

UNIVERSIDADE FEDERAL DO PARANÁ

LUCAS SEIFFERT

UM ALGORITMO DE ANÁLISE ISOGEOMÉTRICA PARA  
PROBLEMAS DO TIPO INTERAÇÃO FLUIDO-ESTRUTURA

CURITIBA

2017



LUCAS SEIFFERT

**UM ALGORITMO DE ANÁLISE ISOGOMÉTRICA PARA  
PROBLEMAS DO TIPO INTERAÇÃO FLUIDO-ESTRUTURA**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel, Curso de Matemática, Departamento de Matemática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Yuan Jin Yun

CURITIBA

2017



# Resumo

Este trabalho tem o objetivo de apresentar um método numérico, descrito em (BAZILEVS et al, 2008) para resolução de sistemas de equações diferenciais que modelam problemas do tipo interação fluido-estrutura. A discretização espacial é feita usando a abordagem de análise isogeométrica, que utiliza as mesmas funções de base que a descrição geométrica do domínio, no caso, NURBS. A integração temporal utiliza um método chamado de  $\alpha$ -generalizado, proposto inicialmente para problemas de mecânica de sólidos. Apresentaremos conceitos necessários para entender o problema e algumas ideias por trás do algoritmo.

**Palavras-chave:** Interação fluido-estrutura, análise isogeométrica, NURBS, método  $\alpha$ -generalizado.



# Abstract

In this work, we introduce a numerical method, as presented in (BAZILEVS et al. 2008), that calculates the solution to a system of nonlinear partial differential equations. These equations arise as a model to problems of the fluid-structure interaction type. The spacial discretisation is done by means of isogeometric analysis, in which the basis for the functions are given by the NURBS basis functions used to parametrise the domain's geometry. As for the time integration steps, the algorithm is an adapted version of the  $\alpha$ -generalised method, which was initially proposed for solid mechanics. We introduce, firstly, concepts necessary for the understanding of the problem and subsequent algorithms.

**Keywords:** Fluid-structure interaction, isogeometric analysis, NURBS,  $\alpha$ -generalised method.



# Sumário

	<b>Introdução</b> . . . . .	<b>11</b>
<b>1</b>	<b>FORMULAÇÃO DO PROBLEMA</b> . . . . .	<b>13</b>
1.1	O que é um problema de interação fluido-estrutura? . . . . .	13
1.2	Alguns conceitos de mecânica de meios contínuos . . . . .	15
1.3	Apresentação das equações do fluido e estrutura . . . . .	17
1.4	Problema fictício para o acoplamento . . . . .	19
1.5	Formulações fracas . . . . .	20
1.6	Aproximação VMS para o fluido . . . . .	22
<b>2</b>	<b>DESCRIÇÃO DO DOMÍNIO COMPUTACIONAL COM NURBS</b> . . . . .	<b>25</b>
2.1	Introdução ao problema de discretização espacial . . . . .	25
2.2	Introdução a NURBS . . . . .	26
2.2.1	Descrição de curvas . . . . .	26
2.2.2	Descrição de superfícies usando NURBS . . . . .	28
2.2.3	Descrição de sólidos usando NURBS . . . . .	30
2.3	Refinamentos da malha . . . . .	30
2.4	Análise Isogeométrica com NURBS . . . . .	31
2.5	Matrizes de conectividade . . . . .	33
<b>3</b>	<b>DISCRETIZAÇÃO TEMPORAL: O MÉTODO <math>\alpha</math>-GENERALIZADO</b> . . . . .	<b>35</b>
3.1	Apresentação do método . . . . .	35
3.2	Caso FSI . . . . .	37
3.3	Matrizes tangentes . . . . .	39
3.4	Shape derivatives . . . . .	42
<b>4</b>	<b>CONCLUSÃO</b> . . . . .	<b>45</b>
<b>5</b>	<b>REFERÊNCIAS</b> . . . . .	<b>47</b>



# Introdução

Trataremos, neste trabalho, de um método numérico descrito em (BAZILEVS et al., 2008) para simulação de sistemas físicos acoplados chamados de interação fluido-estrutura. Como indicado pelo nome, tratam-se de problemas em que um fluido está em contato com um material sólido elástico, de modo que a mudança de um desses sistemas modifica também o outro sistema. Mais especificamente, há acoplamento de momento linear e velocidade na interface de contato entre os sistemas, e o domínio do fluido acompanha o movimento do domínio do material sólido. São exemplos de fenômenos que pode ser modelados dessa forma pára-quedas, asas de avião, vasos sanguíneos e turbinas eólicas.

Ambos os sistemas têm como base a teoria de mecânica de meios contínuos, e a partir das equações de cada sistema em separado, a primeira dificuldade do problema de interação fluido-estrutura é como acoplar um sistema em outro. Mais especificamente, temos duas tarefas: a primeira é conectar a evolução do domínio computacional do fluido com o movimento da estrutura, e a segunda é aplicar a segunda Lei de Newton (balanço de momento linear) e a continuidade da velocidade, ambos na interface. Para o primeiro problema, há pelo menos três abordagens na literatura. A que usaremos é chamada de formulação Lagrangeana-Euleriana Arbitrária (*ALE*, em inglês) e é simples de escrever, mas adiciona uma nova incógnita ao modelo: a velocidade do domínio do fluido. As duas outras abordagens são por bordo imerso (*immersed boundary*, em inglês), que utiliza ideias de teoria cinética da física em conjunto com algoritmos *Lattice-Boltzmann* (LI; FAVIER, 2017), e por conjuntos de nível (COTTET; MAITRE, 2016).

A dificuldade seguinte é a escolha de resolver o sistema de EDPs totalmente acoplado, chamado de abordagem monolítica, ou de separar os campos do fluido e estrutura, resolvendo cada subproblema separadamente e tentar forçar as condições de acoplamento. Usaremos a abordagem monolítica, que possui boas propriedades de estabilidade numérica, apesar de ser mais cara computacionalmente. Além disso, outra desvantagem é o fato de que não podemos utilizar métodos otimizados para cada sistema em separado; devemos criar um novo especialmente para o problema acoplado. Em termos dos métodos em separado, o grande desafio é garantir a convergência e estabilidade numérica (ver, por exemplo, um estudo para caso simplificado em (CAUSIN; GERBEAU; NOBILE, 2005)).

A próxima etapa é escolher como discretizar o domínio espacial, como por exemplo por elementos finitos. O algoritmo que usaremos, entretanto, utiliza uma ideia recente chamada de ‘análise isogeométrica’. Proposta em (HUGHES; COTRELL; BAZILEVS, 2005), ela surgiu como uma alternativa ao *workflow* de elementos finitos em engenharia, em que máquinas ou estruturas são desenhadas em softwares CAD, mas precisam passar por um passo de pré-processamento para a criação da malha computacional necessária para solução das equações diferenciais. Tal processo é reportado no artigo como custoso

e que gera imprecisões no domínio computacional, resultando em erros ainda maiores durante solução das EDPs. Por isso, a proposta é usar as formulações geométricas usadas pelos CAD (como por exemplo, *NURBS*) como malha computacional, e utilizar também a abordagem isoparamétrica. Assim, as bases de *NURBS* serão também utilizadas como base para as funções a serem encontradas. Estudos mais recentes tentam utilizar outro tipo de funções usadas em computação gráfica, como T-splines, que se comportam melhor que *NURBS* em junções de duas superfícies, por exemplo (BAZILEVS et al., 2010).

Ainda assim, a discretização temporal se baseia na aproximação de Petrov-Galerkin. Para o fluido, utilizaremos uma aproximação multi-escala variacional na velocidade, o que permite um melhor controle e precisão sobre turbulências.

