

Tdyn Theory Manual



Tdyn

1. Table of contents

1.	Table of contents	2
2.	Introduction	3
3.	Navier Stokes solver (RANSOL module)	4
	Stability of CFD algorithms	4
	Finite Calculus (FIC) Formulation.....	5
	Stabilized Navier Stokes Equations	6
	Stabilized integral forms.....	7
	Convective and pressure gradient projections.....	8
	Monolithic time integration scheme.....	9
	Compressible flows.....	11
	Turbulence Solvers	11
	Zero equation (algebraic) model: The Smagorinsky model.....	12
	One-equation models: the kinetic energy model (k-model)	13
	Two-equation models: the k- ϵ model	14
	Implicit LES turbulence model.....	14
	Computation of the characteristic lengths.....	15
4.	Overlapping domain decomposition level set (FSURF Module)	17
	Statement of the problem.....	17
	How to calculate a signed distance to the interface	20
	Interfacial boundary conditions	21
	Stabilized FIC-FEM formulation	23
	Overlapping domain decomposition method	24
	Discretization by the finite element method (FEM).....	27
5.	Heat Transfer Solver (HEATRANS Module)	30
	Stabilized Heat Transfer Equations	30
6.	Species Advection Solver (ADVECT Module)	32
7.	References	33

2. Introduction

Tdyn is a numerical simulation solver for multi-physics problem that uses the stabilized (FIC) finite element method. Using this stabilized numerical scheme, Tdyn is able to solve for instance heat transfer problems in both, fluids and solids, turbulent flows, advection of species and free surface problems. This document gives a short overview of the theoretical principles that Tdyn is based on.

3. Navier Stokes solver (RANSOL module)

The incompressible Navier-Stokes-Equations in a given three-dimensional domain Ω and time interval $(0, t)$ can be written as:

$$\left. \begin{aligned} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) + \nabla p - \nabla \cdot (\mu \nabla \mathbf{u}) &= \rho f && \text{in } \Omega \times (0, t) \\ \nabla \mathbf{u} &= 0 && \text{in } \Omega \times (0, t) \end{aligned} \right\} \text{Eq. 1}$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ denotes the velocity vector, $p = p(\mathbf{x}, t)$ the pressure field, ρ the (constant) density, μ the dynamic viscosity of the fluid and f the volumetric acceleration. The above equations need to be combined with the following boundary conditions:

$$\left. \begin{aligned} \mathbf{u} &= \mathbf{u}_c && \text{in } \Gamma_D \times (0, t) \\ p &= p_c && \text{in } \Gamma_P \times (0, t) \\ \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_1 = 0, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_2 = 0, \quad \mathbf{n} \cdot \mathbf{u} &= \mathbf{u}_M && \text{in } \Gamma_M \times (0, T) \\ \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0(\mathbf{x}) && \text{in } \Omega_D \times (0) \\ p(\mathbf{x}, 0) &= p_0(\mathbf{x}) && \text{in } \Omega_D \times (0) \end{aligned} \right\} \text{Eq. 2}$$

In the above equations, $\Gamma := \partial\Omega$, denotes the boundary of the domain Ω , with \mathbf{n} the normal unit vector, and $\mathbf{g}_1, \mathbf{g}_2$ the tangent vectors of the boundary surface $\partial\Omega$. \mathbf{u}_c is the velocity field on Γ_D (the part of the boundary of Dirichlet type, or prescribed velocity type), p_c the prescribed pressure on Γ_P (prescribed pressure boundary). $\boldsymbol{\sigma}$ is the stress field, \mathbf{u}_M the value of the normal velocity and \mathbf{u}_0, p_0 the initial velocity and pressure fields. The union of Γ_D, Γ_P and Γ_M must be Γ ; their intersection must be empty, as a point of the boundary can only be part of one of the boundary types, unless it is part of the border between two of them.

The spatial discretization of the Navier-Stokes equations has been done by means of the finite element method, while for the time discretization an iterative algorithm that can be consider as an implicit two steps "Fractional Step Method" has been used. Problems with dominating convection are stabilized by the so-called "Finite Increment Calculus" method, presented below.

Stability of CFD algorithms

Using the standard Galerkin method to discretize the incompressible Navier-Stokes equations leads to numerical instabilities coming from two sources. Firstly, owing to the advective-diffusive character of the governing equations, oscillations appear in the solution at high Reynolds numbers, when the convection terms become dominant. Secondly, the mixed character of the equations limits the choice of velocity and pressure interpolations such that equal order interpolations cannot be used.

In recent years, a lot of effort has been put into looking for ways to stabilize the governing equations, many of which involve artificially adding terms to the equations to balance the convection, for example, the artificial diffusion method [7].

A new stabilization method, known as finite increment calculus, has recently been developed [5, 6]. By considering the balance of flux over a finite sized domain, higher order terms naturally appear in the governing equations, which supply the necessary stability for a classical Galerkin finite element discretization to be used with equal order velocity and pressure interpolations. This method has been Christianized “Finite Calculus” (FIC).

Finite Calculus (FIC) Formulation

To demonstrate this method, consider the flux problem associated with the incompressible conservation of mass in a 2D source less finite sized domain defined by four nodes, as shown in Figure 1.

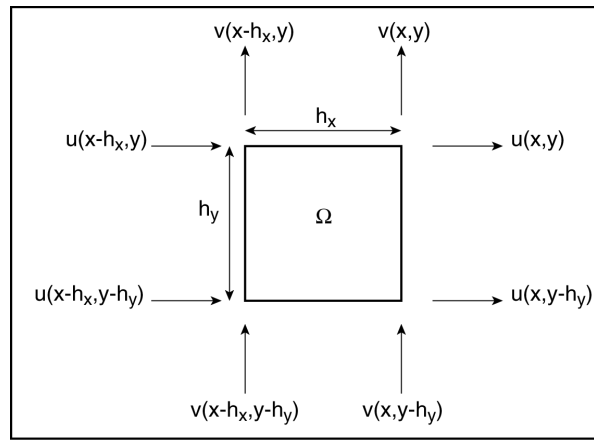


Figure 1 Mass balance in 2 dimensions

Consider the flux problem (mass balance) through the domain, taking the average of the nodal velocities for each surface.

$$\begin{aligned} \frac{h_y}{2} & \left(u(x-h_x, y) + u(x-h_x, y-h_y) - u(x, y) + u(x, y-h_y) \right) \\ & + \frac{h_x}{2} \left(u(x, y-h_y) + u(x-h_x, y-h_y) \right. \\ & \left. - u(x-h_x, y) + u(x, y) \right) = 0 \end{aligned} \quad \text{Eq. 3}$$

Now expand these velocities using a Taylor expansion, retaining up to second order terms. Write $u(x, y) = u$:

$$u(x-h_x, y) = u - h_x \frac{\partial u}{\partial x} + \frac{h_x^2}{2} \frac{\partial^2 u}{\partial x^2} - O(h_x^3) \quad \text{Eq. 4}$$

and similarly for the other two components of the velocity.

Substituting this back into the original flux balance equation gives, after simplification, the following stabilized form of the 2D mass balance equation:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} - \frac{h_x}{2} \left(\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 v}{\partial x \partial y} \right) - \frac{h_y}{2} \left(\frac{\partial^2 u}{\partial x \partial y} - \frac{\partial^2 v}{\partial y^2} \right) = 0 \quad \text{Eq. 5}$$

Note that as the domain size tends to zero, i.e. $h_x, h_y \rightarrow 0$, the standard form of the incompressible mass conservation equation in 2D is recovered. The underlined terms in the equation provide the necessary stabilization to allow a standard Galerkin finite element discretization to be applied. They come from admitting that the standard form of the equations is an unreachable limit in the finite element discretization, i.e. by admitting that the element size cannot tend to zero, which is the basis for the finite element method. It also allows equal order interpolations of velocity and pressure to be used.

Stabilized Navier Stokes Equations

The Finite increment Calculus methodology presented above is used to formulate stabilized forms of the momentum balance and mass balance equations and the Neumann boundary conditions. The velocity and pressure fields of an incompressible fluid moving in a domain $\Omega \subset R^d (d=2,3)$ can be described by the incompressible Navier Stokes equations:

$$\left. \begin{aligned} \rho \frac{\partial u_i}{\partial t} + \rho \frac{\partial u_i u_j}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial s_{ij}}{\partial x_j} &= \rho f_i \\ \frac{\partial u_i}{\partial x_i} &= 0 \end{aligned} \right\} i, j = 1, N \quad \text{Eq. 6}$$

Where $1 \leq i, j \leq d$, ρ is the fluid density field, u_i is the i th component of the velocity field \mathbf{u} in the global reference system x_i , p is the pressure field and s_{ij} is the viscous stress tensor defined by:

$$s_{ij} = 2\nu \left(\epsilon_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad \epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad \text{Eq. 7}$$

The stabilized FIC form of the governing differential Eq. 6 can be written as

$$r_{m_i} - \frac{1}{2} h_{ij}^2 \frac{\partial r_{m_i}}{\partial x_j} = 0 \quad \text{in } \Omega \quad \text{Eq. 8}$$

$$r_d - \frac{1}{2} h_j^d \frac{\partial r_d}{\partial x_j} = 0 \quad \text{in } \Omega \quad \text{Eq. 9}$$

Summation convention for repeated indices in products and derivatives is used unless otherwise specified. In above equations, terms r_{m_i} and r_d denote the residual of Eq. 6, and h_{ij}^m, h_j^d are the *characteristic length* distances, representing the dimensions of the finite domain where balance of mass and momentum is enforced. Details on obtaining the FIC stabilized equations and recommendation for the calculation of the stabilization terms can be found in Oñate 1998.

Let n be the unit outward normal to the boundary $\partial\Omega$, split into two sets of disjoint components Γ_t, Γ_u where the Neumann and Dirichlet boundary conditions for the velocity are prescribed respectively. The boundary conditions for the stabilized problem to be considered are (Oñate 2004):

$$\left. \begin{aligned} n_j \sigma_{ij} - t_i + \frac{1}{2} h_{ij}^m n_j r_{m_i} &= 0 && \text{on } \Gamma_t \\ u_j &= u_j^p && \text{on } \Gamma_u \end{aligned} \right\} \quad \text{Eq. 10}$$

Where t_i, u_j^p are the prescribed surface tractions and σ_{ij} is the total stress tensor, defined as

$$\sigma_{ij} = s_{ij} - p\delta_{ij} \quad \text{Eq. 11}$$

Eq. 8 - Eq. 10 constitute the starting point for deriving stabilized FEM for solving the incompressible Navier-Stokes equations. An interesting feature of the FIC formulation is that it allows using equal order interpolation for the velocity and pressure variables (García Espinosa 2003, Oñate 2004).

