Modeling, Discretization, Optimization, and Simulation of Multiphysics Problems (IIT Indore)

Thomas WICK

thomas.wick@ifam.uni-hannover.de
www.ifam.uni-hannover.de
www.thomaswick.org

AG Wissenschaftliches Rechnen
Institut für Angewandte Mathematik

Leibniz Universität Hannover

Feb 11-12, 2020
Indo-German Winter School, IIT Indore, India

Preface

This class is designed for students and peers who had basic classes in calculus, introduction to numerical methods, differential equations, and finite element discretizations.

The **goal** is to give an introduction to **numerical modeling of multiphysics problems**. These are **nonstationary**, **nonlinear**, **coupled** **partial differential equations**.

This class is organized into five parts:

- Modeling fluid flows and solid mechanics
- Variational formulations and coupling techniques
- Discretization in time and space
- Nonlinear and linear solution
- Numerical simulations and further extensions.

Philosophy of this class: Mixture of very basic techniques that are immediately applied to 'complicated' practical and/or current research problems.

Thomas Wick
(IIT Indore, Feb 2020)

Preliminaries

Schedule

- Class 1: Feb 11: 2pm - 3pm
- Class 2: Feb 11: 3.15pm - 4.15pm
- Class 3: Feb 12: 10am - 11am
- Class 4: Feb 12: 11.15am - 12.15pm
- Class 5: Feb 12: 2pm - 3pm

Literature

Materials presented in this school class are largely based on the following lecture notes:

1. T. Wick; **Numerical methods for partial differential equations**; Hannover: Institutionelles Repositorium der Leibniz Universität Hannover, 2020,
   https://doi.org/10.15488/9248
   Specifically therein: Chapter 8, 13 and 14.

2. T. Wick; **Modeling, Discretization, Optimization, and Simulation of Fluid-Structure Interaction**; TU Munich, 2016,
   Link to TUM-FSI notes

   updated version, Nov 2018:
   Link to LUH-FSI notes (*.pdf)

3. J. Goldenstein, T. Wick; **Goal-oriented a posteriori error estimation and adaptive finite elements**; Lecture notes WS 19/20, Leibniz University Hannover, Feb 6, 2020

Overview

1. Introduction: Motivations and challenges
2. Modeling fluid flows and solid mechanics (class 1)
   - Preliminaries in notation and calculus (OPTIONAL)
   - Brief introduction to continuum mechanics
   - Balance principles for modeling fluid flows and solid mechanics
3. Variational formulations and coupling techniques (class 2)
   - Design of a variational formulation
   - Classifications
   - Coupling techniques
   - Explaining variational-monolithic coupling for two Poisson equations
   - Final variational forms for fluid-structure interaction
4. Discretization in time and space (class 3)
   - Temporal discretization
   - Spatial discretization
   - Adaptivity
5. Nonlinear and linear solution (class 4)
   - Nonlinear solution
   - Linear solution
6. Numerical simulations and further extensions (class 5)
   - FSI program and software
   - Benchmarks, simulations, applications
7. Conclusions and some open questions

What are multiphysics problems?

1. Multiple physical phenomena interact with each other: for instance solids, fluids, temperature variations, chemical reactions, and so forth
2. At least two PDEs (partial differential equations) are involved
3. These may have different physical conservation properties: mass, momentum, angular momentum, energy
4. Information between different PDEs may be exchanged via coefficients, nonlinear coupling terms, right-hand sides, or interfaces

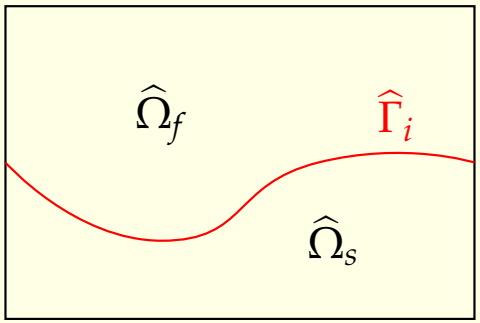
Why are multiphysics problems challenging? In other words: why are they interesting to be investigated?

1. 'True' multiphysics problems are nonlinear and nonstationary and require **extensive developments** for the design of **reliable algorithms**
2. The implementation and software development of such algorithms is tedious; debugging of code takes time! Two examples:
   - Decision whether to couple all equations (monolithic) or solve in an iterative fashion (staggered)
   - Most of the computational cost goes into the linear solver. Developing preconditioners is tedious
3. The numerical analysis of such algorithms is tedious!
4. The mathematical analysis of coupled problems is tedious!
5. Emphasis in all the previous developements should be on **physics-based discretizations**, which maintain as best as possible conservation properties after discretization

Challenges in FSI (fluid-structure interaction) as an example

1. Dealing and coupling of different classes of partial differential equations (PDEs): elliptic, parabolic, hyperbolic that require different tools for both the mathematical analysis and numerical modeling (classes 1-4);
2. Combining different coordinate systems: Eulerian and Lagrangian (class 1);
3. Nonlinearities in various equations and nonlinear coupling terms (class 2);
4. Multidomain character with interface coupling conditions (class 2);

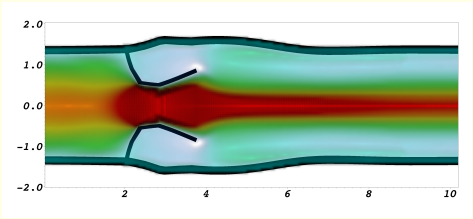
$\widehat{\Omega}_f$: fluid domain
$\widehat{\Omega}_s$: solid domain
$\widehat{\Gamma}_i$: interface

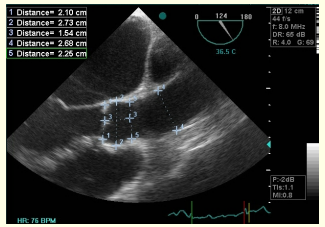


5. Moving interface $\widehat{\Gamma}_i$ requires accurate discretizations and sufficient mathematical regularity (class 2);
6. Designing accurate, robust and efficient numerical methods (monolithic, staggered, semi-explicit, ...) (classes 3-4);
7. Modeling and coupling to other physical phenomena resulting often in multiscale multiphysics problems (class 5).

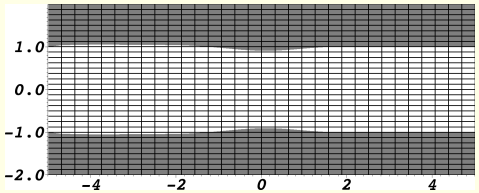
Examples I - IV (multiphysics problems)

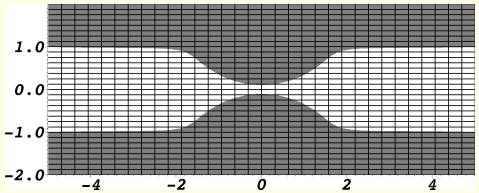
- **Valve dynamics/Flapping: fluids, elasticity**



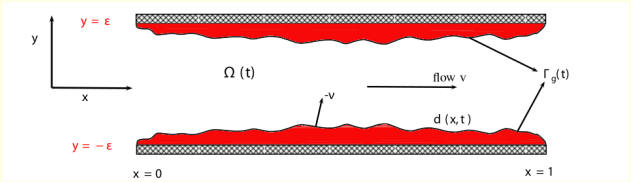


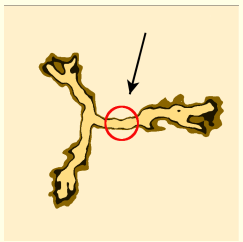
- **Solid growth and clogging: fluids, elasticity, solid growth, contact, chemical reactions**



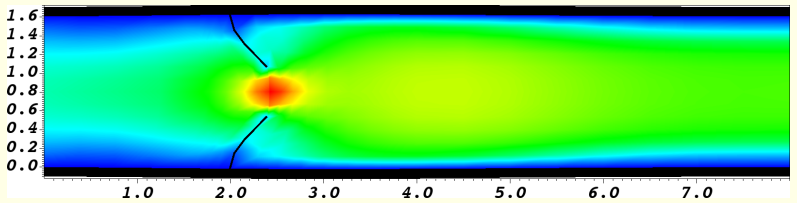


- **Reactive flow in thin channels: fluids, chemical reactions with dissolution and precipitation**



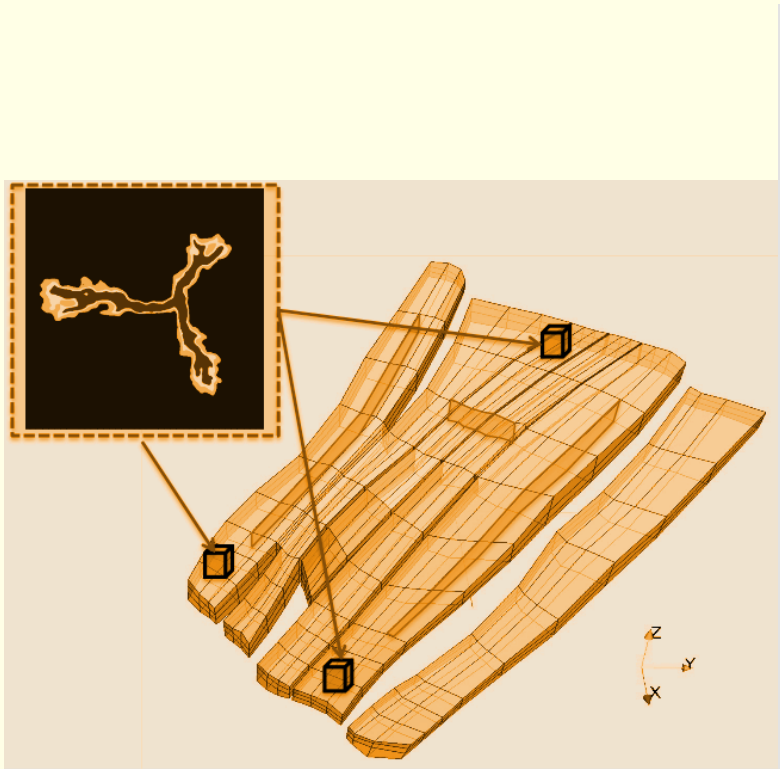


- **Wall stress minimization: fluids, elasticity, optimization**



Example V: Multiscale multiphysics in time and space [1]

- Porous media: geomechanics interacting with Darcy flow: **multiphysics**
- Fractures (possibly growing) are located inside the porous medium
- Fractures are localized: **multiscale in space**. Flow through fracture faster than in surrounding medium: **multiscale in time**



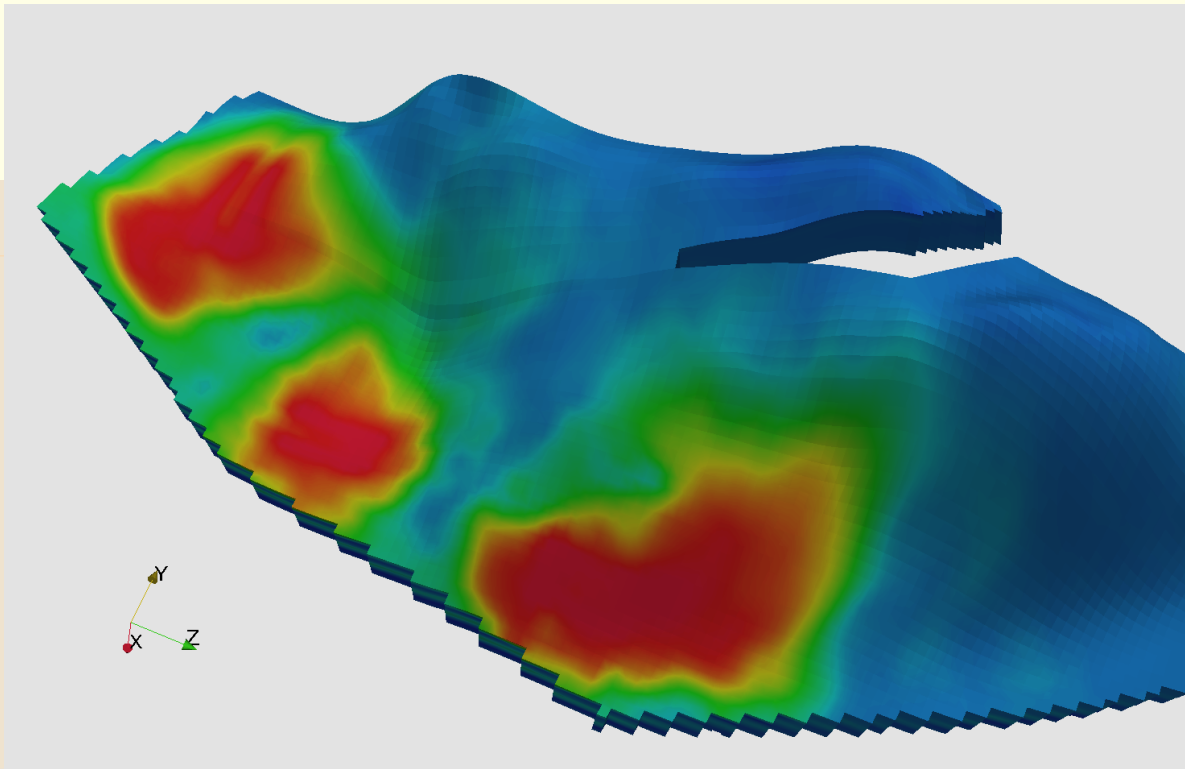


Figure: Fractures are localized phenomena in a 3d large-field porous medium.

[1] Wick/Singh/Wheeler; 2015, SPE Journal

Scientific computing/computational science

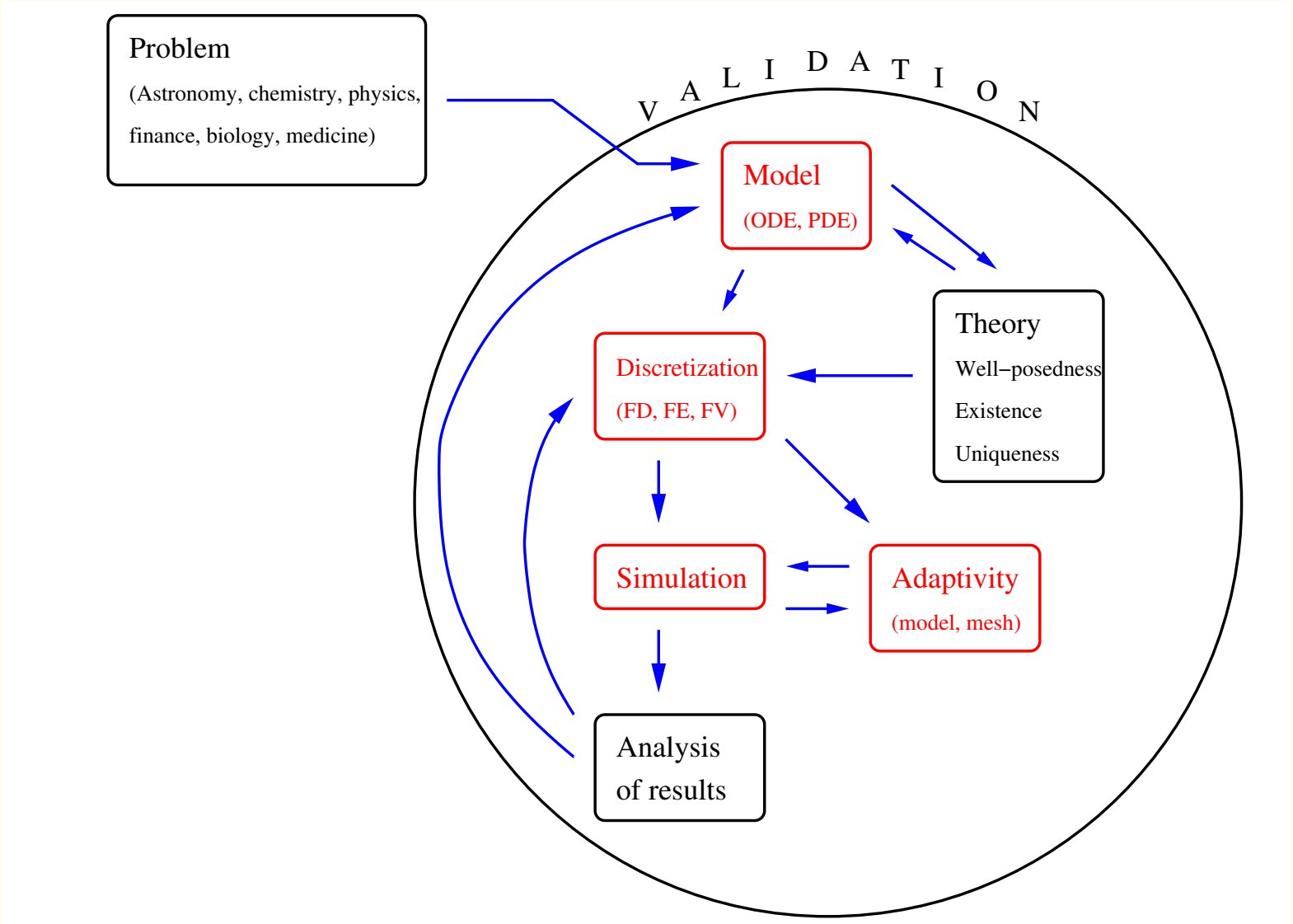


Figure: The third pillar of science between theory and experiments: scientific computing.

Class 1

Notation of functions

- A point in $\mathbb{R}^d$ is denoted by
$$x = (x_1, \ldots, x_d).$$
- The variable for 'time' is denoted by t.
- The euclidian scalar product is denoted by $(x, y) = x \cdot y = \sum_{i=1}^{d} x_i y_i$.
- In these classes, functions are often denoted by
$$u := u(x)$$
if they only depend on the spatial variable $x = (x_1, \ldots, x_d)$.
- If they depend on time and space, they are denoted by
$$u = u(t, x).$$
- Usually in physics or engineering vector-valued and tensor-valued quantities are denoted in bold font size or with the help of arrows. Unfortunately in mathematics, this notation is only sometimes adopted. We continue this crime and do not distinguish scalar, vector, and tensor-valued functions. Thus for points in $\mathbb{R}^3$ we write:
$$x := (x, y, z) = \mathbf{x} = \vec{x}.$$
Similar for functions from a space $u : \mathbb{R}^3 \supseteq U \to \mathbb{R}^3$:
$$u := (u_x, u_y, u_z) = \mathbf{u} = \vec{u}.$$

Domains

- Let $\Omega \subseteq \mathbb{R}^d, d \in \{2,3\}$ be a bounded domain with boundary $\partial\Omega$. We generally assume the boundary to be Lipschitzian [2]
- The outer unit normal vector to $\partial\Omega$ is denoted by n.
- We denote by $\Omega := \Omega(t) \subset \mathbb{R}^d$, $d = 2,3$, the domain of the fluid-structure interaction problem. This domain consists of two time-dependent subdomains $\Omega_f(t)$ and $\Omega_s(t)$. The interface between both domain is denoted by $\Gamma_i(t) = \partial\Omega_f(t) \cap \partial\Omega_s(t)$.
- The initial (or later reference) domains are denoted by $\widehat{\Omega}_f$ and $\widehat{\Omega}_s$, respectively, with the interface $\widehat{\Gamma}_i = \partial\widehat{\Omega}_f \cap \partial\widehat{\Omega}_s$.
- Furthermore, we denote the outer boundary by $\partial\widehat{\Omega} = \widehat{\Gamma} = \widehat{\Gamma}_D \cup \widehat{\Gamma}_N$ where $\widehat{\Gamma}_D$ and $\widehat{\Gamma}_N$ denote Dirichlet and Neumann boundaries, respectively.
- For the convenience of the reader and when we expect no confusion, we omit the explicit time-dependence and we use $\Omega := \Omega(t)$ to indicate time-dependent domains. Throughout these notes, we indicate with 'f' and 's' suffixes, fluid and structure related terms, respectively.
- The time interval is denoted by $I := (0,T)$ with $T > 0$ being the end time value.

[2] Precise definitions in Grisvard, 1985; Adams, 1975

Divergence theorem and integration by parts

Proposition (Gauss' divergence theorem / Gauss-Green theorem)

Let $\Omega \subset \mathbb{R}^n$ be bounded and open, and let $\partial\Omega$ of class C^1. Let $f \in C^1(\bar{\Omega})$. Then,

$$\int_\Omega div f \, dx = \int_{\partial\Omega} f \cdot n \, ds.$$

The outer normal of $\partial\Omega$ is given by n.

Proposition (Partial integration)

Let $f, g \in C^1(\bar{\Omega})$. Then,

$$\int_\Omega \partial_i f \, g \, dx = -\int_\Omega f \partial_i g \, dx + \int_{\partial\Omega} f g n_i \, ds \quad \text{for } i = 1, \ldots, n.$$

Proof.

Apply Gauss' divergence theorem component wise. □

Substitution rule / transformation theorem

Theorem (Integration by substitution / Transformation theorem)

Let $\widehat{\Omega} \subset \mathbb{R}^N$ be open. Let T be a diffeomorphism in $\mathbb{R}^N$. Let $1 \leq p \leq +\infty$. Then $f \in L^p(T(\widehat{\Omega}))$ if and only if $f \circ T \in L^p(\widehat{\Omega})$ and we have

$$\int_{T(\widehat{\Omega})} f \, dx = \int_{\widehat{\Omega}} f \circ T |det(\nabla T)| \, dx.$$

Moreover, if $f \in W^{1,p}(T(\widehat{\Omega}))$ if and only if $f \circ T \in W^{1,p}(\widehat{\Omega})$ and we have

$$(\nabla f) \circ T = ((\nabla T)^{-1})^T \nabla (f \circ T).$$

Chain rule

Definition

Let the functions $g : (a, b) \to \mathbb{R}^{n+1}$ and $f : \mathbb{R}^{n+1} \to \mathbb{R}$ and its composition $h = f \circ g \in \mathbb{R}$ be given and specifically $g(t, x) := (t, x) := (t, x_1, x_2, \ldots, x_n)$:

$$
\begin{aligned}
\frac{d}{dt} h(t, x) &= \frac{d}{dt} f(g(t, x)) = \frac{d}{dt} f(t, x) \\
&= \sum_{k=0}^{n} \partial_k f(g(x)) \cdot \partial_t g_k \\
&= \sum_{k=0}^{n} \partial_k f(t, x_1, \ldots, x_n) \cdot \partial_t x_k, \quad \text{where } x_0 := t \\
&= \partial_t f \cdot \partial_t t + \sum_{k=1}^{n} \partial_k f(t, x_1, \ldots, x_n) \cdot \partial_t x_k \\
&= \partial_t f + \nabla f \cdot (\partial_t x_1, \cdots, \partial_t x_n)^T \\
&= \partial_t f + \nabla f \cdot v.
\end{aligned}
$$

For instance $n = 3$ and time t, this means that we deal with a four-dimensional continuum (t, x, y, z).

Gradient, divergence, trace, Laplace

Well-known in physics, it is convenient to work with the **nabla-operator** to define derivative expressions. The gradient of a single-valued function $v : \mathbb{R}^n \to \mathbb{R}$ reads:

$$\nabla v = \begin{pmatrix} \partial_1 v \\ \vdots \\ \partial_n v \end{pmatrix}.$$

The gradient of a vector-valued function $v : \mathbb{R}^n \to \mathbb{R}^m$ is called **Jacobian matrix** and reads:

$$\nabla v = \begin{pmatrix} \partial_1 v_1 & \dots & \partial_n v_1 \\ \vdots & & \vdots \\ \partial_1 v_m & \dots & \partial_n v_m \end{pmatrix}.$$

The divergence is defined for vector-valued functions $v : \mathbb{R}^n \to \mathbb{R}^n$:

$$\operatorname{div} v := \nabla \cdot v := \nabla \cdot \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} = \sum_{k=1}^{n} \partial_k v_k.$$

The divergence for a tensor $\sigma \in \mathbb{R}^{n \times n}$ is defined as:

$$\nabla \cdot \sigma = \left(\sum_{j=1}^{n} \frac{\partial \sigma_{ij}}{\partial x_j} \right)_{1 \le i \le n}.$$

Gradient, divergence, trace, Laplace

The trace of a matrix $A \in \mathbb{R}^{n\times n}$ is defined as

$$tr(A) = \sum_{i=1}^{n} a_{ii}.$$

Definition (Laplace operator)

The Laplace operator of a two-times continuously differentiable scalar-valued function $u : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\Delta u = \sum_{k=1}^{n} \partial_{kk} u.$$

Definition

For a vector-valued function $u : \mathbb{R}^n \to \mathbb{R}^m$, we define the Laplace operator component-wise as

$$\Delta u = \Delta \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} \partial_{kk} u_1 \\ \vdots \\ \sum_{k=1}^{n} \partial_{kk} u_m \end{pmatrix}.$$

Sobolev spaces

Let $X \subset \mathbb{R}^d, d = 2, 3$ be a time-independent domain. For instance, we later use $X := \widehat{\Omega}_f$ or $X := \widehat{\Omega}_s$. We indicate by $L^p(X), 1 \leq p \leq \infty$ the standard Lebesgue space that consists of measurable functions u, which are Lebesgue-integrable to the p-th power. The set $L^p(X)$ forms a Banach space with the norm $\|u\|_{L^p(X)}$.

$$\|u\|_{L^p(X)} := \left(\int_X |u(x)|^p \, dx \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty,$$
$$\|u\|_{L^\infty(X)} := \text{ess sup } |u(x)|.$$

We obtain the Hilbert space $L^2(X)$ for $p = 2$, equipped with the inner product

$$(u, v)_{L^2(X)} := \int_X u(x) v(x) \, dx.$$

The Sobolev space $W^{m,p}(X), m \in \mathbb{N}, 1 \leq p \leq \infty$ is the space of functions in $L^p(X)$ that have distributional derivatives of order up to m, which belong to $L^p(X)$. This space is equipped with the norm

$$\|u\|_{W^{m,p}(X)} := \left(\sum_{|\alpha| \leq m} \|D^\alpha u\|^p_{L^p(X)} \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty,$$
$$\|u\|_{W^{m,\infty}(X)} := \max_{|\alpha| \leq m} \|D^\alpha u\|_{L^\infty(X)}.$$

Sobolev spaces

The symbol $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$ denotes a multi-index with the properties

$$|\alpha| := \sum_{j=1}^{d} \alpha_j, \quad D^\alpha := \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}.$$

For $p = 2, H^m(X) := W^{m,2}(X)$ is a Hilbert space equipped with the norm with the inner product

$$(u, v)_{H^m(X)} := \sum_{|\alpha| \leq m} (D^\alpha u, D^\alpha v)_{L^2(X)},$$

and the norm $|| \cdot ||_{H^m(X)}$.