A última dificuldade é a escolha da integração no tempo após a discretização espacial. Também conforme (BAZILEVS et al., 2008), usaremos o método  $\alpha$ -generalizado, proposto em (CHUNG; HULBERT, 1993) no contexto de simulação em mecânica de estruturas. Este método, que pode ser classificado como preditor-corretor, permite um melhor controle sobre dissipações numéricas em frequências altas e baixas.

Quanto à organização do trabalho, pretendemos, no **Capítulo 1**, introduzir conceitos de mecânica de meios contínuos para que seja possível a compreensão da modelagem do problema e suas equações diferenciais. Apresentaremos, então, essas equações e indicamos como é feito o acoplamento, que necessita de uma equação diferencial a mais. Por fim, apresentamos a formulação variacional dessas equações e comentamos sobre a aproximação variacional multi-escala que usaremos para o fluido.

No **Capítulo 2**, pretendemos introduzir conceitos de representação de objetos geométricos utilizando a abordagem *NURBS*, para então explicar como é aplicada a análise isogeométrica ao nosso problema. Indicamos, também, alguns detalhes computacionais.

Por fim, descrevemos no **Capítulo 3** o método  $\alpha$ -generalizado. Iniciamos o capítulo indicando as ideias por trás do algoritmo no contexto em que foi criado, de mecânica dos sólidos, para em seguida aplicá-lo ao caso de interação fluido-estrutura. Terminamos o capítulo discutindo sobre questões técnicas, mas importantes, oriundas da linearização das EDPs.

# 1 Formulação do problema

Neste primeiro capítulo, introduzimos as equações diferenciais para um problema do tipo interação fluido-estrutura. Na primeira seção, definimos e comentamos alguns conceitos de mecânica de meios contínuos que farão parte das equações. As referências principais para esta seção são (BATCHELOR, 2000), (CIARLET, 1988) e (MARSDEN; HUGHES, 1983). Na segunda seção, apresentamos as equações para o fluido e para a estrutura, e como é feito o acoplamento entre esses dois sistemas. Considerações sobre o acoplamento levam à seção seguinte, em que introduzimos uma equação diferencial para tratar do movimento do domínio do fluido. Por fim, na seção 1.4 comentamos brevemente sobre as formulações fracas das equações e, na seção 1.5, apresentamos uma formulação de aproximação *variational multiscale* para o fluido.

## 1.1 O que é um problema de interação fluido-estrutura?

A forma mais simples de se compreender o que é um problema do tipo de interação fluido-estrutura é através de um exemplo. Suponha que queremos estudar a dinâmica de uma asa de avião durante o voo. Uma forma de fazer isso é analisar o campo de velocidade do ar que passa pela asa, de um lado, e a deformação da asa, de outro. O que é possível observar experimentalmente é que o ar deforma a asa que, por sua vez, irá modificar o campo de velocidade do ar.

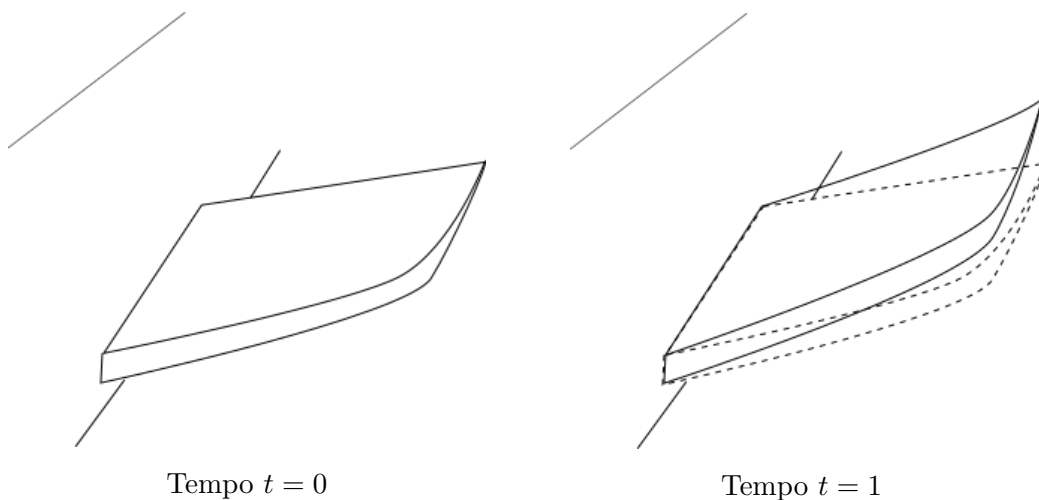


Figura 1 – Movimento hipotético da asa de um avião durante o voo

De fato, a dificuldade do estudo desse tipo de problema está no fato que a “interação” entre os dois sistemas, fluido e estrutura, requer que a análise dos campos das grandezas a

serem estudadas seja acoplada, isto é, estudada em conjunto, através de uma equação que conecte esses dois campos. A simulação do fluido, por exemplo, depende da obtenção da deformação de seu domínio a cada instante de tempo, que depende de como a estrutura está sendo deformada por esse fluido.

Em termos da modelagem matemática do problema, precisamos obter os seguintes componentes:

1. equações de como o campo das grandezas se comporta dentro do fluido ou estrutura, com domínio e valor de contorno fixos. Este item corresponde a um estudo dos problemas de fluido e estrutura ‘em separado’;
2. uma equação, dita de acoplamento, que conecte os campos das grandezas na interface entre o fluido e a estrutura;
3. uma forma de obter a deformação dos domínios no tempo.

Para isso, fixamos a notação para divisão dos domínios e fronteiras do problema, que serão melhor descritos no decorrer do trabalho. Considerando o exemplo da asa de avião acima, e simplificando para uma fatia 2-dimensional, apenas para ilustrar essas notações.

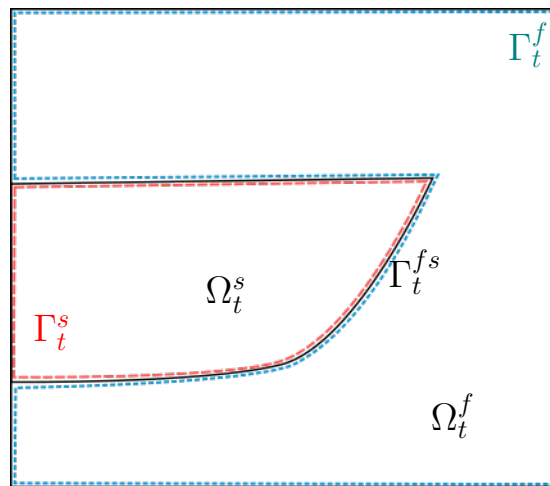


Figura 2 – Notações para domínio e fronteira

Como na figura acima, denotamos por  $\Omega_t^s$  e  $\Omega_t^f$  o domínio da estrutura e do fluido, respectivamente, no tempo  $t$ ;  $\Gamma_t^s$  e  $\Gamma_t^f$  as fronteiras da estrutura e fluido, respectivamente, também em um tempo  $t$ ; e a porção das fronteiras que se intersectam, isto é, a parte de interface entre fluido e estrutura, por  $\Gamma_t^{fs}$ , em um tempo  $t$  fixado.

Seguimos, agora, para a parte efetiva de modelagem de um problema do tipo interação fluido-estrutura.

## 1.2 Alguns conceitos de mecânica de meios contínuos

A modelagem do fenômeno de interação fluido-estrutura pode ser feita através da teoria de mecânica de meios contínuos. Nesta abordagem macroscópica, similar à termodinâmica, assumimos por hipótese que as grandezas que nos interessam têm ordem muito maior que a distância entre as moléculas do objeto a ser estudado (BATCHELOR, 2000). Um caso em que essa abordagem não funciona é o de gases rarefeitos, uma vez que há poucas moléculas distribuídas pelo espaço, de modo que uma teoria mais adequada é a de cinética física.

