Coupling in Time I

Partitioned and Monolithic Schemes

INTRODUCTION

Scenarios which motivate the need of solving a coupled problem:

- **Research projects** in which two or more research fields need to be put together in order to solve a more complex problem.
- **Product development**, where design and verification involve solving problems which obey to different constitutive equations.
- **Commercial software development**, two or more already existing modules for solving a particular physical problem need to be put together and solved in a coupled way.

In all of these cases we depart from the hypothesis that we already know how to solve each of the problems, but now we need to solve their interaction.

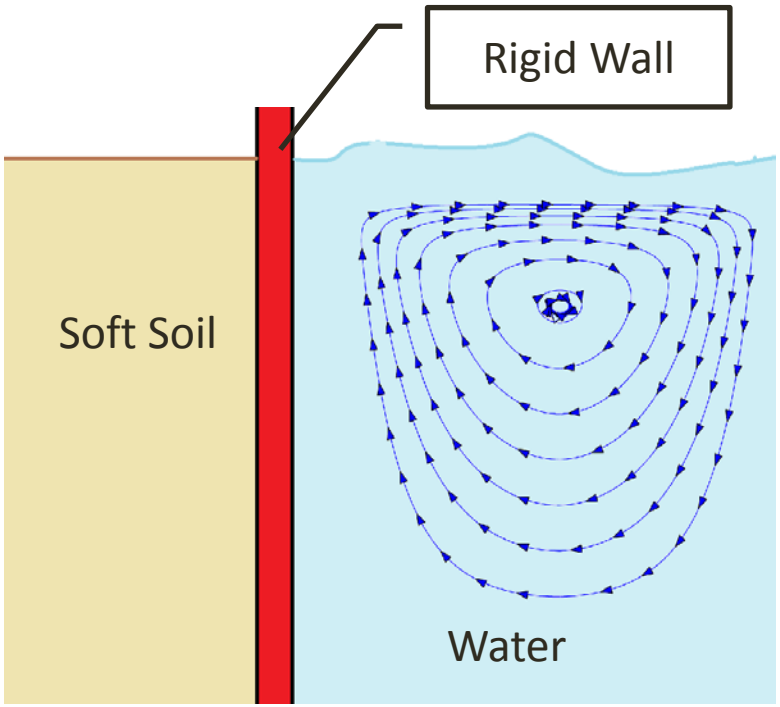
INTRODUCTION

Suppose that we have a coupled dynamical system which we need to evolve in time. There exist three approaches to the simulation of the coupled system:

- **Field elimination**: One or more fields are eliminated, the remaining field is integrated by means of a time stepping scheme.
- **Monolithic or Simultaneous treatment**: The two or more problems to be coupled are treated in a monolithic way, that is, we solve the equations associated to all the problems in a single system, simultaneously.
- **Partitioned System**: Each of the problems to be solved is advanced in time independently. The interaction with the other problems is treated as an external forcing term which are iteratively communicated between the coupled systems.

FIELD ELIMINATION

An example of Field Elimination

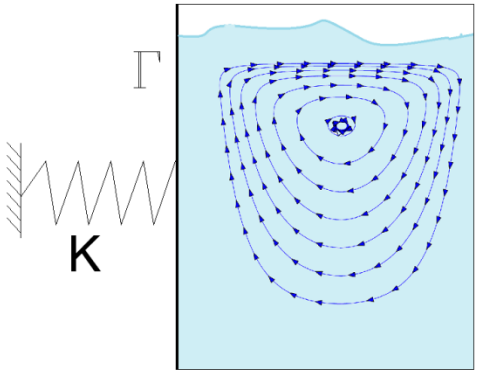


If we are interested in the fluid behavior, we can simplify the problem for the soil and rigid wall domains.

FIELD ELIMINATION

An example of Field Elimination

The effect of the soil and the rigid wall is modeled as a spring:



The boundary condition for the fluid at the wall can now be simplified to:

$$Kd_x = \int_{\Gamma} p$$

The unknowns for the solid domain problem have been eliminated, we only need to solve for the fluid.

FIELD ELIMINATION

However, this cannot be done when facing a general problem.

- Most problems cannot be simplified to few degrees of freedom
- The condensation of the degrees of freedom associated to a general problem can lead to ill-conditioned systems of equations, often involving higher order differential systems in time

Field Elimination is generally restricted to simple linear problems.

A possibility for applying a particular kind of Field Elimination are Reduced-Order Models, but these have complexities of their own.

DISCRETIZATION IN TIME

Let us consider the discretization in time of the heat transfer equation:

$$\begin{aligned} \partial_t u_1 - k\Delta u_1 &= f && \text{in } \Omega_1 \\ u_1 &= \bar{u} && \text{on } \Gamma_D \\ k\boldsymbol{n}\cdot\nabla \boldsymbol{u}_1 &= \bar{t} && \text{on } \Gamma_N \\ u_1(\boldsymbol{x},0) &= u_{10}(\boldsymbol{x}) && \text{in } \Omega_1 \end{aligned}$$

The time dependent variational form of the problem is:

$$(v,\partial_t u) + k(v,u) - < v, \boldsymbol{n}\cdot\nabla u >_\Gamma = < v, f >$$

plus boundary and initial conditions.

DISCRETIZATION IN TIME

We define the finite element space: After discretization in space the problem is:

$$(v_h, \partial_t u_h) + k(\nabla v_h, \nabla u_h) - < v_h, \boldsymbol{n} \cdot \nabla u_h >_\Gamma = < v_h, f > \quad \forall v_h \in V_h(\Omega)$$

Since we are approximating u_h trough shape functions in V_h, the time dependency is found in the coefficients which multiply the test functions:

$$u(\boldsymbol{x}, t) \approx u_h(\boldsymbol{x}, t) = \sum_j u_j(t) N_j(\boldsymbol{x})$$

$$\partial_t u(\boldsymbol{x}, t) \approx \partial_t u_h(\boldsymbol{x}, t) = \sum_j \partial_t u_j(t) N_j(\boldsymbol{x})$$

The associated algebraic problem is:

$$\boldsymbol{M} \frac{d\boldsymbol{U}}{dt} + \boldsymbol{K}\boldsymbol{U} = \boldsymbol{F}$$

We will call ***M*** the mass matrix, ***K*** the stiffness matrix and ***F*** the forcing term.

DISCRETIZATION IN TIME

Time discretization

There exist several possibilities for the time dicretization.

We start by defining a splitting of the time interval $[0, T]$:

$$0 = t^0 < t^1 < t^2 < .. < t^N = T$$

If these time instants are equispaced, then $t^{n+1} = t^n + \delta t$, with $\delta t = T/(N+1)$

The objective is to, given $u_h(\boldsymbol{x}, t^n)$, compute $u_h(\boldsymbol{x}, t^{n+1})$.

We introduce the notation: $u_h^n \coloneqq u_h(\boldsymbol{x}, t^n)$

DISCRETIZATION IN TIME

Time discretization

θ – scheme time integration

We use a finite difference discretization for the time derivative

The θ parameter denotes where the operator is evaluated:

$$\frac{1}{\delta t}(v_h, u_h^{n+1} - u_h^n) + \theta\left[k(\nabla v_h, \nabla u_h^{n+1}) - < v_h, \boldsymbol{n} \cdot \nabla u_h^{n+1} >_\Gamma - < v_h, f^{n+1} >\right]$$
$$+(1-\theta)[k(\nabla v_h, \nabla u_h^n) - < v_h, \boldsymbol{n} \cdot \nabla u_h^n >_\Gamma - < v_h, f^n > \ ] = 0$$

with $0 \leq \theta \leq 1$.

The algebraic form of the problem is now:

$$\boldsymbol{M}\delta_t \boldsymbol{U}^{n+1} + \theta \boldsymbol{K}\boldsymbol{U}^{n+1} + (1-\theta)\boldsymbol{K}\boldsymbol{U}^n = \theta \boldsymbol{F}^{n+1} + (1-\theta)\boldsymbol{F}^n$$

The definition of the several θ methods appears from the choice of θ:

$\theta = 0$:	Forward Euler scheme, explicit, conditionally stable,	$\boldsymbol{\mathcal{O}}(\delta t)$
$\theta = 1/2$:	Crank-Nicolson scheme, implicit, unconditionally stable,	$\boldsymbol{\mathcal{O}}(\delta t^2)$
$\theta = 1$:	Backward Euler scheme, implicit, unconditionally stable,	$\boldsymbol{\mathcal{O}}(\delta t\ )$

DISCRETIZATION IN TIME

Time discretization

Stability of the θ scheme:

Let us consider the expression of the 1D solution in Fourier modes:

$$u^n = \sum_j U_j^n \phi_j$$

Let us also consider the equation without external forces (homogeneous solution).