Derivatives in function spaces

Definition (Directional derivative)

Let f be a mapping from X to Y. Let $a \in X$ and $\delta a \in X$ and $\varepsilon \in \mathbb{R}$. The derivative $f'(a)$ of f is computed as action on vectors of X, i.e.,

$$f'(a)\delta a = \lim_{\varepsilon \to 0} \frac{f(a+\varepsilon\delta a) - f(a)}{\varepsilon} = \frac{d}{d\varepsilon} f(a+\varepsilon\delta a)|_{\varepsilon=0} \in Y.$$

The element $f'(a)\delta a \in Y$ is called the directional derivative of the function f into the direction of the vector $\delta a \in X$ at $a \in X$.

Definition (Gâteaux derivative)

If the directional derivative exists for all directions δa, then $f'(a)$ is called Gâteaux derivative.

Definition (Fréchet derivative)

A mapping $f : \Omega \subset X \to Y$ (Ω open) is called Fréchet differentiable at a point $a \in \Omega$ if there exists an element $f'(a)$ of the space $L(X,Y)$ such that

$$f(a+h) = f(a) + f'(a)h + o(h),$$

with $o(h) = \|h\|\varepsilon(h)$ with $\lim_{h\to 0} \varepsilon(h) = 0$ in Y. Then, the element $f'(a) \in L(X,Y)$ is called the Fréchet derivative at $a \in \Omega$. The Fréchet derivative is necessarily unique and does imply that f is continuous at $a \in \Omega$.

Kinematics with Eulerian and Lagrangian coordinate systems

Definition (Domains)

We denote:

- $\widehat{\Omega}$: the reference/undeformed configuration;
- $\Omega(t)$: the current/deformed configuration.

Furthermore, $\Omega(t=0)$ is the so-called initial configuration. Often, $\widehat{\Omega} := \Omega(t=0)$. Moreover, in these notes we abbreviate $\Omega := \Omega(t)$.

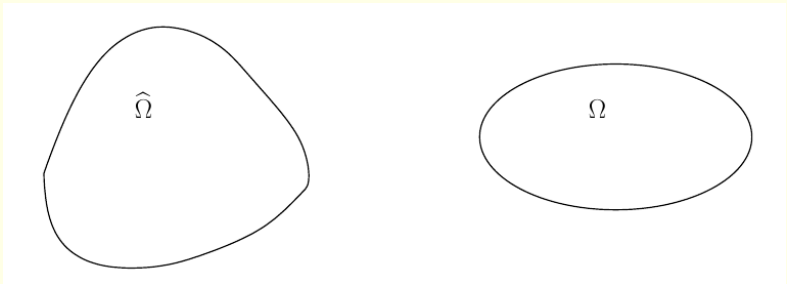


Figure: Domains: $\widehat{\Omega}$ and Ω.

The reasoning to work with two coordinate systems is that

- solids/elasticity are usually described in the **Lagrangian system** $\widehat{\Omega}$;
- fluids (i.e., Navier-Stokes) are usually described in **Eulerian coordinates** Ω.

Deformations

Definition (Deformation field)

A deformation of $\widehat{\Omega}$ is a smooth, one-to-one (i.e., injective), orientation-preserving mapping

$$\widehat{T} : \widehat{\Omega} \to \Omega \quad \text{with } (t, \hat{x}) \mapsto (t, x) = (t, \widehat{T}(t, \hat{x})).$$

This mapping associates each point $\hat{x} \in \widehat{\Omega}$ (of a reference domain) to a new position $x \in \Omega$ (of the physical domain). Consequently, with the help of the deformation, $\widehat{T}$, we can represent the deformed configuration as $\Omega = \widehat{T}(\widehat{\Omega})$.

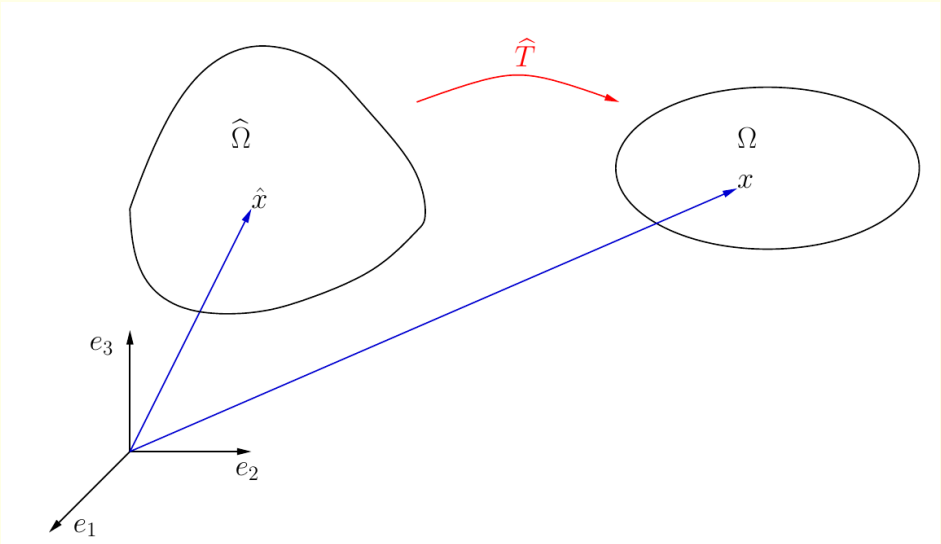


Figure: Definition of the deformation $\widehat{T}$.

Displacements

Definition (Material/Lagrangian description of the displacement field)

$$\hat{u} : (t, \hat{x}) \to \hat{u}(t, \hat{x}) = x(t, \hat{x}) - \hat{x}$$

It relates a particle's position in the reference configuration $\hat{x}$ to its corresponding position in the current configuration x at time t. $\diamond$

Definition (Spatial/Eulerian description of the displacement field)

$$u(t, x) = x - \hat{x}(x, t).$$

We recall that also $x = T(t, \hat{x})$. $\diamond$

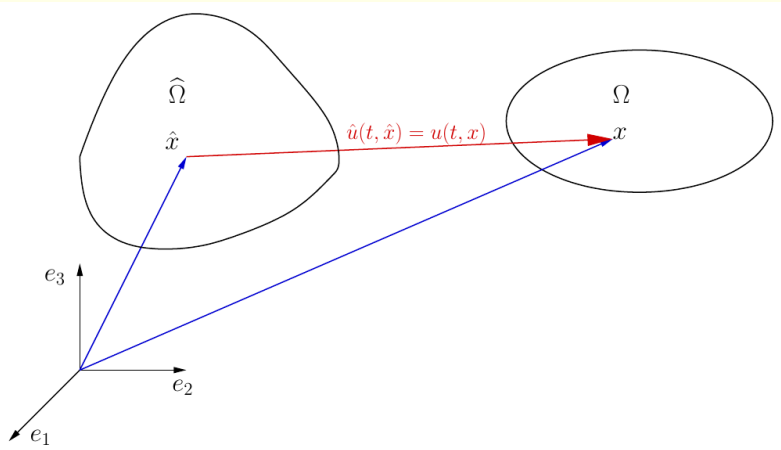


Figure: Descriptions of displacements: Going from the origin via $\widehat{\Omega}$ means $\hat{x} + \hat{u} = x$. Going from the origin via Ω leads to $x - u = \hat{x}$.

Time derivatives (I)

Definition (Total/Material time derivative of a Lagrangian field)

$$\frac{d}{dt}\hat{f}(t,\hat{x}) = \partial_t \hat{f}(t,\hat{x}).$$

The material time derivative measures the rate at which $\hat{f}$ changes in time but following the path line of this article. This means we measure the rate-change in time of exactly the same particle at all times.

Definition (Spatial time derivative of an Eulerian field)

The local time derivative of an Eulerian field is defined as

$$\partial_t f(t,x).$$

The current position x is held fixed while measuring the rate at which f changes in time at this fixed point. This means, at each time, f represents a new particle at x. The spatial time derivative is also known to be the local time derivative.

Time derivatives (II)

Proposition (Total/Material time derivative of an Eulerian field)

Let $f(t,x) : \mathbb{R}^m \to \mathbb{R}$ *and* $x := (x_1, \dots, x_{m-1}) \in \mathbb{R}^{m-1}$. *Then,*

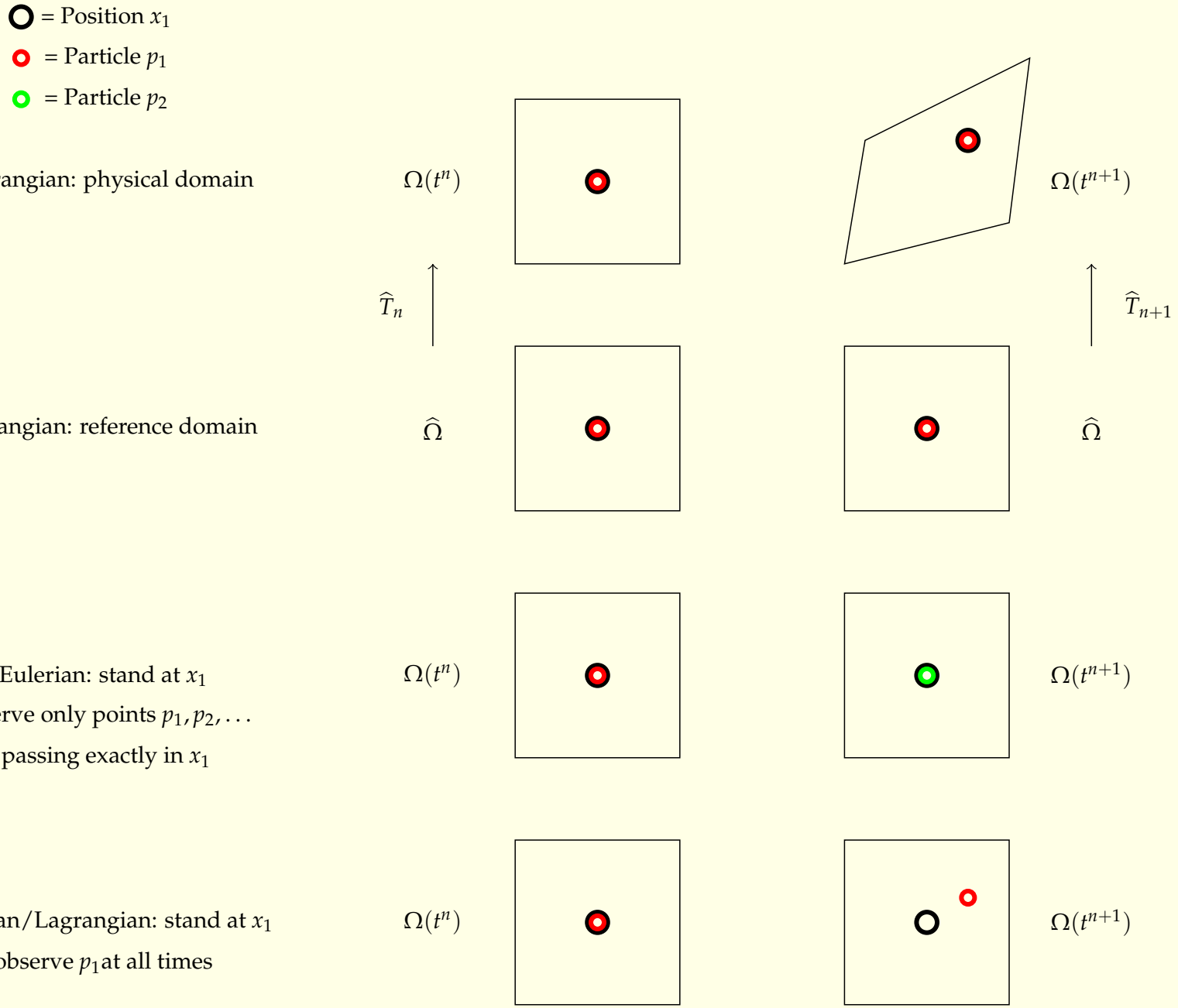
$$\begin{aligned}\frac{d}{dt} f(t,x) &= \frac{d}{dt} f(t, \widehat{T}(t,\hat{x})) \\ &= \partial_t f(t,x) + \nabla f(t,x) \cdot \partial_t \widehat{T}(t,\hat{x}) \\ &= \partial_t f(t,x) + \nabla f(t,x) \cdot \partial_t x(t,\hat{x}) \\ &= \partial_t f(t,x) + \nabla f(t,x) \cdot v(t,x).\end{aligned}$$

The material derivative at a fixed material point $\hat{x}$ *describes the change of an Eulerian variable* f *that is at time* t *at point* x *and travels with the velocity* $v(t,x)$. *Here, the first term prescribes the local change whereas the second term, the convective part, is due to the movement of the particle through the domain.*

Proof.

Follows immediately from the chain rule on page 19. □

Movements of particles in Lagrangian and Eulerian coordinates



Deformation gradient (I)

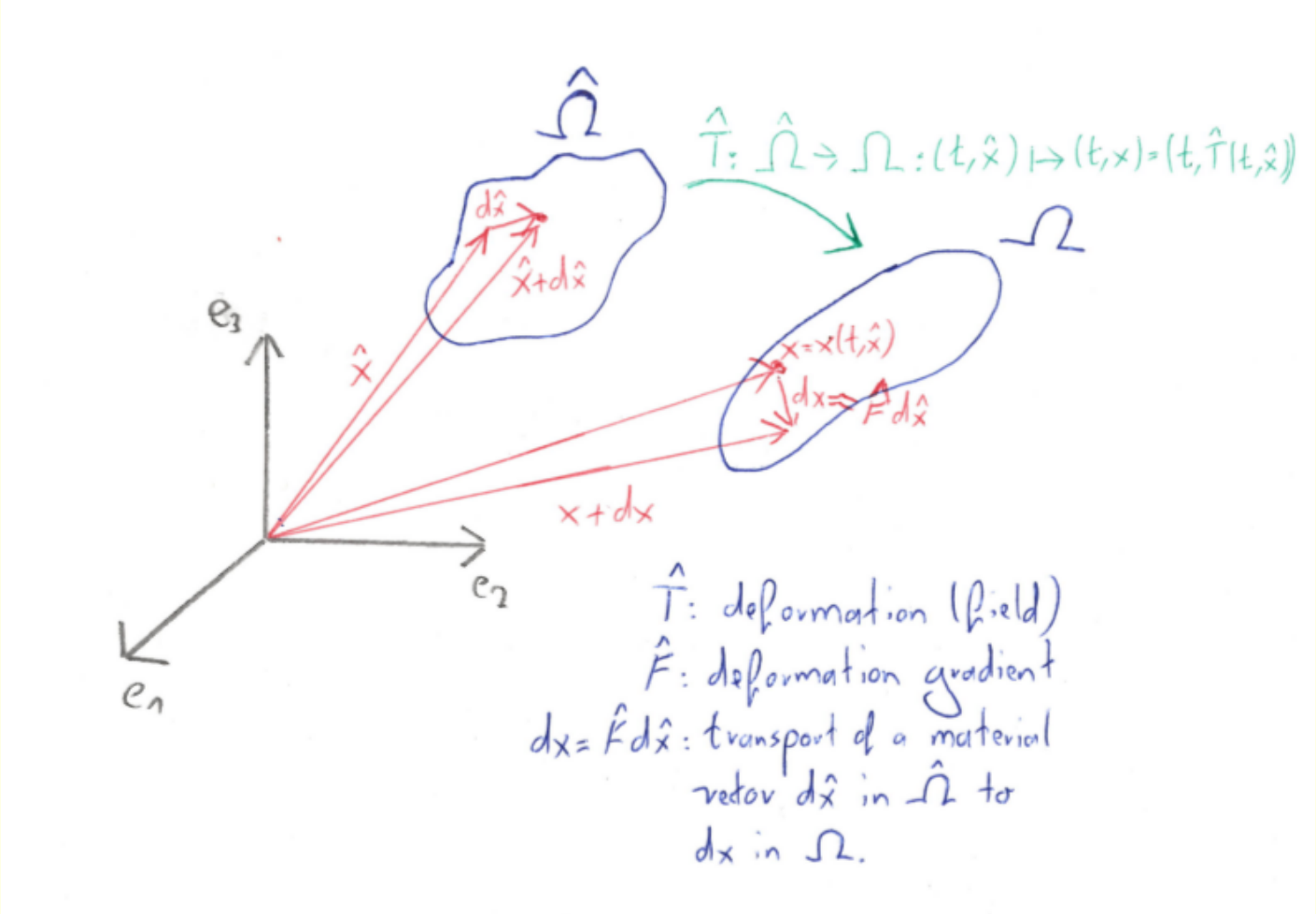


Figure: Transport of materials vectors, transformation $\widehat{T}$, and deformation gradient $\widehat{F}$.

- The key purpose of the deformation gradient $\widehat{F}$ is to link dx and $d\hat{x}$

Deformation gradient (II)

Definition (Deformation gradient $\widehat{F}$)

For $x := x(t, \hat{x}) = \widehat{T}(t, \hat{x}) = \hat{x} + \hat{u}$, it holds for infinitesimal deformations:

$$dx = \widehat{F} \cdot d\hat{x},$$

with $\widehat{F} = \widehat{\nabla} x$, i.e, $\widehat{F}_{ij} = \frac{\partial x_i}{\hat{\partial} \hat{x}_j}$. Using the relationship $x = \hat{x} + \hat{u}$, we therefore obtain

$$\widehat{F} = \frac{\partial x}{\hat{\partial} \hat{x}} = \frac{\partial}{\hat{\partial} \hat{x}} (\hat{x} + \hat{u}) = \frac{\hat{\partial} \hat{x}}{\hat{\partial} \hat{x}} + \frac{\hat{\partial} \hat{u}}{\hat{\partial} \hat{x}} = \hat{I} + \widehat{\nabla} \hat{u}$$

In compact form:

$$\widehat{F} = \widehat{\nabla} \widehat{T} = \hat{I} + \widehat{\nabla} \hat{u}.$$

Transforming spatial derivatives

Proposition

Let $f(t,x) : \Omega \to \mathbb{R}^n$ be an Eulerian function and let $\hat{f}(t,\hat{x}) : \widehat{\Omega} \to \mathbb{R}^n$ a corresponding Lagrangian function. To compute the Eulerian spatial derivative we have the relation:

$$\nabla f = \widehat{\nabla}\hat{f}\,\widehat{F}^{-1}.$$

The inverse $\widehat{F}^{-1}$ exists due to the assumptions on $\widehat{T}$.

Proof.

Chain rule:

$$\widehat{\nabla}\hat{f} = \widehat{\nabla}\hat{f}(t,\hat{x}) = \widehat{\nabla}\hat{f}(t,\widehat{T}(t,\hat{x})) = \sum_{i=1}^{n} \partial_j f(t,\widehat{T}(t,\hat{x})) \cdot \frac{\hat{\partial}\widehat{T}(t,\hat{x})}{\hat{\partial}\hat{x}} = \nabla f\,\widehat{F}.$$

Since $\widehat{F}$ is invertible due to the assumptions on $\widehat{T}$ the assertion follows. □

Strain tensor

Definition (Green-Lagrange strain tensor $\widehat{E}$)

$$\widehat{E} = \frac{1}{2}(\widehat{C} - \hat{I}) = \frac{1}{2}(\widehat{F}^T\widehat{F} - \hat{I}) = \frac{1}{2}(\widehat{\nabla}\hat{u} + \widehat{\nabla}\hat{u}^T + \widehat{\nabla}\hat{u} \cdot \widehat{\nabla}\hat{u}^T),$$

which is symmetric and positive definite for all $\hat{x} \in \widehat{\Omega}$ since $\widehat{C}$ and of course, $\hat{I}$ has these properties, too. $\diamond$

Definition (Linearized Green-Lagrange strain tensor $\widehat{E}_{lin}$)

$$\widehat{E}_{lin} = \frac{1}{2}(\widehat{\nabla}\hat{u} + \widehat{\nabla}\hat{u}^T).$$

$\diamond$

In words:
Strain is a measure of a deformation $\widehat{T}$ representing the displacement $\hat{u}$ between particles in a body relative to a reference length.

Reynolds' transport theorem

Hypothesis

Let us assume:

- $(t, \hat{x}) \mapsto x(t, \hat{x})$ *is continuously differentiable;*
- *For all* $t \geq t_0$ *the mapping* $\hat{x} \mapsto x(t, \hat{x})$ *is invertible;*
- *Orientation-preservation: the Jacobi determinant is always positive; namely* $\hat{J} = \det(\widehat{F}) > 0$.

Proposition (Reynolds' transport theorem)

As before, let $\Omega := \Omega(t)$ *be a time-dependent domain and* $(t, \hat{x}) \mapsto x(t, \hat{x})$ *be given and assume Hypothesis 1. Furthermore, let the functions* $(t, x) \mapsto \partial_t x(t, \hat{x})$ *and* $(t, x) \mapsto f(t, x)$ *be continuously differentiable. Then, it holds*

$$\begin{aligned} \frac{d}{dt} \int_\Omega f(t,x)\, dx &= \int_\Omega \left[\frac{d}{dt} f(t,x) + f(t,x) \nabla \cdot v(t,x) \right] dx \\ &= \int_\Omega \left[\partial_t f(t,x) + \nabla \cdot (f(t,x) \cdot v(t,x)) \right] dx. \end{aligned}$$

Proof.

Sketch: transform to $\widehat{\Omega}$. Interchange differentiation and integration and apply then again the chain rule. Finally, transform back to Ω. □

Balance principles

The equations in continuum mechanics are based on fundamental physical principles:

- Mass
- Momentum
- Angular momentum
- Energy

These laws are formulated in Eulerian coordinates in the following.

Conservation of mass

Definition (Conservation of mass)

$$\frac{d}{dt}\int_\Omega \rho(t,x)\,dx = 0.$$

Applying Reynolds' transport theorem, we obtain the integral formulation

$$\int_\Omega \left[\partial_t \rho(t,x) + \nabla\cdot(\rho(t,x)v(t,x))\right]\,dx = 0$$

and if this holds for any subdomain of Ω and the integrand is sufficiently smooth, we obtain:

$$\partial_t \rho(t,x) + \nabla\cdot(\rho(t,x)v(t,x)) = 0$$

Remark:
When changes in density are time-independent, we have

$$\nabla\cdot(\rho v) = 0.$$

If the density is also spatially constant, we obtain

$$\nabla\cdot v = 0.$$

This is in fact the **continuity condition** for **incompressible fluid flows**; namely the density is constant.

Conservation of momentum

- Newton's second law states:

$$\frac{d}{dt}(mv) = F,$$

  where F is the force that acts on the mass.

- In continuum mechanics Newton's second law reads:

$$\frac{d}{dt}\int_\Omega \rho v\, dx = \int_\Omega \rho f\, dx + \int_{\partial\Omega} g\, ds.$$

- Employing again Reynolds' transport theorem yields:

$$\int_\Omega \left[\partial_t(\rho v) + \nabla\cdot(\rho v v)\right]\, dx = \int_\Omega \rho f\, dx + \int_{\partial\Omega} g\, ds.$$

- Compact form (using Cauchy's theorem for g with $g = \sigma\cdot n$):

$$\int_\Omega (\rho\partial_t v + \rho v\cdot\nabla v - (\nabla\cdot\sigma) - \rho f)\, dx.$$

  For smooth functions ρ, v, σ and arbitray Ω, we obtain the differential form:

$$\rho\partial_t v + \rho(v\cdot\nabla)v - \nabla\cdot\sigma = \rho f.$$

A constitutive law for fluids

Viscous flow with inner friction:

Definition

$$\sigma = -pI + \mu_f(\nabla v + \nabla v^T) + \lambda_f \nabla \cdot vI,$$

where p is the hydrostatic pressure and where $\lambda_f > 0$ and $\mu_s > 0$ are volume and shear viscosity, respectively.

- Fluids with such stress tensors are called *Newtonian fluids*.
- Inner friction means that particles with different velocities interact on the mirco-scale with each other; namely velocity variations do cause friction.
- For incompressible flow, the last term vanishes:

  $$\lambda_f \nabla \cdot vI = 0$$

  because of mass conservation $\nabla \cdot v = 0$.

