ICFD THEORY MANUAL

LSTC-LS-DYNA-ICFD-THE-1.1-1

Incompressible fluid solver in LS-DYNA

Tested with LS-DYNA® v980 Revision Beta

Friday 23rd March, 2012

LSTC
Livermore Software
Technology Corp.

Disclaimer:

The LSTC ICFD group does not accept liability for any errors or omissions in the contents of this document.

Document Information

Confidentiality	**external use**
Document Identifier	LSTC-LS-DYNA-ICFD-THE-1.1-1
Author(s)	Prepared by LSTC® Facundo Del Pin, Iñaki Çaldichoury
Number of pages	33
Date created	Friday 23rd March, 2012
Distribution	LS-DYNA® users

Contents

1 Introduction

1.1 Purpose of this Document

A detailed description of the analytical equations solved by the incompressible fluid solver is given as well as the numerical methods used. Additional descriptions of the equations used for some of the main features of the solver is also provided. The objective of this document is to offer the reader who wishes to understand and use the incompressible fluid solver of LS-DYNA a precise and easy way to understand insight of the formulas, notions and theories employed by the solver.

2 Document Information

Test Case Summary	
Confidentiality	external use
Test Case Name	Incompressible fluid solver in LS-DYNA
Test Case ID	ICFD-THE-1.1
Test Case Status	active
Test Case Classification	Theory document
Metadata	Theory

Table 1: Test Case Summary

3 Notations and physical variables

Table 2 provides the meaning of each symbol and the SI unit of measure while table 3 provides the conventions and operators used :

Symbol :	Meaning :	Units (S.I) :
$\vec{U}$	Fluid velocity	Meters per second
P	Fluid pressure	Pascals
ρ	Fluid density	Kilos per cubic meter
μ	Fluid dynamic viscosity	Pascal seconds
ν	Fluid cinematic viscosity	Square meters per second
α	Thermal diffusivity	Square meter per second

Table 2: Variables and constants

Grad, Curl, Div operators	$\vec{\nabla}$, $\vec{\nabla}\times$, $\nabla\cdot$
Vectors in $\mathbb{R}^3$	Arrow overhead : $v\vec{e}c$
Matrices	Bold : **P**
Vectors in FEM/BEM systems	Small : a_i

Table 3: Operators and conventions

4 The ICFD solver

4.1 Fluid Mechanics equations

4.1.1 The incompressibility condition

The present solver aims to solve the incompressible Navier-Stokes equations. Traditionally, a fluid may be considered incompressible when the Mach number is lower than 0.3 i.e :

$$M = \frac{V}{a} \leq 0.3 \tag{1}$$

where V is the velocity of the flow relative to a fixed object and a is the speed of sound in the medium (In air, we thus have $v \preceq 370KPH$).

In fluid dynamics, the continuity equation states that, in any steady state process, the rate at which mass enters a system is equal to the rate at which mass leaves the system. The continuity equation is analogous to Kirchhoff's current law in electric circuits.

The differential form of the continuity equation is:

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \vec{u}) = 0 \tag{2}$$

If ρ is a constant, as in the case of incompressible flow, the mass continuity equation simplifies to a volume continuity equation:

$$\nabla.(\vec{u}) = 0 \tag{3}$$

4.1.2 Governing set of equations

Conservation of momentum and mass for incompressible Newtonian fluids in the Eulerian conventional form are represented by the Navier-Stokes equations combined with the continuity equations as follows :

$$\rho(\frac{du_i}{dt} + u_j \frac{\partial u_i}{\partial x_j}) = \frac{\partial \sigma_{i,j}}{\partial x_j} + \rho f_i \quad in \quad \Omega \tag{4}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \quad in \quad \Omega \tag{5}$$

The total stress tensor is given by :

$$\sigma_{ij} = -p\delta_{ij} + \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_l}{\partial x_l}\delta_{ij}) \tag{6}$$

For near incompressible flows we have that:

$$\frac{\partial u_i}{\partial x_i} \ll \frac{\partial u_i}{\partial x_j}, \tag{7}$$

and the last term in the brackets may be neglected from Equation (6). Then:

$$\sigma_{ij} \approx -p\delta_{ij} + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{8}$$

In the same way, the term $\partial\sigma_{ij}/\partial x_j$ in the momentum equation may be simplified for near incompressible flows as:

$$\begin{aligned}
\frac{\partial \sigma_{ij}}{\partial x_j} &= -\frac{\partial p}{\partial x_j}\delta_{ij} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] \\
&= -\frac{\partial p}{\partial x_j}\delta_{ij} + \mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) + \mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_j}{\partial x_i}\right) \\
&= -\frac{\partial p}{\partial x_j}\delta_{ij} + \mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) + \mu\frac{\partial}{\partial x_i}\left(\frac{\partial u_j}{\partial x_j}\right) \\
&\approx -\frac{\partial p}{\partial x_j}\delta_{ij} + \mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right).
\end{aligned} \tag{9}$$

The above expression simplifies the viscous term so that we end up with the following system :

$$\rho\left(\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i} + \mu\frac{\partial^2 u_i}{\partial x_j \partial x_j} + \rho f_i \quad in \quad \Omega \tag{10}$$

$$\frac{\partial u_i}{\partial x_i} = 0 \quad in \quad \Omega \tag{11}$$

4.1.3 Boundary conditions

The set of differential equations presented above is incomplete if the appropriate set of boundary conditions and initial conditions is not specified. For the continuum problem, conditions specifying velocities and stress have to be imposed.

The components of the velocity vector will be replaced, according to the constraints related to the boundary, by the proper imposed value. Over a rigid wall, the boundary conditions follow directly from the momentum equations. For a free-slip wall, the normal velocity must vanish; for a non-slip wall, the tangential components must, in addition, vanish. These conditions in general can be defined as:

$$u_i = \overline{v}_i \qquad \text{on} \qquad \Gamma_v, \tag{12}$$

where $\overline{v}_i$ indicates the function that is imposed on the boundary.

The other type of boundary conditions are imposed on the free surface. These are the most difficult to identify as the shape of the free surface has to be computed ahead. The principles that form the basis for the free surface boundary conditions are stated as follow:

- stress tangential to the surface must vanish,
- stress normal to the surface must exactly balance any externally applied normal stress.

In this case these conditions are expressed as:

$$t_i = \sigma_{ij}n_j = \bar{t}_i \qquad \text{on} \qquad \Gamma_f, \tag{13}$$

where n_j is the direction cosine of the outward normal on the boundary with respect to the x_j axis. Both surfaces Γ_v and Γ_f are two disjoint non-overlapping subsets of the boundary Γ.

The set of initial conditions for the Lagrangian Navier-Stokes problem will be given specifying the velocity and pressure at the initial time:

$$\begin{aligned} v_i(x_i,0) &= v_i^0(x_i), && (14)\\ p(x_i,0) &= p^0(x_i), && (15) \end{aligned}$$

where the initial velocity v_i^0 has to satisfy the incompressibility constrain $\partial v_i/\partial x_i = 0$.

4.2 The Fractional Step method

4.2.1 Introduction

For the time integration of the Navier-Stokes equation a projection method was adopted.These methods stand out from classical monolithic schemes as pressure and velocity are uncoupled. Thus four linear systems of equations are obtained, namely three for the momentum equation and one to solve the incompressibility constrain.

The present scheme consists of three steps until the final velocity is achieved. In the first step, a predictor velocity u_i^* is computed. This velocity does not satisfy the incompressibility constrain given by equation (5). Thus, in the second step the velocity u_i is projected into a space of divergence free vector field and a Poisson equation of pressure is obtained. This pressure will correct u_i to get a final step divergence free velocity u_i^{n+1} . The final step consists of moving the particles to their new time step position x^{n+1} . If this is done iteratively then convergence is achieved when the particles always move to the same final position.

4.2.2 Splitting the Momentum Equations

Let us start with the time derivative that is approximated with backward differences of second order as :

$$\frac{du_i}{dt} \approx \frac{1.5u_i^{n+1} - 2u_i^n + 0.5u_i^{n-1}}{\Delta t} \tag{16}$$

However, for clarity purposes, the following demonstrations will use a simple forward differences approximation keeping in mind that the procedure is similar with a backward differences approximation :

$$\frac{du_i}{dt} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t} \tag{17}$$

If we now consider f a generic function of time and f^n the value of f at $t^n = n\Delta t$ or an approximation of it, and let $f^{n+\theta} = \theta f^{n+1} + (1-\theta)f^n$, we can introduce a new variable on the left side of Equation 4 and add and subtract a pressure term on the right hand side as follows :

$$\rho\frac{u_i^{n+1} - u_i^\star + u_i^\star - u_i^n}{\Delta t} = \frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n - \gamma p^n\right]\delta_{ij} + \left[\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) - \rho u_j\frac{\partial u_i}{\partial x_j} + \rho f_i\right]^{n+\theta}, \tag{18}$$

where $u_i^\star$ are fictitious variables called fractional velocities and γ is a scalar value that varies from 0 to 1 and that will be explored later. θ implies a explicit forward Euler scheme, $\theta = 1$ is a implicit backward Euler scheme and $\theta = 0.5$ is the Crank-Nicholson second order scheme. In the present analysis $\theta = 1$ will be adopted. We can see that the pressure term has been taken out from the implicit-explicit term $[\cdot]^{n+\theta}$ and only the implicit part is taken into account.

