicted in 2D plots being time the vertical axis. The temporal scale used to plot the solutions can be selected using the ribbon placed in the **Graphics** section.

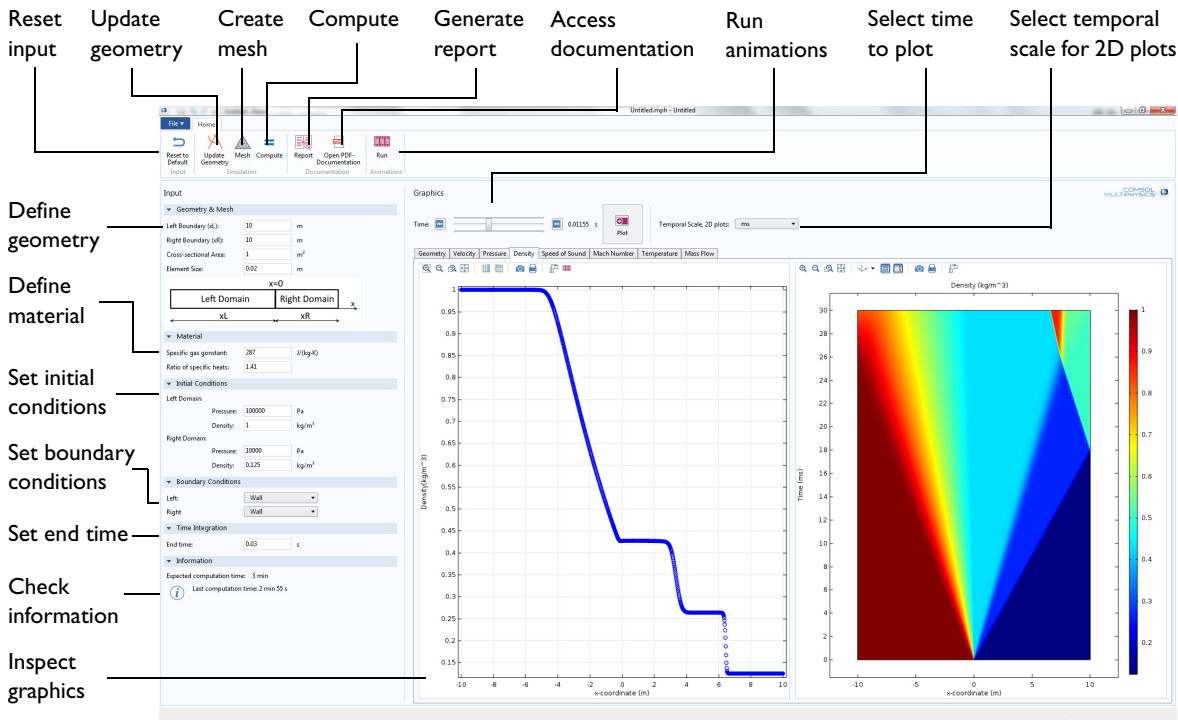


Figure 1: The application's graphical user interface.

The problem is modeled using the Wave Form PDE interface in COMSOL Multiphysics.

**MODEL DEFINITION**

The flow within the tube is modeled by the 1D compressible Euler equations according to

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} &= 0 \\ \frac{\partial \rho u}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} &= 0 \\ \frac{\partial e}{\partial t} + \frac{\partial((e + p)u)}{\partial x} &= 0\end{aligned}$$

where  $\rho$  is the density,  $u$  the velocity,  $p$  the pressure,  $\rho u$  the momentum and  $e$  the energy. This system must be completed with an equation of state. For an ideal gas

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2$$

where  $\gamma$  is the ratio of specific heats for the gas. For diatomic gases,  $\gamma = 1.4$ .

Other quantities of interest, such as the temperature  $T$ , Mach number  $M$ , speed of sound  $c$  and mass flow  $\dot{m}$  can be computed afterwards:

$$\begin{aligned}T &= \frac{p}{\rho R} \\ c &= \sqrt{\gamma \frac{p}{\rho}} \\ M &= \frac{|u|}{a} \\ \dot{m} &= \rho u A\end{aligned}$$

where  $R$  is the specific gas constant (287 J/(kg K) for air) and  $A$  the cross-sectional area of the tube.

The geometry of the shock tube is rather simple (Figure 2). The model is solved in a 1D domain, and the area of the tube is only used to compute the mass flow.

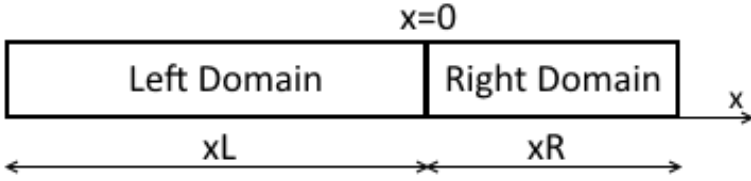


Figure 2: The geometry of the shock tube.