Incompressible Navier-Stokes (fluid flow; Eulerian coordinates)

Isothermal, incompressible, viscous fluid flow is described by the (nonlinear) Navier-Stokes equations.

Formulation (Navier-Stokes equations)

Find vector-valued velocities $v : \Omega \times I \to \mathbb{R}^3$ *and a scalar-valued pressure* $p : \Omega \times I \to \mathbb{R}$*:*

$$\begin{aligned} \rho \partial_t v + \rho (v \cdot \nabla) v - \nabla \cdot \sigma &= \rho f, \\ \nabla \cdot v &= 0, \end{aligned}$$

where

$$\sigma = -pI + \mu_f (\nabla v + \nabla v^T).$$

Here, $\mu_f = \rho_f \nu$ *is the dynamic viscosity, whereas* ν *is the so-called kinematic viscosity. Initial and boundary conditions are added later.*

Elastic deformations (solids; Lagrangian coordinates)

Definition (Second Piola-Kirchhoff stress tensor)

$$\widehat{\Sigma} = \widehat{F}^{-1}\widehat{\Pi} = \hat{J}\widehat{F}^{-1}\sigma_s\widehat{F}^{-T}.$$

◇

- We recall the conservation of momentum:

$$\frac{d}{dt}\int_{\widehat{\Omega}} \hat{\rho}\partial_t\hat{u}\,d\hat{x} = \int_{\widehat{\Omega}} \hat{\rho}\hat{f}\,d\hat{x} + \int_{\partial\widehat{\Omega}} \widehat{F}\widehat{\Sigma}\hat{n}\,d\hat{s}.$$

- Here, the density $\hat{\rho}$ and the volume force $\hat{f}$ are related to the reference configuration. Further calculation (using Gauss' divergence theorem and Reynolds transport theorem) yields

$$\int_{\widehat{\Omega}} \left(\hat{\rho}\partial_t^2\hat{u} - \widehat{\nabla}\cdot\widehat{F}\widehat{\Sigma}\right)\,d\hat{x} = \int_{\widehat{\Omega}} \hat{\rho}\hat{f}\,d\hat{x}.$$

- We notice that the convective term in Reynolds' theorem vanishes since we work in a Lagrangian setting.

Elastic deformations (solids; Lagrangian coordinates)

For sufficiently smooth functions and since (43) holds for each subdomain of $\widehat{\Omega}$, we can write the differential form of the elasto-dynamics equations:

Formulation (Solid equation)

Find $\hat{u} : \widehat{\Omega} \times I \to \mathbb{R}^d$ *such that*

$$\hat{\rho}\partial_t^2 \hat{u} - \widehat{\nabla} \cdot \widehat{F}\widehat{\Sigma} = \hat{f},$$

plus boundary and initial conditions; later more.

A constitutive law: Saint Venant-Kirchhoff-Material

Definition (Saint Venant-Kirchhoff-Material (STVK) / a large displacement-small strain model)

For an isotropic, homogeneous material for which its reference configuration is a natural state; namely $\widehat{\Sigma}_s(0) = 0$, we obtain

$$\widehat{\Sigma}_s := \widehat{\Sigma}_s(\hat{u}) = 2\mu\widehat{E} + \lambda tr(\widehat{E})\hat{I},$$

with the positive Lamé constants μ and λ. In particular, the STVK material is only useful in a neighborhood of the reference configuration $\widehat{C} \approx \hat{I}$ with small strains $\widehat{E}$. We finally mention that the equal sign is (strictly speaking) only an approximation in which higher order terms of $\widehat{E}$ have been neglected.

Prototype configuration

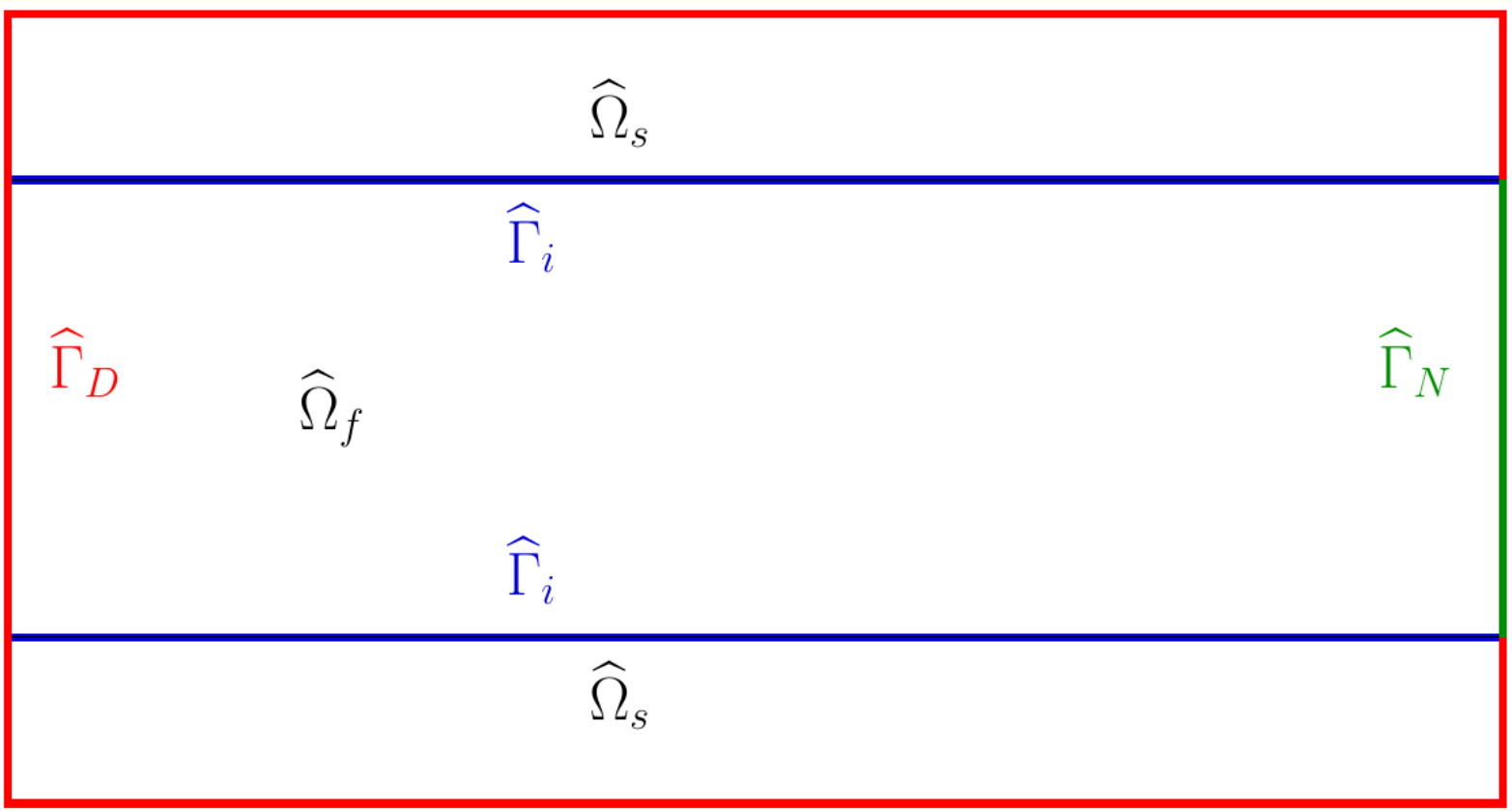


Figure: Definitions of domains and boundaries.

Summary and outlook

Notation, continuum mechanics, modeling fluids and solids ✓

Classifications and deriving a variational formulation

Class 2

Situation

- We have now modeled fluid flows and elastic deformations
- To this end, we have obtained PDEs in the so-called **strong** form
- The goal is to apply **variational** principles:

→ **Weak** form

→ **Galerkin methods** (namely finite elements)[3] for the computer implementation

- Such Galerkin methods are also often used in the mathematical analysis

→ **Goal of class 2:** how do we obtain such variational formulations for multiphysics/fluid-structure interaction?

[3] For an introduction to Galerkin finite elements, I refer to my lecture notes (page 5 of this presentation) on Numerical methods for PDEs `https://doi.org/10.15488/9248` or the references cited therein.

Deriving variational forms: two-step procedure

Deriving a variational form:

- Step 1: Design a function space V that also includes the correct boundary conditions.
- Step 2: Multiply the strong form (D) with a test function from V and integrate.

Remark

*The second operation 'weakens' the derivative information (therefore the name **weak form**) because rather evaluating 2nd order derivatives, we only need to evaluate a 1st order (weak) derivative on the trial function and another 1st order (weak) derivative on the test function.*

Example: Poisson problem

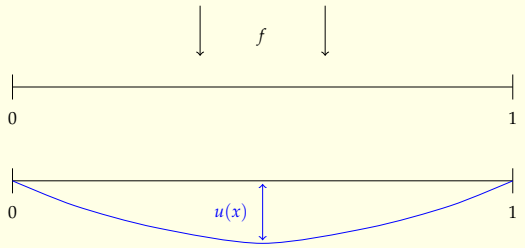


Figure: Poisson problem in 1D: The clothesline problem: a uniform force $f = -1$ acts on a 1D line yielding a displacement $u(x)$.

- Given Poisson (1D) with homogeneous Dirichlet conditions: Find $u : \Omega = (0,1) \to \mathbb{R}$ such that

$$\begin{aligned} -u'' &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

- Step 1: Take space

$$V := \{v |\ v \in C[0,1], v' \text{ is pc. cont. and bound. on } [0,1], v(0) = v(1) = 0\}$$

- We now address Step 2:

$$-u'' = f$$

$$\Rightarrow -\int_\Omega u''\phi\,dx = \int_\Omega f\phi\,dx \qquad \Rightarrow \int_\Omega u'\phi'\,dx - \underbrace{\int_{\partial\Omega} \partial_n u\phi\,ds}_{=0, \text{ because } \phi=0 \text{ on } \partial\Omega} = \int_\Omega f\phi\,dx$$

$$\Rightarrow \int_\Omega u'\phi'\,dx = \int_\Omega f\phi\,dx.$$

- To summarize, we have the following **variational formulation**:

$$\int_\Omega u'\phi'\,dx = \int_\Omega f\phi\,dx$$

Usual abstract notation

Formulation (Variational Poisson problem on the continuous level)

Find $u \in V$ such that

$$a(u, \phi) = l(\phi) \quad \forall \phi \in V,$$

with

$$a(u, \phi) := (u', \phi') := \int_\Omega u' \phi' \, dx,$$
$$l(\phi) := (f, \phi) := \int_\Omega f \phi \, dx.$$

*The unknown function u is called the **trial function** whereas ϕ is the so-called **test function**.*

On the abstract notation

- We just have had: Find $u \in V$ such that

$$a(u, \varphi) = l(\phi) \quad \forall \varphi \in V.$$

  Here $a(u, \varphi) : V \times V \to \mathbb{R}$ is a **bilinear form** and $l(\varphi) \in V^*$ is a linear form (linear functional).

- For **nonlinear problems**, the solution variable $u \in V$ is nonlinear while the test function is still linear. Here we use the notation: Find $u \in V$ such that

$$a(u)(\varphi) = l(\varphi) \quad \forall \varphi \in V.$$

  Here, $a(u)(\varphi)$ is a so-called **semi-linear form**.

- For **(linear) PDE systems**, my notation is: Find $U \in X$ such that

$$A(U, \Psi) = F(\Psi) \quad \forall \Psi \in X.$$

- For **nonlinear PDE systems**, my notation is: Find $U \in X$ such that

$$A(U)(\Psi) = F(\Psi) \quad \forall \Psi \in X.$$

  This last notation is thus used for multiphysics PDEs as in this winter school.

Classifications

1. Dependent variables $(x, y, z, t, ...)$; if only one, then ODE, otherwise PDE
2. Do we deal with steady-state or time-dependent PDEs?
3. Governing physical conservation properties (diffusion, transport, waves, ...)
4. Order of a PDE (in time, space, or both)
5. Single equations and PDE systems
6. Nonlinear problems:
   - Nonlinearity in the PDE
   - The function set is not a vector space yielding a variational inequality
7. Coupled problems and coupled PDE systems:
   - entering model or material parameters,
   - linear or nonlinear coupling terms,
   - right-hand sides,
   - interfaces.

Examples

1. Find $u : \Omega \to \mathbb{R}$:

$$-\Delta u + u^2 = f$$

Properties: nonlinear (semilinear), stationary, scalar-valued.

2. Find $v : \Omega \to \mathbb{R}^n$ and $p : \Omega \to \mathbb{R}$

$$\partial_t v + (v \cdot \nabla)v - \frac{1}{Re}\Delta v + \nabla p = f, \quad \nabla \cdot v = 0$$

with Re being the Reynolds' number. Properties: semilinear, ... For $Re \to \infty$ we obtain the Euler equations. Properties: nonlinear (quasilinear), nonstationary, vector-valued, PDE system.

3. Find $u : \Omega \to \mathbb{R}$ and $\varphi : \Omega \to \mathbb{R}$

$$-\nabla \cdot (a(\varphi)\nabla u) = f,$$
$$a(\varphi)|\nabla u|^2 - \Delta \varphi = g$$

Properties: nonlinear, coupled problem via coefficients, stationary. Equations become linear when solution variables are fixed in the other equation.

Examples

4. Find $u : \Omega \to \mathbb{R}$ and $\varphi : \Omega \to \mathbb{R}$

$$\begin{aligned} -\Delta u &= f(\varphi), \\ |\nabla u|^2 - \Delta \varphi &= g(u) \end{aligned}$$

Properties: nonlinear, coupled problem via right hand sides, stationary. Equations become linear when solution variables are fixed in the other equation.

5. Let Ω_1 and Ω_2 with $\Omega_1 \cap \Omega_2 = 0$ and $\bar{\Omega}_1 \cap \bar{\Omega}_2 = \Gamma$ and $\bar{\Omega}_1 \cup \bar{\Omega}_2 = \Omega$. Find $u_1 : \Omega_1 \to \mathbb{R}$ and $u_2 : \Omega_2 \to \mathbb{R}$:

$$\begin{aligned} -\Delta u_1 &= f_1 \quad \text{in } \Omega_1, \\ -\Delta u_2 &= f_2 \quad \text{in } \Omega_2, \\ u_1 &= u_2 \quad \text{on } \Gamma, \\ \partial_n u_1 &= \partial_n u_2 \quad \text{on } \Gamma. \end{aligned}$$

Properties: linear, coupled problem via interface conditions, stationary.

Hints for developing numerical methods for 'complicated' equations

In case you are given a nonlinear IBVP (initial-boundary value problem) and want to start developing numerical methods for this specific PDE, it is often much easier to start with appropriate simplifications in order to build and analyze step-by-step your final method. Let us say you are given the nonlinear time-dependent PDE

$$\nabla u \partial_t^2 u + u \cdot \nabla u - (\Delta u)^2 = f$$

Then, you could tackle the problem as follows:

1. Consider the linear equation:
$$\partial_t^2 u - \Delta u = f$$
which is nothing else than the wave equation.
2. Add a slight nonlinearity to make the problem semi-linear:
$$\partial_t^2 u + u \cdot \nabla u - \Delta u = f$$
3. Add ∇u such that the problem becomes quasi-linear:
$$\nabla u \partial_t^2 u + u \cdot \nabla u - \Delta u = f$$
4. Make the problem fully nonlinear by considering $(\Delta u)^2$:
$$\nabla u \partial_t^2 u + u \cdot \nabla u - (\Delta u)^2 = f.$$

In each step, make sure that the corresponding numerical solution makes sense and that your developments so far are correct. If yes, proceed to the next step.

Remark for double-checking proofs and debugging codes

Remark

The contrary to the previous example works as well and should be kept in mind! If you have implemented the full nonlinear PDE and recognize a difficulty, you can reduce the PDE term by term and make it step by step simpler. The same procedure holds true for the mathematical analysis.

Situation in terms of coordinate systems

From Class 1, we know that fluids = Eulerian and solids = Lagrangian. Now:

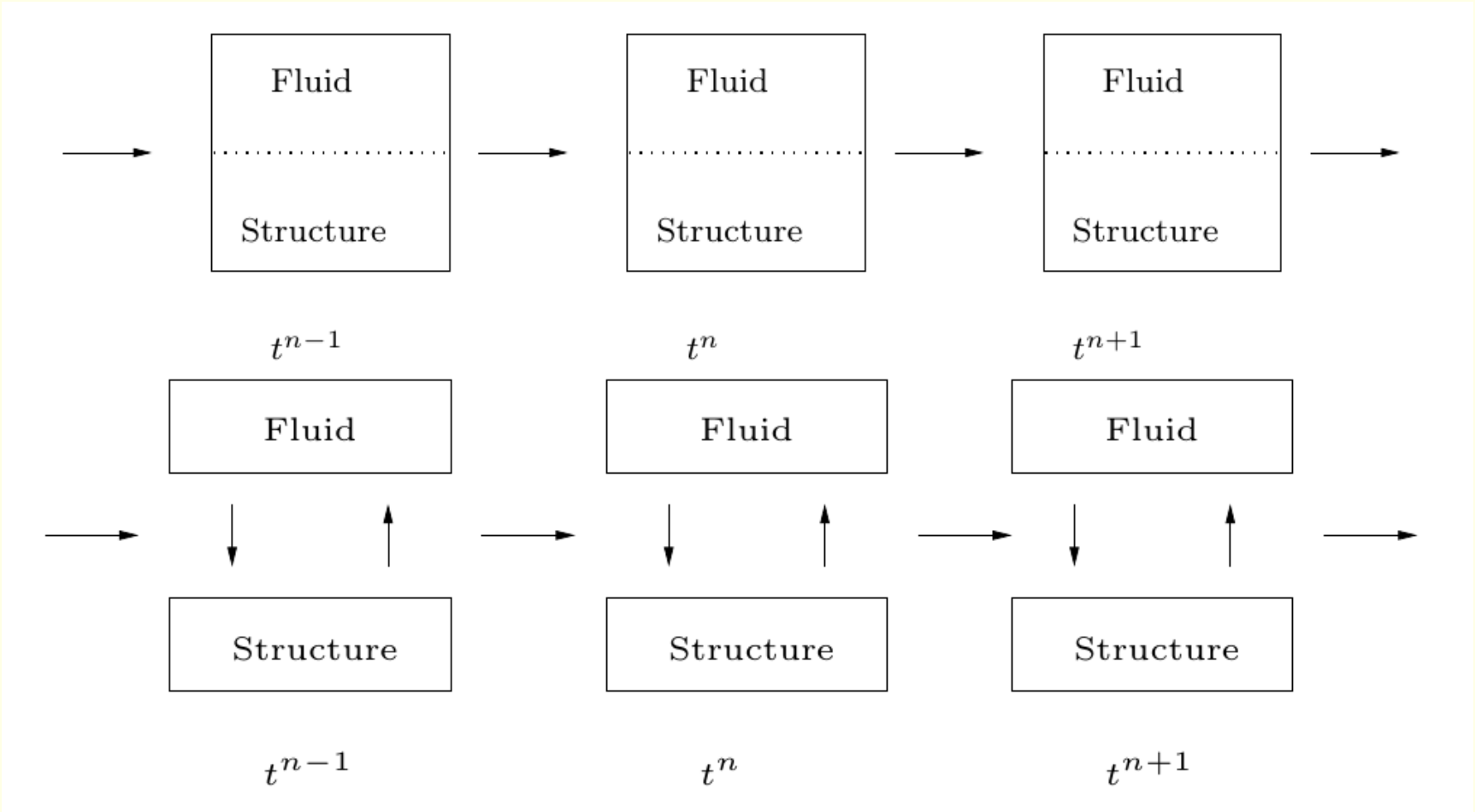
Fluids	Solids	Remarks
Eulerian	Lagrangian	Natural systems
Lagrangian-type	Lagrangian	Resulting in ALE
Eulerian	Eulerian	Fully Eulerian (fixed mesh)
Lagrangian	Eulerian	Unusual; not seen so far

Frameworks (most important):

- Immersed boundary method[4]
- Fictious domain method[5]
- **ALE - arbitrary Lagrangian-Eulerian**[6]
- Deforming-spatial-domain/stabilized space-time [7]
- Fixed-mesh ALE[8]
- Fully Eulerian[9]

[4] Peskin, 2002; Heltai/Costanzo, 2010
[5] Glowinski, 1994
[6] Hughes/Liu/Zimmermann, 1981; Donea/Giuliani/Halleux, 1982; Formaggia/Nobile, 1999
[7] Tezduyar/Sathe/Stein, 2006
[8] Codina/Houzeaux/Coppola-Owen/Baiges, 2009
[9] Dunne, 2006; Richter/Wick, 2010; Frei, 2016

Partitioned versus monolithic



Interface-capturing versus interface-tracking

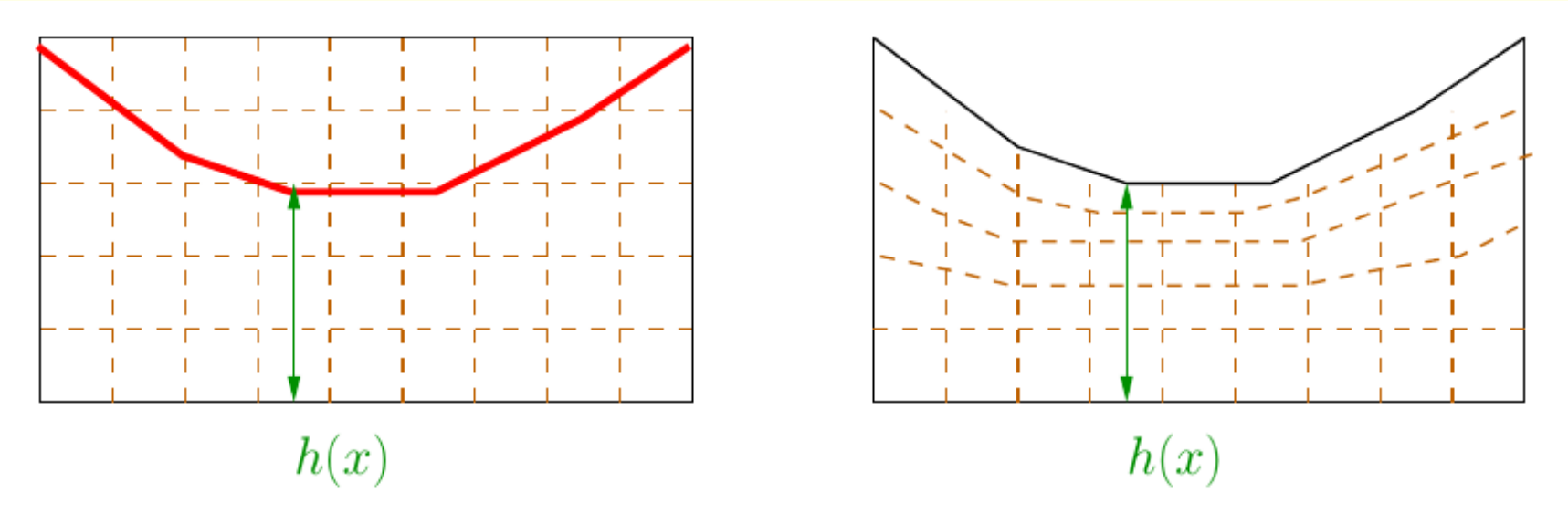


Figure: Left: the mesh is fixed and the interface must be captured. Right: interface-tracking in which the interface is located on mesh edges.

- Interface-tracking may suffer from mesh degeneration, and has problems in contact mechanics, but has a higher accuracy on the interface
- Interface-capturing allows for topology changes and large deformations, but suffers in general from less accurate interface representations.

$\rightarrow$ Variations and improvements of both methodologies are subject to current research in many groups.

The concept of variational-monolithic ALE$_{fx}$

- ALE is an interface-tracking approach

Rough idea of ALE and its definition using the concepts from continuum mechanics:

Definition

ALE is an intermediate state in which the fluid domain is moved according to the solid. This requires a mapping between the deformed state and a reference configuration. However, this is exactly how we worked in Class 1 to introduce the concepts of continuum mechanics. In the following we are going to use $\hat{\mathcal{A}} := \widehat{T}$ and recapitulate all definitions we already have had so far.

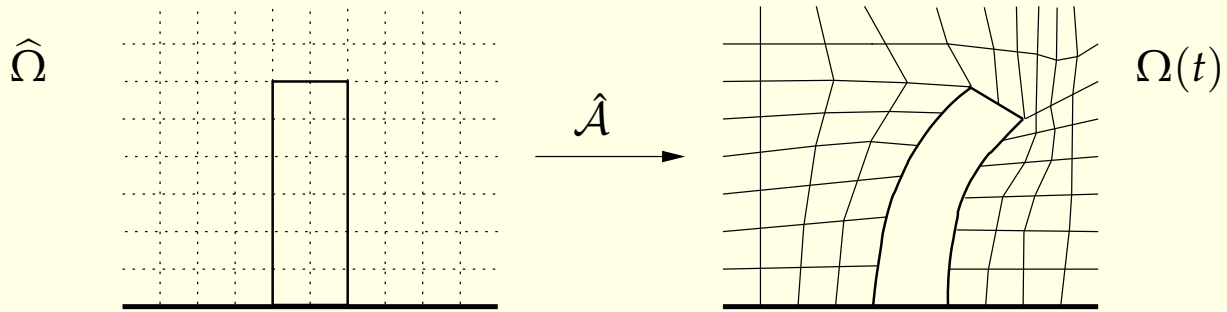


Figure: Defining the ALE transformation.