Rewriting Equation (18), and reordering some terms, we get :

$$u_i^{n+1} - u_i^\star + u_i^\star = \frac{\Delta t}{\rho}\frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n\right]\delta_{ij} + u_i^n - \gamma\frac{\Delta t}{\rho}\frac{\partial p^n}{\partial x_j}\delta_{ij} + \frac{\Delta t}{\rho}\left[\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}\right) + \rho f_i - \rho u_j\frac{\partial u_i}{\partial x_j}\right]^{n+\theta}. \tag{19}$$

The last three terms of Equation (19) will be related to $u_i^\star$ in such a way that we may write:

$$u_i^\star = u_i^n - \gamma\frac{\Delta t}{\rho}\frac{\partial p^n}{\partial x_j}\delta_{ij} + \frac{\Delta t}{\rho}\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^{n+\theta}}{\partial x_j}\right) - \Delta t u_j^{n+\theta}\frac{\partial u_i^{n+\theta}}{\partial x_j} + \Delta t f_i, \tag{20}$$

$$u_i^{n+1} = u_i^\star + \frac{\Delta t}{\rho}\frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n\right]\delta_{ij}, \tag{21}$$

in which f_i is considered constant in time and p^n is the pressure field at time t^n but evaluated at the final position. In the viscous and advection terms, the values of u_i^{n+1} and u_j^{n+1} will be estimated using $u_i^\star$. The term u_i^{n+1} will be estimated as $\bar{u}_i^{n+1}$. The choice of $\bar{u}_i^{n+1}$ will be explicited later. This allows to write Equation (20) as:

$$u_i^\star - \frac{\Delta t}{\rho}\mu\theta\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^\star}{\partial x_j}\right) + \Delta t u_j^\star \frac{\partial \bar{u}_i^{n+1}}{\partial x_j} = u_i^n - \gamma\frac{\Delta t}{\rho}\frac{\partial p^n}{\partial x_j}\delta_{ij} + \frac{\Delta t}{\rho}\mu(1-\theta)\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^n}{\partial x_j}\right) - \Delta t(1-\theta)u_j^n\frac{\partial u_i^n}{\partial x_j} + \Delta t f_i, \quad (22)$$

which is the actual expression to compute the predictor velocity. The parameter γ introduced earlier, determines the amount of pressure splitting. Thus when $\gamma = 1$ it means small pressure splitting and $\gamma = 0$ implies the largest pressure splitting. It is shown in [**?**] that when $\gamma = 0$ a first order scheme in time is obtained. On the contrary when $\gamma = 1$ a second order scheme is found. A second order scheme gives instable values of the pressure field that need to be corrected. The stabilization technique used in this case will be introduced later.

4.2.3 Equation of Incompressibility

The main objective of this phase is to obtain a pressure field that, when replaced in Equation (21), satisfies the incompressibility constraint. Once the intermediate velocity $u_i^\star$ has been computed from Equation (22), the end of step velocity is obtained by adding to $u_i^\star$ the dynamical effect of the still unknown pressure p^{n+1}, which must be determined such that the incompressibility constraint from Equation 5 remain satisfied.

Taking Equation (21) and applying the divergence operator on both sides:

$$\frac{\partial}{\partial x_i}(u_i^{n+1} - u_i^\star) = \frac{\Delta t}{\rho}\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n\right]\delta_{ij}. \quad (23)$$

By means of Equation (5) the first term on the left hand side vanishes and we may write:

$$\frac{\partial}{\partial x_i}(-u_i^\star) = \frac{\Delta t}{\rho}\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n\right]\delta_{ij}. \quad (24)$$

which is a linear elliptic equation of pressure. Solving Equation (24) with the appropriate boundary conditions will determine the new time step pressure.

4.2.4 Three Step Fractional Method

Each iteration inside the time interval Δt will involve to compute three sets of equations, each of them related to different steps of the time integration scheme, namely using $\theta = 1$:

$u_i^\star$:

$$u_i^\star - \frac{\Delta t}{\rho}\mu\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^\star}{\partial x_j}\right) + \Delta t u_j^\star \frac{\partial \bar{u}_i^{n+1}}{\partial x_j} = u_i^n - \gamma\frac{\Delta t}{\rho}\frac{\partial p^n}{\partial x_j}\delta_{ij} + \Delta t f_i, \quad (25)$$

p^{n+1}:

$$\frac{\partial}{\partial x_i}(-u_i^\star) = \frac{\Delta t}{\rho}\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n\right]\delta_{ij}, \quad (26)$$

u_i^{n+1}:

$$u_i^{n+1} = u_i^\star + \frac{\Delta t}{\rho}\frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n\right]\delta_{ij}. \quad (27)$$

4.3 Spatial discretization by the Finite Element Method

4.3.1 The FEM system

Let us first introduce the weak form of the equations of motion that have been already decoupled in Equations (25), (26) and (27). We will use indistinctly the letter N to identify either trial functions or interpolating functions. Multiplying the equations by the trial function N and integrating over the volume Ω:

$u_i^\star$:

$$\int_\Omega N_i u_i^\star d\Omega - \Delta t \int_\Omega N_i \frac{\mu}{\rho}\frac{\partial}{\partial x_j}\left(\frac{\partial u_i^\star}{\partial x_j}\right) d\Omega + \int_\Omega \Delta t N_i u_j^\star \frac{\partial \bar{u}_i^{n+1}}{\partial x_j} d\Omega = \int_\Omega N_i u_i^n d\Omega - \gamma \Delta t \int_\Omega \frac{N_i}{\rho}\frac{\partial p^n}{\partial x_j}\delta_{ij} d\Omega + \Delta t \int_\Omega N_i f_i d\Omega, \tag{28}$$

p^{n+1}:

$$\int_\Omega N \frac{\partial}{\partial x_i}(-u_i^\star) d\Omega = \Delta t \int_\Omega \frac{N}{\rho}\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n\right]\delta_{ij} d\Omega, \tag{29}$$

u_i^{n+1}:

$$\int_\Omega N_i u_i^{n+1} d\Omega = \int_\Omega N_i u_i^\star d\Omega + \Delta t \int_\Omega \frac{N_i}{\rho}\frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n\right]\delta_{ij} d\Omega. \tag{30}$$

where the parameters Δt and γ are considered constant during the integration.

Integrating by parts the stress term in Equation (28) and the pressure term in (29), use of the divergence theorem and the boundary conditions of Equation (13) and (Equation 12) let us arrive to the following set of equations:

$u_i^\star$:

$$\int_\Omega N_i u_i^\star d\Omega + \Delta t \int_\Omega \frac{\mu}{\rho}\frac{\partial N_i}{\partial x_j}\left(\frac{\partial u_i^\star}{\partial x_j}\right) d\Omega + \int_\Omega \Delta t N_i u_j^\star \frac{\partial \bar{u}_i^{n+1}}{\partial x_j} d\Omega = \tag{31}$$

$$\int_\Omega N_i u_i^n d\Omega - \gamma \Delta t \int_\Omega \frac{N_i}{\rho}\frac{\partial p^n}{\partial x_j}\delta_{ij} d\Omega + \Delta t \int_\Omega N_i f_i d\Omega + \int_\Gamma N_i \left[\frac{\partial u_i^\star}{\partial x_j} n_j\right]_{\Gamma_f} d\Gamma, \tag{32}$$

p^{n+1}:

$$\int_\Omega \frac{\partial N}{\partial x_i} u_i^\star d\Omega = \Delta t \int_\Omega \frac{1}{\rho}\frac{\partial N}{\partial x_i}\frac{\partial}{\partial x_j}\left[p^{n+1} d\Omega - \gamma p^n\right]\delta_{ij} + \int_\Gamma N_i \left[u_i^\star + \frac{\Delta t}{\rho}\frac{\partial}{\partial x_j}(-p^{n+1} + \gamma p^n)\delta_{ij}\right]_{\Gamma_v} n_i d\Gamma, \tag{33}$$

u_i^{n+1}:

$$\int_\Omega N_i u_i^{n+1} d\Omega = \int_\Omega N_i u_i^\star d\Omega + \Delta t \int_\Omega \frac{N_i}{\rho}\frac{\partial}{\partial x_j}\left[-p^{n+1} + \gamma p^n\right]\delta_{ij} d\Omega, \tag{34}$$

where n_i is the outward normal vector to the Γ_v boundary. Comparing the boundary term in Equation (33) and the end of step velocity (Equation (34)) we may write:

$$\int_\Gamma N \left[u_i^\star + \frac{\Delta t}{\rho}\frac{\partial}{\partial x_j}(-p^{n+1} + \gamma p^n)\delta_{ij}\right]_{\Gamma_v} n_i d\Gamma = \int_\Gamma N [\tilde{u}_i^{n+1}]_{\Gamma_v} n_i d\Gamma. \tag{35}$$

To complete this set of equations we need to specify the essential and natural boundary conditions for Equation (31) and Equation (33). In the first case the derivative of the velocity normal to the free surface will vanish:

$$\left[\frac{\partial u_i^\star}{\partial x_j} n_j\right]_{\Gamma_f} = 0. \tag{36}$$

Also free-slip or no-slip velocity conditions over the walls may be imposed. In the second case there will be two sets of boundary conditions namely essentials over Γ_f and natural over Γ_v, then we have:

$$p = 0 \qquad \text{on } \Gamma_f, \tag{37}$$

$$\tilde{u}_i^{n+1} = u_i^{n+1} \qquad \text{on } \Gamma_v, \tag{38}$$

which means that the value of pressure will be imposed over the free surface and the normal velocity of the fluid over particles with a given known velocity will take the known normal velocity of the particles.

