BARCELO RENACIMIENTO HOTEL, SEVILLE, SPAIN/14 - 18MAY 2018

IMPLEMENTATION OF A TWO-PHASE PIPE COMPONENT INSIDE THE ESPSS LIBRARY

F. Pinna[1], **M. Leonardi**[2], **F. Nasuti**[3], **J. F. Moral**[4], and **J. Steelant**[5]

[1]Aeronautics an Aerospace Dpt., Chaussée de Waterloo, 72, 1640 Rhode-St-Genèse, Belgium, Email: fabio.pinna@vki.ac.be
[2,3]Dpt. of Mechanical and Aerospace Eng., Università di Roma La Sapienza, Via Eudossiana 18, 00184 Rome, Italy, Email: marco.leonardi@uniroma1.it, francesco.nasuti@uniroma1.it
[4]Empresarios Agrupados, Calle Magallanes, 3. 28015 Madrid, Spain, Email: frj@empre.es
[5]European Space Agency, ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands, Email: Johan.Steelant@esa.int

KEYWORDS: two-phase, AUSM, 6 equations, ESPSS, Propulsion systems, Ecosimpro

ABSTRACT:

One dimensional simulation tools are an important part of the design process of complex aerospace systems. Within this framework, the European Space Propulsion System Simulation (ESPSS) library evolved over time into a full-fledged software capable of dealing with the most common applications in the aerospace domain. To the present date, available two-phase components treated the flow as in equilibrium. In this work, a recent extension of the library to a new non-equilibrium two-phase component is presented.

1 INTRODUCTION

Liquid propulsion systems work under extreme conditions encompassing many complex phenomena pertaining to the two-phase flow domain. Because of their complexity, an analysis of their performance and related unsteady two-phase phenomena requires a tool with a good compromise between reliability, accuracy and speed. A substantial simplification is represented by the homogeneous equilibrium model (HEM). Under this assumption, the flow is considered to be a perfect mixture, where all the fluids move at the same velocity and share the same temperature. Properties like viscosity, thermal conductivity or speed of sound are usually averaged to return an equivalent value for the mixture. Special care has to be taken in the averaging process returning the effective mixture value because it does not necessarily correspond anymore to a thermodynamic function of a fluid; it is rather a model artifact. The resulting equations can then be resolved with standard finite volume techniques with a stable and robust solver. The advantages of this method makes it widespread even for the simulation of complex phenomena, such as, for instance, chill-down (see as an example the Generalized Fluid System Simulation Program in the work of Majumdar and Steadman [1]). On the other hand the simplified homogeneous equilibrium assumption is rarely satisfied during the operative life of a spacecraft; for this reason a more complex treatment is needed in order to reproduce the non-equilibrium between the two phases. Its correct handling is then useful to reduce inaccuracies in the prediction of a propulsion system. Several types of non-equilibrium are possible, namely temperature, velocity, pressure or chemical potential can be different for the two phases. Relaxation processes describe the way the two phases interact with each other toward an equilibrium state and several models could be developed, dealing with different degrees of non-equilibrium, from full to none. It is not within the purpose of this work to review all the models of the last four decades of development addressing this issue, the interested reader could find several exhaustive literature summaries such as in Saurel and Abgrall [2], Lund [3] or Aarsnes *et al.* [4]. The reason for so many models and approaches to exist lies in the complexity of the problem at hand and the difficulty of reaching a clear understanding of the modeling issues. Hence, each physical description is a trade-off between accuracy of the physical representation and computational effort.

Regardless the approach, coupling two (or more)

fluids, or particles, transforms the original set of equations into a model of more difficult solution. Furthermore, the coupling could lead to a non-hyperbolic set of equations, thus hurting the well-posedness of the formulation and eventually preventing the development of an effective numerical scheme. This problem is already visible in Wallis [5] where all the source terms are described by algebraic functions and pressure is the same for both phases. Another results of the two-phase coupling is the appearance of non conservative terms. The difficulty imposed by non-conservativity on converging toward a physical solution is strongly related to non-conservative terms that return different results if an alternative integration path is chosen (see Dinh *et al.* [6]). Most of the models proposed in literature include one or more regularization terms that alter the eigenvalue structure of the system of equations; the final outcome of this manipulation is the achievement of a hyperbolic system (see for instance Tiselj and Petelin [7], [8]). On the other hand, many argues that the introduction of such terms is based more on numerical convenience rather than physical evidence and propose the use of naturally hyperbolic systems: an example is the improvement over the model of Baer and Nunziato [9] performed by Saurel and Lemetayer [10]. Despite this debate, these additional terms are widespread in the community and both virtual mass terms (see Tiselj and Petelin [8]) and interface pressure (see sec. 2) are popular choices. A considerable effort is visible in the literature to treat the flow with specific algorithms for both the flux solver and the time integration scheme. Such specific treatments allows a non standard integration of source and non conservative terms, alleviating their induced stiffness. One dimensional models are often used to predict, for design purposes, the unsteady behavior of complex pipe networks or the interaction of different components. In the nuclear energy sector a tool addressing this need is RELAP5 [11], developed by the Idaho National Laboratory.