A hipótese de modelagem acima nos permite tratar pontos de  $\mathbb{R}^3$  como “elementos de volume”, isto é, que um ponto tem uma quantidade de moléculas suficiente para que as leis de conservação da mecânica newtoniana sejam válidas. Assim, podemos falar de velocidade, densidade e pressão, por exemplo, em um ponto de  $\mathbb{R}^3$ . A essa hipótese é comum dar o nome de “hipótese do contínuo”.

Definimos inicialmente o que queremos dizer com um objeto a ser modelado, e a ideia intuitiva é a de um objeto sólido, que chamaremos de *corpo* (*body*, nos textos em inglês), em  $\mathbb{R}^3$ .

**Definição.** Um corpo  $\mathcal{B}$  é um subconjunto aberto de  $\mathbb{R}^3$ .

Devido à semelhança entre a teoria de deformação de corpos e a de geometria em variedades, é comum também definir um corpo como uma variedade diferenciável de dimensão 3 (TRUESDELL; NOLL, pg.37).

Estamos interessados, então, em modelar o movimento de um corpo  $\mathcal{B}$  durante um intervalo de tempo  $[0, T]$ . Vamos denotar o subconjunto de  $\mathbb{R}^3$  que o corpo ocupa no espaço, num tempo  $t$ , por  $\Omega_t$ . Em particular, considerando  $\mathcal{B}$  como conjunto,  $\Omega_0 \doteq \mathcal{B}$ . A primeira coisa a se fazer é definir uma *função de movimento*. É ela quem fornece em que ponto do espaço se encontra um elemento de volume no tempo  $t$ , que no tempo  $t = 0$  estava em  $\mathbf{x} \in \Omega_0$ .

**Definição.** Uma função de movimento de um corpo  $\mathcal{B}$  é uma aplicação

$$\Phi: \Omega_0 \times [0, T] \rightarrow \mathbb{R}^3.$$

Esta é uma definição genérica, que admite funções de movimento fisicamente impossíveis. Por isso, podemos pedir também que  $\Phi$  seja injetora, para que a deformação obedeça ao princípio de que duas partículas não podem ocupar o mesmo lugar no espaço ao mesmo tempo. Em particular, isso vale para elementos de volume (ou pontos de  $\mathbb{R}^3$ , a partir da hipótese do contínuo). Além disso, queremos que o conjunto imagem da função de movimento seja um conjunto aberto para todo  $t$ , de modo que durante a deformação, um corpo seja deformado em um conjunto que continue sendo um corpo.

Se quisermos obter o quanto o ponto  $\mathbf{x} \in \Omega_0$  se deslocou de sua posição inicial, em um tempo  $t$ , basta utilizarmos a função deslocamento.

**Definição.** A função deslocamento associada a uma deformação  $\Phi$  de  $\mathcal{B}$  é definida por

$$\hat{\mathbf{u}}(\mathbf{x}, t) \doteq \Phi(\mathbf{x}, t) - \mathbf{x}, \quad \forall \mathbf{x} \in \Omega_0, \forall t \in [0, T].$$

Dado um corpo  $\mathcal{B}$  com função de movimento  $\Phi$ , definiremos as aplicações abaixo para facilitar o cálculo das mudanças no volume, comprimento e área durante o movimento do corpo. Essas aplicações aparecem nas equações diferenciais que tentaremos resolver.

**Definição.** Chamamos de gradiente de deformação a aplicação  $\mathbf{F}(\mathbf{x}, t) \doteq \nabla \Phi(\mathbf{x}, t)$ , e denotaremos  $J(\mathbf{x}, t) \doteq \det(\mathbf{F}(\mathbf{x}, t))$ .

**Definição.** Chamamos de tensor de Cauchy-Green (à direita) a aplicação

$$\mathbf{C}(\mathbf{x}, t) \doteq \mathbf{F}(\mathbf{x}, t)^T \mathbf{F}(\mathbf{x}, t).$$

Através de  $\mathbf{F}$ ,  $\mathbf{C}$  e  $J$  conseguimos calcular as variações de comprimentos, áreas e volumes de partes do corpo  $\mathcal{B}$ , em um tempo fixo  $t$ , com relação à configuração inicial  $\Omega_0$  (CIARLET, 1988, pg. 31).

Apresentaremos o cálculo mais simples, de volume. Seja  $A \subset \Omega_0$ . Então, pela fórmula de mudança de variável na integral, o volume em um instante  $t$  dos elementos de volume que estavam em  $A$  se torna

$$\text{vol}(\Phi_t(A)) = \int_A J(\mathbf{x}, t) \, d\mathbf{x}.$$

Não vamos, neste trabalho, deduzir as equações de conservação através da teoria de mecânica de meios contínuos. No entanto, é relevante discutir sobre relações constitutivas, pois são o conceito que dita, a grosso modo, o comportamento do material quando sujeito a forças que tentam deformá-lo.

**Definição.** Um tensor de Cauchy (*stress tensor*, em inglês)  $\boldsymbol{\sigma}$ , é um tensor que associa, a cada vetor unitário  $\mathbf{n}$ , um vetor que indica a força (por unidade de área) no elemento de superfície  $\mathbf{n}$ .

Uma relação constitutiva é o que modela como  $\boldsymbol{\sigma}$  depende da função de movimento. Por exemplo,  $\boldsymbol{\sigma}$  pode depender da velocidade de deformação, ou depender linearmente da função deslocamento. Casos simplificados de uma dimensão permitem uma interpretação mais direta de uma relação constitutiva. Seja  $\mathbf{I} \subset \mathbb{R}$  uma barra de um material, como aço. Neste caso, podemos esticar a barra e medir o quanto ela foi esticada,  $\epsilon$  e o quanto de tração foi necessário,  $\sigma$ . Supondo que a relação constitutiva do aço seja a mesma para

todo ponto em  $\mathbf{I}$ , o gráfico  $\sigma \times \epsilon$ , disponível em livros de engenharia, como (GERE, 2004, pg.), permite mostrar que  $\sigma$  depende linearmente de  $\epsilon$ , em deslocamentos pequenos.

Retornando ao caso tridimensional, em que  $\sigma$  é o tensor de Cauchy e  $\epsilon$  é um vetor que indica o quanto um ponto se deslocou, dizemos que um material é elástico quando  $\sigma$  é uma aplicação linear. Nesse caso,  $\sigma$  pode ser escrita como  $\sigma = \mathcal{C}\epsilon$ , com  $\mathcal{C}$  um 4-tensor (ou aplicação que leva matrizes em matrizes) chamado de módulo de elasticidade. A equação anterior é também chamada de Lei de Hooke generalizada.

### 1.3 Apresentação das equações do fluido e estrutura

Utilizaremos as equações de Navier-Stokes para tentar encontrar os campos de pressão e velocidade de um fluido. Assumimos que o fluido seja incompressível, isto é, que sua densidade  $\rho^f$  não varie com o tempo. Supomos também que o fluido seja Newtoniano, de modo que sua viscosidade  $\mu^f$  seja não-nula e constante no tempo e sua relação constitutiva seja da forma

$$\sigma^f = -p\mathbf{I} + \mu^f(\nabla\mathbf{v} + \nabla\mathbf{v}^T).$$

Em problemas de interação fluido-estrutura, o domínio do fluido deve acompanhar o movimento do sólido, e por isso temos dois problemas:

- i) as equações do fluido devem ser escritas em um domínio que varie com o tempo, e
- ii) para fins de cálculo numérico, devemos encontrar uma função de movimento para este domínio do fluido.

O segundo item é discutido na seção seguinte. Com relação ao primeiro, consideramos a versão de Navier-Stokes na formulação Lagrangeana-Euleriana Arbitrária (de sigla *ALE*, em inglês), que permite um domínio com velocidade  $\hat{\mathbf{v}}$ :

$$\begin{cases} \nabla \cdot \mathbf{v} = 0, & \text{em } \Omega_t \\ \rho^f \frac{\partial \mathbf{v}}{\partial t} + \rho^f (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla \mathbf{v} - \nabla \cdot \sigma^f = \rho^f \mathbf{f}^f, & \text{em } \Omega_t \end{cases}$$

com força, por exemplo gravidade,  $\mathbf{f}^f$ .