The full version of the discrete problem takes place when the unknown fields are approximated over the finite elements. Then within an element, the velocity and pressure will be represented by:

$$u_i = N_l u_{li}, \tag{39}$$

$$p = N_l p_l, \tag{40}$$

where l represents a node of the element. Replacing Equation (39) and Equation (40) in Equation (31) and (34) a final set of equations in a more compact form are obtained:

$$\rho \mathbf{M} u_i^\star + \Delta t \mathbf{K}(\mu) u_i^\star + \mathbf{S}(\rho \bar{u}_i^{n+1}) u_j^\star = \rho \mathbf{M} u_i^n - \gamma \Delta t \mathbf{G} p^n + \Delta t \mathbf{F}, \tag{41}$$

$$\Delta t \mathbf{L}(\frac{1}{\rho}) p^{n+1} = \mathbf{D} u_i^\star + \gamma \Delta t \mathbf{L}(\frac{1}{\rho}) p^n - \tilde{\mathbf{U}}, \tag{42}$$

$$\rho \bar{\mathbf{M}} u_i^{n+1} = \rho \bar{\mathbf{M}} u_i^\star - \Delta t \mathbf{G}(p^{n+1} - \gamma p^n), \tag{43}$$

where the matrices are defined by:

$$\mathbf{M} = \int_\Omega N_q^{n+1} N_l^{n+1} d\Omega, \tag{44}$$

$$\mathbf{K}(\mu) = \int_\Omega \left(\mu \frac{\partial N_q^{n+1}}{\partial x_i} \frac{\partial N_l^{n+1}}{\partial x_i} \right) d\Omega, \tag{45}$$

$$\mathbf{D} = \int_\Omega \left(\frac{\partial N_q^{n+1}}{\partial x_i} N_l^{n+1} \right) d\Omega, \tag{46}$$

$$\mathbf{G} = \int_\Omega \left(N_q^{n+1} \frac{\partial N_l^{n+1}}{\partial x_i} \right) d\Omega, \tag{47}$$

$$\mathbf{L}(\frac{1}{\rho}) = \int_\Omega \left(\frac{1}{\rho} \frac{\partial N_q^{n+1}}{\partial x_i} \frac{\partial N_l^{n+1}}{\partial x_i} \right) d\Omega, \tag{48}$$

$$\mathbf{F} = \int_\Omega N_q^{n+1} f_i d\Omega, \tag{49}$$

$$\mathbf{S}(\rho u_i) = \int_\Omega \left(\rho N_q \bar{u}_i^{n+1} \frac{\partial N_l}{\partial x_i} \right) d\Omega, \tag{50}$$

$$\tilde{\mathbf{U}} = \int_\Gamma N_q^{n+1} u_i^{n+1} n_i d\Gamma. \tag{51}$$

The indices (q,l) denote the nodes of an element and $n+1$ implies that the interpolating functions have been computed at time step t^{n+1}. The matrix $\bar{M}$ in Equation (43) is actually the lumped mass matrix M of Equation (44) obtained by adding the elements of M over a row and placing the result on the diagonal of the matrix.

Equations (41),(42) and (43) form the system that needs to be solved.

4.3.2 Predictor corrector scheme

In the previously defined system of equations, it can be observed that in order to be able to solve Equation (41), an estimation of $\bar{u}_i^{n+1}$ needs to be given. The choice of $\bar{u}_i^{n+1}$ can be the object of discussions as it needs to be the closest possible to the actual value of u_i^{n+1}. An estimated average value of : $\bar{u}_i^{n+1} = 2u_i^n - u_i^{n-1}$ is being used by default for the first solve of the system. However, the solver also provides an option to trigger an iterative procedure on the system. Once u^* and p^{n+1} have been determined using Equation (41) and (42), they can be re-injected in the system in order to form an iterative loop :

$$\rho \mathbf{M} u_k^\star + \Delta t \mathbf{K}(\mu) u_k^\star + \mathbf{S}(\rho u_{k-1}^\star) u_k^\star = \rho \mathbf{M} u^n - \gamma \Delta t \mathbf{G} p_{k-1}^{n+1} + \Delta t \mathbf{F}, \tag{52}$$

$$\Delta t \mathbf{L}(\frac{1}{\rho}) p_k^{n+1} = \mathbf{D} u_k^\star + \gamma \Delta t \mathbf{L}(\frac{1}{\rho}) p^n - \tilde{\mathbf{U}}, \tag{53}$$

where k is the number of iterations (the vector indices do not show for clarity purposes).

This procedure aims to bring more precision to the final pressure and velocity values but is often very time consuming and does not necessarily provide significant additionnal precision. It must be therefore used with caution in specific cases. For stability purposes, this method is authomatically used for the first ICFD time step where ten iterations are conducted. Otherwise, it can be accessed through the ICFD CONTROL SPLIT card (see Keyword manual).

4.4 Integration scheme stabilization

4.4.1 Pressure stabilization

The system formed by Equations (41), (42) and (43) can be solved as such but it can be shown that stability issues arise ([**?**]) depending on the time step size. It can be anticipated that if Δt is very small, stability problems may occur, especially for the second order scheme $\gamma = 1$ used by the solver.
Let's start by observing the $DM^{-1}G$ represents the Laplacian operator at the continuous level. When using discrete elements, this leads to the approximation :

$$\mathbf{D}\mathbf{M}^{-1}\mathbf{G} \simeq \mathbf{L}, \tag{54}$$

Let us now introduce the auxiliary variable Π such as :

$$\Pi = \mathbf{M}^{-1}\mathbf{G}p, \tag{55}$$

This will allow us to introduce a diffusion term in order to stabilize the pressure term that will not lower the order of the scheme :

$$\begin{aligned}
\rho\mathbf{M}u_i^\star + \mathbf{K}(\mu)u_i^\star + \mathbf{S}(\rho\bar{u}_i^{n+1})u_j^\star &= \rho\mathbf{M}u_i^n - \gamma\Delta t\mathbf{G}p^n + \Delta t\mathbf{F}, &(56)\\
\Delta t\mathbf{L}(\frac{1}{\rho})p^{n+1} &= \mathbf{D}u_i^\star + \gamma\Delta t\mathbf{L}(\frac{1}{\rho})p^n - \tilde{\mathbf{U}} - \tau(\mathbf{L}p^{n+1} - \mathbf{D}\pi^n), &(57)\\
\rho\bar{\mathbf{M}}u_i^{n+1} &= \rho\bar{\mathbf{M}}u_i^\star - \Delta t\mathbf{G}(p^{n+1} - \gamma p^n), &(58)\\
\Pi &= \mathbf{M}^{-1}\mathbf{G}p, &(59)
\end{aligned}$$

where τ is a stabilization parameter which depends on the local element sizes and behaves as :

$$\tau \leq C\frac{h^2}{\nu}, \tag{60}$$

where C is a constant.

This stabilization term is consistent with the approximation introduced by Equation (54) which means that the stabilization term will get smaller when using a finer mesh i.e when getting closer to the hypothetical continuous level.