The following works relies on a different approach, pursuing a full integration of the newly build component into the already existing ESPSS (see [12], [13], [14]) library and the multipurpose platform Ecosimpro®. The six-equation model, known as effective field model (see Ishii [15] and Stewart and Wendroff [16]) has been adapted and implemented in ESPSS, according to specific guidelines to match the structure of the previously available two-phase components.
Given the wide variety of components available in ESPSS, a general time integration library had to be chosen. Thus, the retrieved system of equations is solved using the time marching library DASSL by Petzold [17] and the flux solver AUSM+-up proposed by Liou *et al*. [18]. The interpretation of the flux scheme is derived from the stratified flow concept proposed by Stewart and Wendroff [16]) and brings some important modification to the standard implementation of the AUSM+, as originally conceived by Liou [19]. The component is designed to be used similarly to previous ESPSS components and takes advantage of the fluid property library.

2 GOVERNING EQUATIONS

Steward and Wendroff [16] proposed a spatial averaging of the flow field retrieving different system of equations based on the hypotheses applied to a very general model. From this work, Chang and Liou [20] derived a system of equations, imagining the two phases inside a volume completely segregated. The whole concept is referred to as the *stratified flow model*. The averaging procedure originated by this concept leads to a set of equations describing the motion of each of the two phases. By assuming inviscid fluids, the governing equations then read:

$$\begin{aligned} \frac{\partial \alpha_i \rho_i}{\partial t} + \frac{\partial \alpha_i \rho_i u_i}{\partial x} &= 0 \\ \frac{\partial \alpha_i \rho_i u_i}{\partial t} + \frac{\partial \alpha_i \rho_i u_i^2 + \alpha_i p}{\partial x} &= p^{int} \frac{\partial \alpha_i}{\partial x} \\ \frac{\partial \alpha_i \rho_i E_i}{\partial t} + \frac{\partial \alpha_i \rho_i H_i}{\partial x} &= -p^{int} \frac{\partial \alpha_i}{\partial t} \end{aligned} \tag{1}$$

where the index $i = g, l$ represent the generic phase, gas or liquid. Volume fraction of the i^{th} phase is represented by α_i, while total energy and total enthalpy are represented respectively by E_i and H_i. Unlike usual conservation equations for single phase system it is evident the presence of a source term for the momentum and energy equation. Both these terms feature the interface pressure between the two phases p^{int} to describe the inviscid link between the two phases. Yet, it makes the set of equations non hyperbolic. This is a well known issue and many solutions have been proposed over the years. The interface pressure is usually computed as $p_{int} = p - \delta p$ where δp can be expressed in several forms in order to retrieve the hyperbolicity of the system. Three formulations are coded in ESPSS:

$$\delta p = \begin{cases} \rho_g (u_l - u_g)^2 & \text{(2a)} \\ 2 \dfrac{\alpha_g (1-\alpha_g) \rho_g \rho_l}{\alpha_g \rho_l + (1-\alpha_g)\rho_g} |(u_l - u_g)|^2 & \text{(2b)} \\ \min(2\rho_g (1-\alpha_g)|(u_l - u_g)|^2, 0.01P) & \text{(2c)} \end{cases}$$

Expression (2c) is the default in the current implementation, anyway, given its important role in the simulation, the other optional choices are available.

Liou *et al.* [18] already pointed out that the presence of α_g and p_{int} in the source term of the momentum equation has to be accurately described at a discrete level. An additional challenge comes from the fact that this term cannot be expressed as a conservative flux. The discretization proposed by Liou is able to indefinitely preserve a stationary contact discontinuity. Their approach aims at maintaining the equilibrium between the conservative and non-conservative terms that include the pressure. Considering the case of a one-dimensional moving fluid interface, with constant velocity and pressure, all terms cancel out and the proposed equations reduce to

$$\nabla(\alpha_i p) = p\nabla\alpha_i \,, \tag{3}$$

where the term on the LHS is in "conservative" form unlike the counterpart in the RHS. This equation has to be numerically satisfied and consistent with the discretization in the LHS. Chang and Liou [20] showed that the commonly used central differencing does not respect this condition and it is not a wise choice for this case. In general, being this term related purely to the interface inside the cell, only values inside the cell should be taken into account. For a j^{th} cell one has

$$\nabla_j \alpha_i = \alpha_{(k,J+1/2,L)} - \alpha_{(k,J-1/2,R)} \,. \tag{4}$$

A second order discretization shows that the value at the cell interface may have, in general, a different value, while in case of a first order approximation this value would be the same, but still numerically consistent.

3 NUMERICAL METHOD

The equations are discretized with a cell centered Finite Volume Method. The convective flux solver is a flavor of the AUSM family conceived specifically for multiphase flow. This choice is particularly suited for the current application because it can treat fluid phenomena with different scales while, at the same time, it is not depending on the explicit definition of the eigenvalues of the system. This advantage is even more evident for multi-phase flows where the eigenstructure is not always known in closed form. It is not rare to find in literature several workarounds for this problem aiming at modifying the equations in order to retrieve simpler eigenvalues; typically, these methods resort to either a modifications of the system of equations (as in the Baer-Nunziato model [10]) or to a careful selection of the regularization terms, such as the virtual mass, (see for instance Städtke[21]). AUSM in this sense could provide a solution even for cases when a coarser mesh "regularizes" the original solution with the addition of numerical viscosity, without the need to correctly estimate the wave propagation. While this feature may add robustness to the code it is worth noting that, for a strict grid convergence, hyperbolicity should always be maintained. Several types of multi-phase adjusted AUSM flux evaluator have been tested in the earlier development, such AUSMPW+ (Park and Chongam [22]) or AUSM+ (Paillere and Cascales [23]); finally, the method focused on AUSM+-UP, as implemented by Liou *et al.* [18], was used for the production code. Each convective flux at the interface is defined as

$$f_{i,1/2} = \dot{m}_{i,1/2}\vec{\Psi}_{i,L/R} \tag{5}$$

where $\dot{m}_{i,1/2}$ is the common mass flux at the interface for the i^{th} phase and

$$\vec{\Psi}_{i,L/R} = (\alpha_i, \alpha_u u_i, \alpha_i H_i) \,. \tag{6}$$

The meaning of the subscript L/R is linked to the upwinding of certain variables as in the following

$$\vec{\Psi}_{i,L/R} = \begin{cases} \vec{\Psi}_{i,L} & if \dot{m}_{i,1/2} > 0 \\ \vec{\Psi}_{i,R} & otherwise \end{cases} \tag{7}$$