If we insert this solution into the discrete θ-equation, taking into account that all modes are orthogonal, we obtain:

$$\frac{1}{\delta t} M_j \left(U_j^{n+1} - U_j^{n+1} \right) + \theta K_j U_j^{n+1} + (1-\theta) K U_j^n = 0$$

So:

$$U_j^{n+1} = \frac{\frac{1}{\delta t} M_j - (1-\theta) K}{\frac{1}{\delta t} M_j + \theta K} U_j^n$$

We get unconditional stability only if we can assure that for any δt:

$$\left| \frac{\frac{1}{\delta t} M_j - (1-\theta) K}{\frac{1}{\delta t} M_j + \theta K} \right| \leq 1 \Rightarrow \theta \geq \frac{1}{2}$$

DISCRETIZATION IN TIME

Time discretization

Backward differences in time

In this method several of the previous steps in time are used to approximate the time derivative at t^{n+1}.

The schemes are implicit because we evaluate the operator at t^{n+1}:

$$\boldsymbol{M}\delta_t\boldsymbol{U}^{n+1} + \boldsymbol{K}\boldsymbol{U}^{n+1} = \boldsymbol{F}^{n+1}$$

Approximation for $\delta_t\boldsymbol{U}^{n+1}$:

- BDF1: $\delta_t\boldsymbol{U}^{n+1} = (\boldsymbol{U}^{n+1} - \boldsymbol{U}^{n})/\delta t$ Unconditionally stable, $\boldsymbol{\mathcal{O}}(\delta t)$
- BDF2: $\delta_t\boldsymbol{U}^{n+1} = (\frac{\mathbf{3}}{\mathbf{2}}\boldsymbol{U}^{n+1} - \mathbf{2}\boldsymbol{U}^{n} + \frac{\mathbf{1}}{\mathbf{2}}\boldsymbol{U}^{n-1})/\delta t$ Unconditionally stable, $\boldsymbol{\mathcal{O}}(\delta t^2)$
- BDF3: $\delta_t\boldsymbol{U}^{n+1} = (\frac{\mathbf{11}}{\mathbf{6}}\boldsymbol{U}^{n+1} - \mathbf{3}\boldsymbol{U}^{n} + \frac{\mathbf{3}}{\mathbf{2}}\boldsymbol{U}^{n-1} - \frac{\mathbf{1}}{\mathbf{3}}\boldsymbol{U}^{n-2})/\delta t$ Conditionally stable, $\boldsymbol{\mathcal{O}}(\delta t^3)$

Up to order 6.

DISCRETIZATION IN TIME

Time discretization

Adams method (Linear multistep method)

- This method is derivated from polynomial fitting
- It can be either explicit or implicit

Adams-Bashforth (explicit):

ADB1: $\frac{\boldsymbol{M}}{\delta t}(\boldsymbol{U}^{n+1}-\boldsymbol{U}^{n})+\boldsymbol{K}\boldsymbol{U}^{n}=\boldsymbol{F}^{n}$

ADB2: $\frac{\boldsymbol{M}}{\delta t}(\boldsymbol{U}^{n+1}-\boldsymbol{U}^{n})+\frac{3}{2}\boldsymbol{K}\boldsymbol{U}^{n}-\frac{1}{2}\boldsymbol{K}\boldsymbol{U}^{n-1}=\frac{3}{2}\boldsymbol{F}^{n}-\frac{1}{2}\boldsymbol{F}^{n-1}$

…

Adams-Moulton (implicit):

ADM1: $\frac{\boldsymbol{M}}{\delta t}(\boldsymbol{U}^{n+1}-\boldsymbol{U}^{n})+\boldsymbol{K}\boldsymbol{U}^{n+1}=\boldsymbol{F}^{n+1}$

ADM2: $\frac{\boldsymbol{M}}{\delta t}(\boldsymbol{U}^{n+1}-\boldsymbol{U}^{n})+\frac{1}{2}\boldsymbol{K}\boldsymbol{U}^{n+1}+\frac{1}{2}\boldsymbol{K}\boldsymbol{U}^{n}=\frac{1}{2}\boldsymbol{F}^{n+1}+\frac{1}{2}\boldsymbol{F}^{n}$

The method involves the evaluation of the operator at preceding time steps, but they can be stored in memory.

DISCRETIZATION IN TIME

Time discretization
Runge-Kutta methods

Intermediate steps are used for the evaluation of the time derivative

The most popular one is the explicit 4th order Runge-Kutta method:

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \frac{1}{6}\delta t(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 3\boldsymbol{k}_3 + \boldsymbol{k}_4)$$

with:

$$\boldsymbol{k}_1 = \boldsymbol{F}^n - \boldsymbol{K}\boldsymbol{U}^n$$

$$\boldsymbol{k}_2 = \boldsymbol{F}^{n+1/2} - \boldsymbol{K}\left(\boldsymbol{U}^n + \frac{1}{2}\delta t\boldsymbol{k}_1\right)$$

$$\boldsymbol{k}_3 = \boldsymbol{F}^{n+1/2} - \boldsymbol{K}\left(\boldsymbol{U}^n + \frac{1}{2}\delta t\boldsymbol{k}_2\right)$$

$$\boldsymbol{k}_4 = \boldsymbol{F}^{n+1} - \boldsymbol{K}(\boldsymbol{U}^n + \delta t\boldsymbol{k}_3)$$

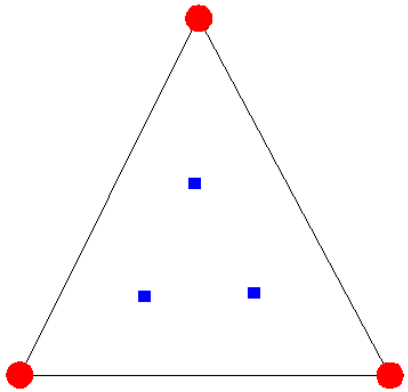
DISCRETIZATION IN TIME

Lumping the Mass matrix

When using explicit time integration schemes, it is convenient to have a diagonal mass matrix to avoid solving a linear system of equations.

The term we need to approximate is: $(v_h, u_h) = \int_\Omega N_i N_j d\Omega U_j$

A diagonal mass matrix can be obtained by using a closed quadrature: Gauss points are on the finite element nodes.



For high-order methods, standard closed quadrature rules yield zero mass for some of the nodes. Special quadrature rules need to be used.

DISCRETIZATION IN TIME

Time discretization

Overview

- Explicit time integration:
  - No linear system to solve
  - Conditionally stable: small time steps

- Implicit time integration schemes:
  - Need to solve linear systems for the implicit system
  - Unconditionally stable up to 2^{nd} order. Larger time steps are possible

Adams and Backward differences methods:

- We only evaluate one function per step
- Need to be started with lower order methods

Runge-Kutta methods

- Several functions need to be evaluated per step
- Better stability and accuracy for the same order
- More expensive

DISCRETIZATION IN TIME

Second derivative in time problems

We are now considering second order in time problems (wave equation, solid mechanics…)

The problem we want to solve is of the form:

$$\partial_{tt}^2 u_h + \widehat{\mathcal{L}}(\partial_t u_h) + \mathcal{L}(u_h) = f$$

We need to approximate the second derivative in time: $\partial_{tt}^2 u_h$.
The most common approach is to introduce and additional variable:

$$y_h = \partial_t u_h$$

And solve the coupled system:

$$\partial_t y_h + \widehat{\mathcal{L}}(y_h) + \mathcal{L}(u_h) = f$$
$$y_h = \partial_t u_h$$

This implicitly introduces an approximation for $\partial_{tt}^2 u_h$. Now all the methods for the first time derivative can be applied.

DISCRETIZATION IN TIME

Second derivative in time problems

A popular implicit scheme for the solid mechanics equations is Newmark's method:

$$u_h^{n+1} = u_h^n + \delta_t u_h^n \delta t + \left[\left(\frac{1}{2} - \beta\right)\delta_{tt}^2 u_h^n + \beta \delta_{tt}^2 u_h^{n+1}\right]\delta t^2$$

$$\delta_t u_h^{n+1} = \delta_t u_h^n + \left[(1-\gamma)\delta_{tt}^2 u_h^n + \gamma \delta_{tt}^2 u_h^{n+1}\right]\delta t$$

γ and β are parameters which define the particular Newmark method:

Stability conditions:

$$\gamma \geq \frac{1}{2}$$

$$\beta \geq \frac{1}{4}\left(\frac{1}{2} + \gamma\right)^2$$

The first condition implies Crank-Nicolson or Backward Euler for the velocity equation.

Second order, stable, non-dissipative scheme: $\gamma = 0.5, \beta = 0.25.$

WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

Let us consider the stationary finite element approximation of the heat transfer equation. We define $V = H^1(\Omega), V_h \subset V$. The problem is: find $u \in V_h$:

$$k(\nabla v_h, \nabla u_h) - k < v_h, \boldsymbol{n} \cdot \nabla u_h >_\Gamma = < v_h, f > \quad \forall v_h \in V_h(\Omega)$$

No boundary conditions have been applied so far. Let us suppose that the boundary conditions are of the form:

$$u = \bar{u} \quad \text{in } \Gamma$$

We decompose u_h into the contribution of shape functions with nodes on Γ and interior shape functions:

$$u_h(\boldsymbol{x}) = u_h^0(\boldsymbol{x}) + u_h^\Gamma(\boldsymbol{x}) = \sum_j U_j^0 N_j^0(\boldsymbol{x}) + \sum_j U_j^\Gamma N_j^\Gamma(\boldsymbol{x})$$

Let us also denote $V_{h,0}$ as the subspace of V_h with functions vanishing on Γ.

WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

The usual approach for enforcing Dirichlet boundary conditions strongly is:

- We test only against test functions $v_{h,0} \in V_{h,0}$.
- Since the value of the unknown in the boundary is known, we can say:

$$U_j^{\Gamma} = \bar{u}$$

- We take into account that, since $v_{h,0}$ vanishes on Γ:

$$k < v_{h,0}, \boldsymbol{n} \cdot \nabla u_h >_{\Gamma} = 0$$

Then, the weak form of the problem with boundary conditions is:

$$k\left(\nabla v_{h,0}, \nabla u_h^0\right) =< v_h, f > -k\left(\nabla v_{h,0}, \nabla u_h^{\Gamma}\right) \qquad \forall v_{h,0} \in V_{h,0}(\Omega)$$

Let us study the stability of this problem.

WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

Poincaré inequality: $\forall u \in H^1(\Omega)$:

$$||u||_{L^2(\Omega)} \leq c_1 L_0 ||u||_{L^2(\Gamma)} + c_2 L_0 ||\nabla u||_{L^2(\Omega)}$$

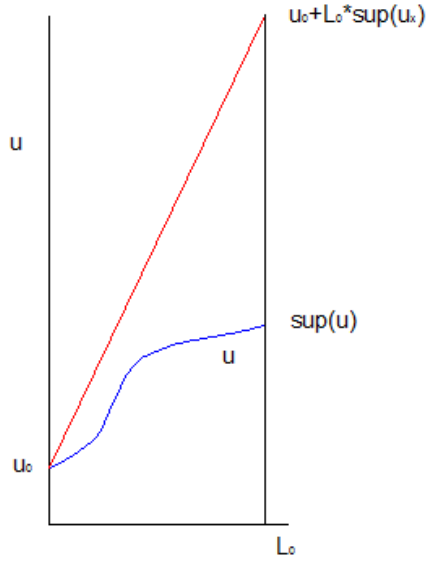
Poincaré inequality states that for a sufficiently smooth u, we can bound the value of the function in the interior of the subdomain through the value in the boundary and its gradient.

In 1D, suppose $u \in C^1(\Omega)$:

$$||u||_{L^\infty(\Omega)} \leq |u_0| + L_0 ||\nabla u||_{L^\infty(\Omega)}$$

Inverse Poincaré inequality:
$\forall u \in H^1(\Omega)$, for a patch of elements *w*:

$$||u||_{L^2(w)} \geq c_i \, h ||\nabla u||_{L^2(w)}$$



WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

Coercivity:

A bilinear form $B(\cdot,\cdot): V \times V \rightarrow R$ is said to be coercive if:

$$B(v,v) \geq k\left|\left|v\right|\right|_V^2 \quad \forall v \in V$$

The coercivity of the bilinear form ensures that the resulting system matrix will be positive definite.

Let us show that the bilinear form with boundary conditions is coercive in the norm:

$$\left|\left|\left|v\right|\right|\right|^2 = k\left|\left|u\right|\right|_{L^2(\Gamma)}^2 + k\left|\left|\nabla u\right|\right|_{L^2(\Omega)}^2 \geq c_1\, k\left|\left|u\right|\right|_{H^1(\Omega)}^2$$

We have:

$$B(v_{h,0}, v_{h,0}) = k(\nabla v_{h,0}, \nabla v_{h,0}) = k\left|\left|\nabla v_{h,0}\right|\right|_{L^2(\Omega)}^2 = k\left|\left|\nabla v_{h,0}\right|\right|_{L^2(\Omega)}^2 + k\left|\left|v_{h,0}\right|\right|_{L^2(\Gamma)}^2$$

WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

Weak imposition of boundary conditions

Instead of restricting to $V_{h,0}$ we want to consider the full V_h space:

$$B(v_h, u_h) = k(\nabla v_h, \nabla u_h) - k < v_h, \boldsymbol{n} \cdot \nabla u_h >_\Gamma = < v_h, f > \quad \forall v_h \in V_h(\Omega)$$

If we study the coercivity of B we see that:

$$\begin{aligned} B(u_h, u_h) &= k\left|\left|\nabla u_h\right|\right|^2_{L^2(\Omega)} - k < u_h, \boldsymbol{n} \cdot \nabla u_h >_\Gamma \\ &\geq k\left|\left|\nabla u_h\right|\right|^2_{L^2(\Omega)} - k\left|\left|u_h\right|\right|_{\mathrm{L}^2(\Gamma)}\left|\left|\nabla u_h\right|\right|_{\mathrm{L}^2(\Gamma)} \\ &\geq k\left|\left|\nabla u_h\right|\right|^2_{L^2(\Omega)} - \frac{c_i}{L_0} k\left|\left|u_h\right|\right|^2_{\mathrm{L}^2(\Gamma)} \end{aligned}$$

Not only we do not have control on $k\left|\left|u_h\right|\right|^2_{L^2(\Gamma)}$ but there are terms which are substracting stability.

WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

The penalty method

In the penalty method this issue is addressed by adding a large enough penalty term:

$$B_p(v_h, u_h) = k(\nabla v_h, \nabla u_h) - k < v_h, \boldsymbol{n} \cdot \nabla u_h >_\Gamma + \alpha(v_h, u_h)_\Gamma$$
$$=< v_h, f > + \alpha(v_h, \bar{u})_\Gamma \qquad \forall v_h \in V_h(\Omega)$$

The additional term is consistent (it enforces the Dirichlet boundary conditions) and will ensure the stability of the weak form.

Coercivity of B_p:

$$B_p(u_h, u_h) = k(\nabla u_h, \nabla u_h) - k < u_h, \boldsymbol{n} \cdot \nabla u_h >_\Gamma + \alpha(u_h, u_h)_\Gamma$$
$$\geq k\left|\left|u_h\right|\right|^2_{L^2(\Omega)} - \frac{c_i}{h} k\left|\left|u_h\right|\right|^2_{\mathrm{L}^2(\Gamma)} + \alpha\left|\left|u_h\right|\right|_{L^2(\Gamma)}$$

It is clear that in order to ensure stability we require:

$$\alpha > \frac{c_i}{h} k$$

WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

The penalty method

Some inconvenients of the penalty method:

- It is non-symmetric even for symmetric problems
- The required value for α is difficult to estimate. In practice it is taken very large (10^6), which can result in ill-conditioned systems of equations.

WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

Nitche's method

A symmetric, better conditioned version of the penalty method:

$$B_N(v_h,u_h) = k(\nabla v_h,\nabla u_h) - k < v_h, \boldsymbol{n}\cdot\nabla u_h >_\Gamma + \alpha\frac{k}{h}(v_h,u_h)_\Gamma$$
$$-k < \boldsymbol{n}\cdot\nabla v_h, u_h >_\Gamma$$
$$=< v_h, f > + \alpha\frac{k}{h}(v_h,\bar{u})_\Gamma - k < \boldsymbol{n}\cdot\nabla v_h, \bar{u} >_\Gamma \qquad \forall v_h \in V_h(\Omega)$$

The term:

$$k < \boldsymbol{n}\cdot\nabla v_h, u_h >_\Gamma - k < \boldsymbol{n}\cdot\nabla v_h, \bar{u} >_\Gamma$$

enforces boundary conditions and makes the method symmetric (although it substracts stability).

It is now sufficient to take: $\alpha > 2c_i$ to ensure stability

c_i depends on the shape of the elements, so for non-stretched elements $c_i = \boldsymbol{\mathcal{O}}(1)$.

WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

Nitche's method

Different expressions for the α parameter need to be devised for each physical problem!

Example: Convection – diffusion equation

$$\begin{aligned} -k\Delta u + \boldsymbol{a}\cdot\nabla u &= f && \text{in } \Omega \\ u &= \bar{u} && \text{on } \Gamma_D \end{aligned}$$

The weak form of the problem is:

$$B(v_h,u_h) = k(\nabla v_h,\nabla u_h) + (v_h,\boldsymbol{a}\cdot\nabla u_h) - k < v_h,\boldsymbol{n}\cdot\nabla u_h >_\Gamma = < v_h,f > \quad \forall v_h \in V_h(\Omega)$$

This weak form requires of stabilization terms in order to ensure that the error is bounded, but coercivity of the bilinear form can be obtained without them.