4.4.2 Convection stabilization

The idea behind the stabilization of the convection term is similar to the pressure stabilization and has been developed in ([**?**]). We will only give the final system here that will consist of :

$$\begin{aligned}
\rho\mathbf{M}u_i^\star + \Delta t\mathbf{K}(\mu)u_i^\star + \mathbf{S}(\rho\bar{u}_i^{n+1})u_j^\star &= \tau(\mathbf{S}_y(u_i^n)y_j^n - \mathbf{S}_u(u_i^n)u_j^{n+1}) + \rho\mathbf{M}u_i^n - \gamma\Delta t\mathbf{G}p^n + \Delta t\mathbf{F}, &(61)\\
\mathbf{M}y_i^n - \mathbf{C}(u_i^\star)u_j^\star &= 0, &(62)\\
\Delta t\mathbf{L}(\frac{1}{\rho})p^{n+1} &= \mathbf{D}u_i^\star + \gamma\Delta t\mathbf{L}(\frac{1}{\rho})p^n - \tilde{\mathbf{U}} - \tau(\mathbf{L}p^n - \mathbf{D}\pi^n), &(63)\\
\rho\bar{\mathbf{M}}u_i^{n+1} &= \rho\bar{\mathbf{M}}u_i^\star - \Delta t\mathbf{G}(p^{n+1} - \gamma p^n), &(64)\\
\Pi &= \mathbf{M}^{-1}\mathbf{G}P, &(65)
\end{aligned}$$

with :

$$\mathbf{S}_u = \int_\Omega \left(\frac{\partial N_q}{\partial x_i} u_i^n \frac{\partial N_l}{\partial x_i} u_i^n \right) d\Omega, \tag{66}$$

$$\mathbf{S}_y = \int_\Omega \left(\frac{\partial N_q}{\partial x_i} u_i^n N_l \right) d\Omega, \tag{67}$$

$$\mathbf{C} = \int_\Omega \left(N_q u_i^n \frac{\partial N_l}{\partial x_i} \right) d\Omega, \tag{68}$$

4.5 Watching and interpreting the analysis

We have seen in the previous sections that the fractional step method consists in segregating the velocity and pressure terms thus resulting in Equation (41) and (42) that must be solved implicitly before finally updating the velocity using Equation (43). When running a CFD only analysis, it is possible for the user to track the solving of these systems through the terminal window. Assuming that the maximum level of output information defined in the ICFD CONTROL OUTPOUT card has been triggered, this kind of information extracted from a 2D case, will appear :

```
21 t 4.6000E-01 dt 2.00E-02 ICFD time step
ICFD::Fluid → Solving...
PFEM:: It 2
PFEM:: It 2
ICFD:: Sub-step Res: 0.000203344 0.000001065
PFEM:: It 5
Frac Step Pres Residual = 0.016417
PFEM:: It 2
PFEM:: It 2
ICFD:: Sub-step Res: 0.000079993 0.000000419
ICFD::Saving Results → Saving...
22 t 4.8000E-01 dt 2.00E-02 ICFD time step
```

After each ICFD time step, the solver will give the number of iteration that were needed in order to solve Equation (41) in the three directions of velocity (two in 2D cases) as well the associated residual. The solver will then proceed to solve Equation (42) and will also output the number of iterations as well as the associated residual. Finally, the solver will give the number of iterations needed in order to solve Equation (43) in the three directions (two in 2D cases) as well as the associated residual. The solver will then proceed to the next ICFD time step. The user can this way make sure that the residuals stay low at each time step and that the analysis is not starting to diverge.

5 Structure and Thermal coupling

5.1 Fluid Structure Interaction (FSI)

5.1.1 Types of FSI coupling

Fluid-structure interaction problems involving an incompressible viscous flow and elastic non linear-structure have been solved in the past using different methods.

The monolithic approach considers the fluid and the solid as a single domain with the fluid and solid equations solved together in a coupled way. However, solving the pressure together with the rest of the unknowns (typically velocities or displacements) is too expensive from the computational point of view : the non-linear system to be solved is large and ill-conditioned with always non-defined positive matrices.

A second approach would be to segregate the pressure from the velocity in the monolithic scheme. This would still imply that the fluid and solid equations are solved in a coupled way in the same system and would therefore require an implementation of the solid equations directly in the ICFD solver. This would not be a practical solution and would contradict the objective of the present solver which is to make fully use of LS-DYNA's mechanical solver capabilities in order to solve complex fluid structure interaction problems.

A third approach would be by using a partitioned (or staggered) method where the fluid and solid equations are uncoupled and that therefore allows using specifically designed codes on the different domains and offer significant benefits in terms of efficiency: smaller and better conditioned subsystems are solved instead of a single problem. It is the method adopted by the present solver in order to solve FSI problems.

In the partitioned approach, two schemes are distinguished : loosely (or weakly) coupled scheme or strongly coupled scheme. Both are available when using the ICFD solver in LS-DYNA. Loosely coupled schemes require only one solution of either field per time step in a sequentially staggered manner and are thus particularly appealing in terms of efficiency. However, they tend to become unstable when the ”added-mass effect” is significant.

In fluid mechanics, added mass or virtual mass is the inertia added to a system because an accelerating or decelerating body must move some volume of surrounding fluid as it moves through it, since the object and fluid cannot occupy the same physical space simultaneously. However the name ”added-mass effect” has been used in the literature to indicate the numerical instabilities that typically occur in the internal flow of an incompressible fluid whose density is close to the structure density. Other causes for instabilities include the elasticity coefficients or a time step too small. The added mass effect therefore does usually not occur in aero-elasticity problems as the solid's density is a lot higher than the fluid's density, but it becomes very important in several other applications such as bio-mechanics where the materials are normally muscles and arteries and the fluid is blood.

Strongly coupled schemes require the convergence of the fluid and solid variables at the interface and give, after an iterative process, the same results as non-partitioned schemes. However, they are also subject to the ”added mass effect” resulting in a non convergence of the solution. Special stabilization techniques must therefore be developed in order to diminish its influence and aim for a wider range of applications for fluid structure interactions. This stabilization method will be developed in the next section. Figure (1) gives a summary of the different FSI couplings possible, those implemented in the ICFD solver and the influence of the ”added mass effect”.

5.1.2 Evaluation of the Laplace Matrix for FSI problems

One approximation induced by the construction of the fractional step scheme is that no stiffness matrix depending on the viscosity term appears (see Equation (42))and that the Laplace matrix **L** simply includes the mass matrix :

$$\mathbf{L} \approx \mathbf{D}\mathbf{M}^{-1}\mathbf{G}, \tag{69}$$

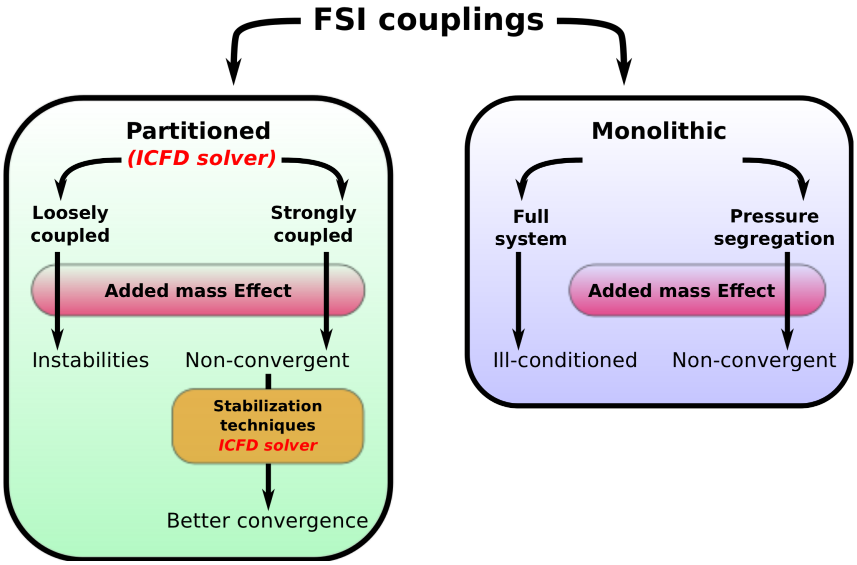

Figure 1: Summary of the different FSI couplings and the numerical troubles induced by the added-mass effect.

However, for FSI problems, a more robust approximation will be needed for stabilization purposes and the Laplace matrix will be expressed as :

$$\frac{\Delta t}{\rho}\mathbf{L}_{\tau_f} = \mathbf{D}\left(\frac{\rho\mathbf{M}}{\Delta t}+\mathbf{K}\right)^{-1}\mathbf{G}, \tag{70}$$

where $\mathbf{K}$ is the stiffness function depending on the viscosity μ in the fluid part. A suitable approximation for the stiffness matrix including the convection term would be :

$$\mathbf{K} \approx \left(\frac{\mu}{h^2}+\frac{\|V\|}{h}\right)\mathbf{M}, \tag{71}$$

where h represents a characteristic element size and V the particle velocity.
The new Laplace matrix can therefore be approached as :

$$\mathbf{L}_{\tau} \approx \mathbf{L}_{\tau_f} + \bar{\mathbf{L}}_{\tau_s}, \tag{72}$$

where $\mathbf{L}_{\tau_f}$ is the Laplace matrix defined in Equation (70) corresponding to the fluid domain including the interfaces :

$$\mathbf{L}_{\tau_f} = \sum(\tau_f \mathbf{L}^e), \tag{73}$$

with $\mathbf{L}^e$ being the Laplace matrix for the element and :

$$\tau_f = \left(\frac{\rho}{\Delta t}+\frac{\mu}{h^2}+\frac{\|V\|}{h}\right)^{-1}, \tag{74}$$

$\bar{\mathbf{L}}_{\tau_s}$ is a Laplace matrix corresponding only to the fluid solid interface :

$$\bar{\mathbf{L}}_{\tau_s} = \sum(\tau_s \bar{\mathbf{L}}^e), \tag{75}$$

with :

$$\begin{aligned}\tau_s &= \left(\frac{\rho_{solid}}{\Delta t}+\frac{\Delta t\lambda}{Jh^2}+\frac{\Delta t G}{Jh^2}\right)^{-1}, &(76)\\ \bar{\mathbf{L}}^e &= \left[\int_\Omega \frac{\partial N_{sf}}{\partial x_j}\frac{\partial N_{sf}}{\partial x_j}d\Omega\right]_s &(77)\end{aligned}$$

where λ and G are Lame parameters, J the Jacobian matrix and $\bar{\mathbf{L}}^e$ is the Laplace matrix of the solid elements evaluated only with the shape function N_{sf} that are different from zero on the fluid-structure interface.