As condições de fronteira se enquadram no problema de tipo misto: podemos ter condições de Dirichlet ou Neumann. Dividimos a fronteira  $\Gamma^f$ , do fluido, num tempo  $t$ , em três partes:  $\Gamma_t^f = \overline{\Gamma_t^{f,s} \cup \Gamma_t^{f,N} \cup \Gamma_t^{f,D}}$ . Os superíndices  $N$  e  $D$  se referem, respectivamente, às partes de Neumann e Dirichlet da fronteira, e  $\Gamma^{f,s}$  é a fronteira interface entre a estrutura e o fluido. Então, as condições de bordo, ainda sem o acoplamento, se tornam, para  $\mathbf{g}^f$  e  $\mathbf{h}^f$  especificados,

$$\begin{cases} \mathbf{v} = \mathbf{g}^f, & \text{em } \Gamma_t^{f,D} \\ \sigma \mathbf{n}^f = \mathbf{h}^f, & \text{em } \Gamma_t^{f,N} \end{cases}$$

Para a estrutura elástica, consideramos um material hiperelástico, isto é, que não é necessariamente elástico mas admite uma função de energia potencial. O modelo utilizado no artigo do algoritmo a ser apresentado, (BAZILEVS et al, 2008), possui relação constitutiva apresentada em detalhes em (SIMO; HUGHES, 1998), e é construído de forma que permita modelar com mais fidelidade situações em que há dilatações ou compressões de maior magnitude. O que queremos encontrar é a função deslocamento  $\mathbf{u}$ , definida no domínio de referência da estrutura, denotado por  $\Omega_0^s$ .

$$\rho_0 \frac{\partial^2 \mathbf{u}}{\partial s^2} - \nabla \cdot (\mathbf{F}\boldsymbol{\Sigma}) = \rho_0 \mathbf{f} \quad \text{em } \Omega_0^s,$$

$$\boldsymbol{\Sigma} = \mu^s J^{-2/3} \left( \mathbf{I} - \frac{1}{3} \text{tr}(\mathbf{C}) \mathbf{C}^{-1} \right) + \frac{1}{2} \kappa^s (J^2 - 1) \mathbf{C}^{-1},$$

$\mu^s$  e  $\kappa^s$  sendo as constantes do material chamadas módulo de cisalhamento e módulo volumétrico, respectivamente, e que podem ser obtidas em testes de laboratório.  $\rho_0 = \rho^s J$ .

A relação constitutiva desse sólido é dada por  $\boldsymbol{\sigma}^s = J^{-1} \mathbf{F} \boldsymbol{\Sigma} \mathbf{F}^T$ .

Para a linearização das equações do sólido, que utilizaremos no algoritmo, devemos indicar também o módulo de elasticidade, que é um 4-tensor:

$$\begin{aligned} \mathbf{C} = & \left( \frac{2}{9} \mu^s J^{-2/3} \text{tr} \mathbf{C} + \kappa^s J^2 \right) \mathbf{C}^{-1} \otimes \mathbf{C}^{-1} + \left( \frac{2}{3} \mu^s J^{-2/3} \text{tr} \mathbf{C} - \kappa^s (J^2 - 1) \right) \mathbf{C}^{-1} \odot \mathbf{C}^{-1} \\ & - \frac{2}{3} \mu^s J^{-2/3} (\mathbf{I} \otimes \mathbf{C}^{-1} + \mathbf{C}^{-1} \otimes \mathbf{I}). \end{aligned}$$

e com produtos tensoriais  $(\mathbf{C}^{-1} \otimes \mathbf{C}^{-1})_{ijkl} = (\mathbf{C}^{-1})_{ij} (\mathbf{C}^{-1})_{kl}$  e

$$(\mathbf{C}^{-1} \odot \mathbf{C}^{-1})_{ijkl} = \frac{(\mathbf{C}^{-1})_{ik} (\mathbf{C}^{-1})_{jl} + (\mathbf{C}^{-1})_{il} (\mathbf{C}^{-1})_{jk}}{2}$$

De forma análoga ao problema do fluido, podemos ter condições de fronteira de Dirichlet ou Neumann. Aqui, colocamos as condições com relação ao domínio de referência,  $\Omega_0^s$ . Então,  $\Gamma_0^s = \overline{\Gamma_0^{fs} \cup \Gamma_0^{s,N} \cup \Gamma_0^{s,D}}$ . Em  $\Gamma^{s,D}$ , temos que o deslocamento é conforme imposto por uma função  $g^s$ , ié,  $u = g^s$  em  $\Gamma^{s,D}$ . No caso de condição de Neumann, é imposta uma tração  $h^s$ . Assim,

$$\begin{cases} \mathbf{u} = \mathbf{g}^s, & \text{em } \Gamma_0^{s,D} \\ \boldsymbol{\sigma}^s \mathbf{n}^s = \mathbf{h}^s & \text{em } \Gamma_0^{s,N}. \end{cases}$$

O acoplamento das equações do fluido com as da estrutura tem duas etapas. Na etapa física, impomos balanço de momento linear e continuidade da velocidade na interface  $\Gamma_t^{fs}$ ,

$$\begin{cases} \mathbf{v} = \frac{\partial \mathbf{u}}{\partial t} \circ \Phi^{-1} \\ \boldsymbol{\sigma}^f \mathbf{n}_t^f + \boldsymbol{\sigma}^s \mathbf{n}_t^s = \mathbf{0}, \end{cases}$$

com  $\boldsymbol{\sigma}^f$ ,  $\boldsymbol{\sigma}^s$  os tensores de Cauchy do fluido e da estrutura, respectivamente, e  $\mathbf{n}_t^f$ ,  $\mathbf{n}_t^s$  os vetores normais à fronteira do fluido e estrutura no tempo  $t$ .

A segunda etapa é chamada de geométrica, e consiste em encontrarmos o novo domínio para o fluido.

## 1.4 Problema fictício para o acoplamento

Queremos achar a função de movimento do domínio do fluido. A princípio, durante a resolução das equações diferenciais do sólido em um instante de tempo  $t$ , obtemos o deslocamento da interface sólido-fluido. Precisamos, então, estender de alguma forma esse deslocamento da fronteira ao domínio todo.

Uma das formas de se fazer isso é usar extensão harmônica, isto é, usar um problema de Laplace com fronteira definida. Mas, no algoritmo que estudamos, usaremos um problema de elasticidade fictício, no domínio de referência  $\Omega_0$ . Isso permite mais graus de liberdade para um maior controle sobre o comportamento da extensão.

