Industrial Training Report

Daniel Díez

June 2, 2017

Implementation of a Two Fluid Navier Stokes element into the general FEM code KratosMultiphysics

Introduction

The present report will explain to a certain degree of detail the tasks that I carried out during my industrial training in CIMNE for my Master's Degree in Numerical Methods For Engineering. They are the following:

1. Formulation of a Navier Stokes element capable of dealing with two fluid phases with different properties using an enrichment for the pressure.
2. Elaboration of a functional for the proposed formulation ready to be implemented in Kratos Multiphysics.
3. Implementation of the element along with the required files into Kratos Multiphysics.

1 Formulation of a two fluid Navier Stokes element.

The Navier Stokes equations for an incompressible fluid in a domain Ω are:

$$\begin{aligned} \rho\partial_t\mathbf{u} + \rho\mathbf{a}\cdot\nabla\mathbf{u} + \nabla p - \nabla\cdot\mathbb{C}\nabla^s\mathbf{u} &= \rho\mathbf{f} \qquad && \mathit{in}\ \Omega\times(0,T) \\ \rho(\nabla\cdot\mathbf{u}) &= 0 && \mathit{in}\ \Omega\times(0,T) \end{aligned} \tag{1}$$

Where ρ is the density of the fluid, which will be considered constant, $\mathbf{u}$ is the unknown velocity, $\mathbf{a}$ is the convection velocity, p is the pressure, $\mathbb{C}$ is the constitutive matrix and $\mathbf{f}$ represent the body forces.

The problem must be completed with suitable initial and boundary conditions. We will split the boundary of the domain $\partial\Omega$ into the Dirichlet boundary Γ_D and the Neumann boundary Γ_N, hence $\partial\Omega = \Gamma_D \cup \Gamma_N$.

$$\begin{aligned} \mathbf{u}(\mathbf{x},0) &= \mathbf{u_0}(\mathbf{x}) \qquad && \mathit{in}\ \Omega, t=0 \\ \mathbf{u}(\mathbf{x},t) &= \mathbf{u_D}(\mathbf{x},t) && \mathit{on}\ \Gamma_D\times(0,T) \\ \mathbf{n}\cdot\boldsymbol{\sigma}(\mathbf{x},t) &= \mathbf{t}(\mathbf{x},t) && \mathit{on}\ \Gamma_N\times(0,T) \end{aligned} \tag{2}$$

Where u_0 corresponds to the initial velocity field, u_D represents the imposed velocity on the Dirichlet boundary, $\boldsymbol{n}$ is the outer normal vector, t is the imposed traction on the Neumann boundary and $\boldsymbol{\sigma}$ is the Cauchy stress tensor.

We test against the functions $\mathbf{w}$ and q and integrate over the domain. The pressure and the viscous term will be integrated by parts obtaining the weak form of the problem. We will aso add an extra equation testing the continuity equation against the enriched shape function q^*:

$$\begin{aligned} (\mathbf{w}, \rho\partial_t\mathbf{u} + \rho\mathbf{a}\cdot\nabla\mathbf{u}) - (\nabla\cdot\mathbf{w}p) + (\nabla\mathbf{w} : \mathbb{C}\nabla^s\mathbf{u}) = (\mathbf{w},\rho\mathbf{f}) + (\boldsymbol{w},t)_{\Gamma_N} \qquad && \forall w\in\mathcal{V} \\ (q,\rho\nabla\cdot\mathbf{u}) = 0 && \forall q\in\mathcal{Q} \\ (q^*,\rho\nabla\cdot\mathbf{u}) = 0 && \forall q^*\in\mathcal{Q}^* \end{aligned} \tag{3}$$

Where $\mathcal{V}\in(\mathcal{H}^1)^d$, $\mathcal{Q}\in\mathcal{L}^2$ and $\mathcal{Q}^*\in\mathcal{L}^2$ represent the following functional spaces:

$$\mathcal{H}^1(\Omega)^d := \left\{ v : \Omega \to \mathbb{R}^d \mid \int_\Omega |v|^2 < \infty, \int_\Omega |\nabla v|^2 < \infty \right\} \tag{4}$$

$$\mathcal{L}^2(\Omega) := \left\{ v : \Omega \to \mathbb{R} \mid \int_\Omega v^2 < \infty \right\} \tag{5}$$