WEAK IMPOSITION OF DIRICHLET BOUNDARY CONDITIONS

Nitche's method

If we study the coercivity of the bilinear form without boundary conditions:

$$B(u_h,u_h) = k\left|\left|\nabla u_h\right|\right|_{L^2(\Omega)} + (u_h, \boldsymbol{a}\cdot\nabla u_h) - k < u_h, \boldsymbol{n}\cdot\nabla u_h >_\Gamma$$

$$\geq k\left|\left|\nabla u_h\right|\right|_{L^2(\Omega)} + (u_h, \boldsymbol{a}\cdot\nabla u_h) - k\frac{c_i}{h}\left|\left|u_h\right|\right|_{L^2(\Omega)}$$

$$= k\left|\left|\nabla u_h\right|\right|_{L^2(\Omega)} + \int_\Omega \boldsymbol{a}\cdot\frac{\nabla u_h^2}{2} - k\frac{c_i}{h}\left|\left|u_h\right|\right|_{L^2(\Omega)}$$

$$= k\left|\left|\nabla u_h\right|\right|_{L^2(\Omega)} + \int_\Omega \nabla\cdot\left(\boldsymbol{a}\frac{u_h^2}{2}\right) - \int_\Omega \nabla\cdot\boldsymbol{a}\left(\frac{u_h^2}{2}\right) - k\frac{c_i}{h}\left|\left|u_h\right|\right|_{L^2(\Omega)}$$

$$= k\left|\left|\nabla u_h\right|\right|_{L^2(\Omega)} + \int_\Gamma \boldsymbol{n}\cdot\boldsymbol{a}\left(\frac{u_h^2}{2}\right) - k\frac{c_i}{h}\left|\left|u_h\right|\right|_{L^2(\Omega)}$$

In the outflow boundary: $\boldsymbol{n}\cdot\boldsymbol{a} > 0$ the convective term is adding stability

But in the inflow $\boldsymbol{n}\cdot\boldsymbol{a} < 0$ it substracts stability!

For Nitsche's method, we will require a stabilization term in the inflow of the form:

$$\left(\alpha\frac{k}{h} - \frac{1}{2}\boldsymbol{n}\cdot\boldsymbol{a}\right)(v_h, u_h - \bar{u})_\Gamma$$

DIRICHLET BOUNDARY CONDITIONS THROUGH LAGRANGE MULTIPLIERS

Another possibility consists in the use of Lagrange Multipliers for the imposition of Dirichlet boundary conditions.

Let us define $V_h \subset H^1(\Omega)$, $Q_h \subset H^{-1/2}(\Gamma)$. Let us consider the weak form of the problem: find $u_h \in V_h, \lambda_h \in Q_h$ such that:

$$\begin{aligned} k(\nabla v_h, \nabla u_h) - < v_h, \lambda_h >_\Gamma = < v_h, f > \qquad & \forall v_h \in V_h \\ < \mu, u_h >_\Gamma = < \mu, \bar{u} >_\Gamma \qquad & \forall \mu_h \in Q_h \end{aligned}$$

The Lagrange multiplier λ_h plays the role of the fluxes $k(\boldsymbol{n} \cdot \nabla u_h)$.

The boundary conditions are tested against μ, the test function for the Lagrange multipliers.

This is very similar to Nitsche's method with some differences:

- There is no penalty parameter.
- The space for the Lagrange multipliers is arbitrary, contrary to the space for $k\boldsymbol{n} \cdot \nabla u_h$ which is linked to the interpolation space for u_h.
- An inf-sup compatibility condition between interpolation spaces needs to be fulfilled.

DIRICHLET BOUNDARY CONDITIONS THROUGH LAGRANGE MULTIPLIERS

Interpolation spaces compatibility conditions

- $\forall \lambda_h \in Q_h \quad \exists v_h \in V_h \mid < v_h, \lambda_h >_\Gamma \geq \|v_h\|_{V_h} \|\lambda_h\|_{Q_h}$
- $\forall u_h \in V_h \mid P_{Q_h}(u_h) = 0 \quad \text{then } \|\nabla u_h\|^2_{\mathrm{L}^2(\Omega)} \geq \beta \|u_h\|^2_{H^1(\Omega)}$

where $P_{Q_h}(u_h)$ denotes the projection of u_h onto the Lagrange multiplier space Q_h. This compatibility conditions are satisfied by equal interpolation spaces

Stabilized Lagrange multipliers

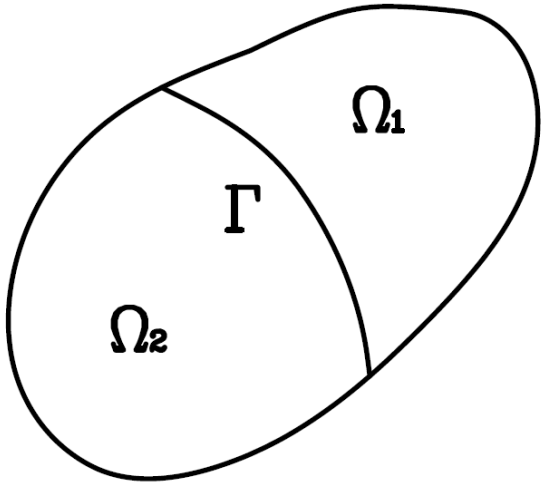
A stabilized formulation which allows arbitrary interpolation spaces can be obtained by adding stabilization terms:

$$k(\nabla v_h, \nabla u_h) - < v_h, \lambda_h >_\Gamma + \alpha < v_h, \lambda_h - k\boldsymbol{n} \cdot \nabla u_h >_\Gamma = < v_h, f > \qquad \forall v_h \in V_h$$
$$\alpha < \mu_h, \lambda_h - k\boldsymbol{n} \cdot \nabla u_h >_\Gamma + < \mu, u_h >_\Gamma = < \mu, \bar{u} >_\Gamma \qquad \forall \mu_h \in Q_h$$

MONOLITHIC APPROACH

Let us consider a time dependent coupled problem, involving two physical subdomains Ω_1 and Ω_2:

$$\partial_t u_1 + \mathcal{L}_1 u_1 = f_1$$
$$\partial_t u_2 + \mathcal{L}_2 u_2 = f_2$$



One or both of the problems might involve a second order time derivative:

$$\partial_{tt}^2 \hat{u}_2 + \hat{\mathcal{L}}_2 \hat{u}_2 = \hat{f}_2$$

This is the case of solid mechanics problems, where the operator $\hat{\mathcal{L}}_2$ depends on the displacement field, while the momenum balance equation depends on the accelerations.

MONOLITHIC APPROACH

Examples:

- Transient heat transfer equation: find u_1 in $\Omega_1 \times (0,T)$:

$$\begin{aligned} \partial_t u_1 - k\Delta u_1 &= f && \text{in } \Omega_1 \\ u_1 &= \bar{u} && \text{on } \Gamma_D \\ k\boldsymbol{n}\cdot\nabla u_1 &= \bar{t} && \text{on } \Gamma_N \\ u_1 &= u_2 && \text{on } \Gamma \\ k\boldsymbol{n}\cdot\nabla u_1 &= k\boldsymbol{n}\cdot\nabla \boldsymbol{u_2} && \text{on } \Gamma \\ u_1(\boldsymbol{x},0) &= u_{10}(\boldsymbol{x}) && \text{in } \Omega_1 \end{aligned}$$

- Dynamic solid mechanics problems: $\boldsymbol{u_1}, \rho_1$ in $\Omega_1 \times (0,T)$:

$$\begin{aligned} \rho_1 d_{tt}^2 \boldsymbol{u}_1 - \nabla\cdot\boldsymbol{\sigma}_1 &= \boldsymbol{f} && \text{in } \Omega_1 \\ \rho_1 &= \rho_{10} J && \text{in } \Omega_1 \end{aligned}$$

$$J = \det|\boldsymbol{F}| \qquad \boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}$$

$$\begin{aligned} \boldsymbol{u}_1 &= \bar{\boldsymbol{u}} && \text{on } \Gamma_D \\ \boldsymbol{n}\cdot\boldsymbol{\sigma_1} &= \bar{t} && \text{on } \Gamma_N \\ \boldsymbol{u}_1 &= \boldsymbol{u}_2 && \text{on } \Gamma \\ \boldsymbol{n}\cdot\boldsymbol{\sigma_1} &= \boldsymbol{n}\cdot\boldsymbol{\sigma_1} && \text{on } \Gamma \\ \boldsymbol{u}_1(\boldsymbol{x},0) &= \boldsymbol{u}_{10}(\boldsymbol{x}) && \text{in } \Omega_1 \\ \rho_1(\boldsymbol{x},0) &= \rho_{10}(\boldsymbol{x}) && \text{in } \Omega_1 \end{aligned}$$

MONOLITHIC APPROACH

Examples:

- Incompressible Navier-Stokes equations: find $\boldsymbol{u_1}, p_1$ in $\Omega_1 \times (0, T)$:

$$
\begin{aligned}
\partial_t \boldsymbol{u}_1 + \boldsymbol{u}_1 \cdot \nabla \boldsymbol{u}_1 - \nu \Delta \boldsymbol{u}_1 + \nabla p_1 &= \boldsymbol{f} && \text{in } \Omega_1 \\
\nabla \cdot \boldsymbol{u}_1 &= 0 && \text{in } \Omega_1 \\
\boldsymbol{u}_1 &= \bar{\boldsymbol{u}} && \text{on } \Gamma_{D1} \\
\boldsymbol{n} \cdot \boldsymbol{\sigma}_1 &= \bar{\boldsymbol{t}} && \text{on } \Gamma_{N1} \\
\boldsymbol{u}_1 &= \boldsymbol{u}_2 && \text{on } \Gamma \\
\boldsymbol{n} \cdot \boldsymbol{\sigma}_1 &= \boldsymbol{n} \cdot \boldsymbol{\sigma}_2 && \text{on } \Gamma \\
\boldsymbol{u}_1(\boldsymbol{x}, 0) &= \boldsymbol{u}_{10} && \text{in } \Omega_1
\end{aligned}
$$

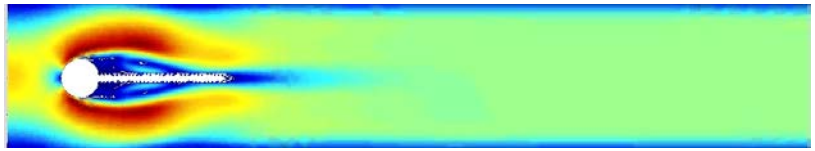
The problems we are solving do not need to be homogeneous in both subdomains (but they can):

- Fluid-Structure Interaction problems
- Incompressible flow – Darcy flow
- The same problem but different materials:
  - Structure-structure problem

MONOLITHIC APPROACH

The monolithic approach is the most straightforward one.

