# The Finite Element Method for Fluid-Structure Interaction with open source software - Doing FSI

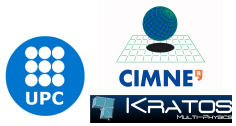
Riccardo Rossi Pooyan Dadvand

Kratos Team

Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE)



July, 2014, TU Munich



# Outline

# Fluid Alone

Let's assume that the solution of the fluid problem alone can be solved by iteratively resolving

$$\begin{pmatrix} \mathbf{F}_{ii} & \mathbf{F}_{i\Gamma} \\ \mathbf{F}_{\Gamma i} & \mathbf{F}_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} d\mathbf{x}_i \\ d\mathbf{x}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{fi} \\ \mathbf{r}_{f\Gamma} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{fi}\left(\mathbf{x}_i^k, \mathbf{x}_\Gamma^k, \mathbf{y}_i^k\right) \\ \mathbf{r}_{f\Gamma}\left(\mathbf{x}_i^k, \mathbf{x}_\Gamma^k, \mathbf{y}_i^k\right) \end{pmatrix} \tag{1}$$

where the indices $i$ and gamma denote parts of the domain on the "interior" $(i)$ or on the FSI interface $\gamma$, $\mathbf{r}_{fi}$ and $\mathbf{r}_{f\Gamma}$ denote the residuals withing the domain and at the interface. The symbol $\mathbf{x}$ is used here to identify all of the unknowns needed for the solution of the fluid problem

**Dirichlet and Neumann conditions**

We will assume that $\mathbf{F}_{ii}, \mathbf{F}_{i\Gamma}, \mathbf{F}_{\Gamma i}$ are modified so to include Dirichlet conditions, and that Neumann conditions are already included in the definition of $\mathbf{r}_{fi}$

clearly depending on the "quality" of the derivatives a Newton-Raphson or Picard scheme can be recovered, leading to different nonlinear convergence properties.

# Structure Alone

The structural problem is here considered in a completely similar fashion as

$$\begin{pmatrix} \mathbf{S}_{ii} & \mathbf{S}_{i\Gamma} \\ \mathbf{S}_{\Gamma i} & \mathbf{S}_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} d\mathbf{y}_i \\ d\mathbf{y}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{si} \\ \mathbf{r}_{s\Gamma} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{si}\left(\mathbf{y}_i^k, \mathbf{y}_\Gamma^k, \mathbf{x}_i^k\right) \\ \mathbf{r}_{s\Gamma}\left(\mathbf{y}_i^k, \mathbf{y}_\Gamma^k, \mathbf{x}_i^k\right) \end{pmatrix} \tag{2}$$

as for the fluid, the indices $i$ and gamma denote parts of the domain on the "interior" ($i$) or on the FSI interface $\gamma$, $\mathbf{r}_{si}$ and $\mathbf{r}_{s\Gamma}$ denote the residuals withing the domain and at the interface. We denote here as $\mathbf{y}$ all the unknowns involved in the solution of the structure

## Dirichlet and Neumann

As before $\mathbf{S}_{ii}, \mathbf{S}_{i\Gamma}, \mathbf{S}_{\Gamma i}$ include Dirichlet conditions, while $\mathbf{r}_{si}$ includes the Neumann conditions

# The discrete FSI problem

Two conditions must be met to ensure that Fluid and Structure Work together:

1. continuity of velocities (and displacements)
2. equilibrium of forces

The easiest way to achieve the continuity of velocities, is to **construct the cinematics imposing that $\mathbf{x}_i \equiv \mathbf{y}_i$**. This is easily accomplished by using a single mesh that encompasses at the same time the fluid and the structure, so that nodes at the interface belong at the same time to the fluid domain AND to the structural domain. Under this conditions the equilibrium of forces at the interface can be expressed as $\mathbf{r}_{fi} + \mathbf{r}_{si} = \mathbf{0}$