In order to be able to use linear interpolations for both the velocity and the pressure we need to add stabilization terms to the Galerkin method. We will basically follow the same lines proposed in [**?**]. Let's consider a subscale decomposition as follows:

$$\mathbf{u} = \mathbf{u_h} + \mathbf{u}_s \tag{6}$$

$$p = p_h + p_s \tag{7}$$

The terms $\mathbf{u}_h$ and p_h represent the part of the solution which belongs to the finite element space $\mathbf{u_h} \in \mathcal{V}_h$ while $\mathbf{u}_s$ and p_s belong to the small scale space such that $\mathcal{V} = \mathcal{V}_h \oplus V_s$ and $\mathcal{Q} = \mathcal{Q}_h \oplus \mathcal{Q}_s$.
As we mentioned in the introduction we will use a enriched pressure field as follows:

$$p_h^{tot} = \sum_i^{nnodes} N_i p_i + \sum_j^{nnodes} N_j^* p_j^* \tag{8}$$

Where N corresponds here to the usual finite elements shape functions while N^* are the enrichment functions. We will have analogously a new set of unknowns p^* for our pressure field. We can conclude that $p_h^{tot} \in \mathcal{Q}_h \cup \mathcal{Q}_h^*$

The enrichment shape functions N^* will be constructed in the following way for an element cut by the interface Γ. The element is splitted by the level set function into two parts, one positive and one negative. The nodes that lay on the positive side have the following formulation:

$$\begin{aligned} N_i^* &= N_i \qquad in\ \Omega_e^{i-} \\ N_i^* &= 0 \qquad in\ \Omega_e^{i+} \end{aligned} \tag{9}$$

While the nodes that lay on the negative side will follow this formulation:

$$\begin{aligned} N_i^* &= 0 \qquad in\ \Omega_e^{i-} \\ N_i^* &= N_i \qquad in\ \Omega_e^{i+} \end{aligned} \tag{10}$$

Notice that defining the enriched shape function in such a way has an important property: they are all equal to zero on the nodes. This will be an important advantage as we will see later.

For the stabilization we use a quasi-static small scales model where $\partial_t u_s = 0$ and:

$$\mathbf{u}_s \approx \tau_1 \mathbf{r_m}(\mathbf{u_h}, p_h^{tot}) \tag{11}$$

$$p_s \approx \tau_2 r_c(\mathbf{u_h}) \tag{12}$$

Obtaining the following weak form with the added stabilization terms.

$$\begin{gathered} (\mathbf{w}_h, \rho\partial_t\mathbf{u_h} + \rho\mathbf{a}\cdot\nabla\mathbf{u_h}) - (\nabla\cdot\mathbf{w_h}p_h) - (\nabla\cdot\mathbf{w}_h, p_h^*) + (\nabla\mathbf{w}_h : \mathbb{C}\nabla^s\mathbf{u_h}) \\ -\sum_{\Omega_e}\int_{\Omega_e}\rho(\mathbf{a}\cdot\nabla\mathbf{w}_h)\tau_1\cdot\mathbf{r_m}(\mathbf{u_h}, p_h^{tot}) - \sum_{\Omega_e}\int_{\Omega_e}\rho(\nabla\cdot\mathbf{w}_h)\tau_2 r_c(\mathbf{u_h}) = (\mathbf{w}_h, \rho\mathbf{f}) + (\boldsymbol{w}, t)_{\Gamma_N} \end{gathered} \tag{13}$$

$$(q_h, \rho\nabla\cdot\mathbf{u_h}) = \sum_{\Omega_e}\int_{\Omega_e}\rho\nabla q_h\tau_1\cdot\mathbf{r_m}(\mathbf{u_h}, p_h^{tot}) \tag{14}$$

$$(q_h^*, \rho\nabla\cdot\mathbf{u_h}) = \sum_{\Omega_e}\int_{\Omega_e}\rho\nabla q_h^*\tau_1\cdot\mathbf{r_m}(\mathbf{u_h}, p_h^{tot}) \tag{15}$$

Notice that as we will use linear elements, the viscous term will vanish in $\mathbf{r_m}$ as it involves second spatial derivatives.