Let us suppose that we want to solve a problem involving two continuum medium (for instance a fluid-structure interaction problem):



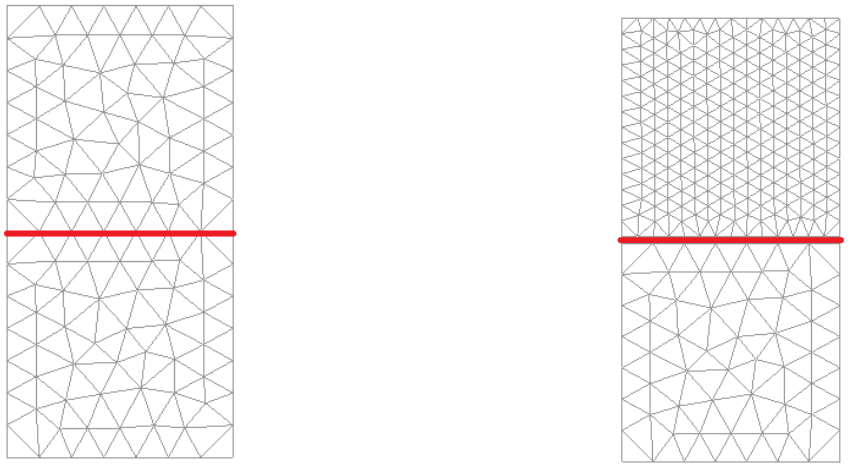
Transmission conditions consist of the continuity of velocities and the normal component of the stresses at the interface:

$$\boldsymbol{u_1} = \boldsymbol{u_2} \qquad \text{on } \Gamma$$
$$\boldsymbol{\sigma}_1 \cdot \boldsymbol{n_1} + \boldsymbol{\sigma}_2 \cdot \boldsymbol{n_2} = \mathbf{0} \qquad \text{on } \Gamma$$

The first equation is usually imposed in a strong manner, while the second one is enforced in a weak sense.

MONOLITHIC APPROACH

The finite element meshes for subdomains 1 and 2, can be matching and non-matching.



- If the meshes are matching, the imposition of the transmission conditions is straightforward.

- For non-matching grids, on the other hand, we need to explicitly enforce them.

MONOLITHIC APPROACH

Let us consider the finite element weak form associated to a general continuum mechanics problem:

$$(\boldsymbol{v}_h, \partial_t \boldsymbol{u}_h)_\Omega + (\nabla \boldsymbol{v}_h, \boldsymbol{\sigma}_h)_\Omega - (\boldsymbol{v}_h, \boldsymbol{n} \cdot \boldsymbol{\sigma}_h)_\Gamma = (\boldsymbol{v}_h, \boldsymbol{f})_\Omega$$
$$\boldsymbol{u}_h = \bar{\boldsymbol{u}} \qquad \text{on } \Gamma$$

where $(\cdot,\cdot)_\Omega$ denotes the L^2-inner product over the domain and $(\cdot,\cdot)_\Gamma$ denotes the product over the boundary. $\boldsymbol{v}_h$ denotes the velocity test functions, each test function corresponding to one node of the finite element mesh.

MONOLITHIC APPROACH

Let us consider a Neumann (1) – Neumann (2) approach.

- The equations for subdomain 1 are (general, before coupling):

$$(\boldsymbol{v}_{1h}, \partial_t \boldsymbol{u}_{1h})_{\Omega^1} + \left(\nabla \boldsymbol{v}_{1h}, \sigma(\boldsymbol{u}_{1h})\right)_{\Omega^1} - < \boldsymbol{v}_{1h}, \boldsymbol{n} \cdot \boldsymbol{\sigma}(\boldsymbol{u}_{1h}) >_{\Gamma_{\text{interface}}} = < \boldsymbol{v}_{1h}, \rho \boldsymbol{b} >_{\Omega^1}$$

- On the other hand, we have the equation for subdomain 2. Here we use the fact that:

$$\boldsymbol{n_2} \cdot \boldsymbol{\sigma}(\boldsymbol{u}_{2h}) + \boldsymbol{n_1} \cdot \boldsymbol{\sigma}(\boldsymbol{u_{1h}}) = \boldsymbol{0}$$

$$(\boldsymbol{v_{2h}}, \partial_t \boldsymbol{u}_{2h})_{\Omega^2} + (\nabla \boldsymbol{v_{2h}}, \boldsymbol{\sigma}(\boldsymbol{u}_{2h}))_{\Omega^2} - < \boldsymbol{v_{2h}}, \boldsymbol{n_2} \cdot \boldsymbol{\sigma}(\boldsymbol{u_{1h}}) >_{\Gamma_{\text{interface}}} = < \boldsymbol{v}_{2h}, \rho \boldsymbol{b} >_{\Omega^2}$$

If the meshes match at the interface and we are using the same interpolation space V_h for $\boldsymbol{u}_h^1$ and $\boldsymbol{u}_h^2$:

$$- < \boldsymbol{v_{2h}}, \boldsymbol{n_2} \cdot \boldsymbol{\sigma}(\boldsymbol{u_{1h}}) >_{\Gamma_{\text{interface}}} = < \boldsymbol{v_{1h}}, \boldsymbol{n_1} \cdot \boldsymbol{\sigma}(\boldsymbol{u_{1h}}) >_{\Gamma_{\text{interface}}} =$$
$$(\boldsymbol{v_{1h}}, \partial_t \boldsymbol{u_{1h}})_{\Omega^1} + (\nabla \boldsymbol{v_{1h}}, \boldsymbol{\sigma}(\boldsymbol{u_{1h}}))_{\Omega^1} - < \boldsymbol{v_{1h}}, \rho \boldsymbol{b} >_{\Omega^1}$$

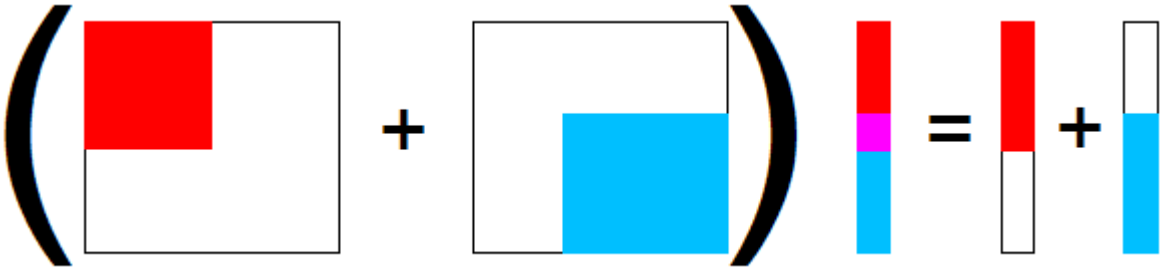
MONOLITHIC APPROACH

So we can write the problem for the second domain as:

$$(\boldsymbol{v_{2h}}, \partial_t \boldsymbol{u_{2h}})_{\Omega^2} + (\nabla \boldsymbol{v_{2h}}, \sigma(\boldsymbol{u_{2h}}))_{\Omega^2} + (\boldsymbol{v_{1h}}, \partial_t \boldsymbol{u_{1h}})_{\Omega^1} + (\nabla \boldsymbol{v_{1h}}, \boldsymbol{\sigma}(\boldsymbol{u_{1h}}))_{\Omega^1} = < \boldsymbol{v_{2h}}, \rho \boldsymbol{b} >_{\Omega^2} + < \boldsymbol{v_{1h}}, \rho \boldsymbol{b} >_{\Omega^1}$$

The equations for the first domain are exactly the same.

- There is no need to integrate boundary terms.
- We can simply assemble the global system and use the unknowns at the interface to account for both velocity values.



MONOLITHIC APPROACH

When the grids are non-matching, these transmission conditions can be enforced in three different manners:

- Dirichlet (1) – Neumann (2)

  We enforce

  $$\boldsymbol{u}_1 = \boldsymbol{u}_2 \qquad \text{on } \Gamma$$

  in the equations for subdomain 1 (Dirichlet boundary condition).

  We enforce

  $$\boldsymbol{\sigma}_1 \cdot \boldsymbol{n} = \boldsymbol{\sigma}_2 \cdot \boldsymbol{n} \qquad \text{on } \Gamma$$

  in the equations for subdomain 2 (Neumann boundary condition).