Um problema de equilíbrio de elasticidade tem como incógnita uma função de deslocamento do material,  $\hat{\mathbf{u}}$ , definida em  $\Omega_0$ , e tem como equações, com condições de contorno mistas em  $\Gamma = \partial\Omega_0 = \overline{\Gamma_0 \cup \Gamma_1}$ ,

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma}^m = \mathbf{f}^m & \text{em } \Omega_0 \\ \hat{\mathbf{u}} = \mathbf{g} & \text{em } \Gamma_0 \\ \boldsymbol{\sigma}^m \mathbf{n}^m = \mathbf{h}^m & \text{em } \Gamma_1, \end{cases}$$

com relação constitutiva

$$\boldsymbol{\sigma}^m(\hat{\mathbf{u}}) = \lambda^m \text{tr}(\boldsymbol{\epsilon}(\hat{\mathbf{u}})) \mathbf{I} + 2\mu^m \boldsymbol{\epsilon}(\hat{\mathbf{u}}),$$

$\mu^m$  e  $\lambda^m$  constantes que dependem do material, e

$$\boldsymbol{\epsilon}(\hat{\mathbf{u}}) = \frac{1}{2}(\nabla \hat{\mathbf{u}} + (\nabla \hat{\mathbf{u}})^T).$$

No caso da evolução do domínio do fluido, consideramos  $\mathbf{f}^m = \mathbf{0}$ . Além disso, como um problema de elasticidade linear é válido apenas para pequenos movimentos, calculamos apenas a mudança no domínio do instante de tempo calculado por último,  $\tilde{t}$ , para o novo  $t$ . Isso significa que vamos substituir na equação, no lugar de  $\hat{\mathbf{u}}$ ,  $\hat{\mathbf{u}}(t) - \hat{\mathbf{u}}(\tilde{t})$ :

$$\nabla \cdot \boldsymbol{\sigma}^m(\hat{\mathbf{u}}(t) - \hat{\mathbf{u}}(\tilde{t})) = 0.$$

Em particular, pela linearidade dos operadores envolvidos, e temos

$$\nabla \cdot \boldsymbol{\sigma}^m(\hat{\mathbf{u}}(t)) = \nabla \cdot \boldsymbol{\sigma}^m(\hat{\mathbf{u}}(\tilde{t})).$$

Na interface sólido-fluido  $\Gamma_t^{fs}$ , impomos a condição de Dirichlet  $\hat{\mathbf{u}} = \mathbf{u}_t \circ \Phi_t^{-1}$ , onde  $\mathbf{u}_t$  é a função deslocamento do sólido no tempo  $t$ , obtida na solução conjunta do sólido com o fluido. No restante, impomos condição de Neumann de tração zero, isto é, que  $\boldsymbol{\sigma}^m(\hat{\mathbf{u}}(t)) \mathbf{n}_t^m = \boldsymbol{\sigma}^m(\hat{\mathbf{u}}(\tilde{t})) \mathbf{n}_t^m$ .

Os coeficientes do material fictício, definidos através de outros coeficientes chamados de *módulo de Young*,  $E_m$ , e *módulo de Poisson*,  $\nu_m$ , são dados por

$$\mu^m = \frac{E_m^h}{2(1 + \nu_m)},$$

$$\lambda^m = \frac{\nu_m E_m^h}{(1 + \nu_m)(1 - 2\nu_m)},$$

$$E_m^h = E_m \left( \frac{J_{x\xi}}{J_{x\xi}^0} \right)^{-\chi},$$

com  $\mu^m$ ,  $\lambda^m$ ,  $\chi$  a serem escolhidos. Conforme (BAZILEVS; TAKIZAWA; TEZDUYAR, 2013, pg. 108),  $E_m$  pode afetar o número de condição da matriz de discretização e, usualmente,  $E_m = 1$ ,  $\mu^m \in [0, 0.5)$  e  $J_{x\xi}^0 = 1$ . Além disso, precisamos que  $\chi > 0$ . O parâmetro  $J_{x\xi}$  é o jacobiano da parametrização da geometria (ver capítulo seguinte sobre NURBS).

Um controle diferenciado sobre a evolução do domínio pode ser dado, por exemplo, quando simulamos um balão. Usamos  $E_m$  como aplicação exponencial em relação à direção radial, de forma que haja maior concentração de *patches* da discretização mais perto da fronteira com o sólido. De forma semelhante, podemos em geral tentar forçar que os *patches* menores do domínio, que estão em regiões onde se espera maior complexidade na dinâmica do fluido, sofram pouco movimento, enquanto que os *patches* maiores, em regiões menos complexas, absorvam a maior parte do movimento do domínio.

## 1.5 Formulações fracas

Colocar um problema de equações diferenciais parciais em sua formulação fraca permite, por um lado, que soluções fisicamente possíveis mas não necessariamente diferenciáveis no sentido estrito sejam solução do problema. Por outro lado, a formulação fraca permite que apliquemos métodos como o de aproximação de Galerkin para resolver numericamente as equações. Para obter a formulação fraca de uma equação, multiplicamos as equações por funções chamadas *teste* convenientes e aplicamos a fórmula de integração por partes (QUARTERONI, 2016, pg. 33).

O ambiente usual para formulações fracas é o dos chamados *espaços de Sobolev*. Como pré-requisito para definir esses espaços, precisaríamos de noções de *Teoria da Medida* e *Teoria de Distribuições*. Por isso, apenas definiremos esses espaços em linhas gerais e indicamos os textos (BREZIS, 2011) e (ADAMS, FOURNIER, 2003).

Definimos uma relação de equivalência entre funções em um espaço de medida, dizendo que  $f$  e  $g$  são equivalentes quando forem iguais em quase-todo-ponto. Então, o espaço  $L^2(\Omega_t)$  é o espaço das classes de equivalência de funções mensuráveis em  $\Omega_t$  tais que a integral de Lebesgue  $\int_{\Omega_t} |f|^2 d\Omega_t$ , em um representante  $f$  da classe, é finita.

Em seguida, definimos o espaço de Sobolev  $H^1(\Omega_t)$  como o espaço das funções em  $L^2(\Omega_t)$  tais que sua primeira derivada (num sentido especial, dito *de distribuições*) também está em  $L^2(\Omega_t)$ . A vantagem de se trabalhar com  $L^2(\Omega_t)$  e  $H^1(\Omega_t)$  está no fato de ambos serem espaços de Hilbert. Informalmente, isso significa que há um produto interno que induz a norma do espaço. Tal produto interno é definido, com a notação  $(\cdot, \cdot)_V$ ,  $V$  espaço

de funções correspondente, como

$$(f, g)_V = \int_V f(\mathbf{x})g(\mathbf{x})dV.$$

Podemos, então, definir os espaços  $\mathcal{S}$  em que procuraremos a solução do problema e os espaços  $\mathcal{V}$  em que se encontram as funções *teste*. Além disso, precisamos impor as condições de Dirichlet nos espaços de funções. Utilizamos a notação  $(\Gamma_t)_g$  para indicar a parte de Dirichlet da borda do espaço  $\Gamma$ , e  $(\Gamma_t)_{g_i}$  para sua coordenada  $i$ .

Para a velocidade  $\mathbf{v}^f$  do fluido, temos

$$\mathcal{S}_v = \left\{ \mathbf{v} \mid \mathbf{v}(\cdot, t) \in (H^1(\Omega_t))^3, u_i = g_i \text{ em } (\Gamma_t)_{g_i} \right\}$$

$$\mathcal{V}_v = \left\{ \mathbf{w} \mid \mathbf{w}(\cdot) \in (H^1(\Omega_t))^3, w_i = 0 \text{ em } (\Gamma_t)_{g_i} \right\}$$

Para a pressão  $p$  do fluido, temos

$$\mathcal{S}_p = \left\{ p \mid p(\cdot) \in L^2(\Omega_t), \int_{\Omega_t} p d\Omega = 0 \text{ se } \Gamma_t = (\Gamma_t)_g \right\}$$

$$\mathcal{V}_p = \mathcal{S}_p$$

Para o deslocamento  $\mathbf{u}^s$  da estrutura, temos

$$\mathcal{S}_u = \left\{ \mathbf{u}^s \mid \mathbf{u}^s(\cdot, t) \in (H^1(\Omega_t))^3, u_i^s = g_i \text{ em } (\Gamma_t)_{g_i} \right\}$$

$$\mathcal{V}_u = \left\{ \mathbf{w} \mid \mathbf{w}(\cdot) \in (H^1(\Omega_t))^3, w_i = 0 \text{ em } (\Gamma_t)_{g_i} \right\}$$

Por fim, para o deslocamento  $\hat{\mathbf{u}}$  do problema fictício do domínio do fluido,

$$\mathcal{S}^m = \left\{ \hat{\mathbf{u}} \mid \hat{\mathbf{u}} \in (H^1(\Omega_t^f))^3, \hat{\mathbf{u}} = \mathbf{u}_t \circ \Phi_t^{-1} \text{ em } \Gamma_t^{fs} \right\}$$

$$\mathcal{V}^m = \left\{ \mathbf{w}^m \mid \mathbf{w}^m \in (H^1(\Omega_t^f))^3, \mathbf{w}^m = 0 \text{ em } \Gamma_t^{fs} \right\}$$