The terms τ_1 and τ_2 are the stabilization parameters which will be taken as:

$$\begin{aligned}\tau_1 =& \mathbf{I}\left(\frac{c_d\rho}{\delta t}+\frac{c_1\mu}{h^2}+\frac{c_2\rho|\boldsymbol{a}|}{h}\right)^{-1}\\ \tau_2 =& \mu+\frac{c_2|\boldsymbol{a}|h}{c_1}\end{aligned} \tag{16}$$

Where h is the characteristic length of the element and c_1 and c_2 are two constants that we have to define for each problem. For linear elements it is oftenly used $c_1 = 4$ and $c_2 = 2$. Similarly, c_d is the dynamic tau coefficient which can be also set in different ways. We use $c_d = \mu$, where μ is the dynamic viscosity of the fluid ($\mu = 0.001$ for water).

2 Elaboration of a functional for the proposed formulation ready to be implemented in Kratos Multiphysics.

The functional for the Navier Stokes problem with Subgrid Scales Stabilization can be constructed in the following way:

$$f(\mathbf{u_h}, p_h, p_h^*, \mathbf{w_h}, q_h, q_h^*) = f_g(\mathbf{u_h}, p_h, p_h^*, \mathbf{w_h}, q_h, q_h^*) + f_{stab}(\mathbf{u_h}, p_h, p_h^*, \mathbf{w_h}, q_h, q_h^*) \tag{17}$$

$$\begin{aligned}f_g(\mathbf{u_h}, p_h, p^*, \mathbf{w_h}, q_h, q_h^*) = &(\mathbf{w_h}, f) - \rho(\mathbf{w_h}, \partial_t\mathbf{u_h} + \mathbf{a}\cdot\nabla\mathbf{u_h}) - (\nabla\mathbf{w_h} : \mathbb{C}\nabla^s\mathbf{u_h}) -\\ &+ (\nabla\cdot\mathbf{w_h}p_h) + (\nabla\cdot\mathbf{w_h}p_h^*) - (q_h, \rho(\nabla\cdot\mathbf{u_h})) - (q_h^*, \rho(\nabla\cdot\mathbf{u_h}))\end{aligned} \tag{18}$$

$$\begin{aligned}f_{stab}(\mathbf{u_h}, p_h, p^*, \mathbf{w_h}, q_h, q_h^*) = &\int_{\Omega_e}\rho(a\cdot\nabla\mathbf{w_h})\tau_1\mathbf{r_m} + \int_{\Omega_e}\rho(\nabla\cdot\mathbf{w_h})\tau_2 r_c -\\ &+ \int_{\Omega_e}\rho\nabla q_h\tau_1\mathbf{r_m} + \int_{\Omega_e}\rho\nabla q_h^*\tau_1\mathbf{r_m}\end{aligned} \tag{19}$$

This functional will be used for each element to obtain its sitfness matrix (lhs) and the loads vector (rhs). These terms will be later assembled to build the global system.

The solution of the problem is such that the residuals of the momentum and mass equations are equal to zero. Adding the stabilization terms we can define the following residuals:

$$R_m^i = r_m^i(u_h, p_h^{tot}) + r_m^{i,stab}(u_h, p_h^{tot}) \tag{20}$$

$$R_c = r_c(u_h, p_h^{tot}) + r_c^{stab}(u_h, p_h^{tot}) \tag{21}$$

$$R_c^* = r_c(u_h, p_h^{tot}) + r_{c^*}^{stab}(u_h, p_h^{tot}) \tag{22}$$

These stabilization terms are:

$$r_m^{i,stab}(u_h, p_h^{tot}) = \partial_{w_i}(\rho(\mathbf{a}\cdot\nabla\mathbf{w}_h)\tau_1\cdot\mathbf{r_m}(\mathbf{u_h}, p_h^{tot}) + \rho(\nabla\cdot\mathbf{w}_h)\tau_2 r_c(\mathbf{u_h})) \tag{23}$$

$$r_c^{stab}(u_h, p_h^{tot}) = \partial_q(\rho\nabla q_h\tau_1\cdot\mathbf{r_m}(\mathbf{u_h}, p_h^{tot})) \tag{24}$$

$$r_{c^*}^{stab}(u_h, p_h^{tot}) = \partial_{q^*}(\rho \nabla q_h^* \tau_1 \cdot \mathbf{r_m}(\mathbf{u_h}, p_h^{tot})) \tag{25}$$