- Neumann (subdomain 2) – Dirichlet (subdomain 1)

- Robin (subdomain 1) – Robin (subdomain 2)

  We enforce:

  $$\alpha(\boldsymbol{u}_1 - \boldsymbol{u}_2) + \beta(\boldsymbol{\sigma}_1 \cdot \boldsymbol{n} - \boldsymbol{\sigma}_2 \cdot \boldsymbol{n}) = 0 \qquad \text{on } \Gamma$$

  in the equations for both subdomains.

  We recall that we need to use different coefficients $\alpha_1, \beta_1, \alpha_2, \beta_2$ so that we ensure that the equations in both subdomains are linearly independent.

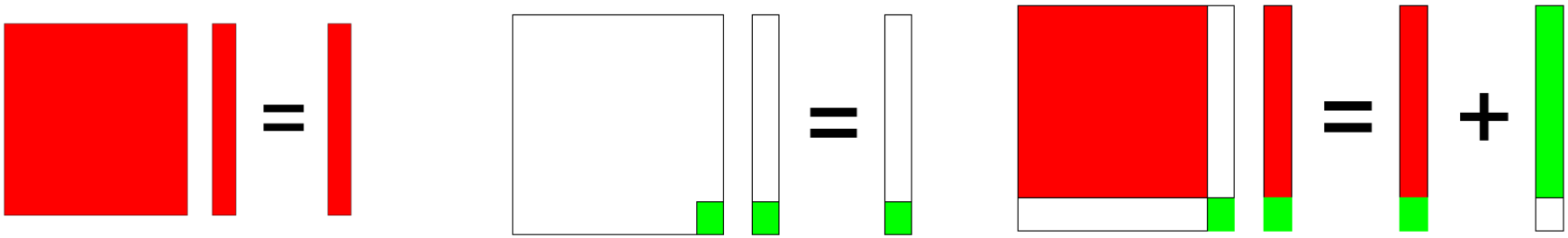
MONOLITHIC APPROACH

Dirichlet boundary conditions on a finite element discretization

We want to enforce Dirichlet boundary conditions on the weak form:

$$(\boldsymbol{v}_h, \partial_t \boldsymbol{u}_h)_\Omega + (\nabla \boldsymbol{v}_h, \boldsymbol{\sigma}_h)_\Omega - (v_h, \boldsymbol{n} \cdot \boldsymbol{\sigma}_h)_\Gamma = (\boldsymbol{v}_h, \boldsymbol{f})_\Omega$$
$$\boldsymbol{u}_h = \bar{\boldsymbol{u}} \qquad \text{on } \Gamma$$

Strong imposition of Dirichlet boundary conditions in a finite element problem:
Original system without b.c. Equations enforcing D.B.C Final system



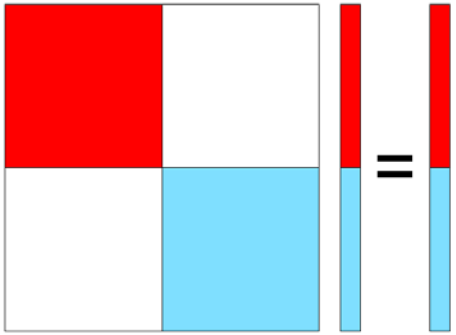
Enforcement can be node to node (matching) or through interpolation.

MONOLITHIC APPROACH

Enforcement of velocity Dirichlet transmission conditions

But now, we need to enforce the velocity of the continuum medium 1 to be equal to the velocity of continuum medium 2 at the interface.

We now represent the global system of equations (which involves the equations for both subdomains), prior to the application of the coupling conditions:



Note that both systems are uncoupled, there is no influence from one to the other. If the systems were well posed, they could be solved independently.

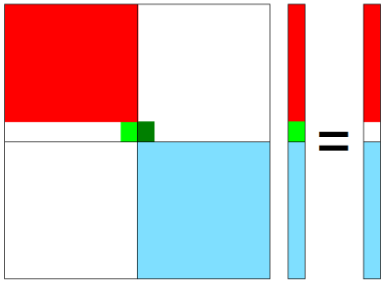
MONOLITHIC APPROACH

Enforcement of velocity Dirichlet transmission conditions

In order to enforce the continuity of velocities (Dirichlet equations):

$$\boldsymbol{u}_1 = \boldsymbol{u}_2 \qquad \text{on } \Gamma$$

We replace the equations for subdomain 1 corresponding to nodes on the interface with the Dirichlet equations:



- We have used the subdomain 1 nodes at the interface to enforce the continuity of velocities.
- Only the equations for those nodes have been modified.
- The subdomain 1 system is no longer symmetric.
- Subdomain 2 problem "knows" nothing about what is going on in subdomain 1 (stress transmission conditions are missing).

MONOLITHIC APPROACH

Neumann boundary conditions on finite elements

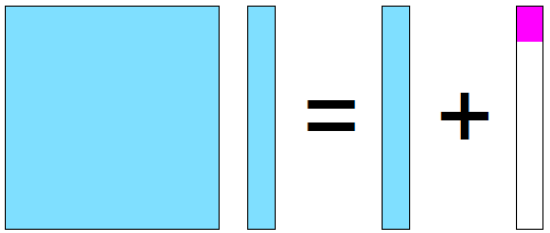
In standard interpolation approaches for solid and fluid mechanics finite elements, stress transmission conditions are enforced weakly.

$$(\boldsymbol{v}_h, \partial_t \boldsymbol{u}_h)_\Omega + (\boldsymbol{v}_h, \boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h) + (\nabla \boldsymbol{v}_h, \boldsymbol{\sigma}_h) - (\boldsymbol{v}_h, \boldsymbol{n} \cdot \sigma_h)_\Gamma = (\boldsymbol{v}_h, \boldsymbol{f})_\Omega$$

The terms in blue correspond to the integral of the tractions. In order to enforce Neumann boundary conditions we simply need to replace these terms by the known traction:

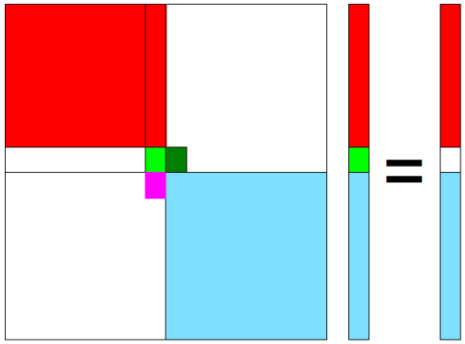
$$(\boldsymbol{v}_h, \partial_t \boldsymbol{u}_h)_\Omega + (\boldsymbol{v}_h, \boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h) + (\nabla \boldsymbol{v}_h, \boldsymbol{\sigma}_h) = (\boldsymbol{v}_h, \boldsymbol{f})_\Omega + (\boldsymbol{v}_h, \boldsymbol{t})_\Gamma$$

For a general continuum mechanics problem, this can be done by adding an additional term to the RHS:



MONOLITHIC APPROACH

Final coupled Monolithic system for a Neumann (2) – Dirichlet (1) coupling scheme



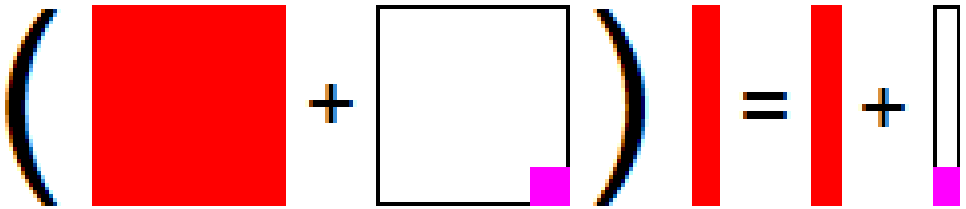
- The system corresponding to subdomain 2 has not been modified.
- Coupling with the subdomain 1 is through the transmitted tractions (magenta terms).
- Dirichlet boundary conditions for subdomain 1 problem have been taken into account so that velocities match at the interface (green terms).
- This can obviously done the other way around: enforce Dirichlet to subdomain 2 and Neumann conditions to subdomain 1.
- The final system is not symmetric.