Upwinding is applied to the definition of the mass flux as well

$$\dot{m}_{i,1/2} = m_{i,1/2}\rho_{i,L/R} \begin{cases} \rho_{i,L} & if u_{i,1/2} > 0 \\ \rho_{i,R} & otherwise \end{cases} \tag{8}$$

The velocity at the interface is consequently defined in the typical AUSM fashion

$$u_{i,1/2} = a_{1/2}\left[\mathcal{M}^{+}_{(4)}(M_{i,L}) + \mathcal{M}^{-}_{(4)}(M_{i,R}) + M_{pk}\right] \tag{9}$$

where

$$M_{i,L/R} = \frac{u_{i,L/R}}{a_{1/2}} \tag{10}$$

$$M_{pk} = -K_p \max(1 - \bar{M_i}^2, 0)\frac{p_R - p_L}{\rho_{k,1/2} a^2_{1/2}} \tag{11}$$

$$\rho_{i,1/2} = \frac{\rho_{i,L} + \rho_{i,R}}{2} \tag{12}$$

$$\bar{M_i}^2 = \frac{M^2_{i,L} + M^2_{i,R}}{2} \tag{13}$$

The coefficient K_p ranges from zero to one but the component sets it to one as default, to increase stability. Unlike in a single-phase solver the speed of

sound is not uniquely defined and an effective definition must be used. In this work the speed of sound at the interface is

$$a_{1/2} = \frac{1}{2}(a_{g,1/2} + a_{l,1/2})\,, \tag{14}$$

and $a_{i_{1/2}} = 0.5(a_{i,R} + a_{i,L})$ The split Mach number $\mathcal{M}^{+}_{(4)}$ is defined as

$$\mathcal{M}^{\pm}_{(4)}(M) = \begin{cases} \mathcal{M}^{\pm}_{(1)} & if|M| \geq 1, \\ \mathcal{M}^{\pm}_{(2)}(1 \mp 2\mathcal{M}^{\mp}_{(2)}) & otherwise \end{cases} \tag{15}$$

To complete the evaluation of the conservative fluxes, pressure should be computed as well:

$$p_{i,1/2} = \mathcal{P}^{+}_{(5)}(M_{i,L})p_L\mathcal{P}^{-}_{(5)}(M_{i,L})p_R + P_{ui}\,, \tag{16}$$

where P_{ui} is the velocity coupling term and it reads

$$P_{ui} = -K_u\mathcal{P}^{+}_{(5)}(M_{i,L})\mathcal{P}^{-}_{(5)}(M_{i,R})\rho_{1/2}a_{1/2}(u_{i,R}-u_{i.L}) \tag{17}$$

where the coefficient K_u could range between zero and one. Similarly to K_p the default is set to one. The pressure split functions is

$$\mathcal{P}^{\pm}_{(5)}(M) = \begin{cases} \frac{1}{M}\mathcal{M}^{\pm}_{(1)} & if|M| \geq 1, \\ \mathcal{M}^{\pm}_{(2)}\left[(\pm 2 - M) \mp 3M\mathcal{M}^{\mp}_{(2)})\right] & o/w. \end{cases} \tag{18}$$

Pressure fluxes must then be multiplied by the volume fraction

$$(\alpha p)_{i,1/2}(j) = \alpha_{i,1/2,L}p_{i,1/2}$$
$$(\alpha p)_{i,1/2}(j+1) = \alpha_{i,1/2,R}p_{i,1/2}\,.$$

Finally the fluxes for the momentum are composed as $f = f(c) + \alpha_{i,R}p_{g,1/2}$ if it is used as the left flux of the j^{th} cell or $f = f(c) + \alpha_{i,L}p_{g,1/2}$ if it is computed as the flux of the right cell.

The time integration is carried out by means of a DASSL solver (see Petzold [17] for more details) integrated inside Ecosimpro®, but all the other available time integration schemes work as well. The use of DASSL allows an automatic estimation of the integration time step, letting the simulation to evolve without little or any user interaction

The calculation of the non conservative terms follows two distinct approaches. The one related to the momentum equations is simply obtained by a spacial derivative and its implementation inside Ecosimpro® is straightforward. The time derivative appearing in the non-conservative terms of the energy equation is computed by means of a separate function which stores the value of the void fraction at the previous time step and then it is evaluated in a discrete Ecosimpro® block. This derivative is computed at first order and therefore does not match the order of the time derivative computed by DASSL. This inconsistency is due to the advanced capability of the time integrator, that evaluates the best advancing time step, while keeping the order as high as possible, according to the problem stiffness. In case an Euler integration is used the order of the integration would be perfectly consistent. On the other hand, these techniques allow the use of the new component with any time integrator available in Ecosimpro® and does not require any extra user effort in the setup. This extra derivative calculation does not interfere with the integration process carried out in the continuous block of the component and it feeds a source term to be added to the energy equation.