Stabilized integral forms

From the momentum equations, the following relations can be obtained (Oñate 2004)

$$\left. \begin{aligned} \frac{\partial r_d}{\partial x_i} &= \frac{h_{ii}^m}{2a_i} \frac{\partial r_{m_i}}{\partial x_j} \\ a_i &= \frac{2\mu}{3} + \frac{u_i h_i^d}{2} \end{aligned} \right\} \quad \text{no sum in } i \quad \text{Eq. 12}$$

Substituting the equation above into Eq. 9 and retaining only the terms involving the derivatives of r_{m_i} with respect to x_i , leads to the following alternative expression for the stabilized mass balance equation

$$r_d - \sum_{i=1}^{n_d} \tau_i \frac{\partial r_{m_i}}{\partial x_i} = 0, \quad \tau_i = \left(\frac{8\mu}{3h_{jj}^m h_j^d} + \frac{2\rho u_i}{h_{ii}^m} \right)^{-1} \quad \text{no sum in } i \quad \text{Eq. 13}$$

The τ_i 's in Eq. 13 when multiplied by the density are equivalent to the *intrinsic time parameters*, seen extensively in the stabilization literature. The interest of Eq. 13 is that it introduces the first space derivatives of the momentum equations into the mass balance equation. These terms have intrinsic good stability properties as explained next.

The weighted residual form of the momentum and mass balance equations is written as

$$\left. \begin{aligned} \int_{\Omega} v_i \left[r_{m_i} - \frac{1}{2} h_j \frac{\partial r_{m_i}}{\partial x_j} \right] d\Omega + \int_{\Gamma_i} v_i \left[n_j \sigma_{ij} - t_i + \frac{1}{2} h_j n_j r_{m_i} \right] d\Gamma = 0 \\ \int_{\Omega} q \left[r_d - \sum_{i=1}^N \tau_i \frac{\partial r_{m_i}}{\partial x_i} \right] d\Omega = 0 \end{aligned} \right\} \text{Eq. 14}$$

where v_i, q are generic weighting functions. Integrating by parts the residual terms in the above equations leads to the following weighted residual form of the momentum and mass balance equations:

$$\left. \begin{aligned} \int_{\Omega} v_i r_{m_i} d\Omega + \int_{\Gamma_i} v_i (n_j \sigma_{ij} - t_i) d\Gamma + \int_{\Omega} \frac{1}{2} h_j \frac{\partial v_i}{\partial x_j} r_{m_i} d\Omega = 0 \\ \int_{\Omega} q r_d d\Omega + \int_{\Omega} \left[\sum_{i=1}^N \tau_i \frac{\partial q}{\partial x_i} r_{m_i} \right] d\Omega - \int_{\Gamma} \left[\sum_{i=1}^N q r_i n_i r_{m_i} \right] d\Gamma = 0 \end{aligned} \right\} \text{Eq. 15}$$

We will neglect here onwards the third integral in Eq. 15 by assuming that r_{m_i} is negligible on the boundaries. The deviatoric stresses and the pressure terms in the second integral of Eq. 15 are integrated by parts in the usual manner. The resulting momentum and mass balance equations are:

$$\begin{aligned} \int_{\Omega} v_i \rho \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) + \frac{\partial v_i}{\partial x_j} \left(\mu \frac{\partial u_i}{\partial x_j} - \delta_{ij} p \right) d\Omega - \int_{\Omega} v_i \rho f_i d\Omega \\ - \int_{\Gamma} v_i t_i d\Gamma + \int_{\Omega} \frac{h_j}{2} \frac{\partial v_i}{\partial x_j} r_{m_i} d\Omega = 0 \end{aligned} \text{Eq. 16}$$

and

$$\int_{\Omega} q \frac{\partial u_i}{\partial x_i} d\Omega + \int_{\Omega} \left[\sum_{i=1}^N \tau_i \frac{\partial q}{\partial x_i} r_{m_i} \right] d\Omega = 0 \text{Eq. 17}$$

In the derivation of the viscous term in Eq. 16 we have used the following identity holding for incompressible fluids (prior to the integration by parts)

$$\frac{\partial s_{ij}}{\partial x_j} = 2\mu \frac{\partial \epsilon_{ij}}{\partial x_j} = \mu \partial^2 u_i / \partial x_j \partial x_j \text{Eq. 18}$$

Convective and pressure gradient projections

The computation of the residual terms are simplified if we introduce the convective and pressure gradient projections c_i and π_i , respectively defined as

$$\begin{aligned}
c_i &= r_{m_i} - u_j \frac{\partial u_i}{\partial x_j} \\
\pi_i &= r_{m_i} - \frac{\partial p}{\partial x_i}
\end{aligned}
\tag{Eq. 19}$$

We can express the residual terms in Eq. 16 and Eq. 17 in terms of c_i and π_i , respectively which then become additional variables. The system of integral equations is now augmented in the necessary number by imposing that the residuals vanish (in average sense). This gives the final system of governing equations as:

$$\begin{aligned}
&\int_{\Omega} v_i \rho \left(\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) + \frac{\partial v_i}{\partial x_j} \left(\mu \frac{\partial u_i}{\partial x_j} - \delta_{ij} p \right) d\Omega - \int_{\Omega} v_i \rho f_i d\Omega - \\
&\quad - \int_{\Gamma_t} v_i t_i d\Gamma + \int_{\Omega} \frac{h_k}{2} \rho \frac{\partial v_i}{\partial x_k} \left(u_j \frac{\partial u_i}{\partial x_j} + c_i \right) d\Omega = 0 \\
&\int_{\Omega} q \frac{\partial u_i}{\partial x_i} d\Omega + \int_{\Omega} \left[\sum_{i=1}^N \tau_i \frac{\partial q}{\partial x_i} \left(\frac{\partial p}{\partial x_i} + \pi_i \right) \right] d\Omega = 0 \\
&\int_{\Omega} b_i \left(u_j \frac{\partial u_i}{\partial x_j} + c_i \right) d\Omega = 0 \quad \text{no sum in } i \\
&\int_{\Omega} w_i \left(\frac{\partial p_i}{\partial x_j} + \pi_i \right) d\Omega = 0 \quad \text{no sum in } i
\end{aligned}
\tag{Eq. 20}$$

Being $i, j, k = 1, N$ and b_i, w_i the appropriate weighting functions.

Monolithic time integration scheme

In this section an implicit monolithic time integration scheme, based on a predictor corrector scheme for the integration of Eq. 20 is presented.

Let us first discretize in time the stabilized momentum Eq. 8, using the trapezoidal rule (or θ method) as (see Zienkiewicz 1995):

$$\begin{aligned}
\rho \left(\frac{u_i^{n+1} - u_i^n}{\delta t} + \frac{\partial}{\partial x_i} (u_i u_j)^{n+\theta} \right) + \frac{\partial p^{n+1}}{\partial x_i} - \frac{\partial s_{ij}^{n+\theta}}{\partial x_j} - \rho f_i^{n+\theta} \\
- \frac{1}{2} \rho h_{mj} \frac{\partial}{\partial x_j} \left(u_j^{n+\theta} \frac{\partial u_i^{n+\theta}}{\partial x_j} + c_i^{n+\theta} \right) = 0
\end{aligned}
\tag{Eq. 21}$$

where superscripts n and θ refer to the time step and to the trapezoidal rule discretization parameter, respectively. For $\theta = 1$ the standard backward Euler scheme is obtained, which has a temporal error of $0(t)$. The value $\theta = 0.5$ gives a standard Crank Nicholson scheme, which is second order accurate in time $0(t^2)$.

An implicit fractional step method can be simply derived by splitting Eq. 21 as follows:

$$\begin{aligned}
\rho \left(\frac{u_i^{*,n+1} - u_i^n}{\delta t} + \frac{\partial}{\partial x_i} (u_i u_j)^{n+\theta} \right) + \frac{\partial p^{n+1}}{\partial x_i} - \frac{\partial s_{ij}^{n+\theta}}{\partial x_j} - \rho f_i^{n+\theta} \\
- \frac{1}{2} \rho h_{mj} \frac{\partial}{\partial x_j} \left(u_j^{n+\theta} \frac{\partial u_i^{n+\theta}}{\partial x_j} + c_i^{n+\theta} \right) = 0 \\
u_i^{n+1} = u_i^{*,n+1} - \frac{\delta t}{\rho} \left(\frac{\partial p^{n+1}}{\partial x_i} - \frac{\partial p^n}{\partial x_i} \right)
\end{aligned} \tag{Eq. 22}$$

On the other hand, substituting the last equation above into Eq. 9 and after some algebra leads to the following alternative mass balance equation

$$\rho \frac{\partial u_i^*}{\partial x_i} - \delta t \frac{\partial^2}{\partial x_i \partial x_j} (p^{n+1} - p^n) + \sum_{i=1}^N \tau_i \frac{\partial}{\partial x_i} \left(\frac{\partial p^{n+1}}{\partial x_i} + \pi_i^{n+1} \right) = 0 \tag{Eq. 23}$$

The weighted residual form of the above equations can be written as follows:

$$\begin{aligned}
\int_{\Omega} v_i \rho \left(\frac{u_i^{*,n+1} - u_i^n}{\Delta t} + u_j^{*,n+\theta} \frac{\partial}{\partial x_j} u_i^{*,n+\theta} \right) d\Omega \\
+ \int_{\Omega} \rho \frac{\partial v_i}{\partial x_j} \left(\frac{\partial u_i^{*,n+\theta}}{\partial x_j} - \delta_{ij} p^n \right) d\Omega - \int_{\Omega} v_i \rho f_i^{n+\theta} d\Omega \\
- \int_{\Gamma_t} v_i t_i^{n+\theta} d\Gamma \\
+ \int_{\Omega} \frac{h_k}{2} \rho \frac{\partial v_i}{\partial x_k} \left(u_j^{*,n+\theta} \frac{\partial u_i^{*,n+\theta}}{\partial x_j} + c_i^n \right) d\Omega = 0 \\
\int_{\Omega} q \left(\rho \frac{\partial u_i^*}{\partial x_i} - \Delta t \frac{\partial^2}{\partial x_i \partial x_j} (p^{n+1} - p^n) \right) d\Omega \\
+ \int_{\Omega} \tau_i \frac{\partial q}{\partial x_i} \left(\frac{\partial p^{n+1}}{\partial x_i} + \pi_i^{n+1} \right) d\Omega = 0 \\
\int_{\Omega} v_i \left(u_i^{n+1} - u_i^* + \frac{\Delta t}{\rho} \left(\frac{\partial p^{n+1}}{\partial x_i} - \frac{\partial p^n}{\partial x_i} \right) \right) d\Omega = 0 \quad \text{no sum in } i
\end{aligned} \tag{Eq. 24}$$

At this point, it is important to introduce the associated matrix structure corresponding to the variational discrete FEM form of Eq. 24:

$$\begin{aligned}
M \frac{1}{\delta t} \bar{U}^{n+1} - M \frac{1}{\delta t} U^n + K(\bar{U}^{n+\theta}) \bar{U}^{n+\theta} + S_1 \bar{U}^{n+\theta} + S_2 C - G P^n \\
+ M_{\Gamma_t} T = F \\
(\delta t L + L^{\tau_2}) P^{n+1} + D^{\tau_2} \Pi + G^T \bar{U}^{n+1} = \delta t L P^n \\
M \frac{1}{\delta t} U^{n+1} - M \frac{1}{\delta t} \bar{U}^{n+1} - G P^{n+1} - G P^n = 0 \\
N U^{n+1} + M C = 0 \\
G P^{n+1} + M \Pi = 0
\end{aligned} \tag{Eq. 25}$$

Where U , P are the vectors of the nodal velocity and pressure fields, T is the vector of prescribed tractions and Π , C the vectors of convective and pressure gradient

projections. Terms denoted by over-bar identify the intermediate velocity obtained from the fractional momentum equation.