# The discrete FSI problem - monolithic form

the result of constraining the kinematics and to impose the equilibrium of the interface is the system

$$\begin{pmatrix} \mathbf{F}_{ii} & \mathbf{F}_{i\Gamma} & 0 \\ \mathbf{F}_{\Gamma i} & \mathbf{F}_{i\Gamma} + \mathbf{S}_{i\Gamma} & \mathbf{S}_{i\Gamma} \\ 0 & \mathbf{S}_{\Gamma i} & \mathbf{S}_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} d\mathbf{x}_i \\ d\mathbf{y}_\Gamma \\ d\mathbf{y}_i \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{fi} \\ \mathbf{r}_{f\Gamma} + \mathbf{r}_{fi} \\ \mathbf{r}_{si} \end{pmatrix} \tag{3}$$

which is known as "monolithic form" of the FSI problem. The monolithic form is equivalent to a "single field approximation" in which two different constitutive models and set of constraints are applied on the structural and fluid parts of the domain.

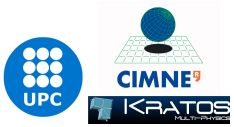
# Monolithic form of FSI

## Advantage of the Monolithic form

The monolithic FSI problem will converge as good as the approximation of the tangent matrix allows (the fact that fluid and structures are treated together will not affect at all the convergence). It is hence **very robust** and it represents the **first choice if it can be done effectively**

**HOWEVER there are difficulties:**

- **the tangent matrix is generally not available in a modular fashion! (and neither the residuals)**
- linear algebra involved may well be nasty, since it will be difficult to construct suitable preconditioners
- Specialized solving procedures can not be used
- cinematics is constrained to be the same, thus limiting the possibility of using different, non conforming, discretizations
- it is **non-modular** in the sense that a single program must implement the fluid and structure solver.



# Alternative Strong-Coupling approaches

An interesting possibility to keep it modular is to keep the fluid and domain solver separated and to combine them by the use of constraints. This is conceptually evident by the use of lagrange multipliers
The key here is to define a mapping (or transfer) operator $\mathbf{T}_{fs}$ of the type $\mathbf{x}_i = \mathbf{T}_{sf}\mathbf{y}_i$ and to consider the system

$$\begin{pmatrix} \mathbf{F}_{ii} & \mathbf{F}_{i\Gamma} & 0 & 0 & 0 \\ \mathbf{F}_{\Gamma i} & \mathbf{F}_{i\Gamma} & 0 & 0 & -I^T \\ 0 & 0 & \mathbf{S}_{ii} & \mathbf{S}_{i\Gamma} & 0 \\ 0 & 0 & \mathbf{S}_{\Gamma i} & \mathbf{S}_{\Gamma\Gamma} & \mathbf{T}_{sf}^T \\ 0 & -I & 0 & \mathbf{T}_{sf} & 0 \end{pmatrix} \begin{pmatrix} d\mathbf{x}_i \\ d\mathbf{x}_\Gamma \\ d\mathbf{y}_i \\ d\mathbf{y}_\Gamma \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{fi} \\ \mathbf{r}_{f\Gamma} \\ \mathbf{r}_{si} \\ \mathbf{r}_{s\Gamma} \\ \mathbf{0} \end{pmatrix} \tag{4}$$

Assuming that the resulting problem is solvable (problem with overconstraint might appear) the use of the lagrangian multipliers identifies the forces needed to guarantee equilibrium at the interface while imposing the continuity of velocities

# Alternative Strong-Coupling approaches

unfortunately, as observed before, the matrix coefficients and the residuals are generally not available in a modular fashion. In order to try to solve this issue, let's write the same problem more compactly as

$$\begin{pmatrix} \mathbf{F} & 0 & \mathbf{T}_f^T \\ 0 & \mathbf{S} & \mathbf{T}_s^T \\ \mathbf{T}_f & \mathbf{T}_s & 0 \end{pmatrix} \begin{pmatrix} d\mathbf{x} \\ d\mathbf{y} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{r}_f \\ \mathbf{r}_s \\ \mathbf{0} \end{pmatrix} \tag{5}$$

with $\mathbf{T}_f := \begin{pmatrix} \mathbf{0} & -\mathbf{I} \end{pmatrix}$ and $\mathbf{T}_s := \begin{pmatrix} \mathbf{0} & \mathbf{T}_{sf} \end{pmatrix}$
We can now "right-precondition" the system. An interesting choice is