The ALE transformation

Definition

The ALE mapping is defined in terms of the fluid mesh displacement $\hat{u}_f$ such that

$$\hat{\mathcal{A}}(\hat{x},t) : \widehat{\Omega}_f \times I \to \Omega_f, \quad \text{with } \hat{\mathcal{A}}(\hat{x},t) = \hat{x} + \hat{u}_f(\hat{x},t).$$

It is specified through the deformation gradient and its determinant

$$\widehat{F} := \widehat{\nabla}\hat{\mathcal{A}} = \hat{I} + \widehat{\nabla}\hat{u}, \quad \hat{J} := \det(\widehat{F}).$$

Furthermore, function values in Eulerian and Lagrangian coordinates are identified by

$$u_f(x) =: \hat{u}_f(\hat{x}), \quad \text{with } x = \hat{\mathcal{A}}(\hat{x},t).$$

The mesh velocity is defined by $\hat{w} := \partial_t \hat{\mathcal{A}}$. The mesh velocity is numerically realized as $\hat{w} = \partial\hat{\mathcal{A}} = \frac{\hat{u}_f - \hat{u}_f^{n-1}}{k}$, where the fluid mesh displacements are computed with the help of an additional PDE.

Two ALE realizations

Definition

Fluid-structure interaction in ALE coordinates can be realized in two ways:

- ALE_{dm} (explicit mesh moving): the fluid equations are computed on the deformed configuration Ω and the mesh is moved explicitely.
- ALE_{fx} (implicit mesh moving): all fluid equations are transformed onto the fixed reference configuration $\widehat{\Omega}$ and the mesh movement is 'hidden' in the transformations $\widehat{F}$ and $\hat{J}$.

In my work, I prefer this second possibility; namely ALE_{fx}.

- In practice the ALE transformation is most often realized by solving an additional PDE (sometimes called mesh motion PDE)
- This auxiliary PDE will move the fluid mesh according to the structural displacements on the interface $\widehat{\Gamma}_i$ where Ω_f and Ω_s touch

$\rightarrow$ Geometric coupling condition: $\hat{u}_f = \hat{u}_s$

- A simple example for the auxiliary PDE is

$$
\begin{aligned}
-\widehat{\Delta}\hat{u}_f &= 0 \quad \text{in } \widehat{\Omega}_f \\
\hat{u}_f &= \hat{u}_s \quad \text{on } \widehat{\Gamma}_i \\
\hat{u}_f &= 0 \quad \text{on } \widehat{\Gamma}_i \setminus \partial\widehat{\Omega}_f.
\end{aligned}
$$

The ALE time derivative (I)

- In a Lagrangian setting, the total and the partial derivatives coincide:

$$\frac{d}{dt}\hat{f}(\hat{x},t) = \partial_t \hat{f}(\hat{x},t).$$

- In an Eulerian framework, we find the following standard relation between the *material time-derivative* (the total time derivative) $\frac{d}{dt}f$ and the partial time derivative $\partial_t f$:

$$\frac{d}{dt}f(x,t) = v \cdot \nabla f + \partial_t f(x,t),$$

  where the additional term $v \cdot \nabla f$ is referred to as a transport term.

- This definition is exactly the same as we have had before on page 30.

The ALE time derivative (II)

Definition (ALE time derivative)

The ALE time derivative is defined as

$$\hat{\partial}_t f(x,t) := \partial_t|_{\hat{\mathcal{A}}} f(x,t) = \partial_t f(x,t) + w \cdot \nabla f,$$

where the transport term appears due to the motion of the computational domain. Moreover (see additionally Figure 11 on page 70):

- In a Lagrangian description, we have $w = v$, i.e., the domain $\Omega(t)$ is moving with the fluid velocity v;
- In a fixed Eulerian setting, it holds $w = 0$, i.e., the domain $\Omega(t)$ is fixed;
- In ALE, we have $0 \leq w \leq v$. Later we will see that $v = w$ at the FSI-interface $\widehat{\Gamma}_i$ and a bit away, we have $0 < w < v$, while far away $w = 0$ (the mesh is not moving anymore). Thus, in ALE, depending on the location in the domain, we use both Eulerian and Lagrangian frameworks with a smooth transition between them.

Illustrations

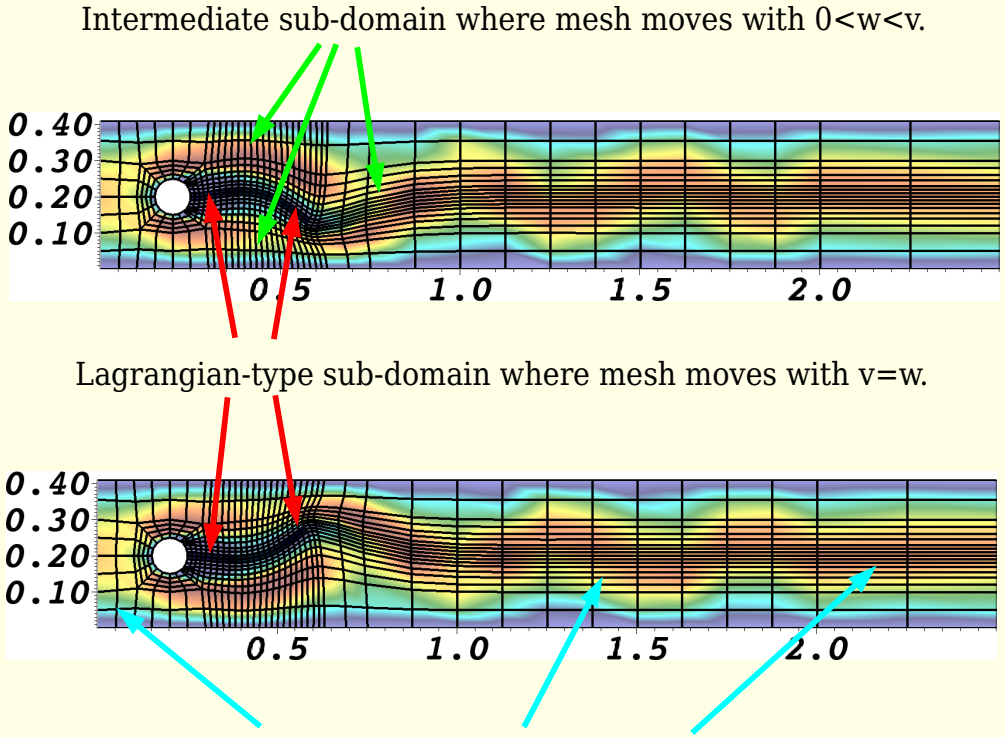


Figure: Explanation of the ALE-approach with the help of snapshots at two different times of a FSI-simulation: consequences of the mesh velocity w on the domain (mesh) movement. In regions where $w = 0$ the domain does not move and is fixed. The solid and the interface $\widehat{\Gamma}_i$ both move with $w = v$. In between (green arrows) the domain moves with $0 < w < v$.

Preparatory example

- Before we attack fluid-structure interaction, we illustrate variational-monolithic coupling again with a simple example.
- Let $u_1 : \Omega_1 \to \mathbb{R}$ and $u_2 : \Omega_2 \to \mathbb{R}$. Consider two Poisson-type problems:

$$\begin{aligned} -\Delta u_1 &= f_1 \quad \text{in } \Omega_1 \\ u_1 &= 0 \quad \text{on } \partial\Omega_1 \setminus \Gamma \\ -\Delta u_2 &= f_2 \quad \text{in } \Omega_2 \\ u_2 &= 0 \quad \text{on } \partial\Omega_2 \setminus \Gamma. \end{aligned}$$

- We need to impose two interface conditions on Γ:

$$\begin{aligned} u_1 &= u_2 \qquad \text{Kinematic condition,} \\ \partial_{n_1} u_1 &= \partial_{n_2} u_2 \qquad \text{Dynamic condition.} \end{aligned}$$

- **Question:** How does a variational form using variational-monolithic coupling look like?

Step 1: design function space

- Define

$$X = \{v \in H^1(\Omega) |\ v_1 = v_2 \text{ on } \Gamma,\ v_1 = 0 \text{ on } \partial\Omega_1 \setminus \Gamma,\ v_2 = 0 \text{ on } \partial\Omega_2 \setminus \Gamma\}$$

- Here, $\Omega := \bar{\Omega}_1 \cup \bar{\Omega}_2$.
- Here, the function v is defined over the entire domain Ω, with

$$v_1 = v|_{\Omega_1}, \quad v_2 = v|_{\Omega_2}$$

- The kinematic coupling condition is built into X.
- And the Dirichlet boundary conditions are build into X as well.

Step 2: Variational formulation

Proposition

Find $U = (u_1, u_2) \in X$ *such that*

$$A(U, \Psi) = F(\Psi) \quad \forall \Psi \in X,$$

where $\Psi = (\varphi_1, \varphi_2) \in X$ *and* $F(\Psi) := (f_1, \varphi_1) + (f_2, \varphi_2)$ *and*

$$A(U, \Psi) := A_1(u_1, \varphi_1) + A_2(u_2, \varphi_2)$$

with

$$A_1(u_1, \varphi_1) = (\nabla u_1, \nabla \varphi_1), \quad A_2(u_2, \varphi_2) = (\nabla u_2, \nabla \varphi_2).$$

Proof.

It holds after partial integration of the strong forms:

$$A_1(u_1, \varphi_1) + \int_\Gamma \partial_n u_1 \varphi_1 \, ds, \quad A_2(u_2, \varphi_2) + \int_\Gamma \partial_n u_2 \varphi_2 \, ds.$$

We sum-up:

$$\begin{aligned} &A_1(u_1, \varphi_1) + \int_\Gamma \partial_n u_1 \varphi_1 \, ds + A_2(u_2, \varphi_2) + \int_\Gamma \partial_n u_2 \varphi_2 \, ds \\ &= A_1(u_1, \varphi_1) + A_2(u_2, \varphi_2) + \int_\Gamma (\partial_n u_1 \varphi_1 + \partial_n u_2 \varphi_2) \, ds. \end{aligned}$$

cont. next slide ...

□

Proof, Part II

Proof continued from previous slide.

... We argue that $\Psi = (\varphi_1, \varphi_2) \in X$. In particular, therein $\varphi_1 = \varphi_2$ on Γ (kinemetic condition!). Also we use the fact that $n_1 = -n_2$. Then:

$$\int_\Gamma (\partial_{n_1} u_1 \varphi_1 + \partial_{n_2} u_2 \varphi_1)\, , ds = \int_\Gamma (\partial_{n_1} u_1 + \partial_{n_1} u_2)\varphi_1 \, ds$$

Our second coupling condition $\partial_{n_1} u_1 + \partial_{n_1} u_2 = 0$ comes into play now and therefore the integral on Γ vanishes:

$$\int_\Gamma (\partial_{n_1} u_1 + \partial_{n_1} u_2)\varphi_1 \, ds = 0.$$

Therefore, the sum of both semi-linear forms does not contain any terms on the interface Γ and consequently, we have

$$A(U, \Psi) := A_1(u_1, \varphi_1) + A_2(u_2, \varphi_2).$$

□

- The last four slides are crucially important to understand variational monolithic coupling!
- In the variational form, we **do not see** anymore the dynamic condition, because it is exactly fulfilled on the variational level, and therefore vanishes explicitly. But implicitly it is of course present!

Recalling the equations from Class 1 (fluids)

- Incompressible, isothermal Navier-Stokes equations:

$$\rho_f \partial_t v_f + \rho_f (v_f \cdot \nabla) v_f - 2\mathrm{div}(\rho_f \nu_f D(v_f)) + \nabla p_f = \rho_f f, \quad \text{in } \Omega_f,\ t \in I,$$
$$\mathrm{div}\, v_f = 0, \quad \text{in } \Omega_f,\ t \in I.$$

- The initial conditions are prescribed as

$$v(0) = v_0 \quad \text{in } \Omega \times \{t = 0\}.$$

- Dirichlet conditions are prescribed on no-slip boundaries and inflow boundaries:

$$v_f = v_{in} \quad \text{on } \Gamma_{f,in}$$
$$v_f = 0 \quad \text{on } \Gamma_{f,D}.$$

  No-slip boundaries might be rigid walls for example in modelling channel flow. We also notice that such a Dirichlet condition is seen in the velocity domain on the interface Γ_i in the case of a fluid-structure interaction setting, i.e.,

$$v_f = v_s \quad \text{on } \Gamma_i.$$

- The second natural type are Neumann boundary conditions (applied stresses):

$$\sigma_f n_f = h \quad \text{on } \Gamma_{f,N} \subset \partial\Omega_f,$$

  with a given vector-valued function $h = h(x, t)$.

Recalling the equations from Class 1 (solids)

- For (nonlinear) elasto-dynamics, we have: Find vector-valued displacements $\hat{u} : \widehat{\Omega} \to \mathbb{R}^3$ such that

$$\hat{\rho}_s \partial_t^2 \hat{u}_s - \widehat{\mathrm{div}}(\widehat{F}\widehat{\Sigma}) = \hat{\rho}_s \hat{f}, \quad \text{in } \widehat{\Omega}_s,\ t \in I,$$

  with $\widehat{\Sigma} := \widehat{\Sigma}_s^{STVK}$.

- Initial conditions:

$$\hat{u}_s(0) = \hat{u}_0 \quad \text{in } \widehat{\Omega}_s \times \{t = 0\}$$
$$\hat{v}_s(0) = \hat{v}_0 \quad \text{in } \widehat{\Omega}_s \times \{t = 0\}$$

- As for the fluid equations, we prescribe Dirichlet boundary conditions (fixing the displacements):

$$\hat{u}_s = \hat{g} \quad \text{on } \widehat{\Gamma}_{s,D} \subset \partial\widehat{\Omega}_s,$$

  where $\hat{g}$ is a given function.

- We can also employ Neumann boundary condition (surface stresses):

$$\widehat{F}\widehat{\Sigma}\hat{n}_s = \hat{J}\widehat{\sigma}_s \widehat{F}^{-T}\hat{n}_s = \hat{h} \quad \text{on } \widehat{\Gamma}_{s,N} \subset \partial\widehat{\Omega}_s,$$

  in which $\hat{h}$ is a given vector-valued function. Such a condition is seen from the structure side on the interface in case of a fluid-structure interaction problem, i.e.,

$$\widehat{F}\widehat{\Sigma}\hat{n}_s = \hat{J}\widehat{\sigma}_f \widehat{F}^{-T}\hat{n}_f \quad \text{on } \widehat{\Gamma}_i.$$

Variational forms for FSI

- We now apply the previous concepts to FSI
- Design function spaces
- Multiply each equation with a test function and integrate
- Little (big!) problem: fluid flows and solid mechanics are defined in different coordinate systems (class 1)
- Use therefore ALE approach: rewrite fluid equations into a Lagrangian-type system

Function spaces

Definition (Spaces for fluid-structure interaction)

We set:

$$\hat{L}_f := L^2(\widehat{\Omega}_f), \quad \hat{L}_s := [L^2(\widehat{\Omega}_s)]^d, \quad \hat{L}_f^0 := L^2(\widehat{\Omega}_f)/\mathbb{R},$$
$$\hat{L}_s^0 := L^2(\widehat{\Omega}_s)/\mathbb{R}, \quad \hat{V}_f^0 := \hat{V}_{\widehat{\Omega}_f}^0, \quad \hat{V}_s^0 := \hat{V}_{\widehat{\Omega}_s}^0.$$

Specifically, we introduce the trial and the test space of the velocity variables (recall that the velocity is d-dimensional) in the fluid domain,

$$\hat{V}_{f,\hat{v}}^0 := \{\hat{v}_f \in [H_0^1(\widehat{\Omega}_f)]^d : \hat{v}_f = \hat{v}_s \text{ on } \widehat{\Gamma}_i\}.$$

Moreover, we introduce the trial and the test spaces for the mesh movement using ALE in the fluid domain,

$$\hat{V}_{f,\hat{u}}^0 := \{\hat{u}_f \in [H_0^1(\widehat{\Omega}_f)]^d : \hat{u}_f = \hat{u}_s \text{ on } \widehat{\Gamma}_i\},$$
$$\hat{V}_{f,\hat{u},\widehat{\Gamma}_i}^0 := \{\hat{\psi}_f \in [H_0^1(\widehat{\Omega}_f)]^d : \hat{\psi}_f = \hat{\psi}_s \text{ on } \widehat{\Gamma}_i \subset \partial X\}.$$

Navier-Stokes in ALE coordinates

- Using ALE, rewrite flow equations into a Lagrangian-type system
- Formulate everything in the reference configuration using the transformation theorem (see page 18), i.e.,

$$\int_\Omega f\,dx = \int_{\widehat{\Omega}} \hat{f}\,\hat{J}\,d\hat{x}, \quad \text{with} \quad \hat{J} = \det \widehat{F}, \quad \widehat{F} = I + \widehat{\nabla}\hat{u}.$$

Formulation (ALE_{fx} fluid problem)

Find $\{\hat{v}_f, \hat{p}_f\} \in \{\hat{v}_f^D + \hat{V}_f^0\} \times \hat{L}_f^0$ *such that the initial data* $\hat{v}_f(0) = \hat{v}_f^0$ *are satisfied, and for almost all time steps* $t \in I$ *holds:*

$$\begin{aligned}\hat{\rho}_f(\hat{J}\partial_t \hat{v}_f, \hat{\psi}^v)_{\widehat{\Omega}_f} + \hat{\rho}_f(\hat{J}\widehat{F}^{-1}(\hat{v}_f - \hat{w})\cdot\widehat{\nabla}\hat{v}_f, \hat{\psi}^v)_{\widehat{\Omega}_f} + (\hat{J}\widehat{\sigma}_f\widehat{F}^{-T}, \widehat{\nabla}\hat{\psi}^v)_{\widehat{\Omega}_f} \\ - \langle \hat{J}\hat{g}_f\widehat{F}^{-T}\hat{n}_f, \hat{\psi}^v\rangle_{\widehat{\Gamma}_{f,N}} - \langle \hat{J}\widehat{\sigma}_f\widehat{F}^{-T}\hat{n}_f, \hat{\psi}^v\rangle_{\widehat{\Gamma}_i} - \hat{\rho}_f(\hat{J}\hat{f}_f, \hat{\psi}^v)_{\widehat{\Omega}_f} = 0, \\ (\widehat{div}\,(\hat{J}\widehat{F}^{-1}\hat{v}_f, \hat{\psi}^p)_{\widehat{\Omega}_f} = 0,\end{aligned}$$

for all $\hat{\psi}^v \in \hat{V}_f^0$ *and* $\hat{\psi}^p \in \hat{L}_f^0$, *and with the transformed Cauchy stress tensor*

$$\widehat{\sigma}_f = -\hat{p}_f\hat{I} + 2\hat{\rho}_f\nu_f\widehat{D}(\hat{v}_f) = -\hat{p}_f\hat{I} + 2\hat{\rho}_f\nu_f(\widehat{\nabla}\hat{v}_f\widehat{F}^{-1} + \widehat{F}^{-T}\widehat{\nabla}\hat{v}_f^T).$$

Coupling conditions on interface Γ_i

- Kinematic coupling condition (physics):

$$\hat{v}_f = \hat{v}_s \quad \text{on } \widehat{\Gamma}_i.$$

- Dynamic coupling condition (physics):

$$\hat{J}\hat{\sigma}_f \widehat{F}^{-T} \hat{n}_f - \widehat{F}\widehat{\Sigma}\hat{n}_s = 0 \quad \text{on } \widehat{\Gamma}_i.$$

- Geometric coupling condition:

$$\hat{u}_f = \hat{u}_s \quad \text{on } \widehat{\Gamma}_i.$$

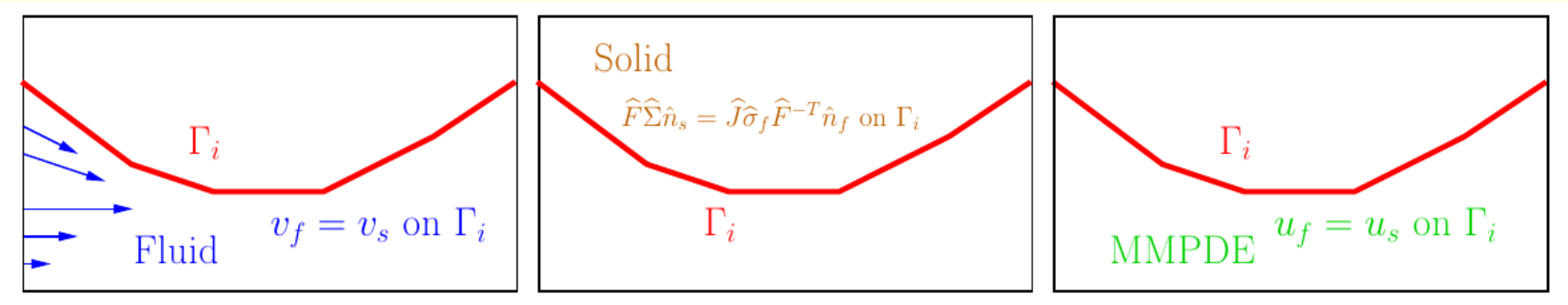


Figure: Illustration of the three coupling conditions on Γ_i (respectively its corresponding definition on the fixed $\widehat{\Gamma}_i$): v_f is required to solve the fluid system, $\widehat{F}\Sigma_s\hat{n}_s$ is required for the solid system, and u_f is necessary for the MMPDE (mesh motion PDE).

Variational-monolithic coupling

- The continuity of flow conditions,

$$v_f = v_s,$$

  and the geometric coupling

$$u_f = u_s,$$

  are incorporated directly in the Sobolev spaces as usually done for Dirichlet conditions:

$$\hat{V}^0_{f,\hat{v}} := \{\hat{v}_f \in H^1_0(\widehat{\Omega}_f) : \hat{v}_f = \hat{v}_s \text{ on } \widehat{\Gamma}_i\},$$
$$\hat{V}^0_{f,\hat{u}} := \{\hat{u}_f \in H^1_0(\widehat{\Omega}_f) : \hat{u}_f = \hat{u}_s \text{ on } \widehat{\Gamma}_i\}.$$

- In variational-monolithic coupling, Neumann-like interface conditions, like the continuity of normal stresses are fulfilled exactly (see page 75) in a weak sense on the continuous level:

$$\langle \hat{J}\widehat{\sigma}_f \widehat{F}^{-T}\hat{n}_f, \hat{\varphi}\rangle - \langle \widehat{F}\widehat{\Sigma}\hat{n}_s, \hat{\varphi}\rangle = 0 \quad \forall \hat{\varphi} \in \hat{V}^0_{f,\hat{v}}.$$

- They are realized through interface integrals (but actually disappear in the later models because of their continuity in the weak (variational) sense), provided that $\varphi_f = \varphi_s$ on $\widehat{\Gamma}_i$ (see again the proof on page 75), as it is guaranteed with the space $\hat{V}^0_{f,\hat{v}}$ here.

Collecting all pieces

In order to derive the weak form a the FSI model, we now put our different pieces together:

Definition (Collecting all pieces for variational-monolithic FSI in ALE_{fx})

To build the FSI model, we use:

- Fluid momentum and mass conservation: the weak form of Navier-Stokes in ALE_{fx}
- Solid momentum in mixed form: the weak form of elasticity
- Fluid mesh motion for realizing the ALE transformation: the weak form $(\widehat{\sigma}_{\text{mesh}}, \widehat{\nabla}\hat{\psi}^u)_{\widehat{\Omega}_f}$ of one of the second-order mesh motion PDEs with $\widehat{\sigma}_{\text{mesh}} := \alpha\widehat{\nabla}\hat{u}_f$, $\alpha > 0$, or alternatively a fourth-order mesh motion PDE

Variational-monolithic ALE fluid-structure interaction

Formulation (Variational-monolithic ALE fluid-structure interaction in $\widehat{\Omega}$)

Find vector-valued velocities, vector-valued displacements and a scalar-valued fluid pressure, i.e., $\hat{U} := \{\hat{v}_f, \hat{v}_s, \hat{u}_f, \hat{u}_s, \hat{p}_f\} \in \hat{X}_D^0 := \{\hat{v}_f^D + \hat{V}_{f,\hat{v}}^0\} \times \hat{L}_s \times \{\hat{u}_f^D + \hat{V}_{f,\hat{u}}^0\} \times \{\hat{u}_s^D + \hat{V}_s^0\} \times \hat{L}_f^0$, *such that* $\hat{v}_f(0) = \hat{v}_f^0$, $\hat{v}_s(0) = \hat{v}_s^0$, $\hat{u}_f(0) = \hat{u}_f^0$, *and* $\hat{u}_s(0) = \hat{u}_s^0$ *are satisfied, and for almost all times* $t \in I$ *holds:*

$$\textit{Fluid momentum} \begin{cases} \hat{A}_1(\hat{U})(\hat{\psi}^v) := (\hat{J}\hat{\rho}_f \partial_t \hat{v}_f, \hat{\psi}^v)_{\widehat{\Omega}_f} + (\hat{\rho}_f \hat{J}(\widehat{F}^{-1}(\hat{v}_f - \hat{w})\cdot\widehat{\nabla})\hat{v}_f), \hat{\psi}^v)_{\widehat{\Omega}_f} + (\hat{J}\widehat{\sigma}_f \widehat{F}^{-T}, \widehat{\nabla}\hat{\psi}^v)_{\widehat{\Omega}_f} \\ + \langle \hat{\rho}_f \nu_f \hat{J}(\widehat{F}^{-T}\widehat{\nabla}\hat{v}_f^T \hat{n}_f)\widehat{F}^{-T}, \hat{\psi}^v \rangle_{\widehat{\Gamma}_{out}} = 0 \quad \forall \hat{\psi}^v \in \hat{V}_{f,\widehat{\Gamma}_i}^0, \end{cases}$$

$$\textit{Solid momentum, v-equation} \left\{ \hat{A}_2(\hat{U})(\hat{\psi}^v) := (\hat{\rho}_s \partial_t \hat{v}_s, \hat{\psi}^v)_{\widehat{\Omega}_s} + (\widehat{F}\widehat{\Sigma}, \widehat{\nabla}\hat{\psi}^v)_{\widehat{\Omega}_s} + (\widehat{\Sigma}_v(\hat{v}_s), \widehat{\nabla}\hat{\psi}^v)_{\widehat{\Omega}_s} = 0 \quad \forall \hat{\psi}^v \in \hat{V}_s^0, \right.$$

$$\textit{Fluid mesh motion} \left\{ \hat{A}_3(\hat{U})(\hat{\psi}^u) := (\widehat{\sigma}_{mesh}, \widehat{\nabla}\hat{\psi}^u)_{\widehat{\Omega}_f} = 0 \quad \forall \hat{\psi}^u \in \hat{V}_{f,\hat{u},\widehat{\Gamma}_i}^0, \right.$$

$$\textit{Solid momentum, u-equation} \left\{ \hat{A}_4(\hat{U})(\hat{\psi}^u) := \hat{\rho}_s(\partial_t \hat{u}_s - \hat{v}_s, \hat{\psi}^u)_{\widehat{\Omega}_s} = 0 \quad \forall \hat{\psi}^u \in \hat{L}_s, \right.$$

$$\textit{Fluid mass conservation} \left\{ \hat{A}_5(\hat{U})(\hat{\psi}^p) := (\widehat{div}\,(\hat{J}\widehat{F}^{-1}\hat{v}_f), \hat{\psi}^p)_{\widehat{\Omega}_f} = 0 \quad \forall \hat{\psi}^p \in \hat{L}_f^0. \right.$$

- The green terms represents the respective terms in the 'standard' coordinate systems

→ True for all solid equations, because they remain in Lagrangian coordinates

→ The fluid has been transformed from Ω to $\widehat{\Omega}$ using the ALE technology.