The system of equations above includes an error due to the splitting of the momentum equation. This error can be eliminated by considering the analogous system of equations

$$\begin{aligned}
 M \frac{1}{\delta t} U_{i+1}^{n+1} - M \frac{1}{\delta t} U^n + K(U_i^{n+\theta}) U_{i+1}^{n+\theta} + S_1 U_{i+1}^{n+\theta} + S_2 C_i \\
 - G P_i^{n+1} + M_{\Gamma_t} T = F \\
 \delta t L(P_{i+1}^{n+1} - P_i^{n+1}) + L^{\tau_2} P_{i+1}^{n+1} + D^{\tau_2} \Pi + G^T U_{i+1}^{n+1} = 0 \\
 N U_{i+1}^{n+1} + M C_{i+1} = 0 \\
 G P_{i+1}^{n+1} + M \Pi_{i+1} = 0
 \end{aligned} \quad \text{Eq. 26}$$

Where i is the iteration counter of the monolithic scheme. Basically, in this final formulation, the convergence of the resulting monolithic uncoupled scheme is enforced by the first term of the second equation of Eq. 27 (see Soto 2001).

Compressible flows

In the previous sections the basics of the incompressible flow solver implemented in Tdyn has been introduced. The extension of that algorithm to compressible flows is straightforward.

Eq. 23 can be easily modified to take into account the compressibility of the flow. The resulting equation is (see M. Vázquez, 1999)

$$\begin{aligned}
 \rho \frac{\partial u_i^*}{\partial x_i} + \frac{\alpha}{\delta t} (p^{n+1} - p^n) - \delta t \frac{\partial^2}{\partial x_i \partial x_i} (p^{n+1} - p^n) \\
 + \sum_{i=1}^N \tau_i \frac{\partial}{\partial x_i} \left(\frac{\partial p^{n+1}}{\partial x_i} + \pi_i^{n+1} \right) = f
 \end{aligned} \quad \text{Eq. 27}$$

In the above equation the terms α and f depend on the compressibility law used.

Turbulence Solvers

At high Reynolds numbers the flow will certainly be turbulent, and the resulting fluctuations in velocities need to be taken into account in the calculations. A process known as Reynolds averaging [3] is applied to the governing equations whereby the velocities, u_i are split into mean and a fluctuating component, where the fluctuating component, u_i' , is defined by:

$$u_i = \bar{u}_i + u_i' \quad \text{Eq. 28}$$

In the expression above:

$$\bar{u}_i(x, t) = \frac{1}{T} \int_{-T/2}^{T/2} u_i(x, t + \tau) d\tau \quad \text{Eq. 29}$$

This leads to extra terms in the governing equations that can be written as a function of ‘Reynolds stresses’ tensor, defined in Cartesian coordinates as:

$$\tau_{ij}^R = -\overline{\rho u_i' u_j'} \quad \text{Eq. 30}$$

Since the process of Reynolds averaging has introduced extra terms into the governing equations, we need extra information to solve the system of equations. To relate these terms to the other flow variables, a model of the turbulence is required. A large number of turbulence models exist of varying complexity, from the simple algebraic models to those based on two partial differential equations. The more complex models have increased accuracy at the expense of longer computational time. It is therefore important to use the simplest model that gives satisfactory results.

Several turbulence models as Smagorinsky, k, k- ϵ , k- ω , k-kt and Spalart Allmaras have been implemented in Tdyn. The final form of the so-called Reynolds Averaged Navier Stokes Equations (RANSE) using these models is:

$$r_{mi} - \frac{h_{mj}}{2} \frac{\partial r_{mi}}{\partial x_j} = 0 \quad \text{on } \Omega, i, j = 1, 2, 3. \text{ No sum in } i \quad \text{Eq. 31}$$

$$r_d - \frac{h_{dj}}{2} \frac{\partial r_d}{\partial x_j} = 0 \quad \text{on } \Omega, j = 1, 2, 3. \quad \text{Eq. 32}$$

Where:

$$r_{mi} = \rho \left[\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} \right] + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j} \left[(\mu + \mu_T) \frac{\partial u_i}{\partial x_j} \right] - \rho f_i \quad \text{Eq. 33}$$

$$r_d = \rho \frac{\partial u_i}{\partial x_i}$$

being μ_T the so-called eddy viscosity.

The theory behind the above mentioned models may be found in the references, but a basic outline of the three most representative models is given below.

Zero equation (algebraic) model: The Smagorinsky model

Zero equation models are the most basic turbulence models. They assume that the turbulence is generated and dissipated in the same place, and so neglect the diffusion and convection of the turbulence. They are based on the idea of turbulent viscosity, and prescribe a turbulence viscosity, ν_t , either by means of empirical equations or experiment.

The Smagorinsky turbulence model is based on the idea of large eddy simulations (LES) in which the coherent large-scale structures are modelled directly within the computational mesh, whilst the small scales are modelled with the concept of eddy

viscosity [4]. The Reynolds stress is written in terms of the turbulence kinetic energy, k , and eddy viscosity as:

$$\overline{u'_i u'_j} = \frac{2}{3} k \delta_{ij} - 2\nu_t \epsilon_{ij} \quad \text{Eq. 34}$$

where

$$\epsilon = \frac{1}{2} \left(\nabla \cdot \underline{u'} + (\nabla \cdot \underline{u'})^T \right) \quad \text{Eq. 35}$$

Smagorinsky proposed that the eddy viscosity depends on the mesh density and velocity gradients, and suggested the following expression:

$$\nu_t = \frac{\mu_t}{\rho} = Ch e^2 \sqrt{\epsilon_{ij} \epsilon_{ij}} \quad \text{Eq. 36}$$

where ν_t is the kinematic eddy viscosity, C is a constant of the order of 0.001, and h^e is the element size.

One-equation models: the kinetic energy model (k-model)

One-equation models attempt to model turbulent transport, by developing a differential equation for the transport of one of turbulent quantities. In the kinetic energy model, the velocity scale of the turbulence is taken as the square root of the turbulence kinetic energy, k . Kolmogorov and Prandtl independently suggested the following relationship between the eddy viscosity, this velocity scale, \sqrt{k} , and length scale, L :

$$\nu_t = C_D \sqrt{kL} \quad \text{Eq. 37}$$

where

$$k = \frac{1}{2} \overline{u'^2} = \frac{1}{2} \overline{u'^2 + v'^2 + w'^2} \quad \text{Eq. 38}$$

and C_D is an empirical constant, usually taken as 1.0.

Manipulation of the Reynolds and momentum equations leads to a differential equation for k [1]. The only gap in this analysis is that left by the length scale, L . This has to be specified either from experiment, or empirical equation. In the analysis used here, it has been specified as 1% of the smallest mesh size. Clearly this is the largest downfall of the analysis, and many consider it insufficiently accurate [3]. However, it is more accurate than the zero-equation models and less computationally expensive than the more accurate 2-equation models, and so it can be a good compromise between computational expense and accuracy, depending on the problem.

Two-equation models: the k - ϵ model

In order to increase the accuracy of our turbulence modelling, it is therefore necessary to develop a second differential transport equation to provide a complete system of closed equations without the need for empirical relationships. The k - ϵ turbulence model develops two such differential transport equations: one for the turbulence kinetic energy, k , and a second for the turbulent dissipation, ϵ . As for the k -model, the k - ϵ model relies on the Prandtl-Kolmogorov expression for the eddy viscosity above. From dimensional arguments, the turbulent dissipation, can be written in terms of the turbulence kinetic energy and the turbulence length scale, and the eddy viscosity written in terms of k and ϵ

$$\epsilon = C_\epsilon \frac{k^{3/2}}{L} \tag{Eq. 39}$$

$$\nu_t = C_\nu \frac{k^2}{\epsilon}$$

Where C_ϵ and C_ν are empirical constants, and are both taken to have a value of 0.09. In a similar way to that of the k -model, differential transport equations can be formulated for k and ϵ [3], thus closing the system of equations, without having to empirically define any of the turbulent quantities.

Since this method directly models the transport of all of the turbulent quantities, it is the most accurate. However, it involves solving 2 differential equations and is so the most computationally expensive. It is therefore necessary to evaluate the importance of turbulence in the problem before a model is selected. In many cases the easiest way to do this may be by trying several methods and using the simplest which gives the most accurate results. If turbulence is a relatively small feature of the flow then it is not necessary to waste computational expense by modelling it with a 2-equation model. At the same time, a zero-equation model would inadequately model a problem with large areas of turbulence, and in many cases this will lead to instabilities in the simulation.

Implicit LES turbulence model

In recent years a significant progress has been carried out in the development of new turbulence models based on the fact that not the entire range of scales of the flow is interesting for the majority of engineering applications. In this type of applications information contained in "the large scales" of the flow is enough to analyze magnitudes of interest as velocity, temperature ... Therefore, the idea that the global flow behaviour can be correctly approximated without the necessity to approximate the smaller scales correctly, is seen by many authors as a possible great advance in the modelling of turbulence. This fact has originated the design of turbulence models that describe the interaction of small scales with large scales. These models are commonly known as Large Eddy Simulation models (LES).

Numerous applications have shown that an extension of the FIC method allows to model low and high Reynolds number flows. In the standard large eddy simulation, a

filtering process is applied to the Navier Stokes Eq. 6. After the filtering, a new set of equations are obtained, variables of this new set of equations are the filtered velocities or also called large scale velocities. As a consequence of the filtering process a new term called the subgrid scale tensor appears into the momentum equation. The subgrid scale tensor is defined in function of the large scale velocities and the subscale velocities too. Then it is necessary to model this term in function only of the large scale velocities. Several models exist for the subgrid scale tensor, but basically all of them propose an explicit description of the subgrid scale, an analytic expression in term of large scale velocities. In conclusion, all of these models define the subgrid scale tensor as a new nonlinear viscous term.