$$\mathbf{P} := \begin{pmatrix} \mathbf{F}^{-1} & 0 & 0 \\ 0 & \mathbf{S}^{-1} & 0 \\ 0 & 0 & \mathbf{I} \end{pmatrix} \tag{6}$$

meaning that we solve instead the system

$$\begin{pmatrix} \mathbf{F}^{-1} & 0 & 0 \\ 0 & \mathbf{S}^{-1} & 0 \\ 0 & 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{F} & 0 & \mathbf{T}_f^T \\ 0 & \mathbf{S} & \mathbf{T}_s^T \\ \mathbf{T}_f & \mathbf{T}_s & 0 \end{pmatrix} \begin{pmatrix} d\mathbf{x} \\ d\mathbf{y} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{F}^{-1} & 0 & 0 \\ 0 & \mathbf{S}^{-1} & 0 \\ 0 & 0 & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{r}_f \\ \mathbf{r}_s \\ \mathbf{0} \end{pmatrix} \tag{7}$$

# Alternative Strong-Coupling approaches

or in more compact form

$$\begin{pmatrix} \mathbf{I} & 0 & \mathbf{F}^{-1}\mathbf{T}_f^T \\ 0 & \mathbf{I} & \mathbf{S}^{-1}\mathbf{T}_s^T \\ \mathbf{T}_f & \mathbf{T}_s & 0 \end{pmatrix} \begin{pmatrix} d\mathbf{x} \\ d\mathbf{y} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{F}^{-1}\mathbf{r}_f \\ \mathbf{S}^{-1}\mathbf{r}_s \\ \mathbf{0} \end{pmatrix} \tag{8}$$

The key point here is that the coefficients of **S** and **F** are **not anymore needed**. We paid for this by introducing the inverses!! Is this a good news??

# Alternative Strong-Coupling approaches

or in more compact form

$$\begin{pmatrix} \mathbf{I} & 0 & \mathbf{F}^{-1}\mathbf{T}_f^T \\ 0 & \mathbf{I} & \mathbf{S}^{-1}\mathbf{T}_s^T \\ \mathbf{T}_f & \mathbf{T}_s & 0 \end{pmatrix} \begin{pmatrix} d\mathbf{x} \\ d\mathbf{y} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{F}^{-1}\mathbf{r}_f \\ \mathbf{S}^{-1}\mathbf{r}_s \\ \mathbf{0} \end{pmatrix} \tag{9}$$

The key point here is that the coefficients of **S** and **F** are **not anymore needed**. We paid for this by introducing the inverses!! Is this a good news??

# Alternative Strong-Coupling approaches

The answer is YES, in the context of using a krylov solver for the FSI problem. We've seen this before, bu let's recall that to solve a system of type

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{10}$$

The GMRES algorithm computes $\omega$ as the effect of matrix A onto a vector $\mathbf{v}_i$ (algorithm as found in the book by Saad)

**ALGORITHM 6.9**: GMRES

1. *Compute* $r_0 = b - Ax_0$, $\beta := \|r_0\|_2$, *and* $v_1 := r_0/\beta$
2. *Define the* $(m+1) \times m$ *matrix* $\bar{H}_m = \{h_{ij}\}_{1 \le i \le m+1, 1 \le j \le m}$. *Set* $\bar{H}_m = 0$.
3. *For* $j = 1, 2, \ldots, m$ *Do:*
4. *Compute* $w_j := Av_j$
5. *For* $i = 1, \ldots, j$ *Do:*
6. $h_{ij} := (w_j, v_i)$
7. $w_j := w_j - h_{ij} v_i$
8. *EndDo*
9. $h_{j+1,j} = \|w_j\|_2$. *If* $h_{j+1,j} = 0$ *set* $m := j$ *and go to 12*
10. $v_{j+1} = w_j / h_{j+1,j}$
11. *EndDo*
12. *Compute* $y_m$ *the minimizer of* $\|\beta e_1 - \bar{H}_m y\|_2$ *and* $x_m = x_0 + V_m y_m$.