4 NEW COMPONENT

Similarly to what is done in the HEM component already existing in the ESPSS library, the implementation provides an abstract component taking care of most of the implementation. It is currently situated in the 1D_Fluid_Flow library. The actual component to be used within an Ecosimpro® schematic inherits the basic information from the parent one, adding few instructions that could vary according to the different implementations. This is a powerful way of defining the library and it has been preserved to allow future introduction of other two-phase components. The new two-phase pipe is represented in fig. 1 and is designed to interact with all the older components in a similar way.

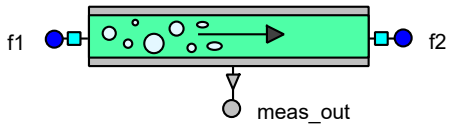

Figure 1: Non-equilibrium two-phase pipe component schematic

Ecosimpro® ESPSS arrange all the connections within a pipe network by means of homogeneous equilibrium model assumptions (HEM). This means that the non-equilibrium information is retained only inside the new component and is not transmitted to the adjacent ones. The main task of the port is to

transfer all the data passed by another component in the network and to retrieve the values for the system of six equations governing the non equilibrium flow. As mentioned, the flow information delivered by the system are tailored for HEM flows and therefore all the flow variables have to be conveniently split. The data available at the port are the volumetric flux $Q[\mathrm{m^3\,s^{-1}}]$, the mass flux $\dot{m}[\mathrm{kg\,s^{-1}}]$ and the enthalpy flux $\dot{m}_h[\mathrm{W}]$ for the mixture. They are used to reconstruct all the needed variables at the boundary of the pipe. Pressure and temperature are estimated at the ghost cells in a similar way to the HEM code. Temperature value at the ghost cell is assumed to be the same for both liquid and gas phase. This pair of thermodynamic values is then used to estimate all the other thermodynamic and transport properties such as density, viscosity and thermal conductivity. Despite the possibility of coding source terms without modifying the parent abstract component, no actual implementation is provided in the non-equilibrium two-phase pipe for any term representing vapor creation, or, in general, any exchange of either mass, momentum or energy. The only coupling between the two phases is achieved by means of the inviscid source term using the interface pressure p^{int} reported in eq. (1).

5 VERIFICATION

In the current ESPSS version, the use of some functions have a limitation in the maximum number of nodes to be used; unfortunately some generic test cases may be particularly hard to grid converge and would require several thousands points (for instance Kitamura water air shock tube [24]) in order to be completely described. For this reason a first code has been implemented outside ESPSS to be used as a reference for the final implementation, while the independent implementation has been compared only against results available in literature. The ESPSS implementation phase has been divided in two different steps: the first one focused explicitly on the flux solver and the overall behavior of the time integration scheme, while a second leg dealt with the connections of the component with the rest of the library. Overall, the homogeneous equilibrium solver seems more robust in handling exceptions, on the other hand most of the difficulties shown by the new component are related to the estimation of the boundary conditions. It is reasonable to assume that when other components will be modified to produce or receive output suited for the non-equilibrium component, a part of these difficulties will vanish. During a preliminary implementation, a stiffened gas equation of state (SGEOS) (as presented by Liou and Chang [18]) has been used to mimic as closely as possible both literature and independent code results. The use of the perfect fluids available in ESPSS returns similar results, therefore, in a following phase, the code has been modified to include the available ESPSS property functions.
This update requires a substantial modification of the methods described by Liou and Chang [18] to compute pressure and void fraction at each time step; while they retrieved closed expression from the SGEOS and the state variables, the ESPSS pipe final implementation relies on an iterative loop that evaluates pressure for the two phases until they have the same value. The current algorithm uses the state variable with a starting void fraction to retrieve a first approximation of the thermodynamic properties for each phase and then it computes the pressure. In case the void fraction falls below or above a fixed threshold the algorithm computes only the dominant phase. Despite its simplicity the code proved to be sufficiently robust even when compared to the closed form solutions. In all the simulations reported hereafter only perfect fluids are used as available in the ESPSS WorkingFluid component: air for the non-condensable gas and water for the liquid.

The present six equations system is naturally displaying a singularity when one of the two phases disappears. At the current state, the solver keeps solving both equations imposing a tolerance below which the vanishing phase gets the same velocity and temperature of the dominant one. While this may seem an arbitrary approach it is a widely adopted method in literature and it does not induce significant error, or at least not more than the one already present in this averaged approach. The independent code implementation was carried with a similar technique and its performance matched single fluid code results.
A selection of the test cases computed with the current component is reported hereafter to highlight the characteristic of the new software.

5.1 COMPONENT WITH SPECIFIC BOUNDARY CONDITIONS

The first part of the implementation has been devoted to reproduce standard test cases without focusing on the port connections with the other components. The choice fell on transmissive boundary conditions without loss of generality, thus simplifying the first part of the testing and freeing the development from the intricacies of more realistic boundaries. It is worth noting that all the test cases presented in this section are not influenced by the implementation or the choice of the boundary conditions as the simulations always stop before a wave can touch the end of the pipe. All the following test

cases are computed on a one dimensional pipe 10m long with the two states separated at 5m.