- The **red** terms are novel due to the ALE transformations.
- The **blue** term is a Neumann condition on the outflow boundary in channel flow, which corrects the outflow profile. Better known as part of the do-nothing condition.[10]

[10] Heywood/Rannacher/Turek, 1996

Compact abstract semi-linear form

Consequently, we can write in short form:

Formulation (Semi-linear form on the time-continuous, space-continuous level)

Find $\hat{U} \in \widehat{X}_0^D$ *such that* $\hat{v}_f(0) = \hat{v}_f^0$, $\hat{v}_s(0) = \hat{v}_s^0$, $\hat{u}_f(0) = \hat{u}_f^0$, *and* $\hat{u}_s(0) = \hat{u}_s^0$ *are satisfied, and for almost all time steps* $t \in I$ *holds:*

$$\hat{A}(\hat{U})(\hat{\Psi}) = 0 \quad \forall \hat{\Psi} \in \widehat{X}$$

with $\hat{\Psi} = \{\hat{\psi}_f^v, \hat{\psi}_s^v, \hat{\psi}_f^u, \hat{\psi}_s^u, \hat{\psi}_f^p\}$ *and* $\widehat{X} = \hat{V}_{f,\hat{v}}^0 \times \hat{L}_f \times \hat{V}_{f,\hat{u},\widehat{\Gamma}_i}^0 \times \hat{V}_s^0 \times \hat{L}_f^0$ *and*

$$\hat{A}(\hat{U})(\hat{\Psi}) := \hat{A}_1(\hat{U})(\hat{\psi}^v) + \hat{A}_2(\hat{U})(\hat{\psi}^v) + \hat{A}_3(\hat{U})(\hat{\psi}^u) + \hat{A}_4(\hat{U})(\hat{\psi}^u) + \hat{A}_5(\hat{U})(\hat{\psi}^p).$$

Summary and outlook

Variational formulation of FSI ✓

Discretization

Class 3

Paradigm of numerical modeling and discretization

- All the previous types of different equations we have seen before have different physical properties and these **properties on the continuous PDE level should be maintained as well as possible after numerical discretization.**
- Thus it is extremely important to **understand the properties of a differential equation** (at least to some extend) on the continuous level (ODE/PDE theory) in order to be able to derive and analyze appropriate algorithmic schemes.

Example

The wave equation (thus the elastodynamic solid equations) satisfy energy conservation on the continuous level. Think of the harmonic oscillator that would never stop if there were no air friction. Thus neglecting physical friction in the PDE equations, also the numerical simulation should not yield any damping on the harmonic oscillator and make it to stop.

Philosophy of discretization

- We explain the following steps in terms of a monolithic scheme (all equations are solved all-at-once)
- Formulation as a common semi-linear form that represents the variational (weak form; principle of virtual work) formulation:

Formulation

Let U be the unknown solution containing all solution variables, here $U = (u, v, p)$, while setting $u = u_f + u_s$ and $v = v_f + v_s$. As before, let X be the common function space. The task is: Find $U \in X$ such that

$$A(U)(\Psi) = 0 \quad \forall \Psi \in X.$$

Omitting the hats^here.

- Four solution steps:
  1. (Adaptive) discretization in time;
  2. (Adaptive) discretization in space;
  3. (Adaptive) nonlinear iteration: Newton's method;
  4. (Adaptive) solution of inner linear systems.

Temporal discretization for a parabolic equation: One-Step-θ

- Explanation in terms of a parabolic problem (the heat equation):

$$\partial_t u - \Delta u = f,$$

be given.

- Time discretization using finite differences yields: Find $u := u^n$:

$$\frac{u - u^{n-1}}{k} - \theta \Delta u - (1-\theta)\Delta u^{n-1} = \theta f + (1-\theta) f^{n-1},$$

where $k = t_n - t_{n-1}$ and with $\theta \in [0,1]$ with $0 \leq \theta < 0.5$ (explicit) and $0.5 \leq \theta \leq 1$ (implicit schemes).

- Re-arranging terms yields:

$$\frac{u - u^{n-1}}{k} - \theta \Delta u = (1-\theta)\Delta u^{n-1} + \theta f + (1-\theta) f^{n-1},$$

where $k = t^n - t^{n-1}$ is the time step size.

- Well-known choices for θ:

$\theta = 0$ Explicit Euler

$\theta = 0.5$ Crank-Nicolson (Trapezoidal rule)

$\theta = 1$ Implicit Euler

One-Step-θ

Formulation (One-Step-θ for PDE multiphysics)

Given $\hat{U}^{n-1}, \widehat{F}^n$, and $\widehat{F}^{n-1}$, we seek $\hat{U}^n = \{\hat{v}_f^n, \hat{v}_s^n, \hat{u}_f^n, \hat{u}_s^n, \hat{p}_f^n\} \in \widehat{X}^0$ by employing One-step-θ splitting in exactly the same way

$$\begin{aligned}&\hat{A}_T(\hat{U}^{n,t})(\hat{\Psi}) + \theta\hat{A}_E(\hat{U}^n)(\hat{\Psi}) + \hat{A}_I(\hat{U}^n)(\hat{\Psi})\\ &= -(1-\theta)\hat{A}_E(\hat{U}^{n-1})(\hat{\Psi}) + \theta\widehat{F}^n(\hat{\Psi}) + (1-\theta)\widehat{F}^{n-1}(\hat{\Psi})\end{aligned}$$

Examples:

- Heat equation from before:

$$\begin{aligned}\hat{A}_T(u^{n,t})(\psi) &= (\frac{u-u^{n-1}}{k}, \psi)\\ \hat{A}_E(u^{n,t})(\psi) &= (\nabla u, \nabla\psi)\\ \widehat{F}^n(\psi) &= (f, \psi)\end{aligned}$$

- Example for an implicit term: for instance pressure term in Navier-Stokes equations:

$$\hat{A}_I((v,p),\psi) = (-pI, \nabla\psi)$$

- The entire FSI system can be classified into these terms and consequently fully time-discretized.
- It remains the question how to choose θ?

Temporal discretization: A-stable schemes

From the model problem

$$y'(t) = \lambda y(t), \quad y(t_0) = y_0,\ \lambda \in \mathbb{C},$$

we know the solution $y(t) = y_0 \exp(\lambda t)$. For $t \to \infty$ the solution is characterized by the sign of $Re\,\lambda$:

$$\begin{aligned} Re\,\lambda < 0 \quad &\Rightarrow |y(t)| = |y_0| \exp(Re\,\lambda) \to 0, \\ Re\,\lambda = 0 \quad &\Rightarrow |y(t)| = |y_0| \exp(Re\,\lambda) = |y_0|, \\ Re\,\lambda > 0 \quad &\Rightarrow |y(t)| = |y_0| \exp(Re\,\lambda) \to \infty. \end{aligned}$$

For a *good* numerical scheme, the first case is particularly interesting whether such a scheme can produce a bounded discrete solution when the continuous solution has this property.

Definition (A-stability)

Let $\{y_n\}_n$ the sequence of solutions of a difference method for solving the ODE model problem. Then, this method is A-stable if for arbitrary $\lambda \in \mathbb{C}^- = \{\lambda : Re(\lambda) \leq 0\}$ the approximate solutions are bounded (or even contractive) for arbitrary, but fixed, step size k. That is to say:

$$|y_{n+1}| \leq |y_n| < \infty \quad \text{for } n = 1, 2, 3, \ldots$$

Temporal discretization: A-stable schemes (cont'd)

- For $\lambda = -10$, we define $R(z) = 1 + z$ where $z = \lambda k$, we have the stability interval for the forward Euler scheme ($\theta = 0$) (not A-stable!!):

$$|1 + z| \leq 1 \quad \Rightarrow \quad |1 - 10k| \leq 1$$

  Thus, we need to choose a k that fulfills the previous relation.

- The backward Euler scheme ($\theta = 1$) and the Crank-Nicolson scheme ($\theta = 0.5$) are A-stable though and do not require a time step restriction [11]

- Example:

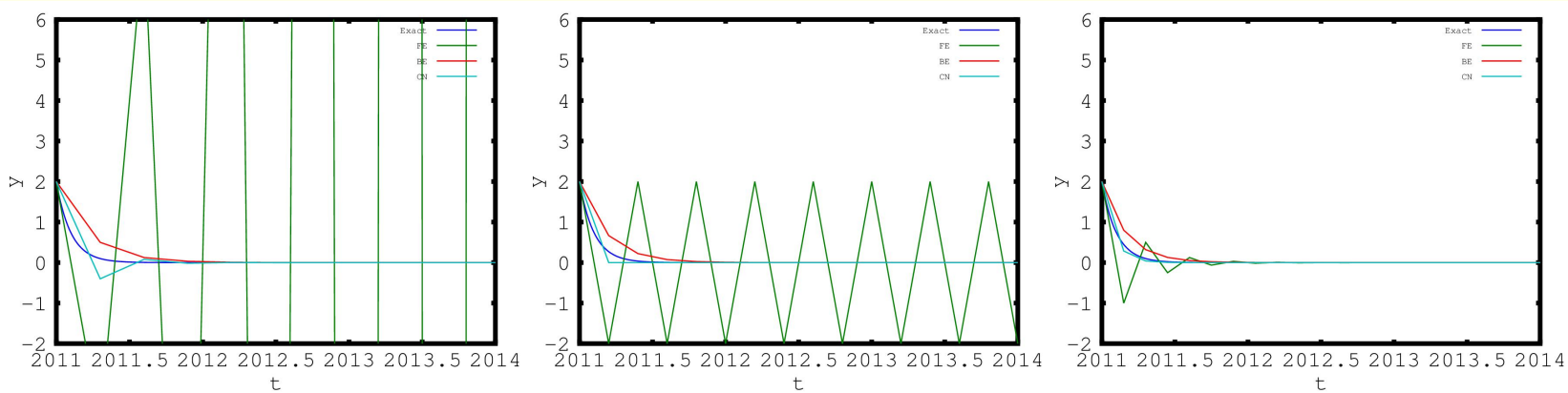


Figure: Example: Blow-up, constant zig-zag non-convergence, and convergence of the forward Euler method.

[11] Exceptions for $\theta = 0.5$ are known; see Rannacher 1986 and below page 97.

Refinements of A-stability (I)

1. **A-stability:**
$$|R(z)| \leq 1, \quad z \leq 0$$
This ensures that the discrete solution remains bounded.

2. **Strict A-stability:**
$$|R(z)| \leq 1 - ck, \quad z \leq -1.$$
The discrete solution is bounded for inhomogeneous right hand sides or irrugular initial data.

3. **Strong A-stability:**
$$|R(z)| \leq \kappa < 1, \quad z \leq -1.$$
Damping of high-frequency errors and robust against local errors.

4. **Numerical dissipation.** For physical oscillations (e.g., wave equation; our solid equation in fact), the numerically introduced dissipation should be as small as possible. This means:
$$R(\pm i) \sim 1.$$

Refinements of A-stability (II)

- It is clear that only implicit schemes can be A-stable and is well-known from lectures for numerical methods for ordinary differential equations.
- For the implicit Euler scheme, it holds in $z = -i$:

$$R(z) = \frac{1}{1-z} \quad \rightarrow \quad |R(-i)| = |\frac{1}{1+i}| = \frac{1}{\sqrt{2}} < 1.$$

- We nicely see that the damping in the implicit Euler scheme is too strong and will damp out physical oscillations. Therefore, this scheme is not suited for wave equations.
- Now, let us consider the Crank-Nicolson scheme:

$$\lim_{z\to\infty} \frac{1+\frac{1}{2}z}{1-\frac{1}{2}z} = -1, \quad \rightarrow \quad |R(-i)| = |\frac{1-\frac{1}{2}i}{1+\frac{1}{2}i}| = 1.$$

- Thus, physical oscillations can be perfectly represented. However, any small disturbances (e.g., even round-off errors), can lead to a blow-up of the solution.
- One possibility is a k-shift towards the implicit side:

$$\theta = 0.5 + k, \qquad \text{for } k \ll 0.5 \quad \text{(characteristic time step size)}$$

This scheme is strictly A-stable and still of second order as proven by Rannacher in 1986.

Illustration of physical oscillations versus numerical instabilities

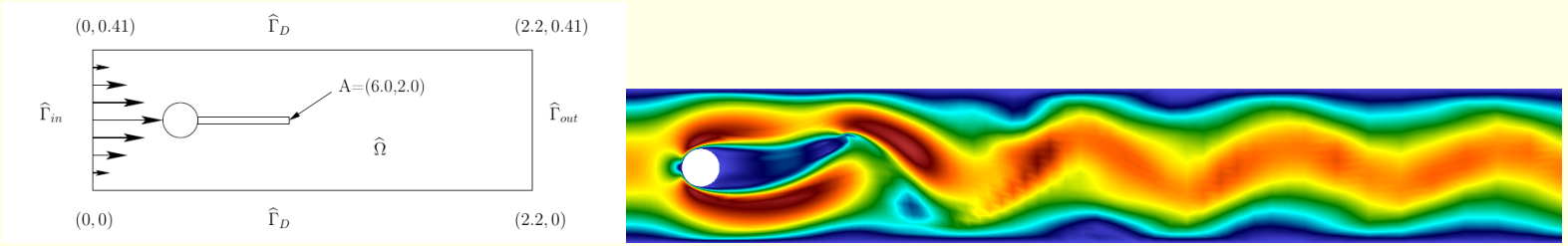
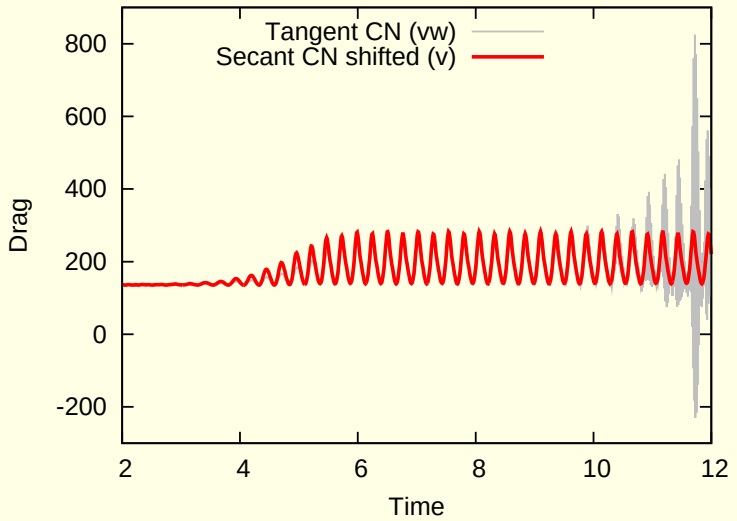


Figure: Fluid flow (Navier-Stokes) interacts with an elastic beam. Due to a non-symmetry of the cylinder, the beam starts oscillating. These oscillations are physical!

- Observe the tip of the elastic beam!

→ Physical oscillations! Shown in red color for a 'good' numerical scheme.

- The grey numerical scheme exhibits at some time around $t \approx 10$ micro-oscillations which are due to numerical instabilites. Finally the grey numerical scheme has a blow-up and yields garbage solutions.



On the other hand, the backward Euler scheme $\theta = 1$ ...

... is too dissipative and suppresses the physical oscillations:

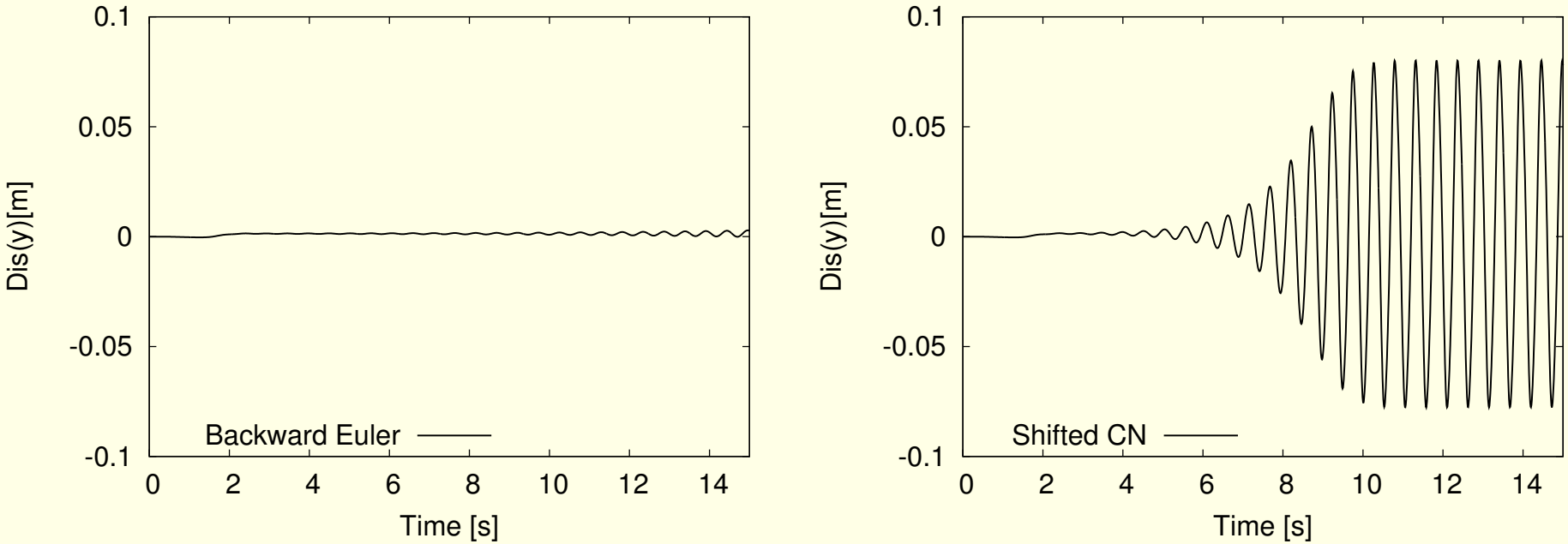


Figure: Comparison of the u_y displacement using the backward Euler scheme (left) and the shifted Crank-Nicolson scheme (right). These findings show that it is important to choose the 'correct' time-stepping scheme. For nonstationary flow, the backward Euler scheme is too dissipative and is not able to compute oscillatory flow as the shifted CN scheme does.

Spatial discretization based on finite elements

Disclaimer: An introduction to finite elements is a one-semester class. Here, we only introduce very little and refer to the lecture notes mentioned on page 5 (T. Wick; Numerical methods for PDEs, 2020).

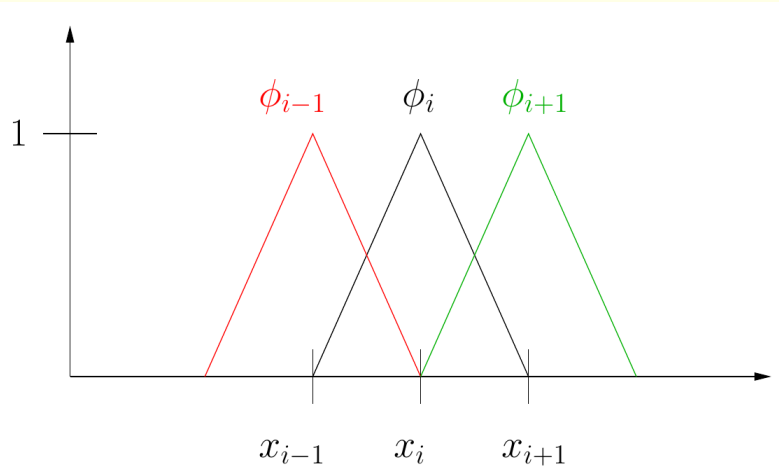
Definition

A finite element is a triple (K, P_K, Σ) where

- K is an element, i.e., a geometric object (in 1D an interval);
- $P_k(K)$ is a finite dimensional linear space of polynomials defined on K;
- Σ, not introduced so far, is a set of degrees of freedom (DoF), e.g., the values of the polynomial at the vertices of K.

These three ingredients yield a uniquely determined polynomial on an element K.

Illustration of finite elements in 1D. The elements are $K_i = [x_i, x_{i+1}]$. The polynomials are linear functions (so-called hat functions since the look like a hat). The DoFs are points x_i, x_{i+1} where the linear function is fixed and therefore uniquely defined.



Finite elements in 2D on quadrilaterals

Let K be an element with four vertices $a^i, i = 1, \ldots 4$ and the sides parallel to the coordinate axes in $\mathbb{R}^2$.

Definition (A basis of Q_1 (bilinear polynomials))

A basis of the space $Q_1(K)$ on an element K is given by

$$Q_1 := Q_1(K) = \{\phi_1, \phi_2, \phi_3, \phi_4\}$$

with the basis functions

$$\phi_1 \equiv 1, \quad \phi_2 = x_1, \quad \phi_3 = x_2, \quad \phi_4 = x_1 x_2.$$

The dimension is $dim(Q_1) = 4$. Clearly, any function from Q_1 can be represented through the linear combination:

$$p(x) = a_{00}\phi_1 + a_{10}\phi_2 + a_{01}\phi_3 + a_{11}\phi_4.$$

Proposition

A Q_1 function is uniquely determined by the values of the nodal points a^i.

Bilinear and biquadatric elements

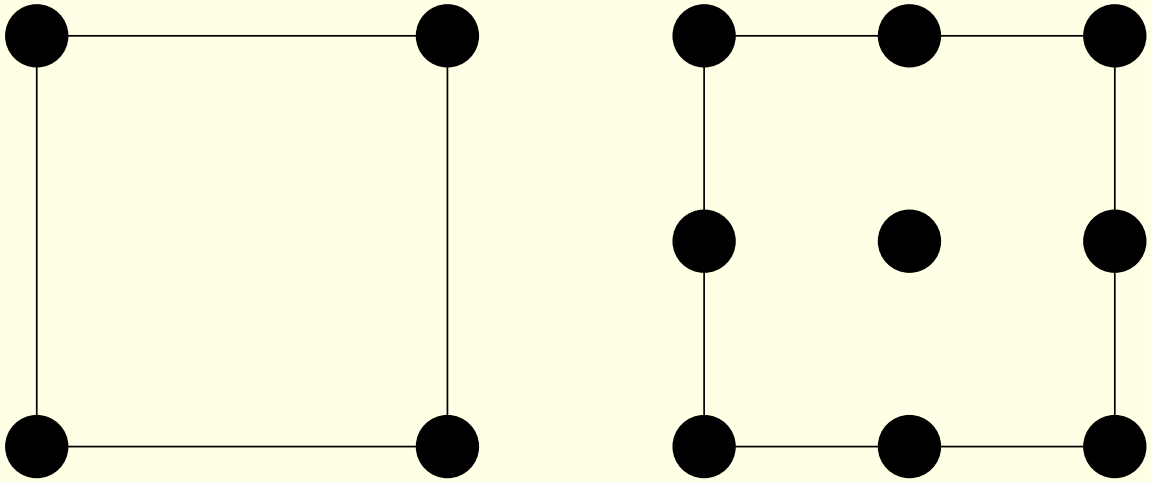


Figure: Q_1 (bilinear) and Q_2 (biquadratic) elements on a quadrilateral in 2D.