Usaremos também notações como  $\nabla_X \mathbf{v}$ , para indicar que a derivada é calculada somente com relação às coordenadas espaciais, e  $\nabla_X^s \mathbf{w} = \frac{1}{2}(\nabla_X \mathbf{w} + (\nabla_X \mathbf{w})^T)$ . Então, temos que as formulações fracas são

**Fluido** (sem abordagem multi-escala): queremos encontrar  $\{\mathbf{v}, p\} \in \mathcal{S}_v \times \mathcal{S}_p$  tais que, para todos  $\{\mathbf{w}^f, q^f\} \in \mathcal{V}_v \times \mathcal{V}_p$ , seja satisfeita

$$B^f(\{\mathbf{w}^f, q^f\}, \{\mathbf{v}, p\}; \hat{\mathbf{v}}) = F^f(\{\mathbf{w}^f, q^f\}),$$

com

$$\begin{aligned} B^f(\{\mathbf{w}^f, q^f\}, \{\mathbf{v}, p\}; \hat{\mathbf{v}}) &= \left( \mathbf{w}^f, \rho^f \frac{\partial \mathbf{v}}{\partial t} \right)_{\Omega_t^f} + \left( \mathbf{w}^f, \rho^f (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla_X \mathbf{v} \right)_{\Omega_t^f} \\ &\quad + (q^f, \nabla_X \cdot \mathbf{v})_{\Omega_t^f} - (\nabla_X \cdot \mathbf{w}^f, p)_{\Omega_t^f} + \left( \nabla_X^s \mathbf{w}^f, 2\mu^f \nabla_X^s \mathbf{v} \right)_{\Omega_t^f}, \end{aligned}$$

$$F^f(\{\mathbf{w}^f, q^f\}) = (\mathbf{w}^f, \rho^f \mathbf{f}^f)_{\Omega_t^f} + (\mathbf{w}^f, \mathbf{h}^f)_{\Gamma_t^{f,N}}.$$

**Sólido:** queremos encontrar  $\mathbf{u} \in \mathcal{S}_{\mathbf{u}}$  tal que, para todo  $\mathbf{w}^s \in \mathcal{V}_{\mathbf{u}}$ , seja satisfeita

$$B^s(\mathbf{w}^s, \mathbf{u}) = F^s(\mathbf{w}^s),$$

com

$$B^s(\mathbf{w}^s, \mathbf{u}) = \left( \mathbf{w}^s, \rho_0^s \frac{\partial^2 \mathbf{u}}{\partial t^2} \right)_{\Omega_0^s} + (\nabla_X \mathbf{w}^s, \mathbf{F} \Sigma)_{\Omega_0^s},$$

$$F^s(\mathbf{w}^s) = (\mathbf{w}^s, \rho_0^s f^s)_{\Omega_0^s} + (\mathbf{w}^s, \mathbf{h}^s)_{\Gamma_0^{s,N}}.$$

**Domínio do fluido:** queremos encontrar  $\hat{\mathbf{u}} \in \mathcal{S}^m$  tal que, para todo  $\mathbf{w}^m \in \mathcal{V}^m$ , tenhamos

$$B^m(\mathbf{w}^m, \hat{\mathbf{u}}) = (\nabla_{\tilde{X}}^s \mathbf{w}^m, 2\mu^m \nabla_{\tilde{X}}^s \hat{\mathbf{u}})_{\Omega_t^f} + (\nabla_{\tilde{X}} \cdot \mathbf{w}^m, \lambda^m \nabla_{\tilde{X}} \cdot \hat{\mathbf{u}})_{\Omega_t^f},$$

$$F^m = B^m(\mathbf{w}^m, \mathbf{u}_t \circ \Phi_t^{-1}).$$

## 1.6 Aproximação VMS para o fluido

Apesar de o método de elementos finitos (e aproximação de Galerkin) ser relativamente bem comportado para equações do tipo elíptico, o mesmo não pode ser dito para outros tipos de equações. Por exemplo, para equações do tipo convecção-difusão, sabe-se que é necessário utilizar métodos de estabilização (QUARTERONI, 2016).

A aproximação *Variational Multiscale* (VMS), apresentada em (HUGHES, 1995), é uma espécie de generalização de algumas abordagens de estabilização de métodos para resolução de equações diferenciais parciais. Decompomos o espaço de Hilbert das possíveis soluções da equação,  $\mathcal{S}$ , em uma soma direta  $\mathcal{S} = S_1 \oplus S_2$ , de um espaço de dimensão finita  $S_1$ , dito de escala “grande” (*coarse*, em inglês), e outro de dimensão infinita,  $S_2$ , dito de escala fina. Utilizaremos  $S_1$  para calcular a solução, e aproximar  $S_2$  através de residuais da equação. Na prática, isso significa que o espaço de dimensão finita, usualmente o espaço de polinômios, é o que utilizamos para aproximar a solução, e o espaço de escala fina terá uma fórmula fechada de cálculo.

Tal ideia pode ser formalizada através da escolha de um operador projeção  $\mathcal{P}: \mathcal{S} \rightarrow S_1$ , tal que se  $\bar{\mathbf{u}} \in \mathcal{S}$  pode ser escrito como a decomposição

$$\bar{\mathbf{u}} = \mathbf{u} + \mathbf{u}', \quad \mathbf{u} \in S_1, \quad \mathbf{u}' \in S_2,$$

então  $\mathbf{u} = \mathcal{P}\bar{\mathbf{u}}$ . (COTTRELL; HUGHES; BAZILEVS, 2009, p. 235).

A aplicação do método VMS para o problema do fluido consiste em decompor os espaços

$$\mathcal{S}_v = S_1 \oplus S_2,$$

$$\mathcal{S}_p = S_{1,p} \oplus S_{2,p},$$

e substituímos  $\bar{\mathbf{v}} = \mathbf{v} + \mathbf{v}' \in \mathcal{S}_v$  e  $\bar{p} = p + p' \in \mathcal{S}_p$  na formulação fraca do fluido.

A fórmula fechada para a aproximação na escala fina é dada a partir do resíduo da aproximação finita multiplicada por uma constante. Isto é, um elemento do espaço de escala fina, é aproximado por

$$\mathbf{v}' = \tau_{SUPS} \left( \rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla \mathbf{v} - \nabla \cdot \boldsymbol{\sigma}^f - \rho^f f^f \right)$$

e

$$p' = \nu_{VLSIC} (\nabla \cdot \mathbf{v}).$$

As constantes de aproximação, neste trabalho, são escolhidas de modo a generalizar estabilizações conhecidas de simulações em fluidos, como *SUPS* (*Streamline-Upwind/Petrov-Galerkin*) e *PSPG* (*Pressure-Stabilizing/Petrov-Galerkin*) (COTTRELL; HUGHES; BAZILEVS, 2009). Essas constantes têm, então, como fórmulas,

$$\tau_{SUPS} = \left( \frac{C_I}{\Delta t^2} + (\mathbf{v} - \hat{\mathbf{v}}) \cdot G(\mathbf{v} - \hat{\mathbf{v}}) + \left( \frac{\mu^f}{\rho^f} \right)^2 G : G \right)^{-1/2},$$

$$\nu_{VLSIC} = (\tau_M \mathbf{g} \cdot \mathbf{g})^{-1},$$

em que é levada em conta a métrica Riemanniana do domínio,

$$G_{ij} = \sum_{k=1}^d \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j},$$

$$(\mathbf{v} - \hat{\mathbf{v}}) \cdot G(\mathbf{v} - \hat{\mathbf{v}}) = \sum_{i,j=1}^d (v_i - \hat{v}_i) G_{ij} (v_j - \hat{v}_j),$$

$$g_i = \sum_{j=1}^d \frac{\partial \xi_j}{\partial x_i},$$

$$\mathbf{g} \cdot \mathbf{g} = \sum_{i=1}^3 g_i g_i,$$

com  $C_I$ , uma constante positiva e  $\frac{\partial \xi}{\partial \mathbf{x}}$  matriz inversa da jacobiana da aplicação de parametrização da geometria.