The initial conditions are:

$$\rho(x, 0) = \begin{cases} \rho_L & x \in [-x_L, 0) \\ \rho_R & x \in (0, x_R] \end{cases}$$

$$u(x, 0) = 0$$

$$e(x, 0) = \begin{cases} \frac{p_L}{\gamma-1} & x \in [-x_L, 0) \\ \frac{p_R}{\gamma-1} & x \in (0, x_R] \end{cases}$$

The user can select two different types of boundary conditions: wall or open boundary. At the walls, the velocity  $u$  is 0. The open boundaries are an outlet boundary and may be used to model a very large tube.

#### NOTES ABOUT THE COMSOL IMPLEMENTATION

The Wave Form PDE interface solves wave equations formulated with first order derivatives in time and space using the explicit Runge-Kutta time stepping in combination with a Discontinuous Galerkin method in space. This method arises from an intelligent combination of the finite element and the finite volume methods, and is specially suited for hyperbolic problems.

The Wave Form PDE interface solves problems that take the form

$$\begin{cases} d_a \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{\Gamma}(u) = f & \text{in } \Omega \\ -n \cdot \mathbf{\Gamma} = g & \text{on } \partial\Omega \end{cases}$$



where  $d_a$  is the matrix of damping or mass coefficients,  $\Gamma$  the conservative flux vector,  $f$  the source term vector, and  $g$  the boundary flux term. The 1D Euler equations can be implemented setting

$$u = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}$$

$$d_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Gamma(u) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (e + p)u \end{bmatrix}$$

and 0 source term if gravity effects are neglected.

The shape functions used are continuous in the interior of each mesh element, but discontinuous across the element boundaries. In order to obtain a meaningful solution, a numerical flux is required on faces between adjacent mesh elements. The numerical flux implemented to solve for the Euler equations is the Osher-Solomon numerical flux (Ref. 2), which is computed from the values of  $\rho$ ,  $u$  and  $p$  at both left and right sides of the interface.

The boundary conditions are imposed through the fluxes at the boundaries. For the open boundary condition, the flux is computed for:

$$\rho_R = \rho_L, u_R = u_L, p_R = p_L$$

The wall boundary condition can be imposed in two different ways. If there is a wall at the right boundary, it will be reflecting a normal shock wave:

$$\rho_R = \rho_L, u_R = -u_L, p_R = p_L$$

Another way to impose a wall is to compute the flux for  $u = 0$ :

$$g(u) = \begin{bmatrix} 0 \\ -pn \\ 0 \end{bmatrix}$$

The time step will depend on the values of the velocity, speed of sound and mesh size  $\Delta x$ :

$$\Delta t = CFL \frac{\Delta x}{\max(|u| + c)}$$

Where  $CFL = 0.3$  is the Courant-Friedrichs-Lewy condition.

## RESULTS

The results obtained will depend on the initial and boundary conditions. For example, the results obtained for

$$x_R = 10m, x_L = 10m$$
$$p(x, 0) = \begin{cases} 10^5 Pa & x \in [-10m, 0) \\ 10^4 Pa & x \in (0, 10m] \end{cases}$$
$$\rho(x, 0) = \begin{cases} 1kg/m^3 & x \in [-10m, 0) \\ 0.125kg/m^3 & x \in (0, 10m] \end{cases}$$

and walls at both sides of the tube are depicted in [Figure 3](#) and [Figure 4](#). The results show the presence of a normal shock, a contact wave, an expansion wave and the reflection of the shock; as well as the interaction between the reflected shock and the contact wave. If open boundaries are used instead of walls, the shock tube has infinite length and the flow tends to evolve towards a uniform solution ([Figure 5](#)).