Equation (72) may also be written as :

$$\mathbf{L}_\tau \approx \mathbf{L}_{\tau_f}+\beta\bar{\mathbf{L}}_{\tau_f}, \qquad (78)$$

with :

$$\beta=\frac{\tau_s}{\tau_f}, \qquad (79)$$

and :

$$\bar{\mathbf{L}}_{\tau_f}=\sum(\tau_f\bar{\mathbf{L}}^e), \qquad (80)$$

This means that the Laplace interface matrix $\bar{\mathbf{L}}_{tau_f}$ may be neglected for small values of the β parameter. This is for instance the case when $\rho_s >> \rho_f$ and the added-mass effect is not present. However, for other physical properties, the β parameter may not be negligible and the Laplace interface matrix must be evaluated in order to obtain good results. It has been shown (**[?]**) that the $\beta\bar{\mathbf{L}}_{tau_f}$ term improves the convergence when $\rho_f \approx \rho_s$, small time steps and when using soft materials.

5.1.3 FSI resolution steps

In section 5.1.1, we have classified the fluid structure interaction types and seen that the ICFD solver allows loosely or strongly coupled schemes. Section 5.1.2 has showed the stabilization steps adopted in order to achieve better convergence when the added-mass effect is important thus allowing the solving of a wider range of FSI applications. It is now time to sum up and present in Figure (3) and (2) the resolution steps adopted by the ICFD solver when solving FSI problems for the strong and loose couplings.

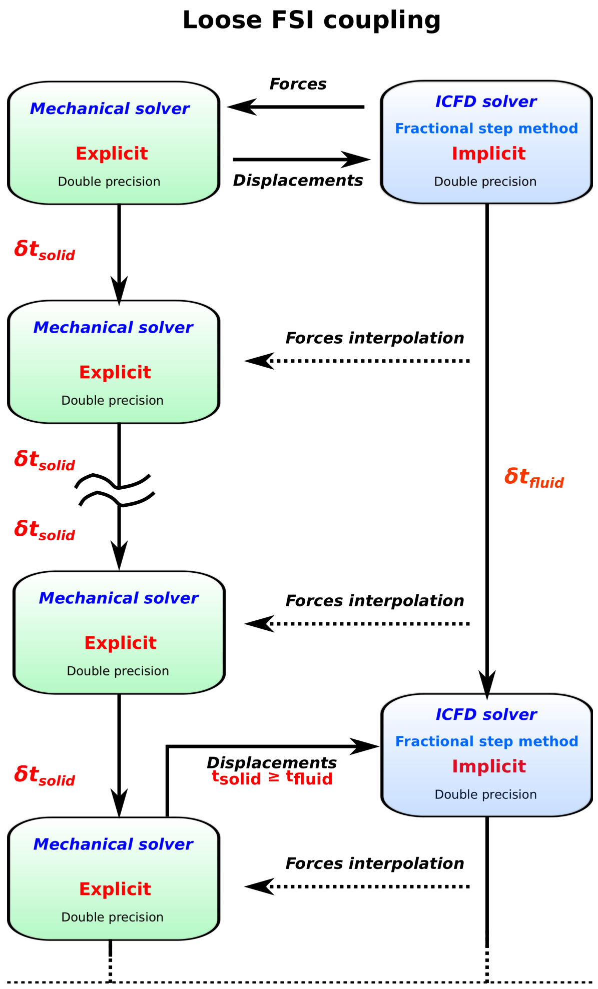

Figure 2: Loose FSI interaction resolution scheme.

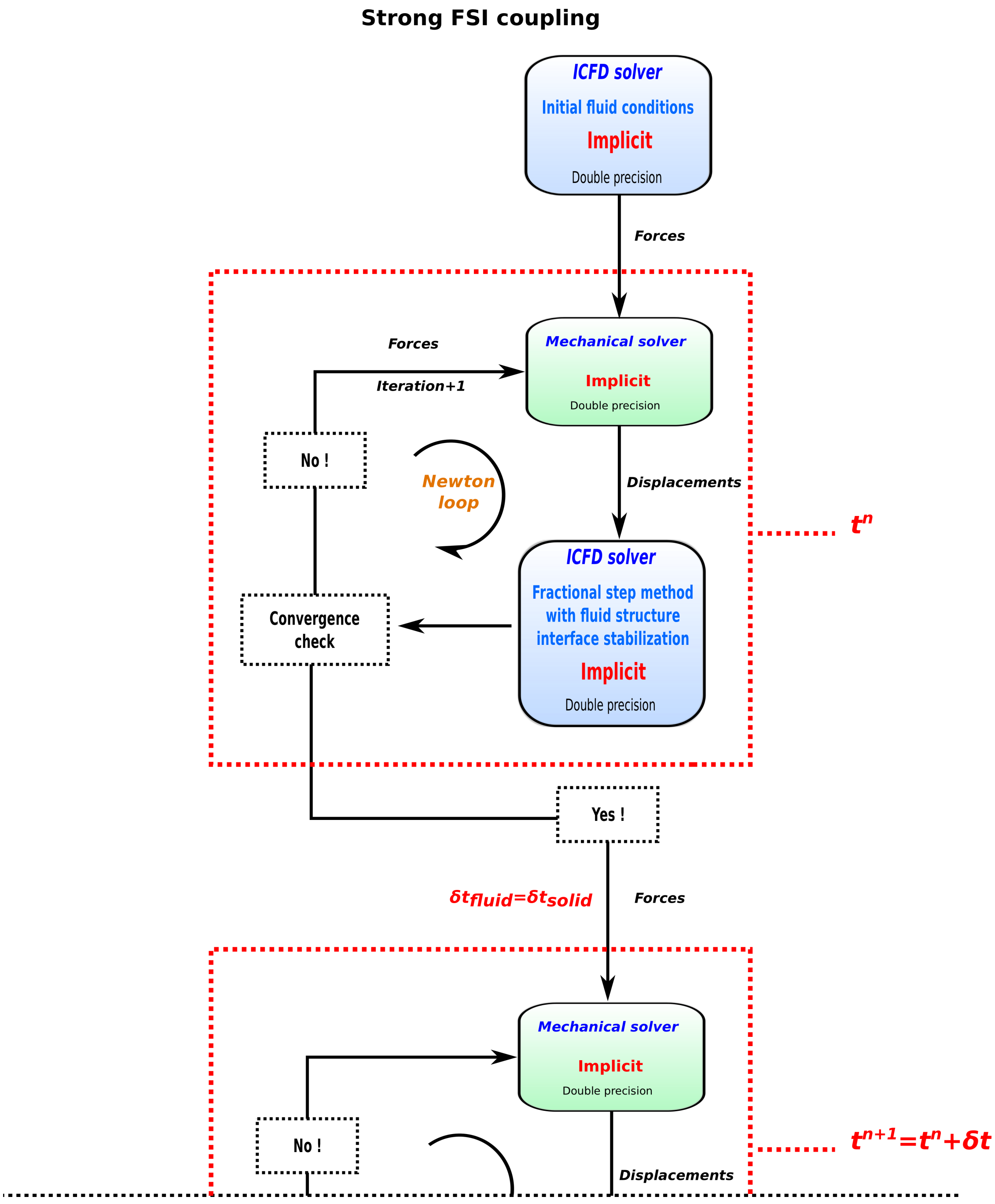

Figure 3: Strong FSI interaction resolution scheme.

5.1.4 Watching and controlling the analysis for FSI cases

5.2 Thermal coupling

5.2.1 Introduction

Heat transfer is a discipline of thermal engineering that concerns the generation, use, conversion, and exchange of thermal energy and heat between physical systems. The ICFD solver offers the possibility to solve and study the behavior of temperature flow in fluids. Potential applications are numerous and include refrigeration, air conditioning, building heating, motor coolants, defrost or even heat transfer in the human body. Furthermore, the ICFD thermal solver is fully coupled with the thermal solver using a monolithic approach which allows to solve complex problems where both heated structures and flows are present and interact together.

5.2.2 The heat equation

The distribution of heat in a given region over time is described by a convection-diffusion equation also called heat equation :

$$\frac{\partial T}{\partial t} + u_j \frac{\partial T}{\partial x_j} - \alpha \frac{\partial^2 T}{\partial x_j x_j} = f \quad in \quad \Omega \tag{81}$$

where α is a positive constant called thermal diffusivity and f is a potential source of heat.

This formulation is incomplete if the appropriate set of boundary conditions and initial conditions is not specified. The user can specify the temperature on the boundaries resulting in Dirichlet conditions :

$$T(\vec{x},t) = T(\vec{x}) \quad on \quad \Gamma \tag{82}$$

Furthermore, if no boundary condition is applied, the solver will automatically apply a Neumann condition :

$$\vec{n}(\vec{x}).\vec{\nabla}T(x,t) = 0 \quad on \quad \Gamma \tag{83}$$

where $\vec{n}(\vec{x})$ is the normalized normal vector at the $\vec{x}$ point.