Apresentamos agora a formulação variacional para o fluido. Denotamos por  $\tilde{\Omega}_t^f$  o produto interno dentro de cada *patch* e  $n_{el}$  o número de *patches* NURBS. A função  $\hat{\mathbf{v}}$  é considerada dada. Queremos encontrar  $\mathbf{v} \in \mathcal{S}_v$  e  $p \in \mathcal{S}_p$  tais que, para todos  $w^f \in \mathcal{V}_v$  e  $q^f \in \mathcal{V}_p$ ,

$$B_{MS}^f(\{\mathbf{w}^f, q^f\}, \{\mathbf{v}, p\}; \hat{\mathbf{v}}) = F_{MS}^f(\{\mathbf{w}^f, q^f\}),$$

onde

$$\begin{aligned} B_{MS}^f(\{\mathbf{w}^f, q^f\}, \{\mathbf{v}, p\}; \hat{\mathbf{v}}) &= B^f(\{\mathbf{w}^f, q^f\}, \{\mathbf{v}, p\}; \hat{\mathbf{v}}) + \sum_{n_{el}} \left( (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla_X \mathbf{w}^f, \mathbf{v}' \right)_{\tilde{\Omega}_t^f} \\ &\quad + \sum_{n_{el}} \left( \nabla_X q^f, \frac{1}{\rho^f} \mathbf{v}' \right)_{\tilde{\Omega}_t^f} + \sum_{n_{el}} (\nabla_X \cdot \mathbf{w}^f \rho^f \nu_{VLSIC}, \nabla_X \cdot \mathbf{v})_{\tilde{\Omega}_t^f} \\ &\quad - \sum_{n_{el}} (\mathbf{w}^f, \mathbf{v}' \cdot \nabla_X \mathbf{v})_{\tilde{\Omega}_t^f} - \sum_{n_{el}} \left( \nabla_X \mathbf{w}^f, \frac{1}{\rho^f} \mathbf{v}' \otimes \mathbf{v}' \right)_{\tilde{\Omega}_t^f} \end{aligned}$$

$$F_{MS}^f(\{\mathbf{w}^f, q^f\}) = F^f(\{\mathbf{w}^f, q^f\}).$$

Por fim, apresentamos a equação variacional para o problema acoplado. Queremos encontrar  $\mathbf{v} \in \mathcal{S}_v$ ,  $p \in \mathcal{S}_p$ ,  $\hat{\mathbf{u}} \in \mathcal{S}^m$  e  $\mathbf{u} \in \mathcal{S}_u$  tais que, para todos  $\mathbf{w}^f \in \mathcal{V}_v$ ,  $q^f \in \mathcal{V}_p$ ,  $\mathbf{w}^s \in \mathcal{V}_u$  e  $\mathbf{w}^m \in \mathcal{V}^m$ ,

$$\begin{aligned} B_{MS}^f(\{\mathbf{w}^f, q^f\}, \{\mathbf{v}, p\}; \hat{\mathbf{v}}) - F_{MS}^f(\{\mathbf{w}^f, q^f\}) + B^s(\mathbf{w}^s, \mathbf{u}) \\ - F^s(\mathbf{w}^s) + B^m(\mathbf{w}^m, \hat{\mathbf{u}}) - F^m(\mathbf{w}^m) = 0 \end{aligned}$$

## 2 Descrição do domínio computacional com NURBS

Neste capítulo, descrevemos alguns conceitos relativos à representação de objetos geométricos através da abordagem NURBS. Na primeira seção, apresentamos o que queremos dizer com NURBS e qual o formato usual de curvas, superfícies e sólidos nessa abordagem. As referências principais são (FARIN, 2003), para apresentação dessa abordagem e (PIEGL; TILLER, 1997), para algoritmos e detalhes sobre construções mais complexas. Enfatizamos que construir uma geometria usando NURBS à mão é trabalhoso, e o mais comum é que isso seja feito softwares CAD e exportado para ser usado no algoritmo.

Na segunda seção apresentamos três técnicas de refinamento para a geometria NURBS, em comparação com técnicas de refinamento usuais do método de elementos finitos. Na terceira seção, explicamos o que significa análise isogeométrica e como ficam as variáveis do problema nessa abordagem. Por fim, comentamos na quarta seção sobre o problema de manter todas as informações sobre estruturas NURBS e do problema de uma forma conveniente para ser usada no algoritmo de solução das EDPs.

### 2.1 Introdução ao problema de discretização espacial

A análise numérica da resolução de equações diferenciais parciais começa, efetivamente, quando escolhemos como lidar com um espaço de funções, de dimensão infinita (e até cardinalidade do contínuo), através de uma aproximação em espaço de dimensão finita. Podemos escolher, por exemplo, espaços de polinômios de grau até certa ordem. Para melhorar a aproximação por polinômios, costuma-se também dividir o domínio em finitos subdomínios, de modo que, nas fronteiras dos subdomínios, seja garantida uma condição de colagem, isto é, continuidade ou regularidades maiores.

Nesse sentido, foi desenvolvido o método chamado de elementos finitos, em que um ponto crucial é a chamada “triangulação” do domínio, que garante resultados de aproximação do espaço de polinômios criados e uma forma relativamente simples de lidar com esses polinômios (QUARTERONI, 2016). Entretanto, a etapa de triangulação é custosa enquanto etapa para a simulação em engenharia, e é uma das motivações para a introdução da *Análise Isogeométrica* (HUGHES; COTTRELL; BAZILEVS, 2004). A ideia dessa nova abordagem é, em vez de discretizar o domínio do problema através de triangulações, utilizar uma discretização “natural” do domínio, que vem com a forma com que *softwares CAD* modelam objetos geométricos.

Uma forma comum de modelagem geométrica em computação gráfica é chamada de NURBS, ou *Non-Uniform Rational B-Splines*.

## 2.2 Introdução a NURBS

### 2.2.1 Descrição de curvas

NURBS é uma forma de representar objetos geométricos, como curvas e superfícies, através de funções racionais. Para sua apresentação, seguiremos a abordagem da evolução histórica das curvas Bézier até chegar às curvas NURBS, conforme (FARIN, 2003).

Começaremos explicando sobre curvas Bézier, e como podemos descrever parábolas usando esse tipo de curvas. A construção é feita usando recursivamente interpolação linear. Vamos considerar um exemplo. Sejam  $\mathbf{b}_0 = (0, 0)$ ,  $\mathbf{b}_1 = (3, 2)$  e  $\mathbf{b}_2 = (6, 0)$  pontos de  $\mathbb{R}^2$ . A primeira etapa de interpolação linear gera os segmentos que ligam esses pontos, e formará o chamado polígono de controle da curva Bézier. Na notação utilizada pelo livro de referência, o índice  $k$  em  $\mathbf{b}_i^k$  indica o ponto  $i$  da iteração  $k$  do algoritmo. Então, para  $t \in [0, 1]$ ,

$$\mathbf{b}_0^1(t) = (1 - t)\mathbf{b}_0 + t\mathbf{b}_1$$

$$\mathbf{b}_1^1(t) = (1 - t)\mathbf{b}_1 + t\mathbf{b}_2.$$

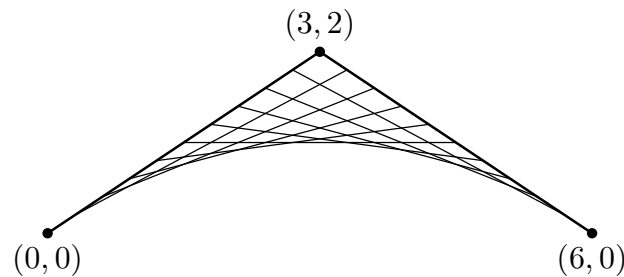


Figura 3 – Exemplo de uma parábola como curva Bézier

A segunda etapa gerará uma parábola, conforme variamos  $t$ :

$$\mathbf{b}_0^2(t) = (1 - t)\mathbf{b}_0^1(t) + t\mathbf{b}_1^1(t) = (1 - t)^2\mathbf{b}_0 + 2t(1 - t)\mathbf{b}_1 + t^2\mathbf{b}_2.$$