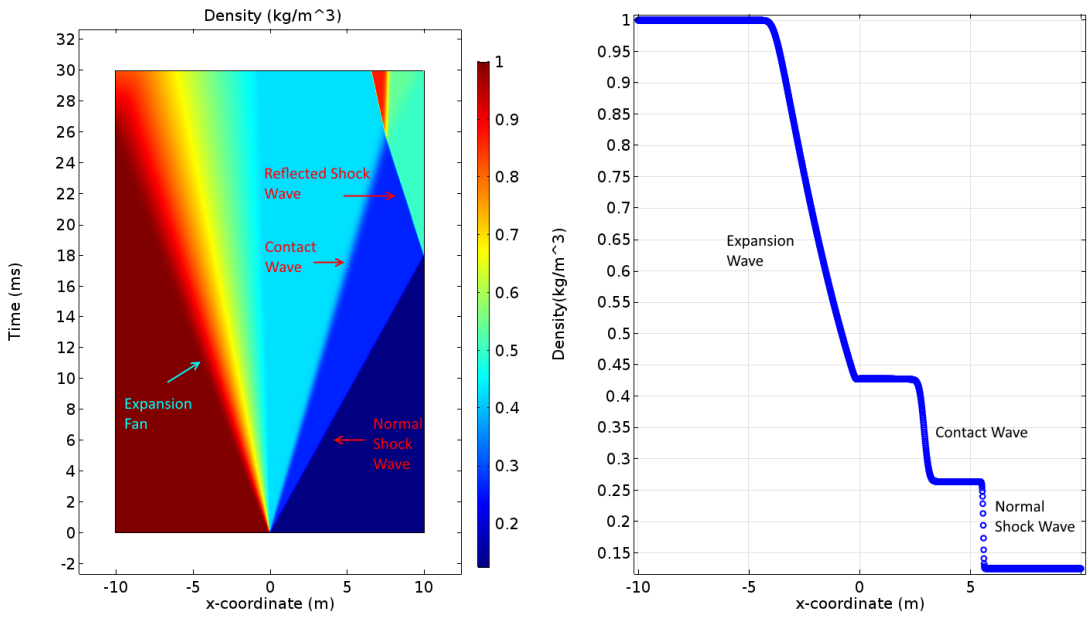


Figure 3: Density (left) and density distribution at  $t = 0.01005$  s (right).

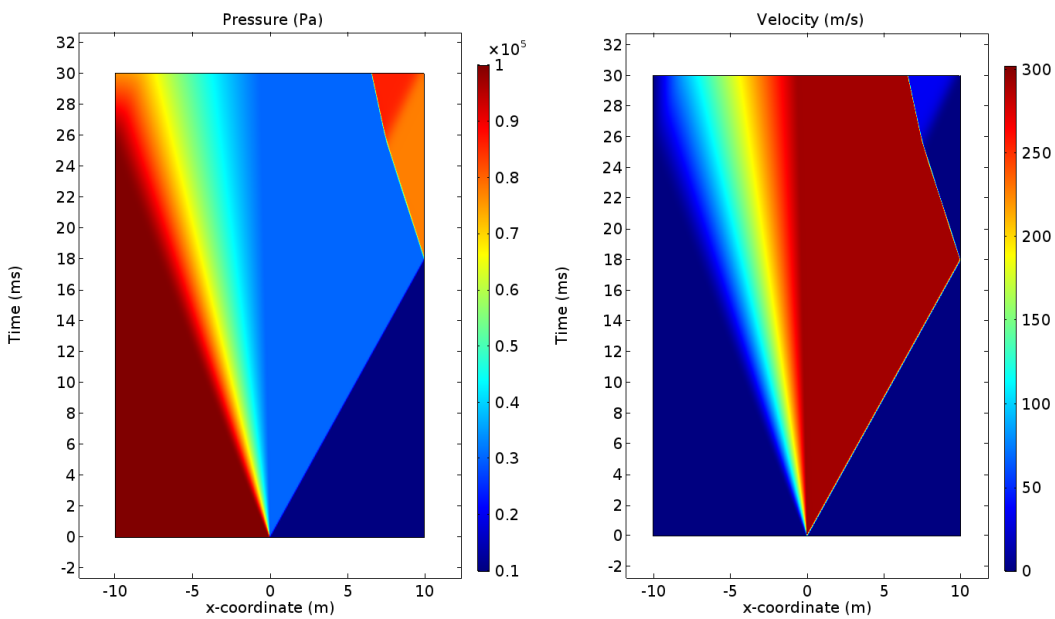


Figure 4: Pressure (left) and velocity (right).

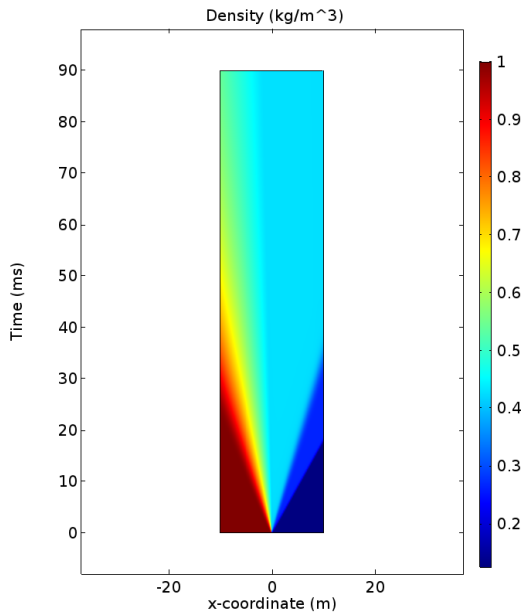


Figure 5: Density using open boundaries.