- Shape-regular meshes.
- A mesh consists of quadrilateral/hexadydra cells $\hat{K}$. They perform a non-overlapping cover of the computation domain $\hat{\Omega} \subset \mathbb{R}^d$, $d = 2, 3$. The corresponding mesh is given by $\hat{\mathcal{T}}_h = \{\hat{K}\}$.
- The discretization parameter is denoted by $\hat{h}$ and is a cell-wise constant that is given by the diameter $\hat{h}_{\hat{K}}$ of the cell $\hat{K}$.

On the spatial stability: the inf-sup condition

- For spatial stability, the velocity space must be sufficiently bigger than the pressure space
- This is known as inf-sup condition[12] and must be satisfied on the continuous and the discrete level
- On the discrete level, the results can be for instance the Taylor-Hood element Q_2^c / Q_1^c or the Q_2^c / P_1^{dc} element.
- In case equal-order elements are used, pressure stabilization must be employed

[12] Girault/Raviart, Finite Element method for the Navier-Stokes equations, 1986

On the spatial stability: the inf-sup condition

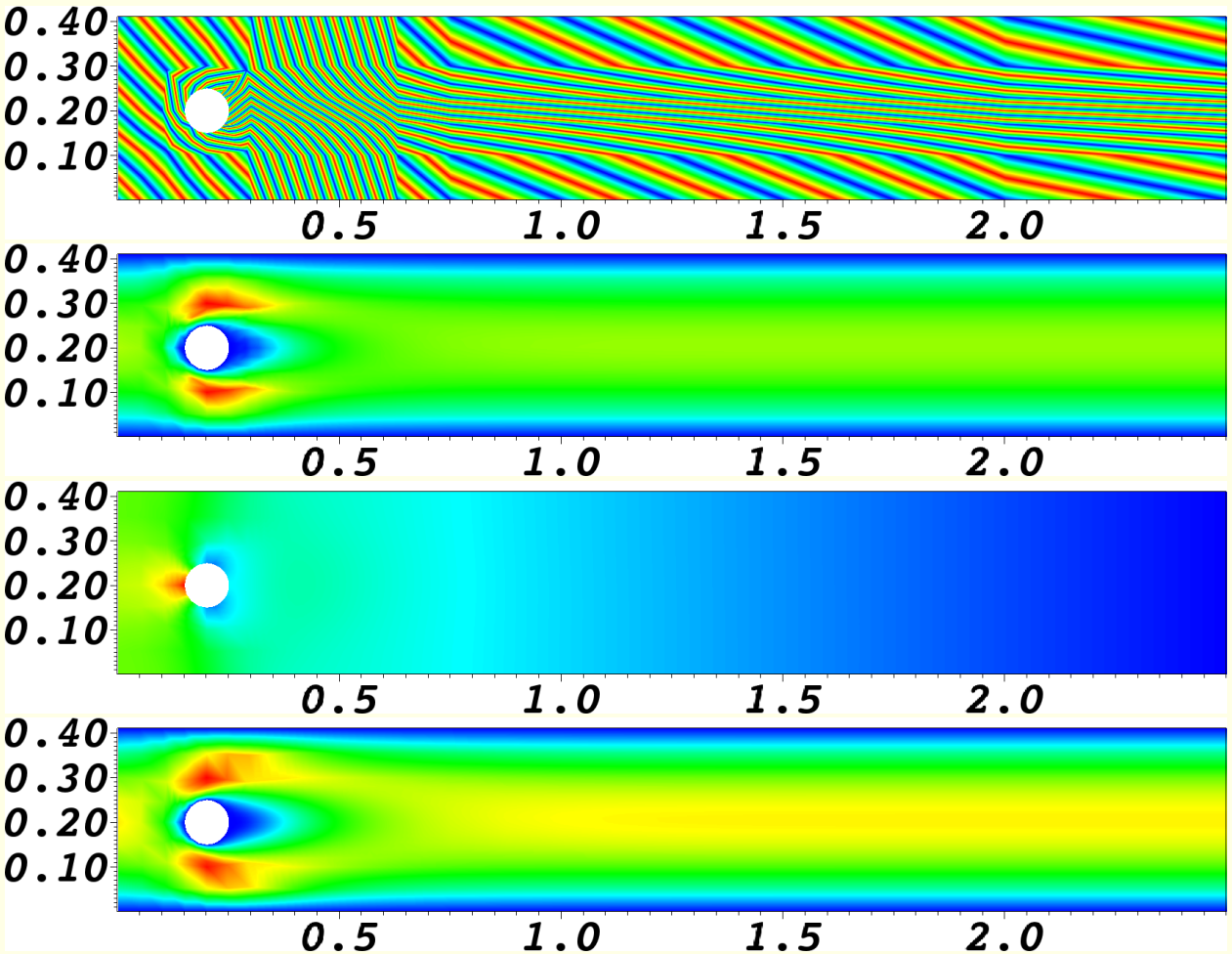


Figure: Fluid flow in channel with a hole: Illustration of the violation of the inf-sup condition using the unstable Q_1^c/Q_1^c discretization: the pressure field oscillates (top left) whereas the corresponding velocity field (top right) is 'more or less' okay in the picture norm. On the bottom, the inf-sup stable Taylor-Hood element Q_2^c/Q_1^c results in a smooth pressure field (bottom left). The corresponding flow field is shown at bottom right. *(As footnote: The picture norm does only provide a rough idea of a sitation. It is neither a proof nor computational evidence nor evidence of numerical convergence or rigorous correctness of a result!)*

Why? How? What?[13]

Why?

- Allowing for flexible algorithms, efficient evaluations of solutions or parts of solutions
- Done by relatively low computational cost
- More important than ever because of big data, multiphysics applications, many unknowns

How?

- Goal-oriented algorithms
- Approach motivated from optimization (Lagrange formalism)
- Adjoint equations
- Combining with parallel computing

What?

- A posteriori error estimates
- Differential equations: temporal and spatial mesh adaptivity; choosing non-uniform time step sizes and spatial meshes for reducing discretization errors
- Controlling stopping criteria in the iterative linear and nonlinear solvers
- Controlling model errors

[13] Goldenstein/Wick; 2020, see page 5 for the full reference

Examples: spatial mesh adaptivity

- The following is for the spatial discretization error U_h. The temporal discretization error U_k could have been considered as well.
- Overall goal: reduction of computational cost, while computing 'something' with sufficient accuracy.
- This 'something' can be the solution itself, the interface in multiphysics problems, or a certain technical quantity of interest (e.g., drag/lift in flow problems).
- How and where to refine the mesh?

  $\rightarrow$ an error estimator η will tell us.
- A general way to express such wishes is in terms of goal functionals $J(U)$
- Example: Drag coefficient for fluid flow (Navier-Stokes): $J(\hat{U}) = \int_{\Gamma_i} \widehat{\sigma}_f \cdot n e_1 \, ds$, where $\widehat{\sigma}_f$ is the fluid's Cauchy stress tensor (class 1), n the normal vector, and e_1 the unit vector in $x - direction$.

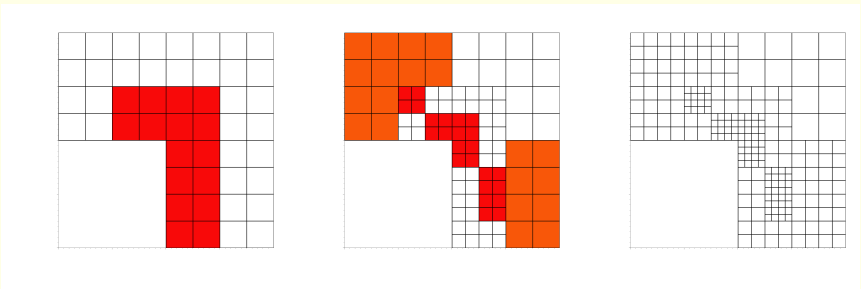


Figure: Meshes on level 0, level 0 and 1, level 0,1,2. Refinining only the regions where 'something' happens which is indicated by an error estimator η.

Example: spatial mesh adaptivity

Definition (Error estimator)

To estimate the iterative or discretization error, we work with an error estimator η. This error estimator is usually based on a posteriori error estimates.

We now have

$$J(U) - J(U_h) \sim \eta$$

A key question is whether η has "something" to do with the true error:

1. Robustness/reliability: $|J(U) - J(U_h)| \leq c_2\,\eta$, where $c_2 > 0$
2. Efficiency: $c_1\,\eta \leq |J(U) - J(U_h)|$, where $c_1 > 0$

This yields:

Definition (Effectivity index)

$I_{eff} := \frac{\eta}{|J(U)-J(U_h)|}$, for which asymptotically, we hope for $I_{eff} \to 1$.

Adaptive scheme:

1. Solve the problem: U_h
2. Estimate error with η
3. Mark regions of domain with high error indicators
4. Adapt the mesh (FEM) or solution algorithm

The adjoint, our friend

- So far we had a goal functional $J(\cdot)$ as well as a Task (PDE, ODE)

$$A(U) = b \quad \text{Problem on continuous level}$$
$$A_h(U_h) = b_h \quad \text{Discretized problem in space and time}$$

- We derive now an estimator η that uses U_h and an additional adjoint equation that measures the sensitivity w.r.t. $J(U_h)$
- To this end, we formulate the optimization problem

$$\min [J(U) - J(U_h)] \quad s.t. \quad A(U) = b$$

- One possiblity to solve such constraint optimization problems is to introduce a Lagrange multiplier such that we can define the Lagrangian $L : V \times V \to \mathbb{R}$:

$$L(U, Z) = (J(U) - J(U_h)) - (A(U) - b)Z$$

  where U is the primal variable and Z the Lagrange multiplier (or adjoint variable).
- The idea is now to compute stationary points as functions of U and Z. These are always saddle points of $L(U, Z)$.

Application to variational formulations

- Given
$$\min(J(U) - J(U_h)) \quad s.t. \quad A(U)(\Psi) = F(\Psi).$$
- We use the solution approach: Find $U, Z \in V$:
$$L(U,Z) = (J(U) - J(U_h)) - A(U)(Z) + F(Z)$$
- A necessary condition for a minimum is
$$L' = 0$$
- It follows for the (Fréchet) derivatives
$$L'_U(U,Z) = J'(U)(\varphi) - A'(U,Z)(\varphi) = 0$$
$$L'_Z(U,Z) = -A'(U,Z)(\Psi) + F(\Psi) = 0$$
- Yielding for the **adjoint** and **primal** problems
$$\text{Find } Z \in V: \quad A'(U,Z)(\Phi) = J'(U)(\Phi) \quad \Phi \in V$$
$$\text{Find } U \in V: A'(U,Z)(\Psi) = F(\Psi) \quad \Psi \in V.$$

A first error estimator

- Set $\Phi = e = U - U_h$. Then:

$$J(e) = A(e, Z) = A(U - U_h, Z) = A(U, Z) - A(U_h, Z) + R^3$$

- A computable error estimator reads:

$$\eta := A(U, Z) - A(U_h, Z) = F(z) - A(U_h, Z)$$

- Until now U and Z are unknown and we cannot evaluate η. We will now derive these unknowns.

  i) The "solution" to the unknown u is more or less easy. We have already

  $$\eta = F(Z) - A(U_h, Z)$$

  where F and U_h are known.

  ii) The only unknown is now Z where the solution is tricky. Just approximating $Z \in V$ by $Z_h \in V_h$ will yield a bad error estimator. The trick ist to use the Galerkin orthogonality and plug-in Z_h: $A(U - U_h, Z_h) = 0$. Then:

  $$\eta = F(Z - Z_h) - A(U_h, Z - Z_h)$$

  Still Z is unknown. Simply using $Z := Z_h$ would yield $Z_h - Z_h \equiv 0$. Thus, we must use a higher-order solution, for instance with higher-order interpolation or a higher-order finite element: $Z := Z_h^{(2)}$. Then:

  $$\eta = F(Z_h^{(2)} - Z_h) - A(U_h, Z_h^{(2)} - Z_h)$$

- With a bit more work, this error estimator can be extended to control also linear and nonlinear iteration errors [14]

[14] Meidner/Vihharev/Rannacher, 2009; Vihharev/Rannacher, 2013

A simplified error estimator for multiphysics

- We finally obtain:

Proposition

The goal functional error between U and U_h can be approximated via

$$J(U) - J(U_h) \approx \eta = F(Z_h^{(2)} - i_h Z^{(2)}) - A(U_h, Z_h^{(2)} - i_h Z^{(2)}),$$

where $F(\cdot)$ is a possible right hand side and $A(\cdot)(\cdot)$ is nothing else than the FSI-semi-linear form from page 86.

- What does this mean for multiphysics (i.e., FSI)?

1. We know $\hat{A}(\hat{U})(\hat{\Psi})$ and can solve it as before
2. Need to solve second problem, the adjoint: Find $\hat{Z}^{(2)}$ such that $\hat{A}'(\hat{U}, \hat{Z}^{(2)})(\hat{\Psi}) = J'(\hat{U})(\hat{\Psi})$ for $\hat{\Psi} \in \widehat{X}^{r+1}$, $r+1$ indicates that we must use a higher order.
3. Plug-in both solutions into the error estimator η.
4. Localize η to obtain indicators on each mesh element
5. Refine the mesh

Final comments to goal-oriented adjoint-based error estimation

Pros

- Adjoint-based error estimation allows to measure precisely at a low computational cost specific functionals of interest $J(U)$ and not only general error norms of the form $\|U - U_{kh}\|$.
- The developed adjoint equation is the same as used in gradient-based optimization[15]
- The adjoint is always a linear equation.
- Can be extended to fully adaptive schemes in which discretization, linear and nonlinear iteration errors can be controlled [16]
- Can we further utilized for multigoal-oriented error estimation (attractive in multiphysics!) [17]

Prize to pay:

- We must compute a second solution $Z \in V$.
- For time-dependent problems, the adjoint is running backwards in time.
- For nonlinear primal problems, the primal solution must be stored since it enters the adjoint solution.
- From the theoretical point of view, we cannot proof convergence of the adaptive scheme for general goal functionals.

[15] Becker, 2004; Becker/Meidner/Vexler, 2007
[16] Meidner/Rannacher/Vihharev, 2009; Rannacher/Vihharev, 2013
[17] Endtmayer/Langer/Wick, 2017-2020

General comments to programming code verification

Three steps (with increasing level of difficulty) after having constructed an a posteriori error estimator η that can be localized and used for local mesh adaptivity. These steps hold true for classical norm-based error-estimation $\|U - U_h\|$ or goal-oriented error estimation $J(U) - J(U_h)$.

1. Does the solution make sense?
   - If possible test your code with an acknowledged benchmark configuration and verify whether $J(U_h)$ matches the benchmark values in a given range and on a sequence of at least **three meshes**. This first step can be performed with uniform and adaptive mesh refinement. In time-dependent problems, please compute at least with three different time step sizes.
   - If no benchmark configuration available, study a **simple, prototype** configuration and observe whether the solution makes sense.
2. Do the true error $J(U) - J(U_h)$ and the error estimator η descrease under mesh refinement?
   - Compute $J(U)$ either on a uniformly-refined super-fine mesh or even analytically (i.e., a manufactured solution). Compute the error $J(U) - J(U_h)$ and observe whether the error is decreasing.
   - If a priori estimates are available, see if the orders of convergence are as expected. But be careful, often goal functionals are nonlinear, for which rigorous a priori error estimates are not available.
3. Compare η and $J(U) - J(U_h)$ in terms of the effectivity index I_{eff}. Do we asymptotically obtain something around $I_{eff} \approx 1$? For nonlinear problems, one easily observes $0.5 \leq I_{eff} \leq 10$, which might be still okay depending on the problem.

Outlook to the next class

Numerical modeling including discretization in time, space and adpative methods ✓

Solution of the finite-dimensional discrete FSI systems $\hat{A}(\hat{U}_h^n)(\hat{\Psi}_h) = 0$

Class 4

What do we have?

- In Class 3, we obtained the temporal and spatially discretized nonlinear form:

$$\hat{A}(\hat{U}_h^n)(\hat{\Psi}_h) = 0$$

- We may also have a right hand side. Then:

$$\hat{A}(\hat{U}_h^n)(\hat{\Psi}_h) = \widehat{F}(\hat{\Psi}_h).$$

- In detail:

Problem

Find $\hat{U}_h^n = \{\hat{v}_{f,h}^n, \hat{v}_{s,h}^n, \hat{u}_{f,h}^n, \hat{u}_{s,h}^n, \hat{p}_{f,h}^n\} \in \widehat{X}_{h,D}^0$, *where*
$\widehat{X}_{h,D}^0 := \{\hat{v}_{f,h}^D + \hat{V}_{f,\hat{v},h}^0\} \times \hat{L}_{s,h} \times \{\hat{u}_{f,h}^D + \hat{V}_{f,\hat{u},h}^0\} \times \{\hat{u}_{s,h}^D + \hat{V}_{s,h}^0\} \times \hat{L}_{f,h}^0$, *for all* $n = 1, 2, \ldots, N$ *such that*

$$\hat{A}(\hat{U}_h^n)(\hat{\Psi}_h) = \widehat{F}(\hat{\Psi}_h) \quad \forall \hat{\Psi}_h \in \widehat{X}_h,$$

with $\hat{\Psi}_h = \{\hat{\psi}_{f,h}^v, \hat{\psi}_{s,h}^v, \hat{\psi}_{f,h}^u, \hat{\psi}_{s,h}^u, \hat{\psi}_{f,h}^p\}$ *and* $\widehat{X}_h = \hat{V}_{f,\hat{v},h}^0 \times \hat{L}_{s,h} \times \hat{V}_{f,\hat{u},\widehat{\Gamma}_i,h}^0 \times \hat{V}_{s,h}^0 \times \hat{L}_{f,h}^0$.

Principle idea

- All this is nothing else than a 'simple' **root-finding problem**
- Here:

$$\hat{A}(\hat{U}_h^n)(\hat{\Psi}_h) - \hat{F}(\hat{\Psi}_h) = 0$$

- Recall classes to the introduction to numerical methods:

Formulation (Root-finding problem)

Let $f : \mathbb{R} \to \mathbb{R}$. The task is to find $x \in \mathbb{R}$ such that

$$f(x) = 0.$$

- Generate sequence of iterates $(x_k)_{k \in \mathbb{R}}$ and hopefully reach at some point

$$|f(x_k)| < TOL, \quad \text{where } TOL \text{ is small, e.g., } TOL = 10^{-10}.$$

- The **key** question is now **how to construct** this sequence? Options are (among others):
  1. Fixed-point schemes: Picard iterations, gradient descent
  2. Newton, quasi-Newton

Newton

- Let us assume that we are at x_k and can evaluate $f(x_k)$.
- Now we want to compute this next iterate x_{k+1} with the unknown value $f(x_{k+1})$.
- Taylor expansion gives us:

$$f(x_{k+1}) = f(x_k) + f'(x_k)(x_{k+1} - x_k) + o(x_{k+1} - x_k)^2$$

- We assume that $f(x_{k+1}) = 0$ (or very close to zero $f(x_{k+1}) \approx 0$).
- Then, x_{k+1} is the sought root and neglecting the higher-order terms we obtain:

$$0 = f(x_k) + f'(x_k)(x_{k+1} - x_k).$$

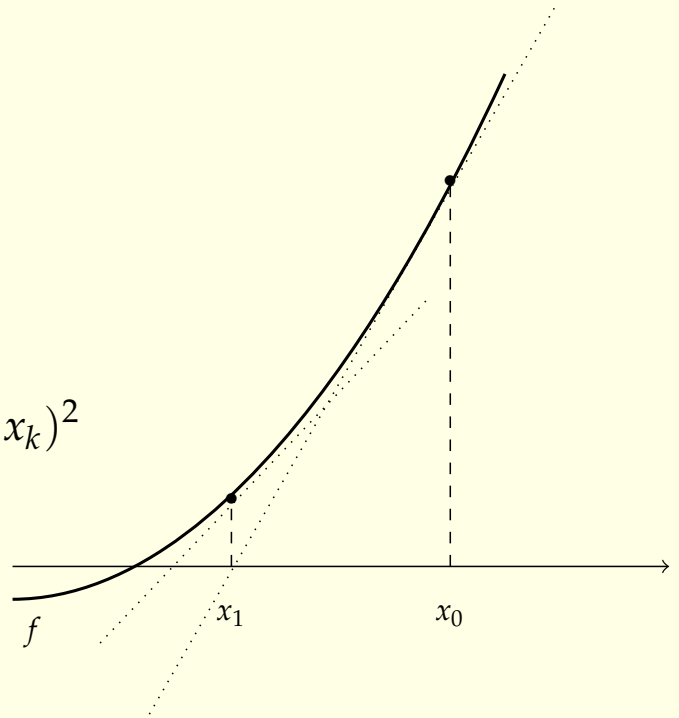


Figure: Geometrical interpretation of Newton's method.

Newton scheme

- Newton scheme, 1st version:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots. \tag{1}$$

  This iteration is possible as long as $f'(x_k) \neq 0$

- Second version: We see that Newton's method can be written a bit more general as

$$x_{k+1} = x_k + d_k, \quad k = 0, 1, 2, \ldots,$$

  where the search direction is

$$d_k = -\frac{f(x_k)}{f'(x_k)}.$$

- The iteration (1) terminates if a stopping criterion

$$\frac{|x_{k+1} - x_k|}{|x_k|} < TOL, \quad \text{or} \quad |x_{k+1} - x_k| < TOL,$$

  or

$$\frac{|f(x_{k+1})|}{|f(x_0)|} < TOL \quad \text{or } |f(x_{k+1})| < TOL$$

  is fulfilled.

- Relative stopping criteria (left ones) are in general recommended.

Newton as defect-correction

Definition (Newton's method as defect-correction scheme)

Compute δx from defect problem, then correct solution to x_{k+1}:

$$f'(x_k)\delta x = d_k, \quad d_k := y - f(x_k),$$
$$x_{k+1} = x_k + \delta x, \quad k = 0, 1, 2, \ldots.$$

The iteration is finished with the same stopping criterion as for the classical scheme. To compute the update δx we need to invert $f'(x_k)$:

$$\delta x = (f'(x_k))^{-1} d_k.$$

This step seems trivial but is the most critical one if we deal with problems in $\mathbb{R}^n$ with $n > 1$ or in function spaces. Because here, the derivative becomes a matrix. Therefore, the problem results in solving a linear equation system of the type $A\delta x = b$ and computing the inverse matrix A^{-1} is an expensive operation.

Going from $\mathbb{R}$ to Banach spaces

- Newton-Raphson (1D), find $x \in \mathbb{R}$ via iterating $k = 0, 1, 2, \ldots$ such that $x_k \approx x$ via:

$$\begin{aligned} \text{Find } \delta x \in \mathbb{R}: \quad & f'(x_k)\delta x = -f(x_k), \\ \text{Update:} \quad & x_{k+1} = x_k + \delta x. \end{aligned}$$

- Newton in $\mathbb{R}^n$, find $x \in \mathbb{R}^n$ via iterating $k = 0, 1, 2, \ldots$ such that $x_k \approx x$ via:

$$\begin{aligned} \text{Find } \delta x \in \mathbb{R}^n: \quad & F'(x_k)\delta x = -F(x_k), \\ \text{Update:} \quad & x_{k+1} = x_k + \delta x. \end{aligned}$$

  Here we need to solve a linear equation system to compute the update $\delta x \in \mathbb{R}^n$.

- Banach spaces, find $u \in V$, with $dim(V) = \infty$, via iterating $k = 0, 1, 2, \ldots$ such that $u_k \approx u$ via:

$$\begin{aligned} \text{Find } \delta u \in V: \quad & F'(u_k)\delta u = -F(u_k), \\ \text{Update:} \quad & u_{k+1} = u_k + \delta u. \end{aligned}$$

  Such a problem needs to be discretized and results again in solving a linear equation system in the defect step.

- Banach spaces, applied to variational formulations, find $U \in X$, with $dim(V) = \infty$, via iterating $k = 0, 1, 2, \ldots$ such that $U_k \approx U$ via:

$$\begin{aligned} \text{Find } \delta U \in V: \quad & A'(U_k)(\delta U, \Psi) = -A(U_k)(\Psi), \\ \text{Update:} \quad & U_{k+1} = U_k + \delta U. \end{aligned}$$

  As before, the infinite-dimensional problem is discretized resulting in solving a linear equation system in the defect step.

Back to FSI ...

- After temporal and spatial discretization, at each nonlinear iteration step, a linear equation system needs to be solved (without any hats):

Formulation

Find $U_h^n \in X_h$ *such that*

$$A(U_h^n)(\Psi_h) = 0 \quad \forall \Psi_h := X_h$$

Formulation (Newton as defect-correction scheme)

For the iteration steps $m = 0, 1, 2, \ldots$, *the Newton update* $\delta U_h^n \in X_h$ *is computed by solving:*

$$A'(U_{h,m}^n)(\delta U_h, \Psi_h) = -A(U_{h,m}^n)(\Psi_h) \quad \forall \Psi \in V_h \times W_h,$$
$$U_{h,m+1}^n = U_{h,m}^n + \omega \delta U_h,$$

with a line search parameter $\omega \in (0, 1]$.