It is our understanding that the stabilized FIC formulation, with an adequate evaluation of the characteristic lengths introduces the same effects of a LES model without giving an explicit expression of the subgrid scale tensor. All the effects related to turbulence modelling are included into the non-linear stabilization parameters.

Computation of the characteristic lengths

The computation of the stabilization parameters is a crucial issue as they affect both the stability and accuracy of the numerical solution. The different procedures to compute the stabilization parameters are typically based on the study of simplified forms of the stabilized equations. Contributions to this topic are reported in (Oñate 1998 and García Espinosa 2005). Despite the relevance of the problem there still lacks a general method to compute the stabilization parameters for all the range of flow situations.

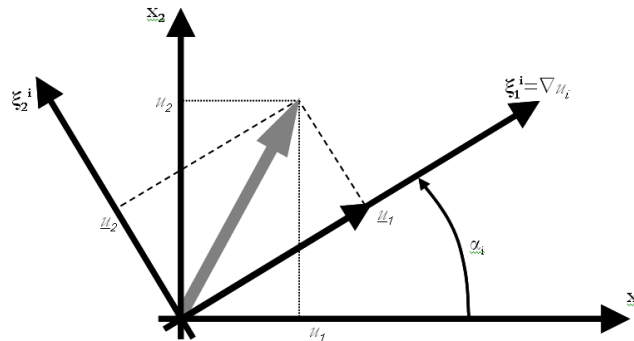


Figure 2 Decomposition of the velocity for characteristic length evaluation

The application of the FIC/FEM formulation to convection-diffusion problems with sharp arbitrary gradients has shown that the stabilizing FIC terms can accurately capture the high gradient zones in the vicinity of the domain edges (boundary layers) as well as the sharp gradients appearing randomly in the interior of the domain (Oñate 1998). The FIC/FEM thus reproduces the best features of both, the so called transverse (cross-wind) dissipation or shock capturing methods.

The approach proposed in this work is based on a standard decomposition of the characteristic lengths for every component of the momentum equation as

$$h_i = \alpha \frac{\mathbf{u}}{|\mathbf{u}|} + \beta \frac{\nabla u_i}{|\nabla u_i|} \quad \text{Eq. 40}$$

Being α the projection of the characteristic length component in the direction of the velocity and β the projection in the direction of the gradient of every velocity component. The decomposition defined by Eq. 40 can be demonstrated to be unique if the characteristic lengths are understood as tensors written in a system of coordinates aligned with the principal curvature directions of the solution. Obviously, Eq. 40 is a simplification of that approach. However, it can be easily demonstrated that the variational discrete FEM form of the resulting momentum equations are equivalent using one or the other approach.

Then, the characteristic lengths are calculated as follows

$$\alpha = \left(\coth \gamma^u - \frac{1}{\gamma^u} \right) u_j l_j, \quad \gamma^u = \frac{u_j l_j}{2\mu} \quad \text{Eq. 41}$$

Where l_j are the maximum projections of the element on every coordinate axis (see Figure 3).

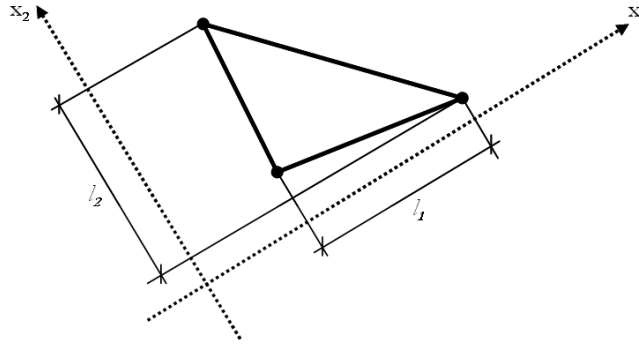


Figure 3 Calculation of l_i in a triangular element

and

$$\beta = \left(\coth \gamma_{ij} - \frac{1}{\gamma_{ij}} \right) l^{\nabla u_i}, \quad \gamma_{ij} = u_i \frac{\nabla u_i}{|\nabla u_i|} \frac{l^{\nabla u_i}}{2\mu} \quad \text{Eq. 42}$$

$$l^{\nabla u_i} = \frac{\nabla u_i}{|\nabla u_i|} l_i, j = 1,2$$

As for the length parameters h_i^d in the mass conservation equation, the simplest assumption $h_i^d = h^d$ has been taken.

The overall stabilization terms introduced by the FIC formulation above presented have the intrinsic capacity to ensure physically sound numerical solutions for a wide spectrum of Reynolds numbers without the need of introducing additional turbulence modelling terms.

4. Overlapping domain decomposition level set (FSURF Module)

This section introduces one of the algorithms implemented in Tdyn for the analysis of free surface flows. The main innovation of this method is the application of domain decomposition concept in the statement of the problem, in order to increase accuracy in the capture of free surface as well as in the resolution of governing equations in the interface between the two fluids. Free surface capturing is based on the solution of a level set equation, while Navier Stokes equations are solved using the iterative monolithic predictor-corrector algorithm presented above, where the correction step is based on the imposition of the divergence free condition in the velocity field by means of the solution of a scalar equation for the pressure.

Statement of the problem

The velocity and pressure fields of two incompressible and immiscible fluids moving in the domain $\Omega \subset R^d (d=2,3)$ can be described by the incompressible Navier Stokes equations for multiphase flows, also known as non-homogeneous incompressible Navier Stokes equations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_j) &= 0 \\ \frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_i u_j) + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} &= \rho f_i \\ \frac{\partial u_i}{\partial x_j} &= 0 \end{aligned} \tag{Eq. 43}$$

Where $1 \leq i, j \leq d$, ρ is the fluid density field, u_i is the i th component of the velocity field u in the global reference system x_i , p is the pressure field and τ is the viscous stress tensor defined by

$$\tau_{ij} = \mu(\partial_i u_j + \partial_j u_i) \tag{Eq. 44}$$

where μ is the dynamic viscosity.

Let $\Omega_1 = \{x \in \Omega \mid x \in \text{Fluid1}\}$ be the part of the domain Ω occupied by the fluid number 1 and let $\Omega_2 = \{x \in \Omega \mid x \in \text{Fluid2}\}$ be the part of the domain Ω occupied by fluid number 2. Therefore Ω_1, Ω_2 are two disjoint subdomains of Ω . Therefore (see Figure 4)

$$\Omega = \text{int}(\overline{\Omega_1 \cap \Omega_2}) \tag{Eq. 45}$$

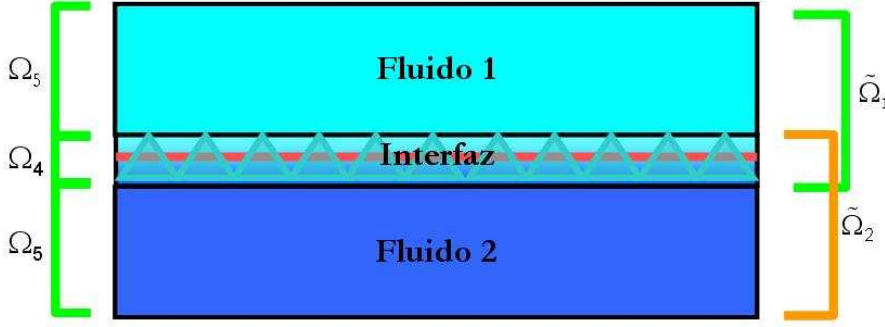


Figure 4 Domain decomposition

where ‘int’ denotes the topological interior and the over bar indicates the topological adherence of a given set, for more details see [18]. The system in Eq. 43 must be completed with the necessary initial and boundary conditions, as shown below.

It is usual in the literature to consider that the first equation in Eq. 43 is equivalent to impose a divergence free velocity field, since the density is taken as a constant. However, in the case of multiphase incompressible flows, density cannot be considered constant in $\Omega \times (0, T)$. Actually, it is possible to define ρ, μ fields as follows:

$$\rho, \mu = \begin{cases} \rho_1, \mu_1 & x \in \Omega_1 \\ \rho_2, \mu_2 & x \in \Omega_2 \end{cases} \quad \text{Eq. 46}$$

Let $\Psi : \Omega \times (0, T) \rightarrow R$ be a function (named Level Set function) defined as follows:

$$\Psi(x, t) = \begin{cases} d(x, t) & x \in \Omega_1 \\ 0 & x \in \Gamma \\ -d(x, t) & x \in \Omega_2 \end{cases} \quad \text{Eq. 47}$$

Where $d(x, t)$ is the distance to the interface between the two fluids, denoted by Γ , of the point x in the time instant t . From the above definition it is trivially obtained that:

$$\Gamma = \{ x \in \Omega \mid \Psi(x, \cdot) = 0 \} \quad \text{Eq. 48}$$

Since the level set 0 identifies the free surface between the two fluids, the following relations can be obtained:

$$n(x, t) = \nabla \Psi|_{(x,t)} ; \kappa(x, t) = \nabla \cdot (n(x, t)) \quad \text{Eq. 49}$$

Where n is the normal vector to the interface Γ , oriented from fluid 1 to fluid 2 and κ is the curvature of the free surface. In order to obtain the above relations it has been assumed that function Ψ fulfils:

$$|\nabla \Psi| = 1 \quad \forall (x, t) \in \Omega \times (0, T) \quad \text{Eq. 50}$$

Therefore, it is possible to redefine density and viscosity as follows:

$$\rho, \mu = \begin{cases} \rho_1, \mu_1 & \Psi > 0 \\ \rho_2, \mu_2 & \Psi < 0 \end{cases} \quad \text{Eq. 51}$$

Let us write the density fields in terms of the level set function Ψ as

$$\rho(x, t) = \rho(\Psi(x, t)) \quad \forall (x, t) \in \Omega \times (0, T) \quad \text{Eq. 52}$$

Then, density derivatives can be written as

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial \Psi} \frac{\partial \Psi}{\partial t}, \quad \frac{\partial \rho}{\partial x_i} = \frac{\partial \rho}{\partial \Psi} \frac{\partial \Psi}{\partial x_i} \quad \text{Eq. 53}$$

Inserting the above relation into the first line of Eq. 43 **Eq. 43** gives

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) &= \frac{\partial \rho}{\partial t} + u_i \frac{\partial \rho}{\partial x_i} = \frac{\partial \rho}{\partial \Psi} \frac{\partial \Psi}{\partial t} + \frac{\partial \rho}{\partial \Psi} u_i \frac{\partial \Psi}{\partial x_i} \\ &= \frac{\partial \rho}{\partial \Psi} \left[\frac{\partial \Psi}{\partial t} + u_i \frac{\partial \Psi}{\partial x_i} \right] = 0 \end{aligned} \quad \text{Eq. 54}$$

which gives as a result that the multiphase Navier Stokes problem is equivalent to solve the following system of equations:

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j} (\rho u_i u_j) + \frac{\partial p}{\partial x_i} - \frac{\partial \tau_{ij}}{\partial x_j} = \rho f_i \quad \text{Eq. 55}$$

$$\frac{\partial u_i}{\partial x_i} = 0$$

coupled with the equation

$$\frac{\partial \Psi}{\partial t} + u_i \frac{\partial \Psi}{\partial x_i} = 0 \quad \text{Eq. 56}$$

Eq. 56 defines the transport of the level set function due to the velocity field obtained by solving Eq. 55

As a conclusion, the free surface capturing problem can be described by the above equations. In this formulation, the interface between the two fluids is defined by the level 0 of Ψ .