5.2.3 Spatial discretization

The time derivative will be approximated with a backward differences of second order as :

$$\frac{1.5T^{n+1} - 2T^n + 0.5T^{n-1}}{\Delta t} + u_j^{n+1} \frac{\partial T^{n+1}}{\partial x_j} - \alpha \frac{\partial^2 T^{n+1}}{\partial x_j x_j} = f^{n+1} \tag{84}$$

The Galerkin method will now be applied to Equation (84) in order to find a weak form that will be integrated over the prescribed domain :

$$\int_\Omega N_i \left(1.5T^{n+1} - 2T^n + 0.5T^{n-1}\right) d\Omega + \int_\Omega \Delta t N_i u_j \frac{\partial T^{n+1}}{\partial x_j} d\Omega - \Delta t \int_\Omega N_i \alpha \frac{\partial}{\partial x_j}\left(\frac{\partial T^{n+1}}{\partial x_j}\right) d\Omega = \Delta t \int_\Omega N_i f^{n+1} d\Omega, \tag{85}$$

Resulting in the matrix form :

$$\mathbf{M}\left(1.5T^{n+1} - 2T^n + 0.5T^{n-1}\right) + \Delta t \mathbf{S}(u_i^n) T^{n+1} - \Delta t \mathbf{K}(\alpha) T^{n+1} = \Delta t \mathbf{F}, \tag{86}$$

As in Section (4.4.2), a similar convection term stabilization process has to be performed in order to get to the final system :

$$\mathbf{M}\left(1.5T^{n+1}-2T^{n}+0.5T^{n-1}\right)+\Delta t\mathbf{S}(u_i^n)T^{n+1}-\Delta t\mathbf{K}(\alpha)T^{n+1}-\tau\left(\mathbf{S}_y(u_i^n)T^n_{proj}-\mathbf{S}_u(u_i^n)T^{n+1}\right)=0, \tag{87}$$

$$\mathbf{M}T^n_{proj}-\mathbf{C}(u_i^n)T^n=\Delta t\mathbf{F}, \tag{88}$$

5.2.4 Coupling with the thermal solver

In Section (5.1.1), the different types of fluid structure couplings were presented. The same vocabulary can be employed to describe the different couplings between the heat equations solved in the fluid by the ICFD solver and the heat equations solved by the thermal solver in the structure. For the thermal coupling, a monolithic approach has been adopted. The coupling between the structure and the fluid is therefore very tight and strong at the fluid-structure interface. The resulting full system includes both the structural and the fluid temperature unknowns (See Figure (4)) and is solved using a direct solver which may in some cases be computer-time consuming.

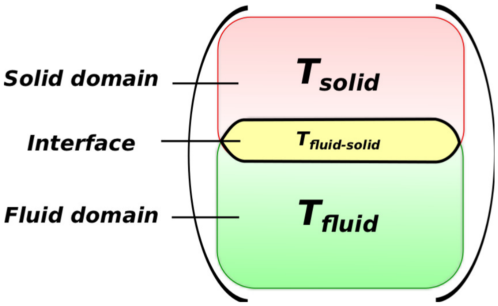

Figure 4: Vector of temperature unknowns when the fluid thermal solver and structure thermal solvers are coupled using a monolithic approach.

5.2.5 Watching and controlling the analysis for thermal analysis.

5.3 Summary

Figure (5) offers a simplified view of the interactions and couplings of the mechanical, thermal and ICFD solvers.

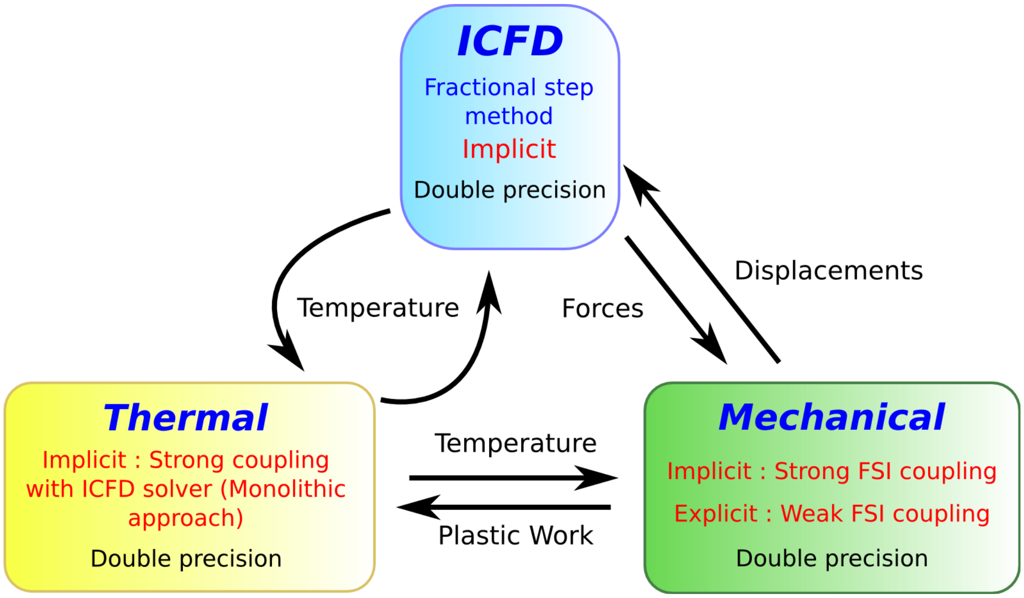

Figure 5: Summary of the different possible interactions between solvers

6 Free surface and mutliphase handling

6.1 The level set method

6.1.1 Introduction

A large collection of fluid problems involves moving interfaces. Applications include air-water dynamics, breaking surface waves and solid bodies penetrating in fluids. In many such applications, the interplay between the interface dynamics and the surrounding fluid motion is subtle, with factors such as density ratios and temperature jumps across the interface, surface-tension effects, topological connectivity, and boundary conditions playing significant roles in the dynamics. The solver uses a level set method based on [**?**], a fast and reliable technique in order to track and correctly represent moving interfaces.

6.1.2 The level set function

Let us first start by introducing the implicit function ϕ whose zero isocontour, $\phi = 0$ represents the interface. The implicit function ϕ is defined throughout the whole computational domain while the isocontour defining the interface is one dimension lower. As a convention the fluid domain where the Navier-Stokes equations will be solved (see Section (4.1)) is defined by $\phi > 0$ while the vacuum is defined by $\phi < 0$. Additionally, ϕ will be described as a distance function resulting in the level set function :

$$\phi(\vec{x}) = min(|\vec{x}-\vec{x}_I|) \qquad on\ \Omega^+\ for\ all\ \vec{x}_I\ \in\ \partial\Omega \tag{89}$$

$$\phi(\vec{x}) = 0 \qquad on\ \partial\Omega\ for\ all\ \vec{x}_I\ \in\ \partial\Omega \tag{90}$$

$$\phi(\vec{x}) = -min(|\vec{x}-\vec{x}_I|) \qquad on\ \partial\Omega^-\ for\ all\ \vec{x}_I\ \in\ \partial\Omega \tag{91}$$

Furthermore, since ϕ is Euclidean distance :

$$|\vec{\nabla}\phi| = 1, \tag{92}$$

This is shown in Figure (6) with the classical dam example where the fluid ($\phi > 0$) and the vacuum ($\phi < 0$) are divided by the interface $\phi = 0$. In the case of two phase fluids, the problem is similar except that the Navier-Stokes equations are solved in the two domains with a smoothing of the density and viscosity values at the interface.

6.1.3 The convection equation

Now, let us suppose that the velocity of each points on the implicit surface is given as $\vec{V}(\vec{x})$ i.e $\vec{V}(\vec{x})$ is known for every point $\vec{x}$ with $\phi(\vec{x}) = 0$. The simplest way to move all the points on the surface with this velocity would be to move the nodes located on the interface by their velocity V through the grid (see Figure (6)). This Lagrangian formulation would not be too hard to accomplish ([**?**]) if the connectivity does not change and the surface elements are not distorted too much. However, even small velocity fields could cause large distortion of the interface elements and the accuracy of the method can deteriorate quickly if a frequent and regular re-meshing of the domain is not applied. This need for frequent re-meshing implies higher computer costs and less scalability when running with multiple processors.

In order to avoid these problems, the implicit function ϕ will be used both to represent the interface and to evolve the interface (see Figure (6)). In order to define the evolution of the implicit function ϕ, the simple convection equation will be used :

$$\frac{\partial\phi}{\partial t} + \vec{V}.\vec{\nabla}\phi = 0, \tag{93}$$

Equation (93) defines the motion of the interface where $\phi(\vec{x}) = 0$. It is an Eulerian formulation of the interface evolution ([**?**]) since the interface is captured by the implicit function ϕ as opposed to being tracked by the displacement of the interface nodes as was done in the Lagrangian formulation.