The shock tube case has been a traditional test for computational methods for compressible flows dating back to the work of Sod [25]. A number of shock tube cases have been adapted to two-phase flow as well. The test conditions for the current calculations are reported in tab. 1 and feature an extremely strong discontinuity in pressure as well a sharp discontinuity in phase. This case has been used as a validation by many authors, in this paper we refer to the one proposed in the interesting two-phase flux solver study performed by Kitamura et al. [24]. The corresponding comparison of pressure and volume fraction is available respectively in fig. 2 and 3. Such a strong discontinuity in the initial conditions is an excellent stress test for the flux solver and the time marching scheme. It has been observed that DASSL is very conservative when estimating time steps for these configurations. A standard CFL condition setup with some basic knowledge about the problem provided bigger time steps, thus reducing considerably the computation time; on the other hand DASSL is able to adapt to many different conditions and when the simulation gets smoother, over time it can increase automatically its time step. All the simulations reported hereafter are obtained with a second order solver. Kitamura [24] reports a difficulty to achieve convergence for some schemes, even when small CFL value are used. A similar behavior was observed as well by the authors of the current work during the implementation of the independent code and it highlights the ability of the new component to return consistent results with the different time integrators available in Ecosimpro®.

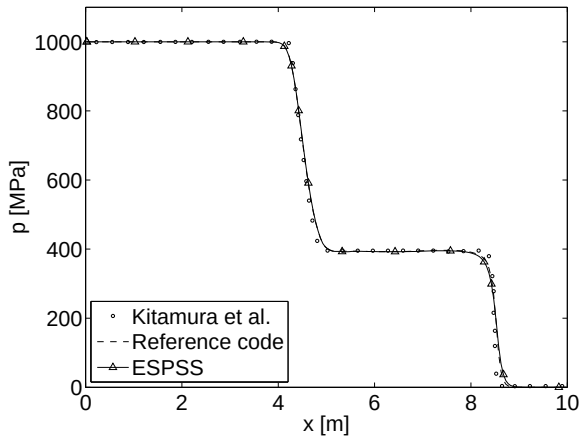

Figure 2: Comparison of pressure against the two-phase shock tube test results from Kitamura *et al.* [24] at $t = 2$ms for 200 cells

The simulation results reported in fig. 2 and 3 correspond to a 200 nodes discretization: the agreement between the independent code, the new ESPSS component and literature is very good. The wave structure shows an expansion front traveling toward the left end of the pipe and a shock moving to the right. It is worth noting that both the expansion fan and the shock in Kitamura's work are slightly sharper and this is most likely due to the settings of the dissipation coefficients. This default value is intended to improve the robustness of the component.

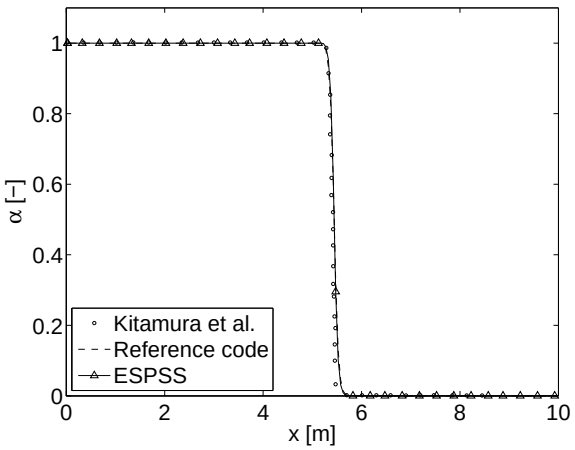

Figure 3: Comparison of volume fraction against the two-phase shock tube test results from Kitamura *et al.* [24] at $t = 2$ms for 200 cells

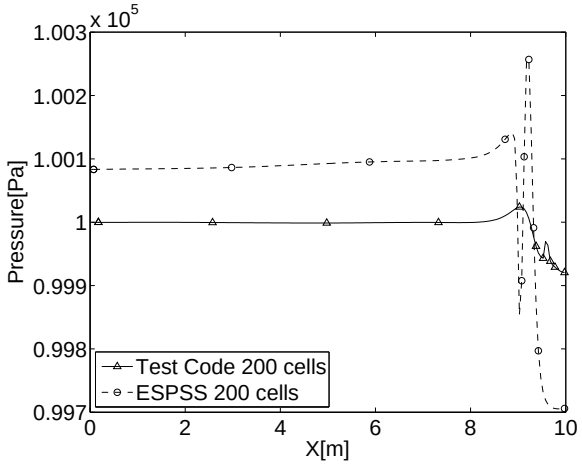

Figure 4: Solutions of the moving contact discontinuity test case: pressure for the ESPSS component and the independent test code for 200 cells at $t = 30$ms

Another interesting test case is the simulation of a moving contact discontinuity. The test conditions are reported in tab. 2. The traveling contact discontinuity is due to an initial sharp phase interface, being all the other parameters equal. This test is meant to

	p[Pa]	α[-]	$u_g = u_l$[m s^{-1}]	$T_g = T_l$[K]
L	10^9	$1-\epsilon$	0	308.15
R	10^5	ϵ	0	308.15

Table 1: Shock tube (air-to-water) test case initial conditions

	p[Pa]	α[-]	$u_g = u_l$[m s^{-1}]	$T_g = T_l$[K]
L	10^5	ϵ	100	308.15
R	10^5	$1-\epsilon$	100	308.15

Table 2: Moving contact discontinuity test case initial conditions

verify the correct implementation of the non conservative terms. As it appears clearly from fig. 4, pressure is not constant up to machine precision, nevertheless it has a reasonably constant value. The relatively bigger error displayed by the ESPSS counterpart is probably due to first order integration of the non conservative term in the equation. On the other hand this error is around 0.1% for the reference code and 0.3% for the ESPSS implementation therefore is deemed acceptable in the limits of the model assumptions. With an initial velocity of 100m s^{-1} the interface, after 30ms is located exactly at 8m. Fig. 5 shows the void fraction computed by both the independent code and the ESPSS implementation; both solvers are able to capture correctly the movement of the discontinuity with a minimal amount of diffusion of the sharp phase interface placing the interface around 8m. This calculation verifies that the discrete representation of the non conservative terms is correctly implemented within the ESPSS library. Any further reduction of the error has to be address from a theoretical point view both at the modeling and discretization scheme level