Remark

The derivative $A'(U_{h,m})(\delta U_h, \psi)$ *is the so-called Jacobian. This operator is obtained by computing the directional derivatives of* $A(U_{h,m})(\Psi)$. *Details are omitted for the convenience of all of us. Details can be found in the literature mentioned on page 5.*

Motivation: work amount in terms of arithmetic operations

Table: Operations for solving $A\delta U = B$ with $A \in \mathbb{R}^{n\times n}$ being large, sparse, and s.p.d.[18]

Scheme	$d=2$	$d=3$
Gauss (direct)	n^3	n^3
Banded-Gauss (direct)	n^2	$n^{7/3}$
Jacobi (iterative)	n^2	$n^{5/3}$
Gauss-Seidel (iterative)	n^2	$n^{5/3}$
CG	$n^{3/2}$	$n^{4/3}$
SOR with opt. ω	$n^{3/2}$	$n^{4/3}$
Multigrid	n	n

- Often, $n \gg 10^4$ up to 10^8 (keep in mind that such systems have to be solved at each time point t_m and for $0 \leq m \leq M$ the end time index M might be big itself, e.g., $M = 1000$)
- This means, we need to opt for iterative or even multigrid schemes
- However they are more difficult to implement!
- For FSI in the following, we propose a mixture of iterative solvers and multigrid.

[18] Here the index n should not be misunderstood with the time point t_n! Both n are different!

Parallel solution of FSI: linear system

- Recall: we are sitting at time step t_n. To solve for U_h^{n+1} at t_{n+1} we utilize Newton's method
- At each Newton step (Index m), we have

$$\underbrace{A'(U_{h,m}^n)(\delta U_h, \Psi_h)}_{=A\delta U} = \underbrace{-A(U_{h,m}^n)(\Psi_h)}_{=B}$$

Thus:

$$A\delta U = B.$$

- Up to 10^5 unknowns, direct solvers work still okay in two-dimensional spatial settings
- For either more unknowns and also for smaller-sized 3D ($\Omega \in \mathbb{R}^3$) problems, we, however, want (must!) employ an iterative method with multigrid preconditioning because of the computational cost
- FSI problem is non-symmetric, therefore GMRES (generalized minimal residual) is a classical choice.
- However, condition number is bad (material and discretization parameters) with $\kappa(A) \gg 1$. Therefore, iterative solvers without preconditioning will practically not work!

→ Construct preconditioner matrix P such that

$$P^{-1}A\delta U = P^{-1}B$$

with $P^{-1} \approx A^{-1}$ such that condition number of $P^{-1}A \approx I$ (close to identity matrix).

More details on the FSI matrix

- In FSI, 3×3 block system:

$$A := \begin{bmatrix} \mathcal{M} & \mathcal{C}_{ms} & 0 \\ \mathcal{C}_{sm} & \mathcal{S} & \mathcal{C}_{sf} \\ \mathcal{C}_{fm} & \mathcal{C}_{fs} & \mathcal{F} \end{bmatrix},$$

  recalling that we have the principal problems (M) mesh motion, (F) fluid, (S) solid.

- In more detail, for a good preconditioner, we need to consider the coupling terms on the interface Γ, here denoted by I:

$$A := \begin{bmatrix}
\mathcal{M}^{\Omega\Omega} & 0 & & \mathcal{M}^{\Omega I} & & & & & \\
0 & I & & & & & & & \\
 & & \mathcal{S}^{\Omega\Omega}_{uu} & \mathcal{S}^{\Omega I}_{uu} & \mathcal{S}^{\Omega\Omega}_{uv} & \mathcal{S}^{\Omega I}_{uv} & & & \\
\mathcal{M}^{I\Omega} & & \mathcal{S}^{I\Omega}_{uu} & \mathcal{S}^{II}_{uu} + \mathcal{M}^{II} & \mathcal{S}^{I\Omega}_{uv} & \mathcal{S}^{II}_{uv} & & & \\
 & & \mathcal{S}^{\Omega\Omega}_{vu} & \mathcal{S}^{\Omega I}_{vu} & \mathcal{S}^{\Omega\Omega}_{vv} & \mathcal{S}^{\Omega I}_{vv} & & & \\
\mathcal{B}^{I\Omega}_{fm} & & \mathcal{S}^{I\Omega}_{vu} & \mathcal{B}^{II}_{fm} + \mathcal{S}^{II}_{vu} & \mathcal{S}^{I\Omega}_{vv} & \mathcal{S}^{II}_{vv} + \mathcal{F}^{II}_{vv} & & \mathcal{F}^{I\Omega}_{vv} & \mathcal{F}^{I}_{vp} \\
0 & 0 & & & & & I & 0 & 0 \\
\mathcal{B}^{\Omega\Omega}_{fm} & 0 & & \mathcal{B}^{\Omega I}_{fm} & & \mathcal{F}^{\Omega I}_{vv} & 0 & \mathcal{F}^{\Omega\Omega}_{vv} & \mathcal{F}^{\Omega}_{vp} \\
\mathcal{B}^{\Omega}_{pm} & 0 & & \mathcal{B}^{I}_{pm} & & \mathcal{F}^{I}_{pv} & 0 & \mathcal{F}^{\Omega}_{pv} & 0
\end{bmatrix}$$

LDU factorization for constructing a preconditioner

- Construct now preconditioner P^{-1}
- Simplified LDU block factorization:

$$A \approx \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \frac{\mathcal{C}_{fm}}{\mathcal{M}} & \frac{\tilde{\mathcal{C}}_{fs}}{\mathcal{S}} & 1 \end{bmatrix} \begin{bmatrix} \mathcal{M} & 0 & 0 \\ 0 & \mathcal{S} & 0 \\ 0 & 0 & \mathcal{X} \end{bmatrix} \begin{bmatrix} I & \frac{\mathcal{C}_{ms}}{\mathcal{M}} & 0 \\ 0 & I & \frac{\mathcal{C}_{sf}}{\mathcal{S}} \\ 0 & 0 & I \end{bmatrix}$$

  where we neglect the coupling term $\mathcal{C}_{sm}$.

- We have $\tilde{\mathcal{C}}_{fs} = \mathcal{C}_{fs} - \frac{\mathcal{C}_{fm}\mathcal{C}_{ms}}{\mathcal{M}}$ and $\mathcal{X} = \mathcal{F} - \frac{\mathcal{C}_{fs}\tilde{\mathcal{C}}_{sf}}{\mathcal{S}}$ and the fluid Schur complement

$$\mathcal{X} = \mathcal{F} - \tilde{\mathcal{C}}_{fs}\,\mathcal{S}^{-1}\,\mathcal{C}_{sf} = \mathcal{F} - (\mathcal{C}_{fs} - \mathcal{C}_{fm}\,\mathcal{M}^{-1}\,\mathcal{C}_{ms})\,\mathcal{S}^{-1}\,\mathcal{C}_{sf}.$$

- Having such a decomposition, it is easy to compute the action of the inverse. In Krylov subspace methods we only need to know the action of matrix-vector multiplications, here P^{-1} on the residual part r
- From linear algebra we know that $P^{-1}r = U^{-1}L^{-1}r$ with $P = LU$ from above.

An FSI preconditioner [20]

- Consecutively solving with L and U yields the following result:
- Algorithm: Evaluation of $P^{-1}r$ (matrix-vector multiplications):
  1. Solve $x_m = \mathcal{M}^{-1} r_m$
  2. Solve $x_s = \mathcal{S}^{-1} r_s$
  3. Solve $x_f = \mathcal{F}^{-1}(r_f - \mathcal{C}_{fm} x_m - \mathcal{C}_{fs} x_s)$
  4. Update $x_s = x_s - \mathcal{S}^{-1} \mathcal{C}_{sf} x_f$
  5. Update $x_m = x_m - \mathcal{M}^{-1} \mathcal{C}_{ms} x_s$
- Parallelization using MPI
- It remains to discuss the solutions of the subproblems $\mathcal{M}^{-1}$, $\mathcal{S}^{-1}$ and $\mathcal{F}^{-1}$:
  1. For $\mathcal{M}^{-1}$ we apply AMG[19] V-cycles with Gauss-Seidel smoother
  2. For $\mathcal{F}^{-1}$: classical Schur complement approch for Navier-Stokes as a preconditioner inside GMRES. The block subproblems therein are solved again with GMRES and AMG preconditioning
  3. For $\mathcal{S}^{-1}$, we have two components u_s and v_s and a full two-by-two block system. Again we derive a Schur complement as preconditioner within a GMRES iteration.

[19] AMG from Trilinos

[20] Jodlbauer/Wick, Chapter 6 in FSI Ricam book, 2017

Summary of linear solution of FSI

- At each Newton step m, solve linear system $A\delta U = B$
- Solution achieved with iterative solver GMRES
- Use preconditioner P^{-1} using Schur complements
- Inside P^{-1} we need to approximate $\mathcal{M}^{-1}, \mathcal{F}^{-1}, \mathcal{S}^{-1}$
- These inverses are 'solved' themselves using GMRES method with, again, Schur complements for $\mathcal{F}^{-1}, \mathcal{S}^{-1}$. Therein, AMG solvers are used to approximate further inverses
- For $\mathcal{M}^{-1}$, we also use a AMG method

The limits of today's simulation power

Consider the simple example of an aircraft with Reynolds number 10^8. A direct numerical simulation (DNS) that captures the smallest scales of order $O(Re^{9/4})$, would lead to a mesh with 10^{18} mesh points. This is simply infeasible with today's supercomputers [21]. Consequently, we have to find approximations with less computational cost.

- → Model order reduction
- → Adaptive methods
- → Parallel computations

[21] Sagout, 2006. This note is inspired by a talk given by Johan Jansson at IWH Heidelberg, Nov 2014.

Summary and outlook

Nonlinear solvers and linear solution using GMRES with Schur complements with multigrid preconditiong ✓

FSI program, software, simulations, further applications

Class 5

FSI program and algorithmic steps[22]

Algorithm 2 FSI program.

Define problem parameters (Δt, θ, material parameters, ...)
Create mesh
Assign degrees of freedom
Set-up constraints (hanging nodes, homogeneous boundary conditions, interface coupling)
Initialize sparsity pattern, Jacobian A and vectors for the right-hand-side r, solution U, newton update δU and previous solution U_{-1}

$t := 0$
Initial guess for current and previous solution $U := 0$, $U_{-1} := 0$

while $t < T$ **do**
    Apply initial conditions of time t to U
    while not converged **do**
        Assemble r
        (if necessary): assemble Jacobian A
        Solve $A\delta U = r$
        Update $U := U + \delta U$
    end while
    Optional:
    - evaluate functionals (drag, lift, displacement, ...)
    - output results for current time-step
    $t := t + \Delta t$
    $U_{-1} := U$
end while

[22] Jodlbauer/Wick; FSI RICAM book, Chapter 6, 2017

Software

- Open-source C++ programming codes
- deal.II: differential equations analysis library, `www.dealii.org`
- DOpElib: Differential and optimization environment library, `www.dopelib.net`

- deal.II step-multiphysics template (ANS, Vol. 1, 2013, pp. 1-19)
  Link to ANS materials

→ An ALE-FSI implementation with the methods learned in these classes can be found there. Therein, FSI benchmarks are computed, see Topic 1 below.

- Similar implementation of ALE-FSI can be found in DOpElib
  `dopelib/Examples/PDE/InstatPDE/Example2`

Simulations and current research

1. 2011: Benchmarking FSI codes: FSI benchmarks from Hron/Turek 2006
2. 2018: with L. Failer: Adaptive Time-Step Control for Nonlinear Fluid-Structure Interaction (JCP)
3. 2018: with D. Jodlbauer and U. Langer: Parallel solution of FSI (in revision)
4. 2016 (with S. Frei and T. Richter): FSI with growth, contact, and chemical reactions (JCP)
5. 2013 (with T. Richter) and 2019 (with W. Wollner): Optimal control and optimal design for FSI

Topic 1

FSI benchmarks from Hron/Turek; 2006 [23] [24]

- For all details, we refer to Chapter 12 (Simulations) in the LUH-FSI notes (*.pdf)

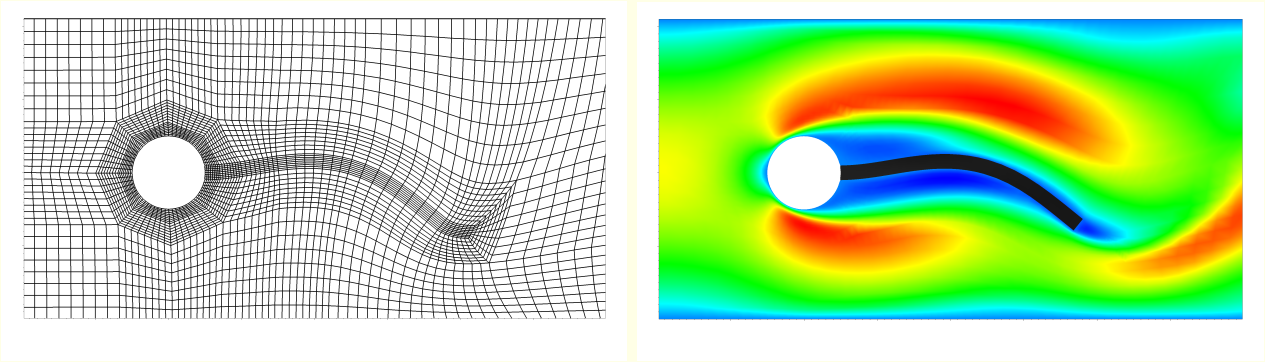


Figure: Results in the deformed configuration $\Omega(t)$: FSI 2 test case: mesh (left) and velocity profile in vertical direction (right) at time $t = 16.14s$.

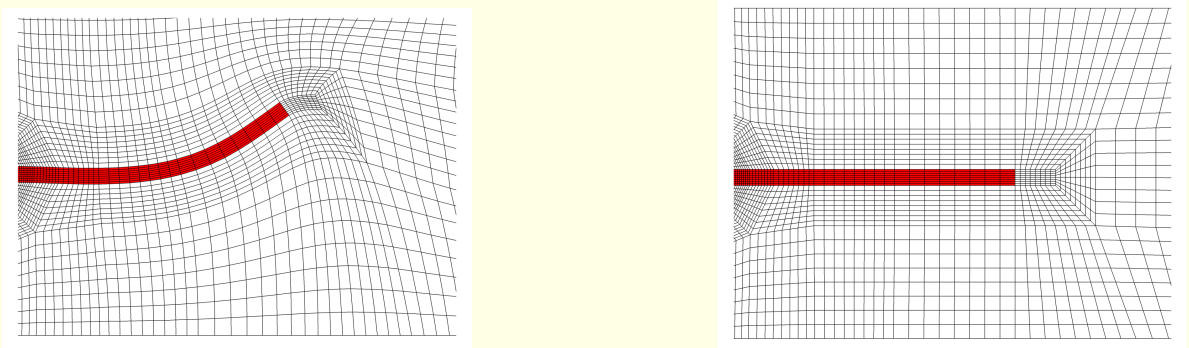


Figure: The current (physical) domain $\Omega(t)$ of the ALE-FSI problem is displayed in the left figure. However, the computations are actually done in the fixed reference domain $\widehat{\Omega}$ (right).

[23] Hron/Turek, 2006

[24] my computations: Wick, Comp. Struct., 2011

Results

Table: Results for the FSI 2 benchmark with the biharmonic mesh motion model and second type of boundary conditions. The mean value and amplitude are given for the four quantities of interest: $u_x, u_y[m], F_D, F_L[N]$. The frequencies $f_1[s^{-1}]$ and $f_2[s^{-1}]$ of u_x and u_y vary in a range of $3.83 - 3.88$ (ref. 3.86) and $1.92 - 1.94$ (ref. 1.93), respectively.

DoF	$k[s]$	$u_x(A)[\times 10^{-3}]$	$u_y(A)[\times 10^{-3}]$	F_D	F_L
27744	3.0e-3	-13.63 ± 11.80	1.27 ± 78.72	207.22 ± 71.13	-0.57 ± 230.6
27744	2.0e-3	-13.72 ± 11.84	1.26 ± 78.38	208.12 ± 71.18	-0.30 ± 232.6
27744	1.0e-3	-13.74 ± 11.85	1.28 ± 78.48	209.46 ± 71.43	-0.06 ± 231.7
27744	0.5e-3	-13.66 ± 11.81	1.28 ± 78.32	208.96 ± 71.60	-0.06 ± 238.2
42024	3.0e-3	-13.34 ± 11.57	1.40 ± 77.08	204.81 ± 68.54	0.79 ± 221.5
42024	2.0e-3	-13.36 ± 11.55	1.28 ± 77.18	205.61 ± 68.67	0.51 ± 223.0
42024	1.0e-3	-13.38 ± 11.58	1.31 ± 77.44	206.11 ± 68.26	0.62 ± 221.2
42024	0.5e-3	-13.27 ± 11.52	1.23 ± 77.25	207.05 ± 68.87	0.30 ± 230.6
72696	3.0e-3	-14.43 ± 12.46	1.35 ± 80.71	212.50 ± 76.40	0.18 ± 234.6
72696	2.0e-3	-14.49 ± 12.44	1.19 ± 80.66	213.49 ± 76.39	0.13 ± 235.7
72696	1.0e-3	-14.49 ± 12.46	1.16 ± 80.63	213.39 ± 75.25	0.23 ± 234.2
72696	0.5e-3	-14.40 ± 12.39	1.25 ± 80.55	213.55 ± 76.06	0.30 ± 240.2
(ref.)	0.5e-3	-14.85 ± 12.70	1.30 ± 81.70	215.06 ± 77.65	0.61 ± 237.8

- To validate programming code with purely computational methods, it is necessary to compute for at least three spatial meshes (DoFs) and three time step sizes k.
- Why? This will show computational qualitative convergence and a provides indications to the accuracy and robustness of the proposed method.

Topic 2

Failer/Wick (JCP, 2018): Adaptive time step control[25]

Problem statement and goals:

- We want to compute a given (physical) quantity of interest $J(U)$ with a certain accuracy at low computational cost.
- As before: $J(U)$ can be a point value, deformation, drag, lift, temperature evaluation etc. but not necessarily in the entire domain!
- Formulate an adjoint problem (Class 3, Adaptivity), **still linear**, **but backward running in time (expensive!)** to determine (local) sensitivity measures to determine $J(U)$ respect to the given PDE

Adjoints are expensive, **but** they do have advantages in error estimation:

$\Rightarrow$ A **robust, time-adaptive, procedure** to calculate functionals of interest with **sufficient accuracy** allowing for the **automated adjustment of time step sizes** where necessary.

$\Rightarrow$ A (global) **error estimator and not only an error indicator**. Therefore, we obtain a guess η about the unknown true error $J(U) - J(U_k)$. Consequently, we know to which accuracy we have computed a certain physical quantity without knowing its exact (analytical) value $J(U)$.

[25] Failer/Wick, JCP, 2018

Adaptive time step control

- Code verification: test code with the help of a manufactured solution (rarely possible!) or with a computationally-obtained referenced solution $U_{ref} =: U$.
- In this work: up to 1 444 384 time steps are used to obtain a numerically-obtained U; wall clock time > 31 days (serial computation in time and space)
- Check by computing the effectivity index (now w.r.t. temporal error):

$$I_{eff} = \frac{\eta}{J(U) - J(U_k)}$$

  where η is a computable error estimator and $J(U) - J(U_k)$ is the true error for some known 'exact' solution U.

- The error estimator reads[26] for M time intervals:

$$\eta := \sum_{m=1}^{M} \eta_m = \ldots + \frac{1}{2}\left(F(Z_k^{(2)} - Z_k^{(1)}) - A(U_k, Z_k^{(2)} - Z_k^{(1)})\right) + \ldots$$

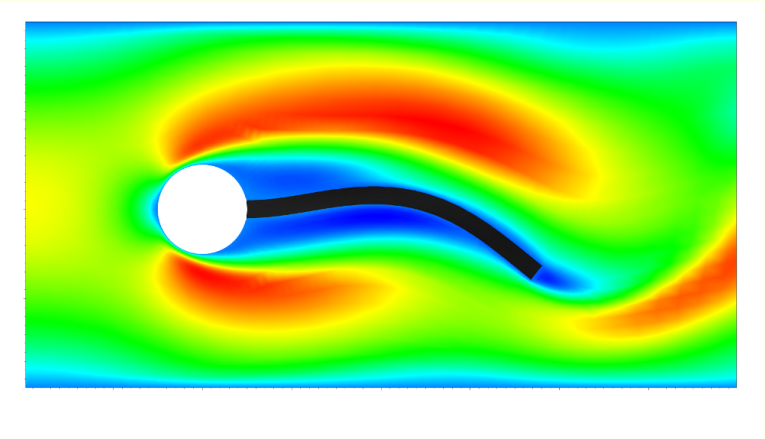
  where we recognize some terms from page 113.

- Once error estimator have been validated with I_{eff} they can be - hopefully - applied to more complicated problems, where no 'exact' (manufactured/analytical) solution is known anymore.

[26] Failer/Wick, JCP, 2018

Adaptive time step control

- Numerical test: FSI-2 benchmark (Hron/Turek, 2006)
- Elastic beam immersed in a fluid (Navier-Stokes)



- Computation of effectivity indices:

M	1128	1482	2322	4176	5844	10518
$J(U_{kh})$	$2.896 \cdot 10^3$	$3.048 \cdot 10^3$	$3.117 \cdot 10^3$	$3.130 \cdot 10^3$	$3.129 \cdot 10^3$	$3.129 \cdot 10^3$
$J(U_{kh}) - J(U_{ref})$	$2.3 \cdot 10^2$	$8.1 \cdot 10^1$	$1.2 \cdot 10^1$	$7.0 \cdot 10^{-1}$	$7.4 \cdot 10^{-1}$	$4.6 \cdot 10^{-1}$
I_{eff}	1.01	1.01	1.00	0.97	1.02	1.04

Table: Effectivity indices I_{eff} for DWR time discretization error estimator with respect to $J(U)$ on adaptively refined time grids.

Topic 3

Jodlbauer/Langer/Wick: Parallel solution of FSI [27]

- Goal: develop a parallel (MPI) solver for nonstationary, nonlinear FSI
- Need Newton's method
- Need iterative linear solver
- Need preconditioner

$\rightarrow$ See class 4

[27] Jodlbauer/Langer/Wick, IJNME, 2019

Parallel solution of FSI: 3D obstacle in a fluid

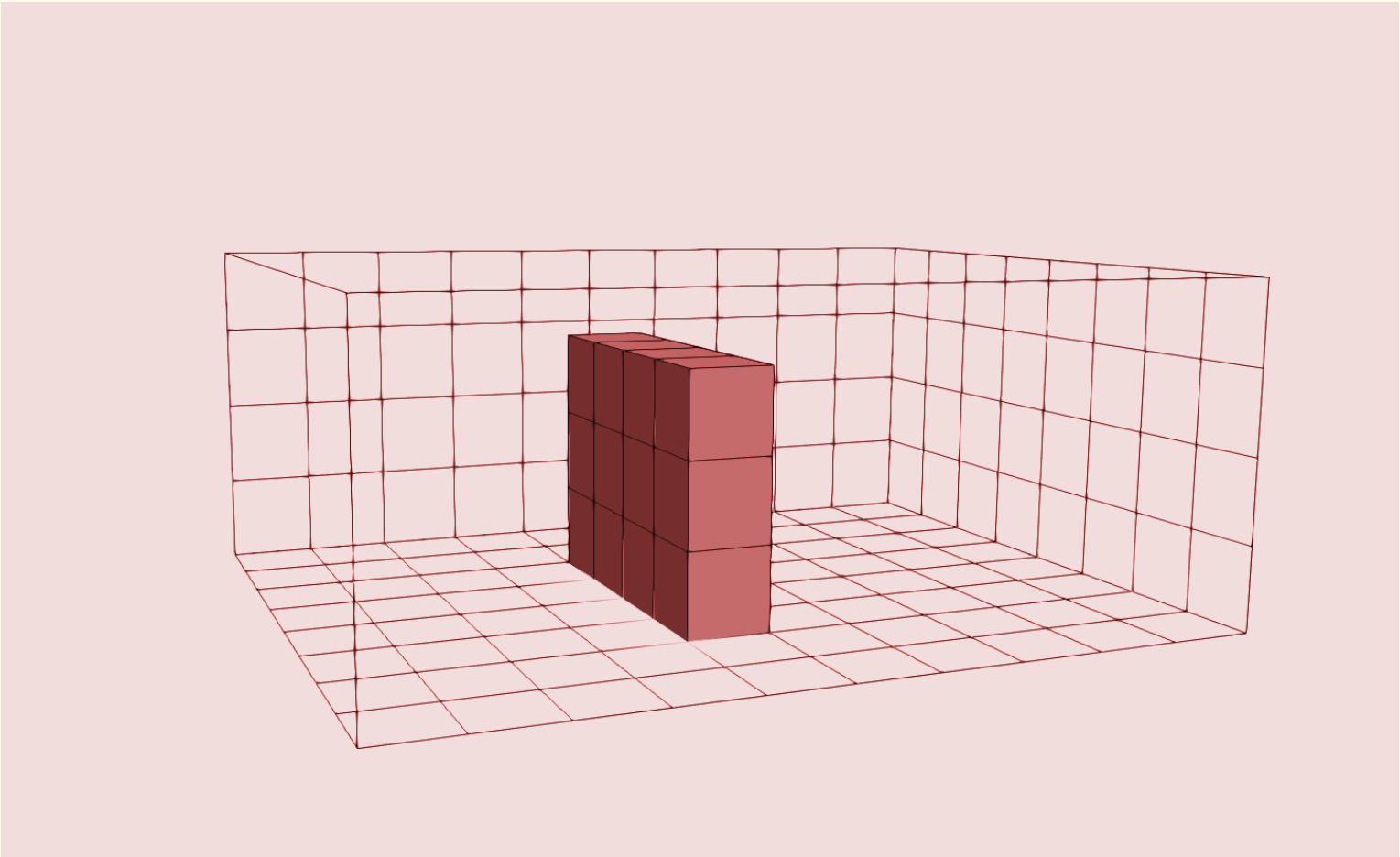


Figure: Graphical illustration of the geometry. The elastic obstacle is displayed in dark color.