### Reference

1. John D. Anderson, Jr.; “Modern Compressible Flow”, *McGraw-Hill*, 2nd Edition, Singapore 1990
2. Felcman, Šolín; “On the construction of the Osher-Solomon scheme for 3D Euler equations”, *East-West J. Num. Math.* (1998), pp. 43 - 64

---

**Application Library path:** CFD\_Module/Applications/shock\_tube

---

# Bibliography

- [1] Comsol Inc. Using the comsol desktop<sup>®</sup> modeling environment, 2016. URL <https://www.comsol.com/video/comsol-desktop-modeling-environment>.
- [2] Comsol Inc. *Introduction to COMSOL Multiphysics*, 2016.
- [3] Comsol Inc. Introducing the application builder in comsol multiphysics, 2016. URL <https://www.comsol.se/video/introducing-application-builder-comsol-multiphysics>.
- [4] Comsol Inc. *Introduction to Application Builder*, 2016.
- [5] Comsol Inc. *COMSOL Multiphysics Physics Builder Manual*, 2016.
- [6] Comsol Inc. Multiphysics software product suite, 2016. URL <https://www.comsol.com/products>.
- [7] J. D. Anderson. *Modern Compressible Flow*. MacGraw-Hill, Singapore, 2<sup>nd</sup> edition, 1990.
- [8] Comsol Inc. Comsol application library: Expansion fan, 2016. URL <https://www.comsol.com/model/expansion-fan-42841>.
- [9] J. F. Lynn. *Multigrid solution of the Euler Equations with Local Preconditioning*. PhD thesis, University of Michigan, 1995.
- [10] Comsol Inc. Comsol application library: 3d supersonic flow in a channel with a bump, 2016. URL <https://www.comsol.com/model/3d-supersonic-flow-in-a-channel-with-a-bump-43741>.
- [11] P. Desevaux Y. Bartosiewicz, Zine Aidoun and Yves Mercadier. Numerical and experimental investigations on supersonic ejectors. *Int. J. of Heat and Fluid Flow*, 26:56–70, 2005.
- [12] A. Bouhangel P. Desevaux and E. Gavignet. Flow visualization in supersonic ejectors using laser tomography techniques. *Int. J. of Refrigeration*, 34:1633–1640, 2010.

- [13] H. Zare-Behtash F. Gnani and K. Kontis. Pseudo-shock waves and their interactions in high-speed intakes. *Progress in Aerospace Sciences*, 42:36–56, 2016.
- [14] Comsol Inc. Comsol application library: Supersonic air-to-air ejector, 2016. URL <https://www.comsol.com/model/supersonic-air-to-air-ejector-43921>.
- [15] Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, Cambridge, 2000.
- [16] J. Felcman and P. Šolín. On the construction of the osher-solomon scheme for 3d euler equations. *East-West J. Num. Math.*, pages 43–64, 1998.
- [17] G.A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Computational Physics*, 27:1–31, 1978.
- [18] S. Turek and J. Hron. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. *Fluid-structure interaction*, 53:371–385, 2006.
- [19] F. Durst E. Krause M. Schäfer, S. Turek and R. Rannache. *Benchmark computations of laminar flow around a cylinder*. Springer, 1996.
- [20] Comsol Inc. Solar panel in periodic flow, 2016. URL <https://www.comsol.se/model/solar-panel-in-periodic-flow-12205>.
- [21] Comsol Inc. Modular mixer-turbulent mixing (k-omega), 2016. URL <https://www.comsol.se/model/modular-mixer-turbulent-mixing-k-omega-16903>.
- [22] Comsol Inc. Naca airfoil optimization, 2016. URL <https://www.comsol.se/model/naca-airfoil-optimization-30681>.
- [23] Comsol Inc. Water purification reactor, 2016. URL <https://www.comsol.se/model/water-purification-reactor-14049>.
- [24] Comsol Inc. Micromixer, 2016. URL <https://www.comsol.se/model/micromixer-320>.
- [25] Comsol Inc. Vacuum drying, 2016. URL <https://www.comsol.se/model/vacuum-drying-42561>.
- [26] Comsol Inc. Transonic flow in a sajbén diffuser, 2016. URL <https://www.comsol.se/model/transonic-flow-in-a-sajben-diffuser-10407>.
- [27] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, New York, 1991.
- [28] T. J. R. Hughes and G. N. Wells. Conservation properties for the galerkin and stabilized forms of the advection-diffusion and incompressible navier-stokes equations. *Computer methods in applied mechanics and engineering*, 194:1141–1159, 2005.