MONOLITHIC APPROACH

Robin – Robin coupling boundary conditions

When using Robin boundary conditions, both the velocity and traction conditions are enforced in a weak sense. For a general continuum mechanics problem, the finite element weak form with Robin-Robin boundary conditions is:

$$(\boldsymbol{v}_h, \partial_t \boldsymbol{u}_h)_\Omega + (\nabla \boldsymbol{v}_h, \boldsymbol{\sigma}_h)_\Omega - (\boldsymbol{v}_h, \boldsymbol{n} \cdot \boldsymbol{\sigma}_h)_\Gamma + \alpha(\boldsymbol{v}_h, \boldsymbol{u}_h - \bar{\boldsymbol{u}})_\Gamma + \beta(\boldsymbol{v}_h, \boldsymbol{\sigma}_h \cdot \boldsymbol{n} - \boldsymbol{t})_\Gamma = (\boldsymbol{v}_h, \boldsymbol{f})_\Omega$$



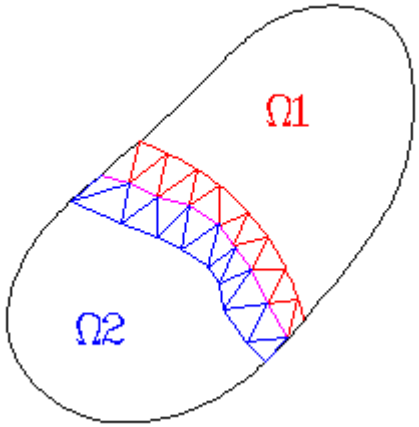
MONOLITHIC APPROACH

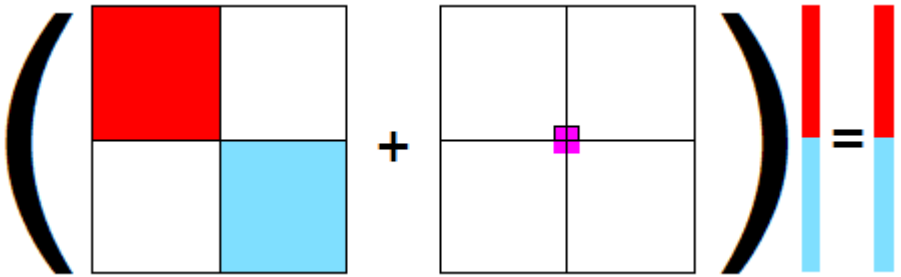
Robin – Robin coupling boundary conditions

The final coupled problem is:

$$(\boldsymbol{v}_{1h}, \partial_t \boldsymbol{u}_{1h})_{\Omega^1} + (\nabla \boldsymbol{v}_{1h}, \boldsymbol{\sigma}_{1h})_{\Omega^1} - (\boldsymbol{v}_{1h}, \boldsymbol{n} \cdot \boldsymbol{\sigma}_{1h})_{\Gamma} + \alpha_1 (\boldsymbol{v}_{1h}, \boldsymbol{u}_{1h} - \boldsymbol{u}_{2h})_{\Gamma} + \beta_1 (\boldsymbol{v}_{1h}, \boldsymbol{\sigma}_{2h} \cdot \boldsymbol{n} - \boldsymbol{\sigma}_{1h} \cdot \boldsymbol{n})_{\Gamma} = (\boldsymbol{v}_{1h}, \boldsymbol{f})_{\Omega^1}$$

$$(\boldsymbol{v}_{2h}, \partial_t \boldsymbol{u}_{2h})_{\Omega^2} + (\nabla \boldsymbol{v}_{2h}, \boldsymbol{\sigma}_{2h})_{\Omega^2} - (\boldsymbol{v}_{2h}, \boldsymbol{n} \cdot \boldsymbol{\sigma}_{2h})_{\Gamma} + \alpha_2 (\boldsymbol{v}_{2h}, \boldsymbol{u}_{1h} - \boldsymbol{u}_{2h})_{\Gamma} + \beta_2 (\boldsymbol{v}_{2h}, \boldsymbol{\sigma}_{2h} \cdot \boldsymbol{n} - \boldsymbol{\sigma}_{1h} \cdot \boldsymbol{n})_{\Gamma} = (\boldsymbol{v}_{2h}, \boldsymbol{f})_{\Omega^2}$$





MONOLITHIC APPROACH

Robin – Robin coupling boundary conditions

Advantages of Robin-Robin Coupling boundary conditions

- The added terms are symmetric.
- There is no need to modify the original systems with coupling conditions: coupling conditions can be added to the system a posteriori (modularity).

Disadvantages of Robin-Robin Coupling conditions

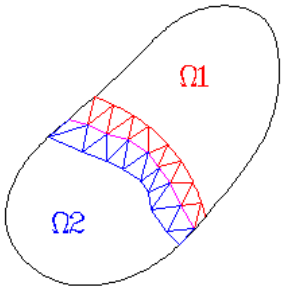
- Coefficients α and β need to be chosen, which is not immediate:
  - If the coefficients are too large, the system becomes ill-conditioned.
  - If the coefficients are too small, boundary conditions are not enforced strongly enough.

MONOLITHIC APPROACH

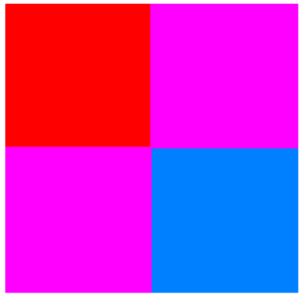
Two physics, one domain

Until now we are considering problems in which each coupled problem takes place in one subdomain.

Transmission conditions occur only at the interface.



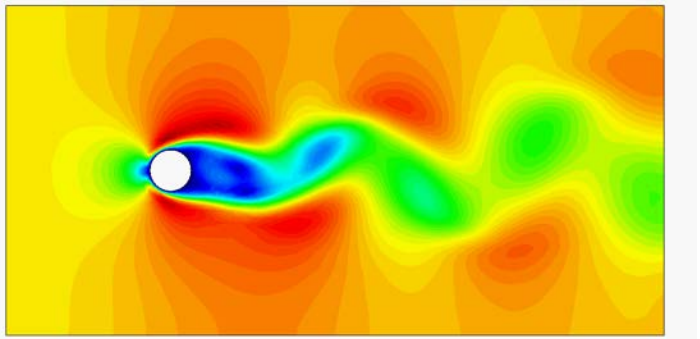
It is also possible to have two coupled physical problems which take place at the same physical domain.
In this case the coupling is not only through values at the interface, but in the whole extension of the computational domain.

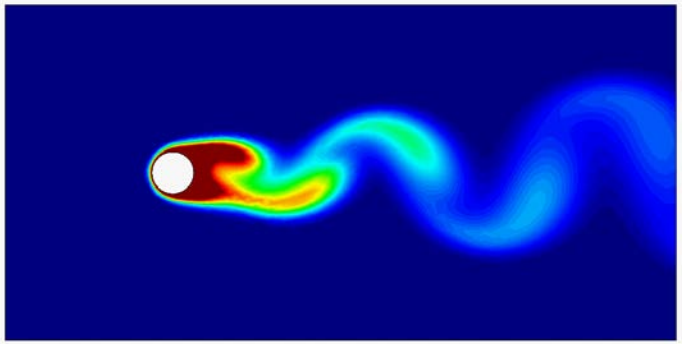


MONOLITHIC APPROACH

Thermally coupled flows

In thermally coupled flows, both physical problems take place in the same domain.





Fluid equations + Boussinesq coupling:

$$(\boldsymbol{v}_h, \partial_t \boldsymbol{u}_h)_\Omega + (\boldsymbol{v}_h, \boldsymbol{u}_h \cdot \nabla \boldsymbol{u}_h) + \nu(\nabla \boldsymbol{v}_h, \nabla \boldsymbol{u}_h)_\Omega - (\nabla \cdot \boldsymbol{v}_h, p_h)_\Omega + (\boldsymbol{v}_h, \boldsymbol{n} \cdot p_h)_\Gamma - \nu(\boldsymbol{v}_h, \boldsymbol{n} \cdot \nabla \boldsymbol{u}_h)_\Gamma = (\boldsymbol{v}_h, \boldsymbol{f})_\Omega + (\boldsymbol{v}_h, \alpha \boldsymbol{g} \Delta\theta)$$

$$(q_h, \nabla \cdot \boldsymbol{u}_h)_\Omega = 0$$

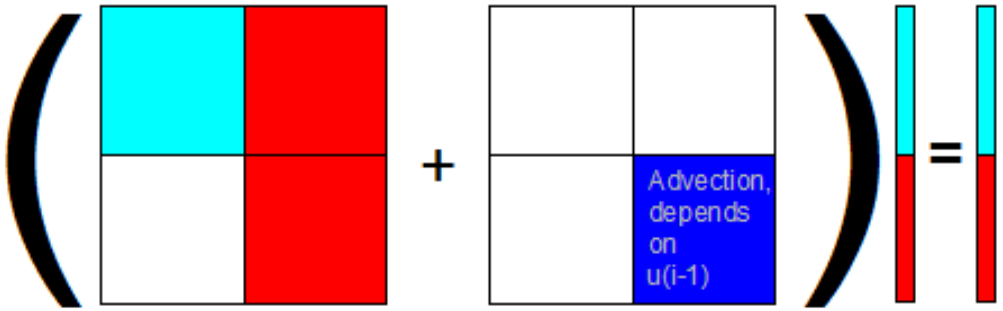
Temperature equation (with advection):

$$(w_h, \partial_t \theta_h)_\Omega + (w_h, \boldsymbol{u}_h \cdot \nabla \theta_h) + \kappa(\nabla w_h, \nabla \theta_h)_\Omega + -\kappa(w_h, n \cdot \nabla \theta_h)_\Gamma = (w_h, f)_\Omega$$

MONOLITHIC APPROACH

Thermally coupled flows

In this case the coupling term for the temperature equation is non-linear.



This makes it impossible to solve the global system in a single step.

It is reasonable to start by solving the temperature equation, and then use the resolved temperature to compute the Boussinesq body forces on the fluid.