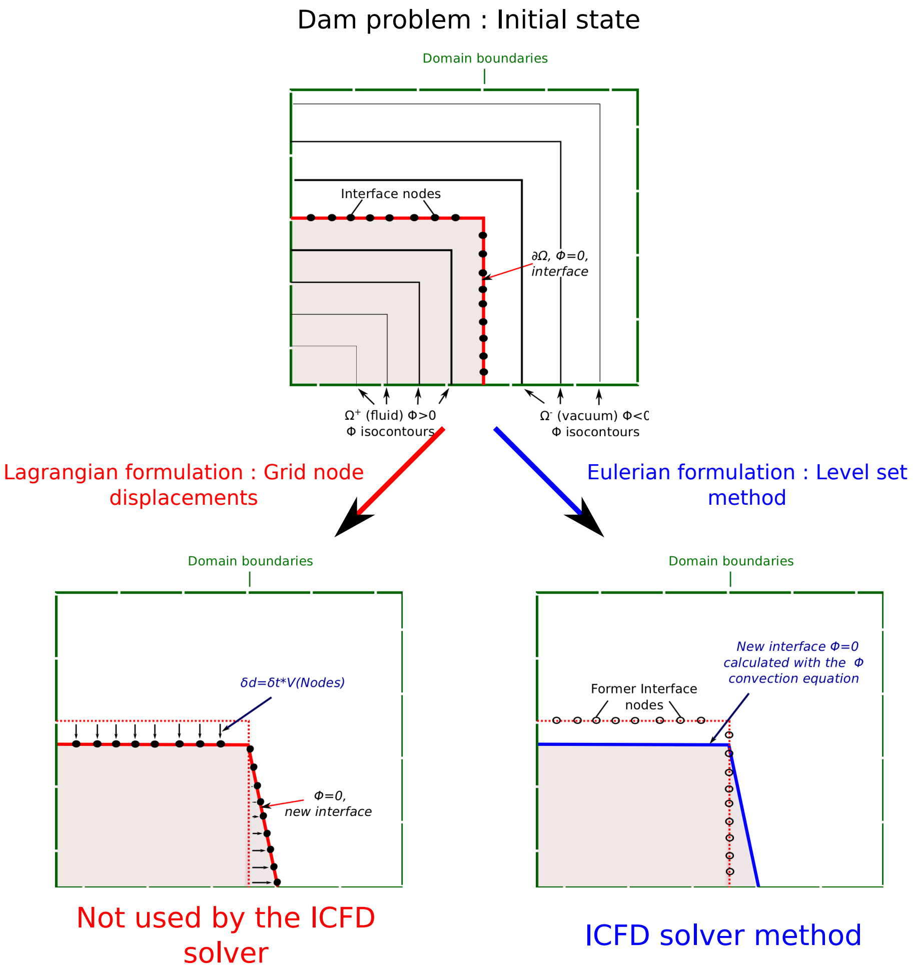

Figure 6: Free surface dam breaking problem : description of the different interface tracking methods

6.1.4 Numerical integration

As in Section (4.3), the Galerkin method will be applied to Equation (93) in order to find a weak form that will be integrated over the prescribed domain :

$$\int_{\Omega} N_i \left(1.5\phi_i^{n+1} - 2\phi_i^n + 0.5\phi_i^{n-1} \right) d\Omega + \int_{\Omega} \Delta t N_i u_j^n \frac{\partial \phi_i^{n+1}}{\partial x_j} d\Omega = 0, \tag{94}$$

Resulting in the matrix form :

$$\mathbf{M} \left(1.5\phi_i^{n+1} - 2\phi_i^n + 0.5\phi_i^{n-1} \right) + \Delta t \mathbf{S}(u_i^n)\phi_j^{n+1} = 0, \tag{95}$$

As in Section (4.4.2), a similar convection term stabilization process has to be performed in order to get to the final system :

$$\mathbf{M} \left(1.5\phi_i^{n+1} - 2\phi_i^n + 0.5\phi_i^{n-1} \right) + \Delta t \mathbf{S}(u_i^n)\phi_j^{n+1} - \tau \left(\mathbf{S}_y(u_i^n)\psi_j^n - \mathbf{S}_u(u_i^n)\phi_j^{n+1} \right) = 0, \tag{96}$$

$$\mathbf{M}\psi_i^n - \mathbf{C}(u_i^n)\phi_j^n = 0, \tag{97}$$

7 Turbulence models

7.1 The RANS model

7.1.1 Introduction

Reynolds-averaged Navier-Stokes (RANS) equations are the most classic approach to turbulence modeling. An time averaged version of the governing equations is solved, which introduces new apparent stresses known as Reynolds stresses. This adds a second order tensor of unknowns for which various models can provide different levels of closure. The incompressible solver provides a k-ε model which is one of the most widely used turbulence models in CFD. This model adds two additional variables, the turbulent kinetic energy *k*, and the turbulent dissipation ε and provides two additional equations that close the model and account for history effects like convection and diffusion of turbulent energy. Since RANS models consider a time averaged version of the governing equations, such models will be more adequate in non dynamic problems where a stationary solution can be reached.

7.1.2 The Reynolds Averaged Equations

When solving Equation (10) and (11), instantaneous quantities are being considered. In order to define the RANS model, the flow will be divided in two parts, a mean(or average) component and a fluctuating component. Thus the instantaneous velocity and pressure can be written as :

$$\begin{aligned} u_i &= \bar{u}_i + u'_i && (98) \\ p &= \bar{P} + p' && (99) \end{aligned}$$

This technique for decomposing the instantaneous motion is referred to as the Reynolds decomposition. Substitution of Equations (98) and (99) in (10) and (11) yields :

$$\begin{aligned} \rho(\frac{\partial \bar{u}_i + u'_i}{\partial t} + (\bar{u}_j + u'_j)\frac{\partial(\bar{u}_i + u'_i)}{\partial x_j}) &= -\frac{\partial(\bar{P} + p')}{\partial x_i} + \mu\frac{\partial^2(\bar{u}_i + u'_i)}{\partial x_j \partial x_j} \quad in \quad \Omega && (100) \\ \frac{\partial(\bar{u}_i + u'_i)}{\partial x_i} &= 0 \quad in \quad \Omega && (101) \end{aligned}$$

Equations (100) and (101) can now be averaged to yield an equation expressing momentum conservation for the average motion. By noting that the average of a derivative is the same as the derivative of the average and that the time average of the fluctuating quantity is zero, resulting in the following Reynolds-averaged Navier Stokes equations (or RANS equations) :

$$\begin{aligned} \rho(\frac{\partial \bar{u}_i}{\partial t} + (\bar{u}_j)\frac{\partial(\bar{u}_i)}{\partial x_j}) &= -\frac{\partial(\bar{P})}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\frac{\partial \bar{u}_i}{\partial x_j} - \rho\overline{u'_i u'_j}\right] \quad in \quad \Omega && (102) \\ \frac{\partial(\bar{u}_i)}{\partial x_i} &= 0 \quad in \quad \Omega && (103) \end{aligned}$$

As can be seen from Equations (102) and (103), the time averaged equations are very similar to the instantaneous equations but for the apparition of a new apparent viscous stress term, $\rho\overline{u'_i u'_j}$ owing to the fluctuating velocity field generally referred to as the Reynolds stresses. The aim of the different RANS models will be, under given hypotheses, to provide additional equations that allow the solving of the system.

7.1.3 The k-ε model

For the moment, the standard $k - \varepsilon$ model is implemented, but more models may be implemented in the future. The $k - \varepsilon$ model is two equation model based on the Boussinesq hypothesis that stipulates that the momentum transfer caused by turbulent eddies can be modeled with an eddy viscosity allowing to model the Reynolds stresses as :

$$\begin{aligned} \rho\overline{u_i'u_j'} &= 2\mu_t S_{i,j} - \frac{2}{3}\rho k \delta_{i,j} && (104)\\ \mu_t &= \rho C_\mu \frac{k^2}{\varepsilon} && (105)\\ S_{i,j} &= \frac{1}{2}\frac{\partial \bar{u}_i}{\partial x_j} - \frac{1}{3}\frac{\partial \bar{u}_k}{\partial x_k}\delta_{i,j} && (106) \end{aligned}$$

where μ_t is the eddy viscosity, $S_{i,j}$ is the mean strain rate, k the mean turbulent kinetic energy and ε the turbulent dissipation. The $k-\varepsilon$ model then provides the two mandatory transport equations in order to close the model :

$$\begin{aligned} \frac{\partial \rho k}{\partial t} + \frac{\partial(\rho k \bar{u}_i)}{\partial x_i} &= \frac{\partial}{\partial x_j}\left[(\mu + \frac{\mu_t}{\sigma_k})\frac{\partial k}{\partial x_j}\right] + \mu_t S^2 - \rho\varepsilon. && (107)\\ \frac{\partial \rho \varepsilon}{\partial t} + \frac{\partial(\rho \varepsilon \bar{u}_i)}{\partial x_i} &= \frac{\partial}{\partial x_j}\left[(\mu + \frac{\mu_t}{\sigma_\varepsilon})\frac{\partial \varepsilon}{\partial x_j}\right] + C_{1\varepsilon}\frac{\varepsilon}{k}\mu_t S^2 - C_{2\varepsilon}\rho\frac{\varepsilon^2}{k} && (108) \end{aligned}$$

with the following default constants that can be changed by the user through the ICFD CONTROL TURBULENCE card :

$$\begin{aligned} C_{1\varepsilon} &= 1.44 && (109)\\ C_{2\varepsilon} &= 1.92 && (110)\\ \sigma_\varepsilon &= 1.3 && (111)\\ \sigma_k &= 1.3 && (112)\\ C_\mu &= 0.09 && (113)\\ & && (114) \end{aligned}$$

7.2 The LES model

7.2.1 Introduction

As the power of computer increases Large Eddy Simulation (LES) models have become a popular technique in order to simulate turbulence. Those models are based on the assumption that large eddies contain most of the kinetic energy of the flow and depend on the geometry while the smaller ones are considered more universal and independent of the flow's geometry. Therefore, instead of using time averaged quantities as in RANS models, LES models divide each flow variable into a large-scale (resolved) component, which is still time-dependent allowing the analysis of instantaneous quantities even in flows that are stationary in the average, and into a subgrid (estimated) component. The filtering is usually mesh size dependent and the model used by the ICFD solver is the classical Smagorinsky model.