Denoting prescribed values by an over-bar, the boundary conditions to be considered for the above presented problem are

$$\begin{aligned} u &= \bar{u} \quad \text{on } \Gamma_u \\ p &= \bar{p}, \quad n_j \tau_{ij} = \bar{t}_i \quad \text{on } \Gamma_p \\ \left. \begin{aligned} u_j n_j &= \bar{u}_n, \\ n_j \tau_{ij} g_i &= \bar{t}_1 \\ n_j \tau_{ij} s_i &= \bar{t}_2 \end{aligned} \right\} \quad \text{on } \Gamma_\tau \end{aligned} \quad \text{Eq. 57}$$

Where the boundary $\partial\Omega$ of the domain Ω has been split in three disjoint sets: Γ_u, Γ_p where the Dirichlet and Neumann boundary conditions are imposed and Γ_τ where the Robin conditions for the velocity are set. In the above, vectors g, s span the space tangent to Γ_τ . In a similar way, the boundary conditions for Eq. 56 are defined

$$\Psi = \bar{\Psi} \quad \text{on } \Gamma_u \quad \text{Eq. 58}$$

Finally, the initial conditions of the problem are

$$u = u_0 \quad \text{on } \Omega, \quad \Psi = \Psi_0 \quad \text{on } \Omega \quad \text{Eq. 59}$$

Where $\Gamma_0 = \{x \in \Omega \mid \Psi_0(x) = 0\}$ defines the initial position of the free surface between the two fluids.

How to calculate a signed distance to the interface

Based on flow properties as density or viscosity, we can calculate an initial guess for Ψ as follows

$$\Psi(x, t) = \begin{cases} 1 & \text{if } \rho(x, t) = \rho_1 \\ 0 & \text{if } x \in \Gamma(t) \\ -1 & \text{if } \rho(x, t) = \rho_2 \end{cases} \quad \text{Eq. 60}$$

Clearly, the level set function obtained from Eq. 60 is not a signed distance to Γ . We need to *reinitiate* Ψ in order to guarantee that Ψ fulfils the definition. We use a technique based on the following property to reinitiate the level set function as a signed distance to Γ . If Ψ is a signed distance function to Γ then,

$$\|\nabla\Psi\| = 1 \quad \forall (x, t) \in \Omega \times (0, T] \quad \text{Eq. 61}$$

where $\|\cdot\|$ denotes the Euclidian norm in \mathbb{R}^d .

We use Eq. 61 to recalculate Ψ . In case that Eq. 61 is not satisfied, then

$$1 - \|\nabla\Psi\| = r \quad \text{Eq. 62}$$

The residual r in Eq. 62 can be understood as the time derivate of Ψ with respect to a pseudo-time τ , i.e.

$$r := \partial_\tau\Psi \quad \text{Eq. 63}$$

Then, the stationary solutions of Eq. 63 (i.e. when $\partial_\tau\Psi = 0$) satisfy Eq. 61. In summary, for a given time $t \in [0, T]$, we calculate Ψ as the stationary solution of the following hyperbolic problem.

$$\begin{aligned} \partial_\tau\Psi + \|\nabla\Psi\| &= 1 \quad \text{on } \Omega \\ \Psi(\tau = 0) &= \Psi(\cdot, t) \\ \Psi(x, \tau) &= 0 \quad \forall x \in \Gamma(t) \end{aligned} \quad \text{Eq. 64}$$

The problem stated in **Eq. 64** can be rewritten in a more convenient manner as

$$\partial_t \Psi + (\mathbf{w} \cdot \nabla) \Psi = \text{sign}(\Psi); \quad \mathbf{w} = \text{sign}(\Psi) \frac{\nabla \Psi}{\|\nabla \Psi\|} \quad \text{Eq. 65}$$

Since the level set values identify the free surface between the two fluids, the following relations can be obtained:

$$\mathbf{n}(\mathbf{x}, t) = \nabla \Psi|_{\mathbf{x}, t}; \quad \kappa(\mathbf{x}, t) = \nabla \cdot (\mathbf{n}(\mathbf{x}, t)) \quad \text{Eq. 66}$$

where \mathbf{n} is the normal vector to the interface Γ , oriented from fluid 2 to fluid 1 and κ is the local curvature of the interface.

From equation Eq. 65 it can be easily shown that the reinitialization process begins from the interface and it propagates to the whole domain. This property is attractive from the computational point of view, since we are only interested in accurate values of Ψ in the region close to the interface. Then we only need to reach the steady state solution of Eq. 65 in a narrow band close to the interface as proposed in [13]. The reinitialization algorithm used in this work has been optimized using the concept of nodal levels described next.

Given a level set distribution Ψ , for every time step t one can assign a level l_i to the node \mathbf{x}_i based on the following rules:

- A nodal point i of coordinates \mathbf{x}_i belongs to the level $l_i = 1$, if $\Psi(\mathbf{x}_i, t) \geq 0$, and it is connected to at least one node j with $\Psi(\mathbf{x}_j, t) < 0$
- A nodal point i of coordinates \mathbf{x}_i belongs to the level $l_i = -1$, if $\Psi(\mathbf{x}_i, t) < 0$, and it is connected to at least one node j with $\Psi(\mathbf{x}_j, t) \geq 0$
- A nodal point i of coordinates \mathbf{x}_i belongs to the level $l_i > 1$, if $\Psi(\mathbf{x}_i, t) > 0$, and it is connected to at least one node of level l_{i-1}
- A nodal point i of coordinates \mathbf{x}_i belongs to the level $l_i < -1$, if $\Psi(\mathbf{x}_i, t) < 0$, and it is connected to at least one node of level l_{i+1}

Once the levels have been assigned to the mesh points, the reinitialization is performed in a narrow band of n levels around the interface. Eq. 65 is solved using a forward Euler scheme until the steady state in the n levels is achieved. k iterations with $k > n$ are performed in such a way that for every iteration $i = 1, \dots, k$. The iterations are carried out only for those nodes j of the mesh with $|l_j| \leq i$.

Due to the evolution of the interface defined by Eq. 56, the level set function does not fulfil Eq. 61 after some time. It is therefore recommended to apply the reinitialization scheme described above at every time step.

Interfacial boundary conditions

At a moving interface the jump condition applies. For immiscible fluids we can write:

$$\begin{aligned} [\mathbf{n} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n})] &= \gamma \kappa & \text{on } \Gamma \\ [\mathbf{s} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n})] &= 0 & \text{on } \Gamma \end{aligned} \quad \text{Eq. 67}$$

where γ is the coefficient of surface tension (a constant of the problem), \mathbf{s} is any unit vector tangent to the interface and $[\cdot]$ defines the jump across the interface.

$$[A] = A_1 - A_2 \quad \text{Eq. 68}$$

Taking into account that $\boldsymbol{\sigma}_i = -p\mathbf{I} + \mu_i(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ $i = 1,2$ and using taking into account that $\mathbf{s} \cdot \mathbf{n} = 0$, the second equation in Eq. 67 yields

$$\mathbf{s} \cdot [(\mu_1 - \mu_2)(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \cdot \mathbf{n}] = 0 \quad \text{Eq. 69}$$

From Eq. 69 we can conclude that $(\mu_1 - \mu_2)(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \cdot \mathbf{n} \in \langle \mathbf{s} \rangle^\perp$, where $\langle \mathbf{v} \rangle^\perp$, denotes the orthogonal subspace to \mathbf{v} . Then it is clear that

$$(\mu_1 - \mu_2)(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \cdot \mathbf{n} = \lambda \mathbf{n}, \quad \lambda \in \text{Spec}(\boldsymbol{\tau}_1 - \boldsymbol{\tau}_2) \quad \text{Eq. 70}$$

Integrating over any closed surface ∂M containing Γ in both sides of Eq. 70 yields

$$\int_{\partial M} (\mu_1 - \mu_2) \mathbf{n} \cdot (\nabla\mathbf{u} + \nabla\mathbf{u}^T) = \int_{\partial M} \lambda \mathbf{n} \quad \text{Eq. 71}$$

Applying the Gauss theorem to the right hand side of Eq. 71 we can deduce that $\lambda = 0$ or equivalently

$$\mathbf{n} \cdot (\nabla\mathbf{u} + \nabla\mathbf{u}^T) = 0 \quad \text{on } \Gamma \quad \text{Eq. 72}$$

Introducing Eq. 72 into the first condition in Eq. 67 the following equivalent condition is obtained:

$$p_1 \mathbf{n} = p_2 \mathbf{n} + \gamma \kappa \mathbf{n} \quad \text{Eq. 73}$$

This form of the jump condition is very useful in the development of the method presented in the following sections.

In the classical level set formulation for multiphase flow problems [15] the jump condition Eq. 67 can be introduced into the momentum equation by adding an additional source term, i.e.

$$\begin{aligned} \partial_t(\rho\mathbf{u}) + \nabla(\rho\mathbf{u} \otimes \mathbf{u}) - \nabla\boldsymbol{\sigma} &= \rho\tilde{\mathbf{f}} \\ \tilde{\mathbf{f}} &= \mathbf{f} + \gamma \kappa \mathbf{n} \delta(\Gamma) \end{aligned} \quad \text{Eq. 74}$$

Where $\delta(\Gamma)$ is a Dirac's delta function on Γ . Since the properties of the fluids change discontinuously across the interface, the direct solution of Eq. 74 has important numerical drawbacks: inaccurate resolution of the pressure at the interface, fluid mass losses, inaccurate interface location, etc. A common approach to overcome these difficulties is to smooth the discontinuous transition of the flow properties and the Dirac's delta functions across the interface using of a smoothed Heaviside function. This function is usually expressed as:

$$H_\epsilon(\Psi) = \begin{cases} 0 & \Psi < -\epsilon \\ \frac{1}{2} \left[1 + \frac{\Psi}{\epsilon} + \frac{1}{\pi} \sin\left(\frac{\pi\Psi}{\epsilon}\right) \right] & |\Psi| \leq \epsilon \\ 1 & \Psi > \epsilon \end{cases} \quad \text{Eq. 75}$$

where 2ϵ is the thickness of the transition area. Then a property η is approximated as

$$\eta(\phi) = \eta_1 + (\eta_1 - \eta_2)H_\epsilon(\phi) \quad \text{Eq. 76}$$

and a Dirac's delta function as

$$\delta(\Gamma) = d_\Psi H_\epsilon \quad \text{Eq. 77}$$

This approach also presents several deficiencies: the accuracy of the results depends on the *ad hoc* thickness of the transition area 2ϵ , the jump condition is not properly captured, the interface propagation velocity is not correctly calculated, etc.