Take in acccount that the terms that we added to the residuals that correspond to the stabilization depend also on the residuals of the Navier Stokes equations, so if the residuals $\boldsymbol{r_m}$ and r_c are equal to zero, these new residuals R_m, R_c and R_{c*} will be equal to zero as well ensuring consistency.

We want to impose that the residuals of the problem with the added stabilization terms are equal to zero. $R_m = 0$, $R_c = 0$ and $R_{c*} = 0$. We can see now that we can impose that from our functional by solving:

$$\frac{\partial f(\boldsymbol{u})}{\partial v_i} = R_i(\boldsymbol{u}) = 0 \tag{26}$$

where v_i are the test functions of our problem: $\mathbf{w_h}, q_h$ and q_h^* and $\boldsymbol{u}$ represents the unknowns of the problem: $\mathbf{u_h}, p_h$ and p_h^*. Using Newton-Raphson to solve equation 26 we get:

$$\frac{\partial R_i(\boldsymbol{u})}{\partial u_j} \triangle u_j = R_i(\boldsymbol{u}) \tag{27}$$

or equivalently:

$$\frac{\partial^2 f}{\partial v_i \partial u_j} \triangle u_j = \frac{\partial f}{\partial v_i} \tag{28}$$

This allows us to identify the element contributions as:

$$lhs = \frac{\partial^2 f}{\partial v_i \partial u_j} = \frac{\partial R_i}{\partial u_j} \tag{29}$$

$$rhs = \frac{\partial f}{\partial v_i} = R_i \tag{30}$$

The system is therefore:

$$\begin{pmatrix} \frac{\partial R_m}{\partial u_h} & \frac{\partial R_m}{\partial p_h} & \frac{\partial R_m}{\partial p_h^*} \\ \frac{\partial R_c}{\partial u_h} & \frac{\partial R_c}{\partial p_h} & \frac{\partial R_c}{\partial p_h^*} \\ \frac{\partial R_{c^*}}{\partial u_h} & \frac{\partial R_{c^*}}{\partial p_h} & \frac{\partial R_{c^*}}{\partial p_h^*} \end{pmatrix} \begin{pmatrix} u_h \\ p_h \\ p_{h^*} \end{pmatrix} = \begin{pmatrix} R_m \\ R_c \\ R_{c^*} \end{pmatrix} \tag{31}$$

Written in a more compact way:

$$\begin{pmatrix} K & V \\ H & K_{ee} \end{pmatrix} \begin{pmatrix} \mathbf{u_h}, p_h \\ p_h^* \end{pmatrix} = \begin{pmatrix} b + f_v \\ f_e \end{pmatrix} \tag{32}$$

Where:

$$K = \begin{pmatrix} \frac{\partial R_m}{\partial u_h} & \frac{\partial R_m}{\partial p_h} \\ \frac{\partial R_c}{\partial u_h} & \frac{\partial R_c}{\partial p_h} \end{pmatrix} \qquad V = \begin{pmatrix} \frac{\partial R_m}{\partial p_h^*} \\ \frac{\partial R_c}{\partial p_h^*} \end{pmatrix} \tag{33}$$
$$H = \begin{pmatrix} \frac{\partial R_{c^*}}{\partial u_h} & \frac{\partial R_{c^*}}{\partial p_h} \end{pmatrix} \qquad K_{ee} = \begin{pmatrix} \frac{\partial R_{c^*}}{\partial p_h^*} \end{pmatrix}$$

$$b + f_v = \begin{pmatrix} R_m \\ R_c \end{pmatrix} \tag{34}$$
$$f_e = \begin{pmatrix} R_{c^*} \end{pmatrix}$$