Parallel solution of FSI: scalability

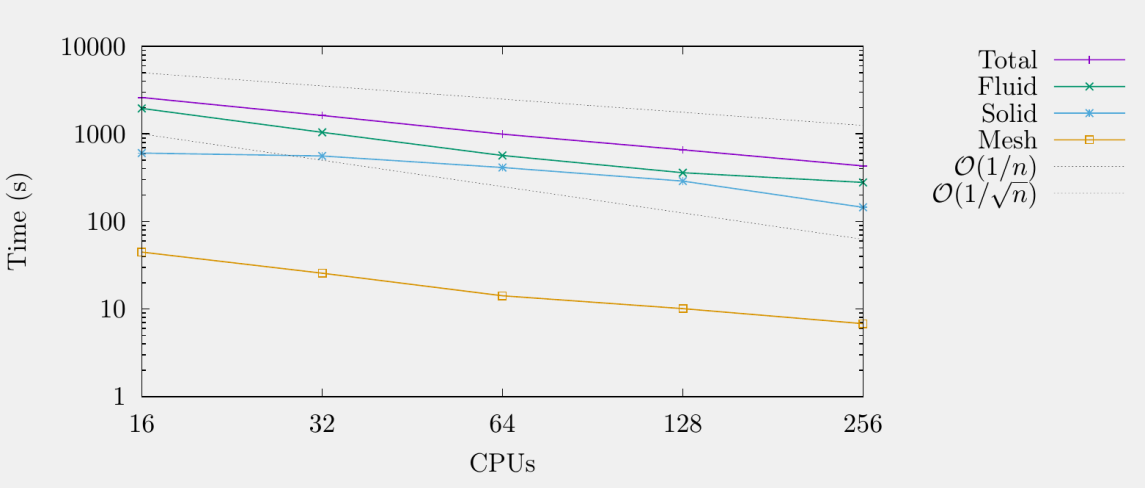


Figure: Strong scalability using the preconditioned GMRES for approximately $14 \cdot 10^6$ dofs $(r = 4)$ in $3d$. Average time given in seconds for the solution of one linear system.

- Specifically, for 16 cores, the total CPU time to solve the linear problem with $14 \cdot 10^6$ dofs at a single time step is 2605 seconds, i.e., 43 minutes.
- On 256 cores, the computational cost decreases to 431 seconds, i.e., 7.2 minutes.
- Reduction by 84% of the computational time.

Topic 4

Frei/Richter/Wick (JCP, 2016): Mechano-chemical FSI with solid contact [28]

- Contact of solids (current topic! - several groups are working on these type of problems).
- Multiscale multiphysics problem:
  - Multiscale: Different temporal time scales: two-scale approach
  - Multiphysics: FSI couples with solid growth and a ODE on the interface for chemical reactions

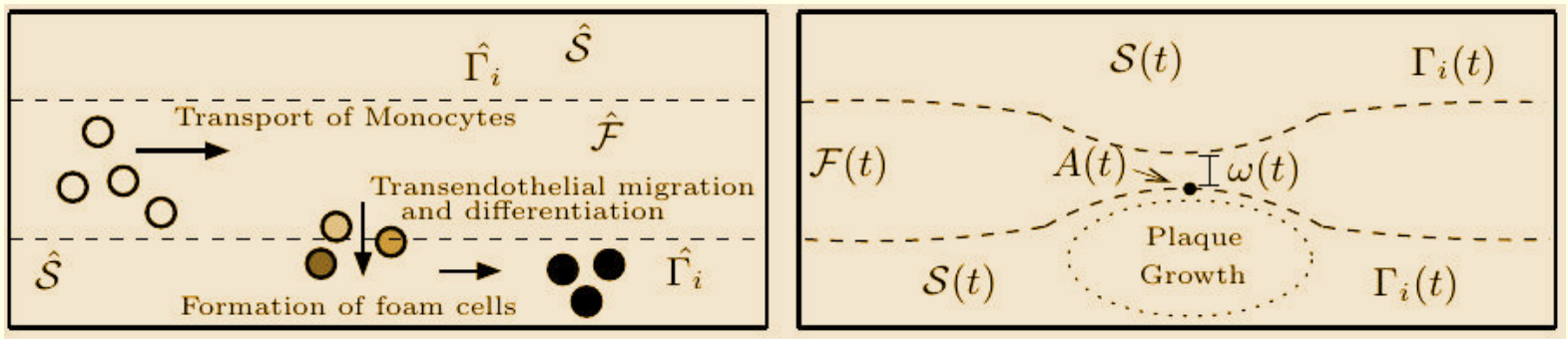


Figure: Configuration of the domain and mechanism of plaque formation. Left: Domain in reference configuration split into fluid part $\hat{\mathcal{F}}$ and solid $\hat{\mathcal{S}}$ divided by the interface $\hat{\Gamma}_i$. Right: Domain in the current (Eulerian) description with plaque formation and narrowing of the vessel.

[28] Frei/Richter/Wick, JCP, 2016

Multiple scales in time

- Short scale: heart does beat once in about every $[1]s$;
- Long scale: plaque growth takes place in a time span of months, i.e. $[> 1\,000\,000]s$

$\Rightarrow$ a numerical simulation will not be able to resolve each detail while following the long-term process

$\Rightarrow$ consider an averaged flow problem and focus on the long-scale dynamics

$\Rightarrow$ to incorporate effects of the short-scale dynamics, we compute effective wall stresses with the help of isolated small-scale simulations

- Accurate handling of the different time-scales is an open problem.

Solution algorithms: Long-scale/short-scale

Initialize $\vec{v}^0 = 0$, $\vec{u}^0 = 0$, $g^0 = 0$ and the vessel-width $\omega^0 = 2$. Set time-step $k_l = [1]day = [86\,400]s$. Iterate for $n = 1, 2, \ldots$:

1.a) Solve quasi-stationary long-scale problem:

$$\{c_s^{n-1}, \omega^{n-1}\} \mapsto \{\vec{v}^n, \vec{u}^n, p^n\}$$

1.b) Compute the vessel width in the point $A(\tau_n)$

$$\omega^n = 2 - 2\vec{u}_{s,2}^n(A(\tau_n), \tau_n)$$

2.a) Set $\vec{v}^{s,0} = \vec{v}^n$, $\vec{u}^{s,0} = \vec{u}^n$ and solve the short-scale problem in $I_n = [[\tau_n]days, [\tau_n]days + [1]s)$

$$\{\vec{v}^{s,0}, \vec{u}^{s,0}, c_s^{n-1}, \omega^n\} \mapsto \{\vec{v}^{s,m}, \vec{u}^{s,m}, p^{s,m}\},\ m = 1, \ldots, N_s$$

2.b) Compute average wall stress in main stream direction

$$\boldsymbol{\sigma}_{WS}^n = \frac{1}{N_s} \sum_{m=1}^{N_s} \int_{\Gamma_i} |\boldsymbol{\sigma}_f(\vec{v}^{s,m}, p^{s,m})\vec{n} \cdot \vec{e}_1| \, \mathrm{d}o$$

2.c) Update the foam cell concentration

$$c_s^n = c_s^{n-1} + k_l \gamma_0 \left(1 + \boldsymbol{\sigma}_{WS}^n / \overline{\boldsymbol{\sigma}}\right)^{-1}$$

Long-scale problem: clogging - the limits of ALE

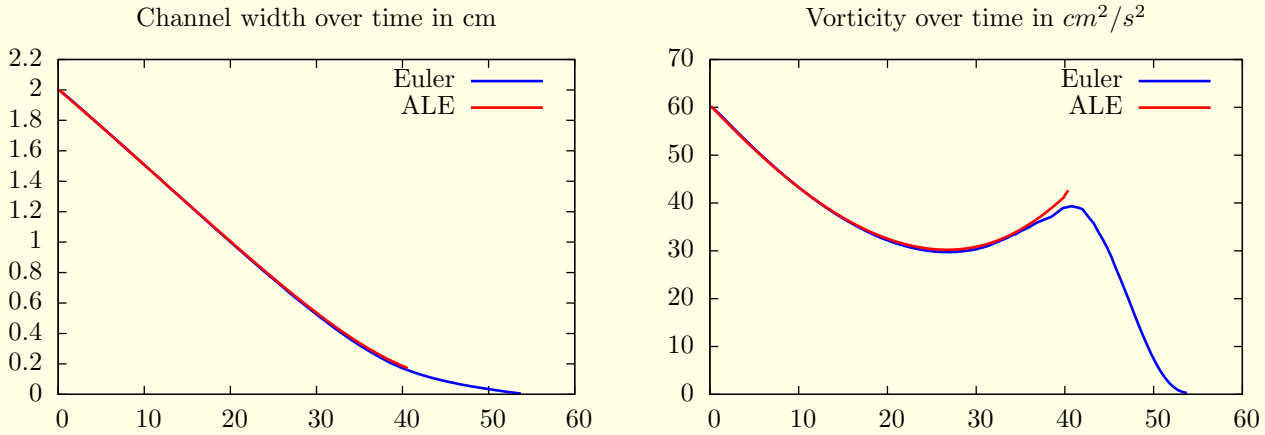


Figure: Channel width and vorticity for a long-scale simulation with reduced inflow velocity. The inflow velocity goes to zero when the channel closes. This makes the complete closure of the channel possible.

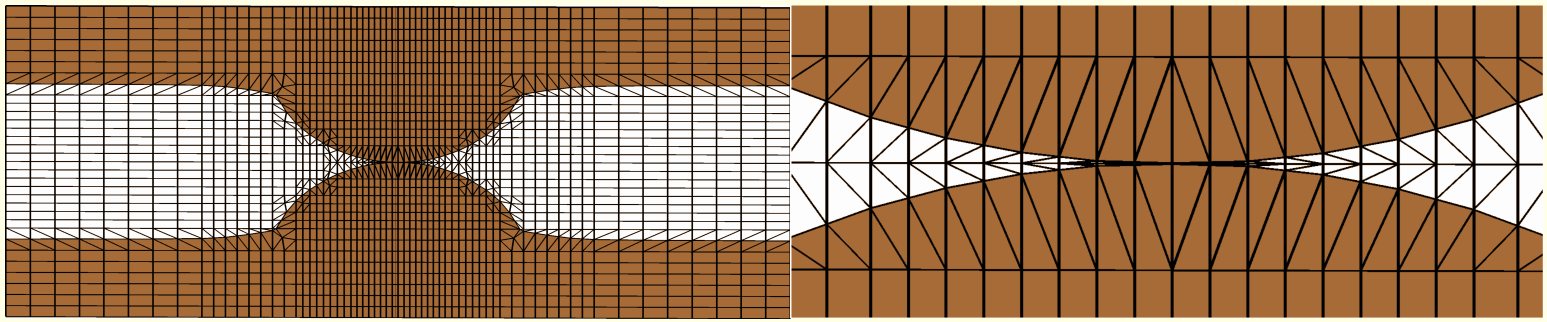


Figure: **Fully Eulerian deformation** when the channel is completely closed at $\tau = [55.8]days$. The standard classical ALE technique cannot close the channel!

Topic 5

Wollner/Wick (2019-2020): FSI with optimization [29]

- Drag minimization, while controlling Lamé coefficient in the valves
- The cost functional is given by:

$$J(q, \hat{U}) = F(\widehat{\Gamma}_{\text{opt}}, T) + \frac{\alpha}{2}|q - q_d|^2$$

where T is the end time value

- Here, $F(\cdot)$ is the drag functional defined as

$$F(\widehat{\Gamma}_{\text{opt}}, T) := \int_{\widehat{\Gamma}_{\text{opt}}} (\widehat{\sigma}_f \cdot \hat{n}) \cdot e_1 \, \mathrm{d}s$$

where $\hat{n}$ is the unit normal vector pointing outward of the domain $\widehat{\Omega}_s$ and e_1 the first unit vector in $\mathbb{R}^2$.

- The boundary part, where the drag is evaluated is

$$\widehat{\Gamma}_{\text{opt}} := \{2 \leq x \leq 8;\ y = 0\}.$$

[29] Wick/Wollner, arXiv 1910.03424, 2019

Gradient method

Algorithm (Gradient method)

Let $q^0 \in \mathbb{R}^p$ be an initial guess, and pick parameters $\gamma \in (0, 1/2)$ and $\beta \in (0,1)$. For $k = 0, 1, \ldots$ until $\|\nabla_q \mathcal{J}(q^k)\|_Q < TOL$ iterate

1. *Solve the (nonlinear) primal FSI problem to obtain $\hat{U}_h \in \widehat{X}_h^N$ using Newton's method from Class 4*
2. *Solve a (linear) adjoint problem (similar to Class 3, subsection Adaptivity) to obtain $\hat{Z}_h \in \widehat{X}_h^N$.*
3. *Compute the gradient $\nabla \mathcal{J}(q^k)$ using (2) from below.*
4. *Find the largest $l \in \{0, 1, \ldots\}$ such that (Armijo-rule)*

$$\mathcal{J}(q^k - \beta^l \nabla \mathcal{J}(q^k)) \leq \mathcal{J}(q^k) - \gamma \beta^l \|\nabla \mathcal{J}(q^k)\|^2$$

   holds and set $\beta_k = \beta^l$.

5. *Update*

$$q^{k+1} = q^k - \beta_k \nabla \mathcal{J}(q^k).$$

$$(\nabla J(q), \delta q) = \frac{\mathrm{d}}{\mathrm{d}q} J(q, \hat{U}) \delta q \qquad \forall\, \delta q \in \mathbb{R}^p \tag{2}$$

Configuration and mesh

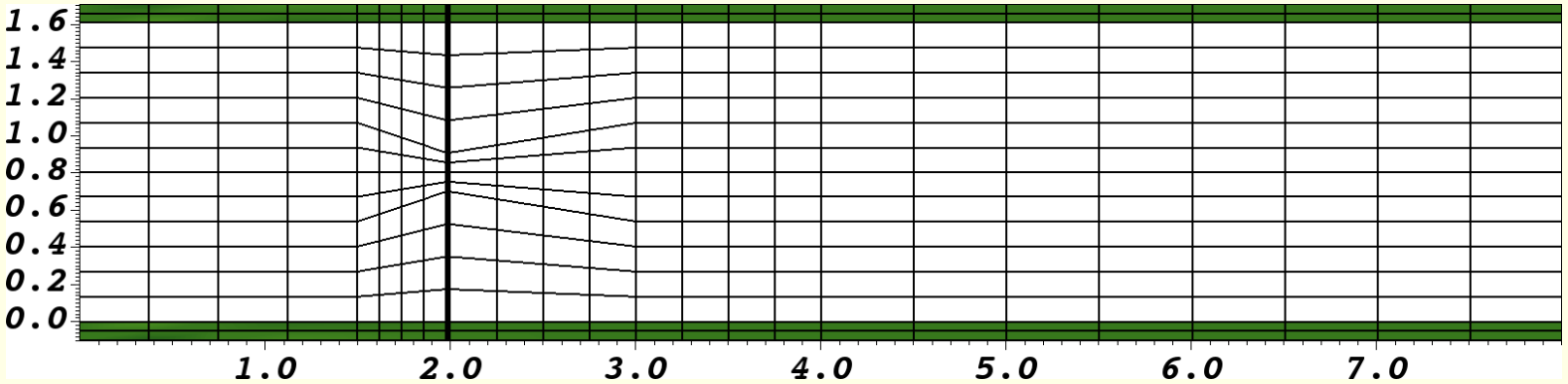


Figure: The mesh for the flapping membranes example at the initial time step. All geometric values are given in cm. The solid boundaries are colored in dark green. The flaps are located at $1.9788\,cm \leq x \leq 2.0\,cm$.

- Configuration very challenging because of small meshes of the elastic flaps

$\Rightarrow$ Will degenerate ALE transformation!

$\Rightarrow$ Good mesh motion model PDE required! Here biharmonic model $\Delta^2 u = 0$.

Performance of optimization algorithms

Table: Optimization results for the flapping membrane example with $\alpha = 1$ and $q_d = 5 \cdot 10^6$. The initial Residual in $q_0 = 2 \cdot 10^7$ is $|\nabla \mathcal{J}(q^0)| = 1.686 \cdot 10^7$

	Gradient method			BFGS method		
Iter	$\mathcal{J}(q^k)$	q^k	$\frac{\lvert\nabla \mathcal{J}(q^k)\rvert}{\lvert\nabla \mathcal{J}(q^0)\rvert}$	$\mathcal{J}(q^k)$	q^k	$\frac{\lvert\nabla \mathcal{J}(q^k)\rvert}{\lvert\nabla \mathcal{J}(q^0)\rvert}$
0	$1.265 \cdot 10^{14}$	$2 \cdot 10^7$	$1.0000 \cdot 10^{-0}$	$1.265 \cdot 10^{14}$	$2 \cdot 10^7$	$1.0000 \cdot 10^{-0}$
1	$1.9517 \cdot 10^{12}$	$3.13665 \cdot 10^6$	$1.2422 \cdot 10^{-1}$	$1.952 \cdot 10^{12}$	$3.13665 \cdot 10^6$	$1.242 \cdot 10^{-1}$
2	$3.0118 \cdot 10^{10}$	$5.23147 \cdot 10^6$	$1.5432 \cdot 10^{-2}$	$8.346 \cdot 10^2$	$5 \cdot 10^6$	$< 10^{-11}$
3	$4.6476 \cdot 10^8$	$4.97125 \cdot 10^6$	$1.9170 \cdot 10^{-3}$			
4	$7.1728 \cdot 10^6$	$5.00357 \cdot 10^6$	$2.3813 \cdot 10^{-4}$			
5	$1.1151 \cdot 10^5$	$4.99956 \cdot 10^6$	$2.9582 \cdot 10^{-5}$			
6	$2.5424 \cdot 10^3$	$5.00006 \cdot 10^6$	$3.6747 \cdot 10^{-6}$			
7	$8.6090 \cdot 10^2$	$4.99999 \cdot 10^6$	$4.5649 \cdot 10^{-7}$			
8	$8.3495 \cdot 10^2$	$5 \cdot 10^6$	$5.6707 \cdot 10^{-8}$			

- BFGS = Broyden-Fletcher-Goldfarb-Shanno[30] (quasi-Newton method in which the Newton matrix is approximated through lower-order terms)

[30] see e.g., Nocedal/Wright; Numerical optimization, 2006

Optimal flow field

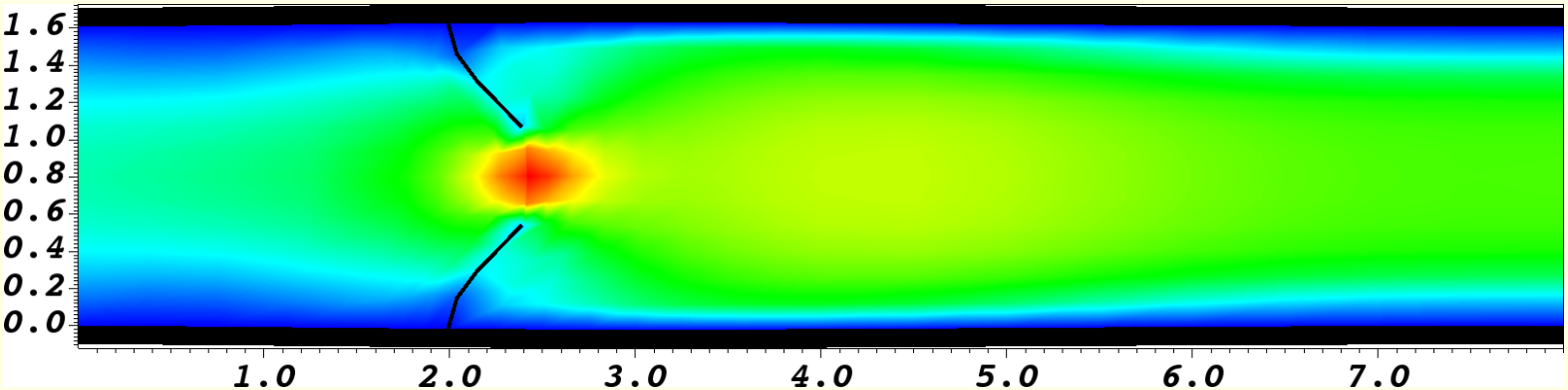


Figure: Flow field in the optimal controlled state.

Recent summary in 2017

Fluid-structure interaction: Book on latest results

- 2016 Conference with Winter school am RICAM Linz (Organizers: Stefan Frei, Bärbel Holm (geb. Janssen), Thomas Richter, Thomas Wick, Huidong Yang)
- Nov 2017: Buch (eds. S. Frei et al.): 'Fluid-Structure Interaction: Modeling, Adaptive Discretisations and Solvers' Radon Series on Computational and Applied Mathematics 20, de Gruyter
- Also in 2017, the following book appeared: T. Richter, **Fluid-structure interactions: models, analysis, and finite elements**, Springer, 2017

DE G DE GRUYTER

Stefan Frei, Bärbel Holm, Thomas Richter, Thomas Wick, Huidong Yang (Eds.)

FLUID-STRUCTURE INTERACTION

Modeling, Adaptive Discretisations and Solvers

With contrib. by Basting, Steffen / Birken, Philipp / Canic, Sunica / Colciago, Claudia / Deparis, Simone / Forti, Davide / Glowinski, Roland / Hoffman, Johan / Jodlbauer, M.Sc. Daniel / Langer, Ulrich / Monge, Azahar / Quaini, Annalisa / Shokrpour Roudbari, M.Sc. Mahnaz / Simsek, M.Sc. Görkem / van Brummelen, E. Harald / van der Zee, Kristoffer G.

This monograph discusses modeling, adaptive discretisation techniques and the numerical solution of fluid structure interaction. An emphasis in part I lies on innovative discretisation and advanced interface resolution techniques. The second part covers the efficient and robust numerical solution of fluid-structure interaction.

In part III, recent advances in the application fields vascular flows, binary-fluid-solid interaction, and coupling to fractures in the solid part are presented. Moreover each chapter provides a comprehensive overview in the respective topics including many references to concurring state-of-the art work.

Contents
Part I: Modeling and discretization
On the implementation and benchmarking of an extended ALE method for FSI problems
The locally adapted parametric finite element method for interface problems on triangular meshes
An accurate Eulerian approach for fluid-structure interactions

Part II: Solvers
Numerical methods for unsteady thermal fluid structure interaction
Recent development of robust monolithic fluid-structure interaction solvers
A monolithic FSI solver applied to the FSI 1,2,3 benchmarks

Part III: Applications
Fluid-structure interaction for vascular flows: From supercomputers to laptops
Binary-fluid–solid interaction based on the Navier–Stokes–Cahn–Hilliard Equations
Coupling fluid-structure interaction with phase-field fracture: Algorithmic details

Radon Series on Computational and Applied Mathematics 20

xiv, 372 pages, 53 Figures (bw), 106 Figures (c)

Hardcover:
RRP *€ [D] 119.95 / *US$ 137.99 / *GBP 109.00
ISBN 978-3-11-049527-0

eBook:
RRP *€ [D] 119.95 / *US$ 137.99 / *GBP 109.00
PDF ISBN 978-3-11-049425-9
EPUB ISBN 978-3-11-049258-3

Date of Publication: November 2017

Language of Publication: English

Subjects:
Applied Mathematics
Differential Equations and Dynamical Systems
Mechanics and Fluid Dynamics
Numerical and Computational Mathematics

Of interest to: Graduate students and researchers in mathematics, physics, and engineering.

*Prices in US$ apply to orders placed in the Americas only. Prices in GBP apply to orders placed in Great Britain only. Prices in € represent the retail prices valid in Germany (unless otherwise indicated). Prices are subject to change without notice. Prices do not include postage and handling if applicable. Free shipping for non-business customers when ordering books at De Gruyter Online. RRP: Recommended Retail Price.

Order now! orders@degruyter.com

degruyter.com

Conclusions

- ✓Brief introduction to continuum mechanics
- ✓Modeling of fluids and solids
- ✓Variational formulations of coupled PDEs
- ✓Numerical modeling of nonstationary, nonlinear, coupled PDEs (FSI)
- ✓Relationships to 'simpler' equations to establish understanding
- ✓Benchmarking, simulations, applications, and optimization

Almost done ...

Some open questions and ongoing ideas

1. Rigorous benchmarking of 3D fluid-structure interaction
2. Parallel space-time goal-oriented adaptive fluid-structure interaction
3. Efficient algorithms for fluid-structure interaction optimization and uncertainty quantification (optimal control, inverse problems, optimal design, parameter estimation)
4. Coupling fluid-structure interaction to other equations and/or physical phenomena such as chemical reactions, heat diffusion, ...

Funding acknowledgements

The support by DAAD (German Academic Exchange Service)
is gratefully acknowledged.

The End

Thank's for your valuable time!

Thanks a lot to the organizers, Kapil Ahuja,
for establishing this winter school, and
Rajendra Choudhary for general help at all times.

Questions?

Thomas WICK
thomas.wick@ifam.uni-hannover.de
www.ifam.uni-hannover.de
www.thomaswick.org