# A Neumann-Neumann algorithm

for example if we were to apply the algorithm to a the preconditioned system, we could notice that the iterations typically start with a zero initial guess *for the variation*, hence the first residual would be

$$\mathbf{r}_0 = \begin{pmatrix} \mathbf{F}^{-1}\mathbf{r}_f \\ \mathbf{S}^{-1}\mathbf{r}_s \\ \mathbf{0} \end{pmatrix} \tag{11}$$

effectively telling that each of the two domains behaves as if the other did not exist

Note that one may easily start from a kinematically compatible situation which is not in equilibrium. This could be achieved for example by applying the solution of the structure as a dirichlet constraint on the structure to get an inital guess.

# A Neumann-Neumann algorithm

During the iterations assuming an iteration vector of the type $v_j := \begin{pmatrix} \mathbf{v}_F & \mathbf{v}_S & \mathbf{v}_\lambda \end{pmatrix}^T$ we need to compute

$$\omega := \begin{pmatrix} \mathbf{I} & 0 & \mathbf{F}^{-1}\mathbf{T}_f^T \\ 0 & \mathbf{I} & \mathbf{S}^{-1}\mathbf{T}_s^T \\ \mathbf{T}_f & \mathbf{T}_s & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v}_F \\ \mathbf{v}_S \\ \mathbf{v}_\lambda \end{pmatrix} \tag{12}$$

this can be done via auxiliary steps defining $\mathbf{f}_f := \mathbf{T}_f^T \mathbf{v}_\lambda$ and $\mathbf{f}_s := \mathbf{T}_s^T \mathbf{v}_\lambda$ as the forces acting on either side due to the action of the constraint.
In this context it is hence clear that the iteration vector becomes

$$\omega := \begin{pmatrix} \mathbf{F}^{-1}\mathbf{f}_f \\ \mathbf{S}^{-1}\mathbf{f}_s \\ \mathbf{T}_f\mathbf{v}_f + \mathbf{T}_s\mathbf{v}_s \end{pmatrix} \tag{13}$$

where we can interprete $\mathbf{F}^{-1}\mathbf{f}_f$ and $\mathbf{S}^{-1}\mathbf{f}_s$ as the variation of velocities on either side induced by the change in the forces at the interface. Since Neumann conditions (forces) are applied here on both sides, this denotes a Neumann-Neumann algorithm.

# The idea of Robin-Robin approaches

Of course many techniques other techniques exist for preconditioning.
The drawback of the Neumann-Neumann approach is obviously that the stiffness of the other domain is NOT taken into account when each domain is solved.
To try to cope with this problem one may recast the lagrangian multiplier problem into an augmented lagrangian problem.
This corresponds to considering a system of the type

$$\begin{pmatrix} \mathbf{F}_{ii} & \mathbf{F}_{i\Gamma} & 0 & 0 & 0 \\ \mathbf{F}_{\Gamma i} & \mathbf{F}_{i\Gamma} & 0 & 0 & -\mathbf{I}^T \\ 0 & 0 & \mathbf{S}_{ii} & \mathbf{S}_{i\Gamma} & 0 \\ 0 & 0 & \mathbf{S}_{\Gamma i} & \mathbf{S}_{\Gamma\Gamma} & \mathbf{T}_{sf}^T \\ 0 & -\mathbf{I} & 0 & \mathbf{T}_{sf} & \epsilon\mathbf{I} \end{pmatrix} \begin{pmatrix} d\mathbf{x}_i \\ d\mathbf{x}_\Gamma \\ d\mathbf{y}_i \\ d\mathbf{y}_\Gamma \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{fi} \\ \mathbf{r}_{f\Gamma} \\ \mathbf{r}_{si} \\ \mathbf{r}_{s\Gamma} \\ \mathbf{0} \end{pmatrix} \tag{14}$$

where $\epsilon$ represents a **suitably small** penalty such that $\frac{1}{\epsilon}$ is "large" when compared to $\mathbf{F}_{i\Gamma}$ and $\mathbf{S}_{i\Gamma}$