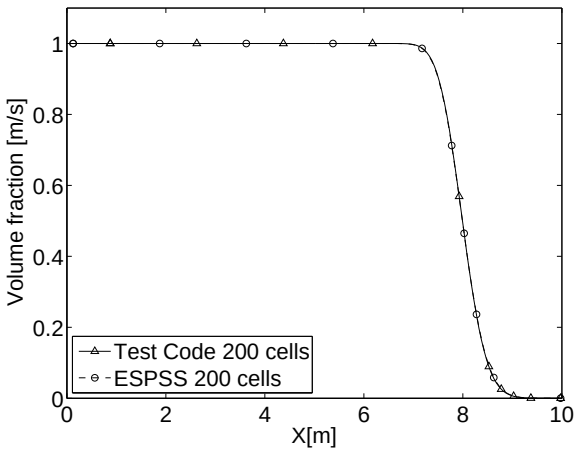

Figure 5: Solutions of the moving contact discontinuity test case: volume fraction for the ESPSS component and the independent test code for 200 cells at $t = 30$ms

In tab. 3 the condition for a cavitation test case are reported. It is worth noting that, being both fluids perfect, there is no evaporation and no mass transfer from the liquid to the gas phase. This case does not target the reproduction of a physical relevant case but aim at testing the behavior of a strong expansion within the component. These two expansion fronts are created by an initial condition where the fluids on the left and the right side of the component leave their position, moving toward the closest exit with an opposite direction but same velocity magnitude. This test case highlight the difficulty related to the vanishing phase; the void fraction cannot be set to zero or to ϵ as this would prevent the algorithm to converge. The strain imposed on the solver has been found in both the independent and ESPSS implementation. It is plausible that a more realistic simulation of the evaporation due to a low pressure would alleviate this difficulty, allowing for a smaller initial void fraction to be used. In such a case the void created by the two expansion fronts would be filled by the vapor generated in the middle of the pipe.

The chosen scheme and its current implementation capture correctly the cavitation behavior as reported in fig. 6 and 7. Kitamura reports constantly less dissipative results, with sharper gradients than the ones in the current work. On the other hand, both the independent code and ESPSS implementation return the same output. The density comparison in fig. 7 show a small spike in Kitamura's result that is not present in our calculation. This is most probably associated with the lower dissipation imposed in that work; as a matter of fact current standard settings in ESPSS avoid this kind of results, increasing overall robustness at the price of a slightly more dissipative calculation.

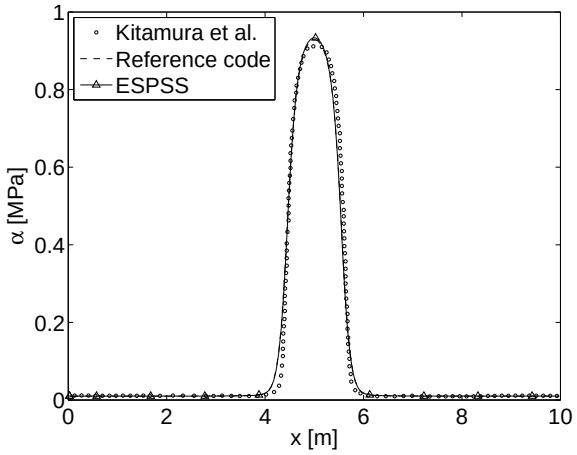

Figure 6: Cavitation test case: void fraction for the ESPSS component, the independent test code and Kitamura [24] at $t = 5$ms for 200 cells

	p[Pa]	α[-]	u_g[m s^{-1}]	u_l[m s^{-1}]	T_g[K]	T_l[K]
L	10^5	10^-2	-100	-100	300.00	300.00
R	10^5	10^-2	100	100	300.00	300.00

Table 3: Cavitation test initial conditions

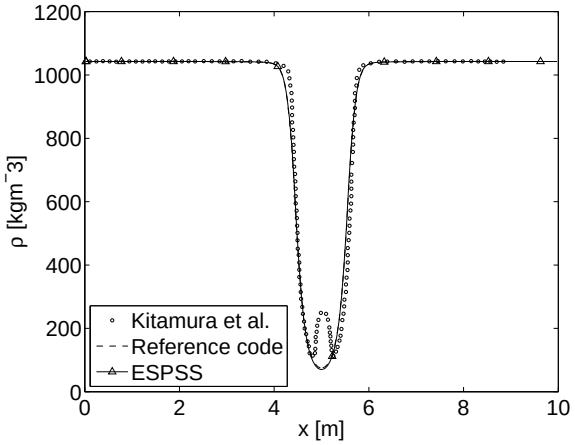

Figure 7: Cavitation test case: comparison of mixture density for the ESPSS component, the independent test code and Kitamura [24] at $t = 5$ms for 200 cells

5.2 COMPONENT WITH PORTS

All the discontinuous cases in the previous section are a great test for the implementation of the flux solver, non-conservative terms and their link with the time integrators. On the other hand the verification of the port behavior in a component is a fundamental step. A minimum viable schematic in ESPSS is obtained by attaching the new component to two fixed volumes by means of two intermediate junctions, as displayed in fig. 8.