Notice that the matrix K and b are the usual stiffness matrix and the load vector respectively that we would obtain if we did not use an enrichment for the pressure. As we mentioned before, the enrichment shape functions are equal to zero on the nodes and as consequence the value of the pressure in the nodes is only defined in terms of p_h and not p_h^*. There is no need then to calculate or store now these values p_h^* as the nodal values p_h still have physical meaning. Therefore, we can simply condensate the degrees of freedom corresponding to ${p_h}^*$ obtaining:

$$(K - VK_{ee}^{-1}H)(\mathbf{u_h}, p_h) = b + f_v - VK_{ee}^{-1}f_e \tag{35}$$

After evaluating these expressions in every Gauss point we would have the element LHS and RHS ready to be assembled into the global system.

Functional Implementation

We have to take in account that the implementation must be valid for both cut and non cut elements. That is why we calculate first K and b and then the enrichment matrices which will be only added in case the element is cut.

Galerkin Functional

$$\begin{aligned}\mathbf{rv\_galerkin} =& \mathbf{w_h^T}\rho\mathbf{f} - \mathbf{w_h^T}\rho\mathbf{acc_h} - \rho\mathbf{convective\_term\ w} + \mathbf{grad\_sym\_w^T}\boldsymbol{\sigma} + \\ &+ \mathbf{div\_w^T p_h} - \rho\mathbf{q_h div\_v}\end{aligned}$$

Stabilization Functional

$$\mathbf{vel\_residual} = \rho\mathbf{f} - \rho\mathbf{acc_h} - \rho\mathbf{convective\_term^T} - \mathbf{grad\_p} \tag{36}$$

$$\begin{aligned}\mathbf{rv\_stab} &= \tau_1\rho\mathbf{grad\_q^T vel\_residual}\\ \mathbf{rv\_stab+} &= \tau_1\rho\mathbf{vel\_residual^T grad\_w\ vconv}\\ \mathbf{rv\_stab-} &= \tau_2\rho\mathbf{div\_w\ div\_v}\end{aligned}$$

Enrichment Functional

$$\mathbf{vel\_residual\ enr} = \mathbf{vel\_residual} - \mathbf{grad\_penr} \tag{37}$$

$$\begin{aligned}\mathbf{rv\_enriched} &= -\mathbf{qenr_h}\rho\mathbf{div\_v} + \mathbf{div\_w\ penr} + \tau_1\rho\mathbf{grad\_qenr^T\ vel\_residual\_enr}\\ \mathbf{rv\_enriched+} &= \tau_1\rho\mathbf{grad\_qenr^T\ vel\_residual\_enr}\\ \mathbf{rv\_enriched-} &= \tau_1\rho\mathbf{grad\_penr^T\ grad\_w\ vconv}\\ \mathbf{rv\_enriched-} &= \tau_1\rho\mathbf{grad\_q^T grad\_penr}\end{aligned}$$

3 Implementation of a solver into KratosMultiphysics specially created to be used along with the previously mentioned two fluid Navier Stokes element.

Finally, once the formulation was ready and the element was created, it was implemented into Kratos. The framework is divided in different modular applications designed to be used to solve different problems present in engineering. The proposed element in this work was implemented inside the FluidDynamicsApplication as a derivation of the already existing element for Navier Stokes.

In order to be able to use this element some other files were required:

- triangle_enrichment_utilities: The elements that are cut by the interface have to follow a special procedure to calculate their contributions. They are splitted into sub-elements with their corresponding enrichment shape functions. There was alredy implemented an utility for this purpose in 3D. Following the same principles an utility was created to do the same for the 2D case.
- two_fluid_navier_stokes_solver: In order to be able to use our element, it was necessary to create a new solver derived from the monolithic_navier_stokes solver that could deal with multiphase flows. Inside the solver, there were calls to the alreading existing level set convection process and redistance process so as to be able to move the interface according to the calculated velocities from the fluid calculation.

- volume_conservation_process: Finally another process was created in order to slightly move the interface at each time step ensuring that there was no lose of mass after doing the convection and redistance.