# The idea of Robin-Robin approaches

The advantage is that we can now symbolically obtain $\lambda = \frac{1}{\epsilon} d\mathbf{x}_\Gamma - \frac{1}{\epsilon}\mathbf{T}_{sf}\mathbf{y}_\Gamma$ and substitute it in the system to get

$$\begin{pmatrix} \mathbf{F}_{ii} & \mathbf{F}_{i\Gamma} & 0 & 0 \\ \mathbf{F}_{\Gamma i} & \mathbf{F}_{i\Gamma} + \frac{1}{\epsilon}\mathbf{I} & 0 & -\frac{1}{\epsilon}\mathbf{T}_{sf}^T \\ 0 & 0 & \mathbf{S}_{ii} & \mathbf{S}_{i\Gamma} \\ 0 & -\frac{1}{\epsilon}\mathbf{T}_{sf} & \mathbf{S}_{\Gamma i} & \mathbf{S}_{\Gamma\Gamma} + \frac{1}{\epsilon}\mathbf{T}_{sf}\mathbf{T}_{sf}^T \end{pmatrix} \begin{pmatrix} d\mathbf{x}_i \\ d\mathbf{x}_\Gamma \\ d\mathbf{y}_i \\ d\mathbf{y}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{fi} \\ \mathbf{r}_{f\Gamma} \\ \mathbf{r}_{si} \\ \mathbf{r}_{s\Gamma} \end{pmatrix} \tag{15}$$

which gives equivalent results for sufficiently low values of the penalty parameter

# The idea of Robin-Robin approaches

An attractive possibility at this point would be to construct a right preconditioner of the type

$$\mathbf{P}_\epsilon := \begin{pmatrix} \mathbf{F}_\epsilon^{-1} & 0 \\ 0 & \mathbf{S}_\epsilon^{-1} \end{pmatrix} \tag{16}$$

with

$$\mathbf{F}_\epsilon := \begin{pmatrix} \mathbf{F}_{ii} & \mathbf{F}_{i\Gamma} \\ \mathbf{F}_{\Gamma i} & \mathbf{F}_{\Gamma\Gamma} + \frac{1}{\epsilon}\mathbf{I} \end{pmatrix} \quad \mathbf{S}_\epsilon := \begin{pmatrix} \mathbf{S}_{ii} & \mathbf{S}_{i\Gamma} \\ \mathbf{S}_{\Gamma i} & \mathbf{S}_{\Gamma\Gamma} + \frac{1}{\epsilon}\mathbf{T}_{sf}\mathbf{T}_{sf}^T \end{pmatrix} \tag{17}$$

The nice thing of such preconditioner is that it corresponds to the application of both a velocity and a force at the interface, the two conditions competing one with the other. Such preconditioner is hence known as Robin-Robin preconditioner and is known to behave well in hard FSI problems.

# Dirichlet-Neumann Preconditioner

An alternative possibility is finally to impose the velocity of the FSI boundary as a Dirichlet condition, while still applying the corresponding interaction forces on the structure. Schematically this corresponds to solving the system

$$\begin{pmatrix} \mathbf{F}_{ii} & \mathbf{F}_{i\Gamma} & 0 & 0 \\ 0 & \mathbf{I} & 0 & \mathbf{T}_{sf}^{T} \\ 0 & 0 & \mathbf{S}_{ii} & \mathbf{S}_{i\Gamma} \\ 0 & -\mathbf{T}_{sf} & \mathbf{S}_{\Gamma i} & \mathbf{S}_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} d\mathbf{x}_i \\ d\mathbf{x}_\Gamma \\ d\mathbf{y}_i \\ d\mathbf{y}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{r}_{fi} \\ 0 \\ \mathbf{r}_{si} \\ \mathbf{r}_{s\Gamma} \end{pmatrix} \tag{18}$$

This is probably the most widely spread preconditioner. Such preconditioner **only works well when the structure is "heavy" or very rigid**

A very dangerous characteristic of such technique is that the block $\begin{pmatrix} \mathbf{F}_{ii} & \mathbf{F}_{i\Gamma} \\ 0 & \mathbf{I} \end{pmatrix}$ is singular in many cases, namely when the fluid domain is completely closed. In the cases in which it is well defined, it is still convenient to use it together with a GMRES scheme for the interface

# Conclusions about FSI

**FSI problems are as nonlinear as the subproblems...**

The problem in FSI is the convergence of the underlying domain decomposition problem!!! - **Learn to solve the linear problem to be faced at each iteration and you'll learn how to solve the complete problem**