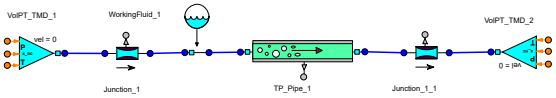

Figure 8: ESPSS schematic of the Tank discharge problem

Such a configuration could represents, in a simple way, the discharge of a tank to a fixed environmental pressure as shown in Fig. 9. This is a well suited test case as it features a need for the connection to two components, simulating the tank and the environment, and it has a closed form solution. Unfortunately such a solution exist for a single fluid, either liquid or gas, so we are going to show the gas-only calculation. As already mentioned, the constant presence of a vanishing phase represents an additional difficulty and its wrong handling would be visible in the final output.

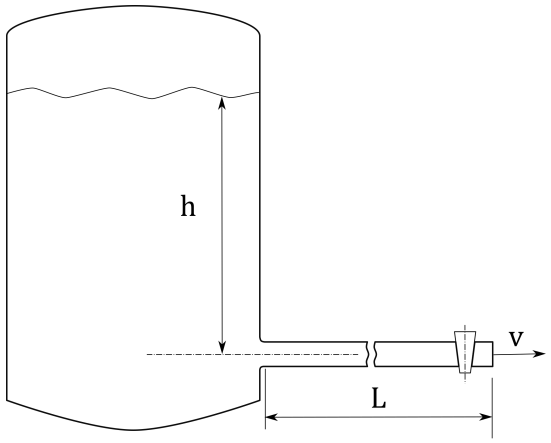

Figure 9: Schematic representation of the tank discharge problem

The equation representing the velocity at the exit of the pipe is obtained from the integration along a streamline. The derivation of the equation is not further detailed here but can be found in several fluid dynamics text book. The velocity at the exit of the pipe is:

$$v_{ex} = \sqrt{gh}\tanh(t/\tau)\,, \tag{19}$$

where $\tau = L/\sqrt{gh}$ and g is the gravitational acceleration and h represent the height difference of the fluid.
In case one does not want to simulate a liquid, where the hydrostatic pressure is relevant, it is sufficient to simulate the pressure difference to compute the asymptotic speed of the fluid. The case shown in fig. 10 and fig. 11 represent a discharge of air from a tank at $P_{tank} = 0.5$MPa to another environment with $\Delta P = 0.1$MPa. The current test case represents a pipe with length $L = 0.1$m. The asymptotic velocity is 223.6m s^{-1} and it is retrieved by the component, if both junctions have backward and forward loss coefficient set to $\zeta_f = \zeta_b = 0.55$. The analytical solution has a smooth unrealistic behavior and it should be intended as an idealized case. A real flow would see traveling waves within the pipe and this phenomenon is correctly duplicated by both the independent code and the ESPSS implementation already with 20 nodes. The code is further

compared with a single fluid implementation which shows the same flow features. The ESPSS component displays constantly a shift in time, and it takes a bit more time for it to reach the analytically predicted velocity. This behavior is mostly associated with the junction components at the two sides of the non-equilibrium pipes. Their presence is mandatory because of ESPSS and Ecosimpro® design. The peculiar implementation of the boundary conditions along with the junction is prone to cause such a shift. It could be argued that a 20 nodes mesh is under resolved for the case but a 50 nodes simulation, reported in fig. 11, shows that the simulations is, basically, grid converged.

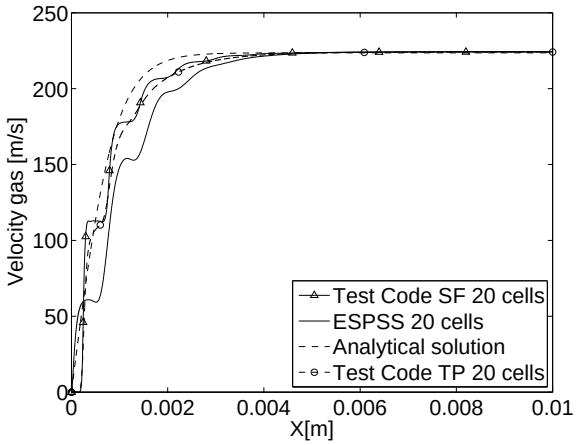

Figure 10: Comparison of different solver against the analytical solution for a tank discharge with 20 nodes. The data reported are retrieved from the last node of the calculation

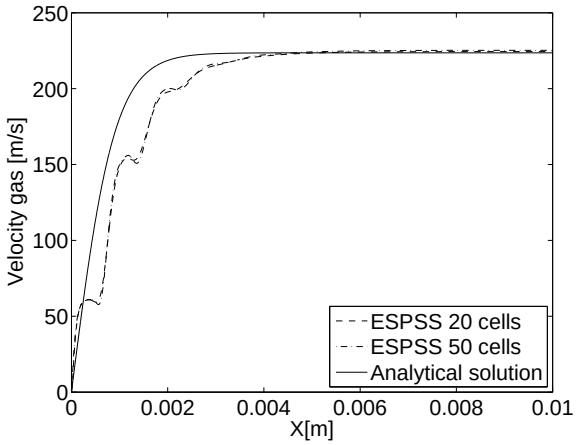

Figure 11: Comparison of different ESPSS solver with a 20 nodes and 50 nodes discretization against analytical solution. The data reported are retrieved from the last node of the calculation

6 CONCLUSIONS

A new non-equilibrium two-phase component has been developed inside the European Space Propulsion System Simulation library. Unlike the previously implemented two-phase component, it features a six-equation model able to treat many non-equilibrium phenomena with the exception of pressure difference between the phases. The model uses a regularization interface pressure term and several ways of computing it have been made available to the user. The code is able to treat real fluid properties as provided by the ESPSS library and it is compatible with all the already existing components. The behavior of the new two-phase pipe has been verified satisfactorily against literature and analytical test cases. Its current implementation features only inviscid interactions between the phases and it is intended to serve as a framework for future developments. Most of the basic structure of the solver is already laid out, therefore the detection of different two-phase regimes or the exchange of mass, momentum and energy, as it happens during the interactions of the different phases, is limited to the implementation of the corresponding correlations and source terms.