Other schemes, such as the Ghost Cell method or lagrangian methods as the Particle FEM Method, attempt to overcome the problem of properties smoothing at the interface region. However, these methods have several drawbacks, such us limitations of conservative and convergence properties in some cases or an increase of the computational effort. As a conclusion, a method able to deal with a discontinuous transition of properties across the interface, the jump interface condition and computationally affordable for large models is required.

The overlapping domain decomposition method described in the following sections has been developed to overcome these difficulties.

Stabilized FIC-FEM formulation

The stabilized FIC form of the governing differential equations Eq. 55 and Eq. 56 is written as

$$\underline{r_m - \frac{1}{2}(\mathbf{h}_m \cdot \nabla)r_m} = 0, \quad \underline{r_d - \frac{1}{2}(\mathbf{h}_d \cdot \nabla)r_d} = 0, \quad \underline{r_\Psi - \frac{1}{2}(\mathbf{h}_\Psi \cdot \nabla)r_\Psi} = 0 \quad \text{Eq. 78}$$

The boundary conditions for the stabilized FIC problem are written as

$$\begin{aligned} \mathbf{u} &= \bar{\mathbf{u}} \quad \text{on } \Gamma_D \\ \mathbf{n} \cdot \boldsymbol{\sigma} - \underline{\frac{1}{2} \mathbf{n} \cdot \mathbf{h}_m \cdot \mathbf{r}_m} &= \bar{\mathbf{t}} \quad \text{on } \Gamma_N \\ \mathbf{n} \cdot \mathbf{u} = \bar{u}_n, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \underline{\mathbf{g} - \frac{1}{2} \mathbf{n} \cdot \mathbf{h}_m \cdot \mathbf{r}_m \cdot \mathbf{g}} &= \bar{\mathbf{t}}_1 \quad \text{on } \Gamma_M \end{aligned} \quad \text{Eq. 79}$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} \cdot \underline{\mathbf{s} - \frac{1}{2} \mathbf{n} \cdot \mathbf{h}_m \cdot \mathbf{r}_m \cdot \mathbf{s}} = \bar{\mathbf{t}}_2$$

$$\Psi = \bar{\Psi} \quad \text{on } \Gamma_{inlet}$$

The underlined terms in Eq. 78 and Eq. 79 introduce the necessary stabilization for the numerical solution of the Navier Stokes problem [2], [4], [6].

Note that terms \mathbf{r}_m , \mathbf{r}_d and \mathbf{r}_Ψ denote the residuals of Eq. 55 and Eq. 56, i.e.

$$\begin{aligned}\mathbf{r}_m &:= \partial_t(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u}\otimes\mathbf{u}) + \nabla p - \nabla \cdot \boldsymbol{\sigma} - \rho\mathbf{f} \\ \mathbf{r}_d &= \nabla \cdot \mathbf{u} \\ \mathbf{r}_\Psi &:= \partial_t\Psi + (\mathbf{u} \cdot \nabla)\Psi\end{aligned}\tag{Eq. 80}$$

The characteristic lengths vectors \mathbf{h}_m , \mathbf{h}_d and \mathbf{h}_Ψ contain the dimensions of the finite space domain where the balance laws of mechanics are enforced. At the discrete level these dimensions are of the order of the element or grid dimension used for the computation. Details on obtaining the FIC stabilized equations and the recommendation for computing the characteristic length vectors can be found in [4]-[6].

Overlapping domain decomposition method

To overcome the drawbacks related to the discontinuity of fluid properties across the interface and to impose the interfacial boundary condition we split the domain Ω into two overlapping subdomains. Based on this subdivision of the domain, we apply a Dirichlet-Neumann overlapping domain decomposition technique with appropriate boundary conditions in order to satisfying the interfacial condition.

Let $K = \bigcup_{e=1}^{\#K} (N^e, K^e, \Theta^e)$ be a finite element partition of domain the Ω , where N^e are element nodes, K^e is the element spatial domain, Θ^e are element shape functions and K is the total number of elements in the partition.

We assume that K satisfies the following approximation property:

$$\text{dist}\left(\partial\Omega, \partial\left(\bigcup_{e=1}^{\#K} K^e\right)\right) \leq \max_{1 \leq e \leq \#K} \{\text{diam}(K^e)\}\tag{Eq. 81}$$

for a fixed instant $t, t \in (0, T]$. Let us consider a domain decomposition of domain Ω into three disjoint sub domains $\Omega_3(t), \Omega_4(t)$ and $\Omega_5(t)$ in such a way that $\Omega_3(t) = \bigcup_e K_3^e$, $\Omega_5 = \bigcup_e K_5^e$, where K_3^e are the elements of the finite element partition K , such as $\forall \mathbf{x} \in K_3^e | \psi(\mathbf{x}, t) > 0$ and K_5^e are the elements of the finite element partition such as $\forall \mathbf{x} \in K_5^e | \psi(\mathbf{x}, t) < 0$. The geometrical domain decomposition is completed with

$$\Omega_4(t) = \Omega \setminus (\Omega_3(t) \cup \Omega_5(t))\tag{1}$$

From this partition we define the two overlapping domains:

$$\bar{\Omega}_1(t) := \text{int}(\overline{\Omega_3(t) \cup \Omega_4(t)}), \quad \bar{\Omega}_2(t) := \text{int}(\overline{\Omega_4(t) \cup \Omega_5(t)})\tag{Eq. 82}$$

Some comments about the $\tilde{\Omega}_1 - \tilde{\Omega}_2$ partition are given next.

In order to simplify the notation we will omit the time dependency of domains in the rest of the section.

Let $\partial\tilde{\Omega}_i, i = 1, 2$ be the boundary of $\tilde{\Omega}_i$, then:

$$\partial\tilde{\Omega}_i \cap \partial\Omega = \underbrace{(\partial\tilde{\Omega}_i \cap \Gamma_D)}_{\Gamma_{iD}} \cup \underbrace{(\partial\tilde{\Omega}_i \cap \Gamma_N)}_{\Gamma_{iN}} \cup \underbrace{(\partial\tilde{\Omega}_i \cap \Gamma_M)}_{\Gamma_{iM}} \quad \text{Eq. 83}$$

It has to be noted that $\partial\tilde{\Omega}_i$ contains an additional term $\tilde{\Gamma}_i$ that is not included into $\partial\Omega$. This term comes from the presence of an interface inside Ω . Since we are using a capturing technique, the position of the interface will not lay on mesh nodes, as usually happens in a tracking technique. Therefore some elements can be intersected by the interface. Then $\tilde{\Gamma}_i$ is defined as follows:

$$\tilde{\Gamma}_i = \partial\Omega_4 \cap \tilde{\Omega}_{i+(-1)^{i-1}} \quad \text{Eq. 84}$$

Note that $\tilde{\Gamma}_i$ is not coincident with Γ , but the following condition is satisfied:

$$\text{dist}(\Gamma, \tilde{\Gamma}_i) \leq \max \{\text{diam}(K^e) | K^e \subset \Omega_4\} \quad \text{Eq. 85}$$

Summarizing, we have

$$\partial\tilde{\Omega}_i = \Gamma_{iD} \cup \Gamma_{iN} \cup \Gamma_{iM} \cup \tilde{\Gamma}_i$$

Figure 5 **Geometrical domain decomposition** Figure 5 is a sketch of the domain partition described above.

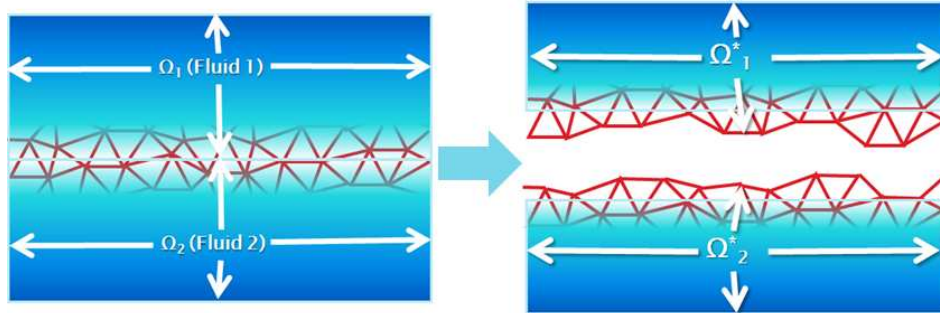


Figure 5 Geometrical domain decomposition

We have two restrictions for the definition of the boundary condition on $\tilde{\Gamma}_i$: fluid velocities must be compatible at the interface and the jump boundary condition Eq. 67 must be satisfied on Γ . The domain decomposition technique chosen is of Dirichlet-Neumann type and it allows us to fulfil both restrictions.

Let us introduce a standard notation. We denote by x_i a variable related to $\tilde{\Omega}_i$. For example \mathbf{u}_1 is the velocity field solution of the problem in Eq. 79, where the domain Ω has been replaced by $\tilde{\Omega}_1$.

We apply the Dirichlet conditions on $\tilde{\Gamma}_1$ and the Neumann conditions on $\tilde{\Gamma}_2$. For the boundary $\tilde{\Gamma}_1$, we make use of the compatibility of velocities at the interface:

$$\mathbf{u}_1 = \mathbf{u}_2 \quad \text{on } \tilde{\Gamma}_1 \quad \text{Eq. 86}$$

For the boundary $\tilde{\Gamma}_2$ we make use of the jump boundary condition. We are looking for a traction vector $\tilde{\mathbf{t}}$ such that:

$$\mathbf{n}_2 \cdot \boldsymbol{\sigma}_2 = \tilde{\mathbf{t}} \quad \text{on } \tilde{\Gamma}_2 \quad \text{Eq. 87}$$

and $\tilde{\mathbf{t}}$ must guarantee that the jump boundary condition Eq. 67 holds, that is to say:

$$p_1 \mathbf{n} = p_2 \mathbf{n} + \gamma \kappa \mathbf{n} \quad \text{on } \Gamma \quad \text{Eq. 88}$$

Let us write the variational problem considering the domain decomposition above described.