7.2.2 LES equations

The quantities will be divided in a resolvable scale part U_i and in a subgrid scale part u_i^{sub} :

$$\begin{aligned} u_i &= \widetilde{U}_i + u_i^{sgs} && (115)\\ p &= \widetilde{P} + p^{sgs} && (116) \end{aligned}$$

Applying this decomposition to Equation (10) yields after filtering the resulting equation :

$$\rho(\frac{\partial \widetilde{U_i}}{\partial t} + (\widetilde{U_j})\frac{\partial(\widetilde{U_i})}{\partial x_j}) = -\frac{\partial \widetilde{P}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu \frac{\partial \widetilde{U_i}}{\partial x_j} + \rho \tau_{i,j}\right] \quad in \quad \Omega \tag{117}$$

with the extra stress term due to the sub grid scales (SGS) :

$$\tau_{i,j} = \widetilde{U_i}\widetilde{U_j} - \widetilde{U_i U_j} \tag{118}$$

Subgrid-scale turbulence models usually employ the Boussinesq hypothesis in order to express the SGS stress as a turbulent viscosity μ_{sgs} :

$$\rho(\frac{\partial \widetilde{U_i}}{\partial t} + (\widetilde{U_j})\frac{\partial(\widetilde{U_i})}{\partial x_j}) = -\frac{\partial \widetilde{P}}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu \frac{\partial \widetilde{U_i}}{\partial x_j} + \mu_{sgs}\frac{\partial \widetilde{U_i}}{\partial x_j}\right] \quad in \quad \Omega \tag{119}$$

7.2.3 The Smagorinsky model

The Smagorinsky model describes the turbulent viscosity as :

$$\mu_{sgs} = \rho(C_s\Delta)^2|\widetilde{S}| \tag{120}$$

where the filter width is usually taken as :

$$\Delta = (Volume)^{\frac{1}{3}} \tag{121}$$

and :

$$\widetilde{S} = \sqrt{2S_{i,j}S_{i,j}} \tag{122}$$

C_s is a constant named Smagorinsky constant. Its default value in the solver is 0.1 and can be changed in the ICFD CONTROL TURBULENCE card.

7.3 Wall function

8 The Volume Mesher

8.1 Introduction

The ICFD solver uses an automatic volume mesher for the fluid domains. This greatly simplifies the pre-processing stage. For this feature a good quality body-fitted surface mesh has to be provided. For FSI simulations, the solver uses an ALE approach for mesh movement. In the cases where FSI simulations result in large displacements the solver can automatically re-mesh to keep an acceptable mesh quality. This section will focus on giving the basic steps of how the volume mesh is built as well as describe some local refinement and re-meshing tools.

8.2 The Delaunay criteria

Before describing the steps leading to the construction of the volume mesh, let us introduce the so-called Delaunay criterion as its application will be fundamental in the construction of the volume mesh. In mathematics and computational geometry, a Delaunay triangulation for a set of nodes **P** in a plane is a triangulation DT(**P**) such that no node is inside the circumcircle of any triangle DT(**P**). For instance, in Figure (7), the Delaunay criterion is violated since the node $\mathbf{P}_4$ is inside the circumcircle of the triangle composed of $\mathbf{P}_1$, $\mathbf{P}_2$ and $\mathbf{P}_3$.

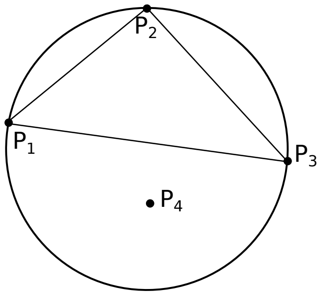

Figure 7: Violation of the Delaunay criterion, the point P_4 is inside the circumcircle of the triangle $P_1P_2P_3$

8.3 Initial Volume mesh building steps

- **Step one** : The solver reads the surface nodes and elements given by the user. The list of surfaces have to be non-overlapping and should not leave any gaps or open spaces between the surface boundaries. The nodes on the boundary of two neighbor surfaces have to be uniquely defined (no duplicate nodes) and should match exactly on the interface. Each surface element has to be orientable i.e a normal vector defining the interior/exterior faces of the surface element can be associated to it. If one of those conditions is not met, the solver will return an error.

- **Step two** : Using the initial surface nodes, the solver will then join the surface nodes in order to build an initial volume mesh. The tetrahedras or triangles in 2D thus created all need to respect the Delaunay criteria. Figure (**??**) shows an example of an initial volume mesh created.

- **Step three** : The solver will progressively add nodes to the volume mesh. Every time a new node is added, its ”host” tetrahedra or triangle will be divided in smaller tetrahedras or triangles respecting the Delaunay criteria. This procedure will be repeated until the volume elements have reached their desired size based on a linear interpolation of the surface sizes that define the volume enclosure.

8.4 Local refinement tools

8.4.1 Imposing the mesh size on a volume domain

During the geometry set up, the user can define surfaces that will be used by the mesher to specify a local mesh size inside the volume. Two cards and methods are available :

- For the MESH SIZE card, a surface with the specified element size must be defined in the input deck and will be read along with the other initial surfaces (Step one). This surface must be entirely defined in the volume mesh but does not need to be necessarily closed. The nodes of this specific surface will directly be used when building the initial volume mesh (Step two). A linear interpolation will then be used to define the element sizes between this specifically added surface and the classic surfaces defining the volume enclosure.

- For the MESH SIZE SHAPE card, a local element size can be defined by the user in specific zones corresponding to given geometrical shapes (box, sphere, cylinder). This element size will be used as reference in the specified zone when progressively adding nodes and building the final volume mesh (Step three). This zone does not necessarily need to be entirely defined in the volume mesh. However, in cases where a boundary surface mesh is defined in this zone, it is advised to use a similar element size for the shape zone and the boundary surface mesh in order to keep a good quality mesh.

8.4.2 The boundary layer mesh

It is also possible to specify several anisotropic elements to be added to the boundary layer in order to better represent close-to-the-wall effects. The solver will extrude the specified surface mesh in the normal direction of the surface elements by a certain distance proportional to their size. The desired number of boundary layer elements will then be added to this thickness by always dividing the element closest to the surface thus resulting in a boundary layer mesh that gets finer when approaching the wall.

8.5 Volume mesh remeshing criteria

8.5.1 Mesh deformation criteria

For FSI simulations, the solver uses an ALE approach for mesh movement which means that large deformations of the fluid mesh can occur. By default, the solver, only rebuilds the mesh if elements get inverted. Inversion of elements usually occur in rotational problems such as wind turbine analyzes. However, it is also possible through the ICFD CONTROL ADAPT SIZE card to trigger a re-meshing of elements that have been distorted by the mesh movement algorithm and that no longer respect the initial mesh size. This frequently occurs if problems involving bodies moving in translation.

8.5.2 Using adaptivity

8.6 Watching the mesh quality

References

A Consistent units for ICFD

	USI	Equivalence ($([kg]^\alpha * [m]^\beta * [s]^\gamma)$)			ex 1	ex 2
Mass	kg	$[kg]^\alpha$	$[m]^\beta$	$[s]^\gamma$	g	g
Length	m				mm	mm
Time	s				s	ms
Energy	J	1	2	-2	$1e^{-9}$	$1e^{-3}$
Force	N	1	1	-2	$1e^{-6}$	1
Stress/Pressure	Pa	1	-1	-2	1	$1e^{6}$
Velocity	$\frac{m}{s}$	0	1	-1	$1e^{-3}$	1
Density	$\frac{kg}{m^3}$	1	-3	0	$1e^{6}$	$1e^{6}$
Dynamic Viscosity	Pa.s	1	-1	-1	1	$1e^{3}$
Thermal Diffu.	$\frac{m^2}{s}$	0	2	-1	$1e^{-6}$	$1e^{-3}$
Heat capacity	$\frac{J}{kgK}$	0	2	-2	$1e^{-6}$	1
Thermal Cond.	$\frac{J}{m.s}$	1	1	-3	$1e^{-6}$	$1e^{3}$

Table 4: Consistent units for the ICFD solver