7 ACKNOWLEDGMENTS

This work has been performed under the ESA contract 4000111093/14/NL/PA, phase 4 of the European Space Propulsion System Simulation Tool Project.

References

[1] Alok Majumdar and Todd Steadman. Numerical modeling of thermofluid transients during chilldown of cryogenic transfer lines. *SAE Technical Papers*, February 2003.

[2] Richard Saurel and Rémi Abgrall. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computing*, 21(3):1115–1145, 1999.

[3] Halvor Lund. A hierarchy of relaxation models for two-phase flow. *SIAM Journal on Applied Mathematics*, 72(6):1713–1741, 2012.

[4] Ulf Jakob F. Aarsnes, Tore Flåtten, and Ole Morten Aamo. Review of two-phase flow models for control and estimation. *Annual Reviews in Control*, 42:50 – 62, 2016.

[5] G. B Wallis. *One-dimensional two-phase flow*. New York : McGraw-Hill, 1969.

[6] T. N. Dinh, R. R. Nourgaliev, and T. G. Theofanous. Understanding the ill-posed two-fluid model. In *10^{th} International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-10), Seoul, Korea*, pages 5–9, 2003.

[7] Iztok Tiselj and Stojan Petelin. First and Second-Order Accurate Schemes for Two-Fluid Models, journal = Journal of Fluids Engineering, year = 1998, volume = 120, number = 2, doi = 10.4208/cicp.020813.190214a, publisher = ASME,.

[8] Iztok Tiselj and Stojan Petelin. Modelling of two-phase flow with second-order accurate scheme. *Journal of Computational Physics*, 136(2):503 – 521, 1997.

[9] M.R. Baer and J.W. Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (ddt) in reactive granular materials. *International Journal of Multiphase Flow*, 12(6):861 – 889, 1986.

[10] Richard Saurel and Olivier Lemetayer. A multiphase model for compressible flows with interface, shock, detonation waves and cavitation. *Journal of Fluid Mechanics*, 431.

[11] Relap5 website. `https://www.relap.com/`.

[12] M. Leonardi, F. Nasuti, F. Di Matteo, J. Steelant, F. Moral Moral, and J. Vila Vacas. Improving combustion chamber and pipe components of the european space propulsion system simulation (espss) library with ausm scheme. In *Space Propulsion*, number SP2016-3125133, Rome, Italy, May 02-06 2016.

[13] J. Moral Moral, J. Vilá, V. Fernàndez-Villacé, F. Di Matteo, and J. Steelant. Espss model of a simplified combined-cycle engine for supersonic cruise. In *Space Propulsion*, number SP2016-3125351, Rome, Italy, May 02-06 2016.

[14] Di Matteo F. and Steelant J. Multi-disciplinary propulsion simulations at engineering level by means of the european space propulsion system simulation espss. In *RTO/AVT/VKI Lecture Series on Fluid Dynamics Associated to Launcher Developers*, number EN-AVT-206-06, St. Genesius-Rode, Belgium, April 15-17 2013.

[15] M. Ishii. *Thermo-fluid dynamic theory of two-phase flow*, volume 75. Paris, Eyrolles, 1975.

[16] H. B. Stewart and B. Wendroff. Two-phase flow: Models and methods. *Journal of Computational Physics*, 56:363–409, 1984.

[17] Linda R Petzold. Description of DASSL: a differential/algebraic system solver. *Proc. 10th IMACS World Congress*, 1982.

[18] M.-S. Liou, C.-H. Chang, L. Nguyen, and T. G. Theofanous. How to solve compressible multifluid equations: a simple, robust, and accurate method. *AIAA Journal*, 46(9):2345–2356, 2008.

[19] Meng-Sing Liou. A sequel to ausm: Ausm+. *Journal of Computational Physics*, 129(2):364 – 382, 1996.

[20] Chih-Hao Chang and Meng-Sing Liou. A new approach to the simulation of compressible multifluid flows with ausm+ scheme. In *16th AIAA Computational Fluid Dynamics Conference*, 2003.

[21] Herbert Städtke. *Gasdynamic aspects of two-phase flow: hyperbolicity, wave propagation phenomena, and related numerical methods*. WILEY-VCH, 2006.

[22] Jin Seok Park and Chongam Kim. Extension of ausmpw+ scheme for two-fluid model. 17, 09 2013.

[23] H. Paillère, C. Corre, and J.R. García Cascales. On the extension of the ausm+ scheme to compressible two-fluid models. *Computers & Fluids*, 32(6):891 – 916, 2003.

[24] Keiichi Kitamura, Meng-Sing Liou, and Chih-Hao Chang. Extension and Comparative Study of AUSM-Family Schemes for Compressible Multiphase Flow Simulations. *Communications in Computational Physics*, 16(3):632674, 2014.

[25] G.A. Sod. Survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.; (United States)*.