Domain 1

$$\begin{aligned} & (\rho_1 \partial_t \mathbf{u}_1, \mathbf{v})_{\tilde{\Omega}_1} + \langle \rho_1 (\mathbf{u}_1 \cdot \nabla) \mathbf{u}_1, \mathbf{v} \rangle_{\tilde{\Omega}_1} + (\boldsymbol{\sigma}_1, \nabla \mathbf{v})_{\tilde{\Omega}_1} \\ & \quad + \frac{1}{2} (\mathbf{r}_{1m}, (\mathbf{h}_{1m} \cdot \nabla) \mathbf{v})_{\tilde{\Omega}_1} + (\tilde{\mathbf{t}}, \mathbf{v})_{\Gamma_{1N}} + (\tilde{\mathbf{t}}_1 \mathbf{g} + \tilde{\mathbf{t}}_2 \mathbf{s}, \mathbf{v})_{\Gamma_{1M}} \\ & = \langle \rho_1 \mathbf{f}, \mathbf{v} \rangle_{\tilde{\Omega}_1} \end{aligned} \quad \text{Eq. 89}$$

$$(q, \nabla \cdot \mathbf{u}_1)_{\tilde{\Omega}_1} + \frac{1}{2} (r_{1d}, (\mathbf{h}_{1d} \cdot \nabla) q)_{\tilde{\Omega}_1} = 0$$

$$\mathbf{u}_1 = \mathbf{u}_2 \quad \text{on } \tilde{\Gamma}_1$$

Domain 2

$$\begin{aligned} & (\rho_2 \partial_t \mathbf{u}_2, \mathbf{v})_{\tilde{\Omega}_2} + \langle \rho_2 (\mathbf{u}_2 \cdot \nabla) \mathbf{u}_2, \mathbf{v} \rangle_{\tilde{\Omega}_2} + (\boldsymbol{\sigma}_2, \nabla \mathbf{v})_{\tilde{\Omega}_2} \\ & \quad + \frac{1}{2} (\mathbf{r}_{2m}, (\mathbf{h}_{2m} \cdot \nabla) \mathbf{v})_{\tilde{\Omega}_2} + (\tilde{\mathbf{t}}, \mathbf{v})_{\Gamma_{2N}} + (\tilde{\mathbf{t}}_1 \mathbf{g} + \tilde{\mathbf{t}}_2 \mathbf{s}, \mathbf{v})_{\Gamma_{2M}} \\ & = \langle \rho_2 \mathbf{f}, \mathbf{v} \rangle_{\tilde{\Omega}_2} \end{aligned} \quad \text{Eq. 90}$$

$$(q, \nabla \cdot \mathbf{u}_2)_{\tilde{\Omega}_2} + \frac{1}{2} (r_{2d}, (\mathbf{h}_{2d} \cdot \nabla) q)_{\tilde{\Omega}_2} = 0$$

An iterative strategy between the two domains is used to reach a converged global solution in the whole domain Ω . It is expected that this global solution will satisfy both restrictions presented above. For this reason we define the following stop criteria:

$$\begin{aligned} \|u_1 - u_2\| &\leq \text{tol}_u \quad \text{on } \Omega \\ |\mathbf{n} \cdot (\boldsymbol{\sigma}_1 - \boldsymbol{\sigma}_2) \cdot \mathbf{n} - \gamma\kappa| &\leq \text{tol}_\sigma \quad \text{on } \Gamma \end{aligned} \quad \text{Eq. 91}$$

The global velocity solution obtained from Eq. 90 is used as the convective velocity for the transport of the level set function, i.e.

$$(\partial_t \Psi, \zeta)_\Omega + \langle (\mathbf{u} \cdot \nabla) \Psi, \zeta \rangle_\Omega + \frac{1}{2} (r_\Psi, (\mathbf{h}_\Psi \cdot \nabla) \zeta)_\Omega = 0 \quad \text{Eq. 92}$$

Unfortunately there is not theoretical result that can ensure the convergence of this method. However our experience in applying this method to a wide range of problems involving moving interfaces has showed that the method is stable and robust.

An expected property of the method is its dependency with the mesh size around the interface. This issue is directly related with how good is Γ approximated by $\tilde{\Gamma}_i$, as it has been point out in property Eq. 85. A fine mesh close to the interface reduces significantly the amount of iterations needed to reach a converged global solution, as well as the accuracy of the velocities and the pressure at the interface.

This methodology can be viewed as a combination of domain decomposition and level set techniques. This justifies the name given to the new method: ODDLS (by Overlapping Domain Decomposition Level Set).

Discretization by the finite element method (FEM)

In this section we present the final discrete system of equations associated to problem Eq. 89 and Eq. 90 using the FEM method. Let us consider a uniform partition of the time interval of analysis $[0, T]$, with time step size δt . We will denote by a superscript the time step at which the algorithmic solution is computed. We assume $\mathbf{u}_1^n, p_1^n, \mathbf{u}_2^n$ and p_2^n are known. If $\theta \in [0, 1]$, the trapezoidal rule applied to Eq. 89 and Eq. 90 consists of finding $\mathbf{u}_1^{n+\theta}, p_1^{n+\theta}, \mathbf{u}_2^{n+\theta}, p_2^{n+\theta}$ as the solution of the problem

$$\begin{aligned} (\rho_i^n \delta_t \mathbf{u}_i^n, \mathbf{v})_{\tilde{\Omega}_i} + \langle \rho_i^n (\mathbf{u}_i^{n+\theta} \cdot \nabla) \mathbf{u}_i^{n+\theta}, \mathbf{v} \rangle_{\tilde{\Omega}_i} + (\boldsymbol{\sigma}_i^{n+\theta}, \nabla \mathbf{v})_{\tilde{\Omega}_i} \\ + \frac{1}{2} (\mathbf{r}_{im}^{n+\theta}, (\mathbf{h}_{im}^{n+\theta} \cdot \nabla) \mathbf{v})_{\tilde{\Omega}_i} + (\tilde{\mathbf{t}}, \mathbf{v})_{\Gamma_{iN}} \\ + (\tilde{\mathbf{t}}_1 \mathbf{g} + \tilde{\mathbf{t}}_2 \mathbf{s}, \mathbf{v})_{\Gamma_{iM}} + (i-1) (\tilde{\mathbf{t}}, \mathbf{v})_{\tilde{\Gamma}_i} = \langle \rho_i^n \mathbf{f}, \mathbf{v} \rangle_{\tilde{\Omega}_i} \end{aligned} \quad \text{Eq. 93}$$

$$(q, \nabla \cdot \mathbf{u}_i^{n+\theta})_{\tilde{\Omega}_i} + \frac{1}{2} (r_{id}^{n+\theta}, (\mathbf{h}_{im}^{n+\theta} \cdot \nabla) q)_{\tilde{\Omega}_i} = 0 \quad i = 1, 2$$

where $x^{n+\theta} := \theta x^{n+1} + (1-\theta)x^n$ and $\delta_t x^n := \frac{1}{\delta t} (x^{n+1} - x^n)$.

This problem is nonlinear due to the convective term and the evolution of the free surface. Prior to the finite element discretization we linearize it using a Picard method, which leads to an Oseen problem for each iteration step. Several strategies can be adopted to deal with the two iterative algorithms involved in the ODDLS method: the

overlapping domain decomposition iterative scheme and the linearization scheme (Picard). In our case, for each iteration step of the linearization scheme the domain decomposition scheme is also updated. This simultaneous iterative strategy requires to complete Eq. 91 with a stop criteria for the Picard iteration. The same iteration index is used for both schemes.

The adopted strategy reduces noteworthy the computational effort versus other nested iterative schemes. We denote by a superscript n the time step at which the algorithmic solution is computed and by a superscript j the iteration step of the domain decomposition scheme (which for a simultaneous iterative strategy will be the same as for the Picard iteration step). The form of the iterative scheme is as follows

$$\begin{aligned}
& (\rho_i^{n,j-1} \delta_t \mathbf{u}_i^{n,j}, \mathbf{v})_{\tilde{\Omega}_i} + \langle \rho_i^n (\mathbf{u}_i^{n+\theta,j-1} \cdot \nabla) \mathbf{u}_i^{n+\theta,j}, \mathbf{v} \rangle_{\tilde{\Omega}_i} + (\boldsymbol{\sigma}_i^{n+\theta,j}, \nabla \mathbf{v})_{\tilde{\Omega}_i} \\
& \quad + \frac{1}{2} (\mathbf{r}_{im}^{n+\theta,j}, (\mathbf{h}_{im}^{n+\theta,j} \cdot \nabla) \mathbf{v})_{\tilde{\Omega}_i} + (\tilde{\mathbf{t}}, \mathbf{v})_{\Gamma_{iN}} \\
& \quad + (\tilde{\mathbf{t}}_1 \mathbf{g} + \tilde{\mathbf{t}}_2 \mathbf{s}, \mathbf{v})_{\Gamma_{iM}} + (i-1) (\tilde{\mathbf{t}}, \mathbf{v})_{\tilde{\Gamma}_i} = \langle \rho_i^{n,j-1} \mathbf{f}, \mathbf{v} \rangle_{\tilde{\Omega}_i} \\
& (q, \nabla \cdot \mathbf{u}_i^{n+\theta,j})_{\tilde{\Omega}_i} + \frac{1}{2} (r_{id}^{n+\theta,j}, (\mathbf{h}_{im}^{n+\theta,j} \cdot \nabla) q)_{\tilde{\Omega}_i} = 0 \quad i = 1, 2
\end{aligned} \tag{Eq. 94}$$

The Galerkin approximation of Eq. 94 is straightforward and stable, thanks to the FIC stabilized formulation [2-7]. Equal polynomial order can be used for the approximation of the velocities and the pressure. Based on the finite element partition introduced in Eq. 81, let $V_h \subset V$ and $Q_h \subset Q$ be two finite dimensional conforming finite element spaces, being h the maximum of the elements partition diameters.