MONOLITHIC APPROACH

Advantages of Monolithic approaches

- We solve all the coupled problems in a single step.
- No convergence or iterative scheme issues.
- Boundary conditions are applied implicitly in time, the final solution automatically satisfies them.

Disadvantages of Monolitihic approaches

- The system of equations to be solved is larger.
- The system of equations to be solved can be ill-conditioned (involves different physical problems, with different consitutive laws and physical parameters).
- Modularity is difficult to attain: equations for each problem need to be assembled in a global matrix. We cannot couple independent pieces of software.

PARTITIONED SCHEMES

Motivations for a partitioned scheme

- **Customization:** Each field can be treated with algorithms which are known to perform well for the isolated system.
- **Independent modelling:** Partitioned schemes facilitate the use of non-matching meshes.
- **Software reuse:** Independent pieces of sofware can be used to deal with each of the problems.
- **Modularity:** Each physical problem can be packed in a software module, all implementations for the particular problem are localized in its module.

PARTITIONED SCHEMES

Let us suppose that we have a time-dependent coupled problem to be solved, which we have already discretized in time:

$$\boldsymbol{A}_X \boldsymbol{X}^{n+1} + \boldsymbol{B}_X \boldsymbol{Y}^{n+1} = \boldsymbol{F}_X + \boldsymbol{C}_X \boldsymbol{X}^n$$
$$\boldsymbol{A}_Y \boldsymbol{Y}^{n+1} + \boldsymbol{B}_Y \boldsymbol{X}^{n+1} = \boldsymbol{F}_Y + \boldsymbol{C}_Y \boldsymbol{Y}^n$$

In the monolithic approaches the previous system is solved altogether. On partitioned approaches, we want to iteratively solve systems for only ***X*** or ***Y***.

PARTITIONED SCHEMES

Explicit coupling

- Staggered Approach

We send the off-diagonal terms to the RHS through a prediction:

$$\boldsymbol{A}_X \boldsymbol{X}^{n+1} = \boldsymbol{F}_X + \boldsymbol{C}_X \boldsymbol{X}^n - \boldsymbol{B}_X \tilde{\boldsymbol{Y}}^{n+1}$$
$$\boldsymbol{A}_Y \boldsymbol{Y}^{n+1} = \boldsymbol{F}_Y + \boldsymbol{C}_Y \boldsymbol{Y}^n - \boldsymbol{B}_Y \tilde{\boldsymbol{X}}^{n+1}$$

Variables $\tilde{\boldsymbol{Y}}^{n+1}$ and $\tilde{\boldsymbol{X}}^{n+1}$ are the predicted values.

1st order approximation: $\tilde{\boldsymbol{Y}}^{n+1} = \boldsymbol{Y}^n$ 2nd order Taylor expansion: $\tilde{\boldsymbol{Y}}^{n+1} = 2\boldsymbol{Y}^n - \boldsymbol{Y}^{n-1}$

- Substitution

Another possibility consists in solving first for ***X***, and using the resolved variable for the solution of the ***Y*** equations:

First we solve for $\boldsymbol{X}^{n+1}$:

$$\boldsymbol{A}_X \boldsymbol{X}^{n+1} = \boldsymbol{F}_X + \boldsymbol{C}_X \boldsymbol{X}^n - \boldsymbol{B}_X \tilde{\boldsymbol{Y}}^{n+1}$$

Now we use $\boldsymbol{X}^{n+1}$ in the equation for ***Y***:

$$\boldsymbol{A}_Y \boldsymbol{Y}^{n+1} = \boldsymbol{F}_Y + \boldsymbol{C}_Y \boldsymbol{Y}^n - \boldsymbol{B}_Y \boldsymbol{X}^{n+1}$$

PARTITIONED SCHEMES

Iteration

The previous procedure can be iterated (with iteration counter *i*) until convergence. If we reach convergence, we recover the solution of the monolithic problem.

While convergence is not reached, iterate (i):

$$\boldsymbol{A}_X \boldsymbol{X}^{n+1,i} = \boldsymbol{F}_X + \boldsymbol{C}_X \boldsymbol{X}^n - \boldsymbol{B}_X \boldsymbol{Y}^{n+1,i-1}$$
$$\boldsymbol{A}_Y \boldsymbol{Y}^{n+1,i} = \boldsymbol{F}_Y + \boldsymbol{C}_Y \boldsymbol{Y}^n - \boldsymbol{B}_Y \boldsymbol{X}^{n+1,i}$$

Convergence or stability of any of these schemes is not guaranteed!

PARTITIONED SCHEMES

Stability for a simplified substitution scheme

Let us consider the algebraic form of a coupled heat transfer problem, discretized with a backward Euler scheme:

$$\begin{bmatrix} \frac{\boldsymbol{M}_{XX}}{\delta t} + \boldsymbol{K}_{XX} & \boldsymbol{K}_{XY} \\ \boldsymbol{K}_{YX} & \frac{\boldsymbol{M}_{YY}}{\delta t} + K_{YY} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_X^{n+1} \\ \boldsymbol{U}_Y^{n+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_X^{n+1} \\ \boldsymbol{F}_Y^{n+1} \end{bmatrix} + \begin{bmatrix} \frac{\boldsymbol{M}_{XX}}{\delta t} \boldsymbol{U}_X^n \\ \frac{\boldsymbol{M}_{YY}}{\delta t} \boldsymbol{U}_Y^n \end{bmatrix}$$

Let us simplify this problem by considering:

a. δt is small

b. $\boldsymbol{F} = 0$

$$\begin{bmatrix} \frac{\boldsymbol{M}_{XX}}{\delta t} & \boldsymbol{K}_{XY} \\ \boldsymbol{K}_{YX} & \frac{\boldsymbol{M}_{YY}}{\delta t} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_X^{n+1} \\ \boldsymbol{U}_Y^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{\boldsymbol{M}_{XX}}{\delta t} \boldsymbol{U}_X^n \\ \frac{\boldsymbol{M}_{YY}}{\delta t} \boldsymbol{U}_Y^n \end{bmatrix}$$

PARTITIONED SCHEMES

Stability for a simplified substitution scheme

The associated substitution scheme is:

$$\frac{\boldsymbol{M}_{XX}}{\delta t}\boldsymbol{U}_X^{n+1} = \frac{\boldsymbol{M}_{XX}}{\delta t}\boldsymbol{U}_X^{n} - \boldsymbol{K}_{XY}\boldsymbol{U}_Y^{n}$$
$$\frac{\boldsymbol{M}_{YY}}{\delta t}\boldsymbol{U}_Y^{n+1} = \frac{\boldsymbol{M}_{YY}}{\delta t}\boldsymbol{U}_Y^{n} - \boldsymbol{K}_{YX}\boldsymbol{U}_X^{n+1}$$

Which yields:

$$\boldsymbol{U}_Y^{n+1} = \boldsymbol{U}_Y^{n} - \delta t\boldsymbol{M}_{YY}^{-1}\boldsymbol{K}_{XY}\boldsymbol{U}_X^{n} - \delta t^2\boldsymbol{M}_{YY}^{-1}\boldsymbol{K}_{XY}\boldsymbol{M}_{XX}^{-1}\boldsymbol{K}_{YX}\boldsymbol{U}_Y^{n}$$

Discarding the δt^2 terms, and making the approximation (valid in Neumann-Neumann and grid-matching coupling schemes):

$$\boldsymbol{K}_{YX}\boldsymbol{U}_X^{n} \approx \boldsymbol{K}_{XY}\boldsymbol{U}_Y^{n}$$

We have:

$$\boldsymbol{U}_Y^{n+1} = \boldsymbol{U}_Y^{n}(\boldsymbol{I} - \delta t\boldsymbol{M}_{YY}^{-1}\boldsymbol{K}_{XY})$$

This scheme is only stable if $\left|\boldsymbol{I} - \delta t\boldsymbol{M}_{YY}^{-1}\boldsymbol{K}_{XY}\right| \leq 1$

PARTITIONED SCHEMES

Substepping

If both problems have different scales in time, it can be convenient to use different time steps for each of the coupled problems.

This is specially important if we want to use explicit time integration schemes. The limitation in the time step due to stability reasons will define the maximum time step size for each problem.

We solve for $\boldsymbol{X}^{n+1}$ with a prediction for $\widetilde{\boldsymbol{Y}}^{n+1}$:

$$\boldsymbol{A}_X \boldsymbol{X}^{n+1} = \boldsymbol{F}_X + \boldsymbol{C}_X \boldsymbol{X}^n - \boldsymbol{B}_X \widetilde{\boldsymbol{Y}}^{n+1}$$

Then we solve for the substeps of $\boldsymbol{Y}$ using an interpolation $\boldsymbol{X}$ at the given substep:

$$\boldsymbol{A}_Y \boldsymbol{Y}^{n+i\alpha} = \boldsymbol{F}_Y + \boldsymbol{C}_Y \boldsymbol{Y}^{n+(i-1)\alpha} - \boldsymbol{B}_Y (\boldsymbol{X}^{n+1}(i\alpha) + \boldsymbol{X}^n(1-i\alpha))$$