Let $\tilde{V}_{h,i}^0 := \{ \mathbf{v} \in V_h \mid \mathbf{v}|_{\partial \tilde{\Omega}_i} = \mathbf{0} \}$, $Q_{h,i}^0 := \{ q \in Q_h \mid q|_{\partial \tilde{\Omega}_i} = 0 \}$ be the finite element spaces for the test functions. The discretized form of Eq. 94 is

$$\begin{aligned}
& (\rho_{h,i}^{n,j-1} \delta_t \mathbf{u}_{h,i}^{n,j}, \mathbf{v}_h)_{\tilde{\Omega}_i} + \langle \rho_{h,i}^n (\mathbf{u}_{h,i}^{n+\theta,j-1} \cdot \nabla) \mathbf{u}_{h,i}^{n+\theta,j}, \mathbf{v}_h \rangle_{\tilde{\Omega}_i} \\
& \quad + (\boldsymbol{\sigma}_{h,i}^{n+\theta,j}, \nabla \mathbf{v}_h)_{\tilde{\Omega}_i} + \frac{1}{2} (\mathbf{r}_{h,im}^{n+\theta,j}, (\mathbf{h}_{h,im}^{n+\theta,j} \cdot \nabla) \mathbf{v}_h)_{\tilde{\Omega}_i} \\
& \quad + (\tilde{\mathbf{t}}, \mathbf{v}_h)_{\Gamma_{iN}} + (\tilde{\mathbf{t}}_1 \mathbf{g} + \tilde{\mathbf{t}}_2 \mathbf{s}, \mathbf{v}_h)_{\Gamma_{iM}} + (i-1) (\tilde{\mathbf{t}}, \mathbf{v}_h)_{\tilde{\Gamma}_i} \\
& \quad = \langle \rho_{h,i}^{n,j-1} \mathbf{f}, \mathbf{v}_h \rangle_{\tilde{\Omega}_i} \\
& (q_h, \nabla \cdot \mathbf{u}_{h,i}^{n+\theta,j})_{\tilde{\Omega}_i} + \frac{1}{2} (r_{h,id}^{n+\theta,j}, (\mathbf{h}_{h,im}^{n+\theta,j} \cdot \nabla) q_h)_{\tilde{\Omega}_i} = 0 \quad i = 1, 2
\end{aligned} \tag{Eq. 95}$$

$$\forall (\mathbf{v}_h, q_h) \in \mathbf{V}_h \times Q_h$$

Remark

Updating of the level set equation is done once the convergence of the Navier-Stokes equations is obtained. Therefore the complete iteration loop is as follows:

- Step 1. Solve equations (54) in $\tilde{\Omega}_1$.
- Step 2. Solve equations (54) in $\tilde{\Omega}_2$.

- Step 3. Check convergence on pressure and velocity fields. If it is satisfied, go to Step 4, otherwise go to Step 1.
- Step 4. Solve equation (51).
- Step 5. Check convergence on level set equation. If it is satisfied, advance to the next time step, otherwise go to Step 1.

5. Heat Transfer Solver (HEATRANS Module)

Tdyn solves the transient Heat Transfer Equations in a given domain $\Omega := \{\Omega_F \cup \Omega_S\}$ (being Ω_F the fluid domain and Ω_S the solid domain) and time interval $(0, t)$:

$$\begin{aligned} \rho_C C_P \left(\frac{\partial \theta}{\partial t} + (\mathbf{u} \cdot \nabla) \theta \right) - \nabla \cdot (\kappa_F \nabla \theta) &= \rho_F q_F \quad \text{in } \Omega_F \times (0, t) \\ \rho_S C \frac{\partial \theta}{\partial t} - \nabla \cdot (\kappa_S \nabla \theta) &= \rho_S q_S \quad \text{in } \Omega_S \times (0, t) \end{aligned} \quad \text{Eq. 96}$$

where $\theta = \theta(\mathbf{x}, t)$ denotes the temperature field, ρ_S the (constant) solid density, k_S the solid thermal conduction tensor, k_F the fluid thermal conduction constant, C_P the fluid specific heat constant, C the solid specific heat constant and q_S and q_F the solid and fluid, volumetric heat source distributions respectively. The above equations need to be combined with the following boundary conditions:

$$\begin{aligned} \theta &= \theta_c && \text{in } \Gamma_\theta \times (0, t) \\ n \kappa_F \nabla \theta &= f_F && \text{in } \Gamma_F \times (0, T) \\ n \kappa_S \nabla \theta &= f_S && \text{in } \Gamma_S \times (0, T) \\ \theta(x, 0) &= \theta_0(x) && \text{in } \Omega \times \{0\} \end{aligned} \quad \text{Eq. 97}$$

In the above equations, $\Gamma := \partial\Omega$ denotes the boundary of the domain Ω , with \mathbf{n} the normal unit vector, θ_c is the temperature field on Γ_θ (the part of the boundary of Dirichlet type, or prescribed temperature type), f_F the prescribed heat flux on Γ_F (prescribed heat flux fluid boundary), f_S the prescribed heat flux on Γ_S (prescribed heat flux solid boundary) and θ_0 the initial temperature field.

The spatial discretization of the heat transfer equations is done by means of the finite element method, while for the time discretization implicit first and second order schemes have been implemented.

Stabilized Heat Transfer Equations

The Finite Increment Calculus theory presented above is also valid for the heat transfer balance equations Eq. 96. The stabilized form of the governing differential equations in 3 dimensions is written as follows:

$$\begin{aligned} r_F - \frac{h_{Fj}}{2} \frac{\partial r_F}{\partial x_j} &= 0 && \text{in } \Omega_F, j = 1, 2, 3 \\ r_S - \frac{h_{Sj}}{2} \frac{\partial r_S}{\partial x_j} &= 0 && \text{in } \Omega_S, j = 1, 2, 3 \end{aligned} \quad \text{Eq. 98}$$

Where

$$r_F = \rho C_P \left(\frac{\partial \theta}{\partial t} \right) - \kappa_F \Delta \theta - \rho q_F$$

$$r_S = \rho_S C \frac{\partial \theta}{\partial t} - \nabla \cdot (\kappa_S \nabla \theta) - \rho_S q_S$$

Eq. 99

and the new boundary conditions are:

$$\theta = \theta_c \quad \text{in } \Gamma_\theta \times (0, t)$$

$$n \kappa_F \nabla \theta + n \frac{h_{Fj}}{2} r_F = f_F \quad \text{in } \Gamma_F \times (0, T)$$

$$n \kappa_S \nabla \theta + n \frac{h_{Sj}}{2} r_S = f_S \quad \text{in } \Gamma_S \times (0, T)$$

$$\theta(x, 0) = \theta_0(x) \quad \text{in } \Omega \times \{0\}$$

Eq. 100

The stabilized Heat transfer balance equations are discretized in time in the standard way, to give the following equations:

$$\theta^{j+1, n+1} = \theta^n - \frac{\Delta t}{\rho C_P} [\rho C_P (\mathbf{u} \cdot \nabla) \theta - \nabla \cdot (\kappa_F \nabla \theta) - \rho q_F]^{j, n+\theta}$$

$$\theta^{j+1, n+1} = \theta^n - \frac{\Delta t}{\rho C} [\nabla \cdot (\kappa_S \nabla \theta) - \rho q_S]^{j, n+\theta}$$

Eq. 101

6. Species Advection Solver (ADVECT Module)

Tdyn solves the transient Species Advection Equations in a given fluid domain Ω_F for a number of different species, and time interval $(0, t)$:

$$\frac{\partial \varphi}{\partial t} + (\mathbf{u} \cdot \nabla)\varphi + (c \cdot \varphi \cdot g \cdot \nabla)\varphi - \nabla \cdot (\kappa_P \nabla \varphi) = q_F \quad \text{in } \Omega_F \times (0, t) \quad \text{Eq. 102}$$

where $\varphi = \varphi(\mathbf{x}, t)$ denotes the concentration of species field, c the decantation coefficient, k_S the total diffusion coefficient (including turbulent effects) and q_P a volumetric source. The above equations need to be combined with the standard boundary conditions.

As in the previous examples, the spatial discretization of the Species Advection equations has been done by means of the finite element method, while for the time discretization implicit first and second order schemes have been implemented. Problems with dominating convection are stabilized by the Finite Calculus (FIC) method, presented above.

7. References

- [1] R. Codina "A discontinuity capturing crosswind dissipation for the finite element solution of the convection diffusion equation", *Comp. Meth. Appl. Mech. Eng.*, pages 110:325-342, 1993.
- [2] J. García, "A Finite Element Method for the Hydrodynamic Analysis of Naval Structures" (in Spanish), Ph.D. Thesis, Univ. Politècnica de Catalunya, Barcelona, December 1999.
- [3] C. Hirsch "Numerical computation of internal and external flows", John Wiley and Sons Ltd., Chichester, UK, 1991.
- [4] C. Kleinstreuer "Engineering Fluid Dynamics, An Interdisciplinary Systems Approach", Cambridge University Press, Cambridge, UK, 1997.
- [5] E. Oñate "A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. Research report No. 150", CIMNE, Barcelona, January 1999.
- [6] García, J., Oñate, E., Sierra, H., Sacco, C. and Idelsohn, S. "A Stabilized Numerical Method for Analysis of Ship Hydrodynamics". *Proceedings Eccomas Conference on CFD*, 7-11 September 1998, Athens, John Wiley, 1998.
- [7] E. Oñate, M. Manzan, "Stabilization techniques for finite element analysis of convection-diffusion problems", *Publication CIMNE No. 183*, Barcelona, February 2000.
- [8] O.C. Zienkiewicz, T.L. Taylor "The finite element method". McGrawHill Book Company, Volume 1, Maidenhead, UK, 1989.
- [9] E. Oñate, J. García and S. Idelsohn, "On the stabilisation of numerical solutions of advective-diffusive transport and fluid flow problems". *Comp.Meth. Appl. Mech. Engng*, 5, 233-267, 1998.
- [10] J.E. Bardina, P.G. Huang and T.J. Coakley, "Turbulence Modelling Validation, Testing, and Development". NASA technical Memorandum 110446, Moffett Field, California, April 1997.
- [11] D.C. Wilcox "Turbulence Modelling for CFD". DCW industries, Inc. 5354 Palm Drive, La Cañada, California, 1993.
- [12] M. Vázquez, R. Codina and O.C. Zienkiewicz "Numerical modelling of Compressible Laminar and Turbulent Flow: The CBS Algorithm". *Monograph CIMNE N° 50*, Barcelona, April 1999.
- [13] J. García-Espinosa and E. Oñate. "An Unstructured Finite Element Solver for Ship Hydrodynamics". *Jnl. Appl. Mech.* Vol. 70: 18-26. (2003).

- [14] E. Oñate, J. García-Espinosa y S. Idelsohn. "Ship Hydrodynamics". Encyclopedia of Computational Mechanics. Eds. E. Stein, R. De Borst, T.J.R. Hughes. John Wiley & Sons (2004).
- [15] O.C. Zienkiewicz and Codina, R., "A general algorithm for compressible and incompressible flow. Part I, The Split, Characteristic Based Scheme", *Int. Journ. Num. Meth. Fluids*, 20, 869-885 (1995).
- [16] O. Soto, R. Lohner, J Cebral. An Implicit Monolithic Time Accurate Finite Element Scheme for Incompressible Flow Problems. 15th AIAA Computational Fluid Dynamics Conference. Anaheim (USA) 2001.
- [17] J. García-Espinosa, E. Oñate and J. Bloch Helmers. "Advances in the Finite Element Formulation for Naval Hydrodynamics Problems". International Conference on Computational Methods in Marine Engineering (Marine 2005). June 2005. Oslo (Norway).
- [18] J. García-Espinosa, A. Valls and E. Oñate. "ODDLS: A new unstructured mesh finite element method for the analysis of free surface flow problems". *Int. Jnl. Num. Meth. Engng.* 76(9): 1297-1327 (2008).