As iterações continuam até a iteração  $n-1$ , onde  $n$  é o número de pontos no polígono de controle dado. Assim, um polígono de 3 pontos gera uma parábola, e uma de 4 gera uma curva polinomial de grau 3. O algoritmo recursivo, em sua forma genérica, é conhecido como algoritmo de *de Casteljau* e é dado, em cada etapa  $j \in \{0, \dots, n-1-k\}$  e iteração  $r$ , por

$$\mathbf{b}_j^r(t) = (1 - t)\mathbf{b}_j^{r-1}(t) + t\mathbf{b}_{j+1}^{r-1}(t), \text{ para } r \in \{1, \dots, n\} \text{ e } j \in \{0, \dots, n-r\}.$$

A vantagem das curvas de Bézier é a sua invariância por transformações afins. Isso significa que a curva não muda qualitativamente quando são aplicadas rotações, translações ou multiplicação por constantes. Dessa forma, aplicar a transformação afim no

polígono de controle e depois calcular os pontos da curva é o mesmo que calcular cada ponto e em seguida sua imagem sobre a transformação afim.

No entanto, a descrição de curvas mais complexas pode requerer graus cada vez mais altos do polinômio e, por consequência, de um número cada vez maior de pontos no polígono de controle. Por isso, o passo seguinte para representação de curvas em computação gráfica é o uso de splines.

**Definição.** Seja  $\Xi = [\xi_1, \xi_2, \dots, \xi_n]$  um vetor de  $n + 1$  nós distintos e ordenados de um intervalo  $[a, b]$ , isto é,  $a = \xi_0 < \xi_1 < \dots < \xi_n = b$ . Uma função  $s_k: [a, b] \rightarrow \mathbb{R}$  é dita *spline* de grau  $k$  ( $k \geq 1$ ) relativa ao vetor de nós  $\Xi$  se

- i)  $s_k|_{[\xi_j, \xi_{j+1}]} \in \mathbb{P}_k, \quad j \in 1, \dots, n - 1$
- ii)  $s_k \in C^{k-1}[a, b]$

Nesse caso, cada subintervalo  $[\xi_i, \xi_{i+1}]$  pode ser descrito por meio de uma curva Bézier de grau  $k$ . A dificuldade, agora, é como garantir que a curva  $s_k$  tenha a regularidade necessária. Detalhes sobre como isso é feito se encontram na referência (FARIN, 2003).

Um caso mais específico é o de curvas definidas através de B-Splines, isto é, as que são escritas por meio de uma base de funções de suporte local (QUARTERONI, et al, 2014).

**Definição.** Seja  $\Xi = [\xi_1, \dots, \xi_{i+p+1}]$  um vetor de nós conforme a definição de *splines*. Então, o elemento da base de funções de grau  $p$  e suporte em  $[\xi_i, \xi_{i+p+1}]$  é denotado por  $N_{i,p}$  e definido recursivamente como

$$N_{i,0}(\xi) = \begin{cases} 1, & \text{se } \xi_i \leq \xi < \xi_{i+1} \\ 0, & \text{caso contrário} \end{cases}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi),$$

A fórmula recursiva acima é chamada de fórmula de *Cox-de Boor*, e pode-se mostrar que essa base forma uma partição de unidade para  $[0, 1]$ . Podemos, então, representar uma curva em  $\mathbb{R}^d$ , usando B-Splines. Dados pontos  $\mathbf{B}_i \in \mathbb{R}^d$ , chamados de pontos de controle da curva, definimos uma curva

$$C(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{B}_i.$$

Como o objetivo principal de softwares CAD é o de representar objetos e situações de engenharia, o requerimento da curva ser  $C^k$  em todo o seu domínio não faz sentido. Sobre isso, temos duas observações

- i) Quando precisamos que a curva de grau  $p$  passe exatamente sobre um dos pontos de controle, repetimos o nó relativo a esse ponto de controle  $p$  vezes. Por exemplo, é

comum pedir que a curva passe exatamente sobre o ponto inicial e final do polígono de controle. Utilizando B-Splines de grau 3, e supondo intervalo  $[0, 1]$ , isso pede que o vetor  $\Xi$  seja da forma  $\Xi = [0, 0, 0, \xi_4, \dots, \xi_{n-3}, 1, 1, 1]$ . Detalhes sobre por que isso funciona, substituindo esse vetor  $\Xi$  na fórmula de Cox-de Boor podem ser encontrados em (FARIN, 2003).

- ii) Fora dos pontos de controle, a curva deve ter o requerido grau de regularidade, e isso se mantém quando fizermos refinamentos (seção abaixo), isto é, colocarmos mais nós ou aumentarmos o grau dos polinômios utilizados.

Cada intervalo dado pelo vetor de nós  $\Xi$  é chamado de *patch*, e tem semelhanças com o que chamamos de elementos na abordagem de elementos finitos. De fato, sabemos que há, no máximo,  $p$  funções de base com suporte em um *patch*, e detalhes como este serão contabilizados quando quisermos organizar a geometria do domínio dado, em uma matriz (ver seção abaixo).

Apresentamos, por fim, a forma de representar curvas usando NURBS, que foram criadas para que seja possível representar curvas cônicas, como circunferências e elipses, de forma exata. São duas as ideias principais. A primeira é utilizarmos as funções de base B-Spline para gerar funções racionais. Observamos que é possível utilizar funções racionais em Bézier ou mesmo splines. A segunda ideia é a de utilizarmos geometria projetiva, isto é, fazermos uma projeção de uma curva em  $\mathbb{R}^4$  para  $\mathbb{R}^3$ , no caso de dimensão 3.

Não daremos detalhes sobre essa construção, e apenas indicaremos que, na prática, isso implica em funções base com pesos  $w_i > 0$ :

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{j=1}^n N_{j,p}(\xi)w_j}.$$

Como no caso de B-Splines, uma curva  $C$  em NURBS será dada por pontos de controle  $\{\mathbf{B}_i\}$  e vetor de nós  $\Xi$ , através de

$$C(\xi) = \sum_i^n R_i^p(\xi)\mathbf{B}_i.$$

## 2.2.2 Descrição de superfícies usando NURBS

A abordagem que usaremos para descrever superfícies pode ser também utilizada com curvas Bézier ou B-splines. A diferença, novamente, é com relação à base de funções utilizada.

O que faremos é representar superfícies em  $\mathbb{R}^d$  como um produto tensorial de duas curvas. Isso quer dizer que uma curva representará a geometria na direção de um parâmetro  $\xi$ , e a outra, na direção de um parâmetro  $\eta$ .

Por exemplo, dadas duas curvas, uma com 3 *patches* e outra com 2 *patches*, escolhamos a com 3 para representar o parâmetro semelhante ao eixo  $Ox$  e a outra, para o parâmetro

fazendo o papel do eixo  $Oy$ . A imagem abaixo tenta mostrar o comportamento de uma superfície que seria o produto tensorial dessas duas curvas. À esquerda, temos os intervalos de parametrização gerando os 6 *patches* mencionados, e à direita, temos uma superfície parametrizada com domínio nesses *patches*.

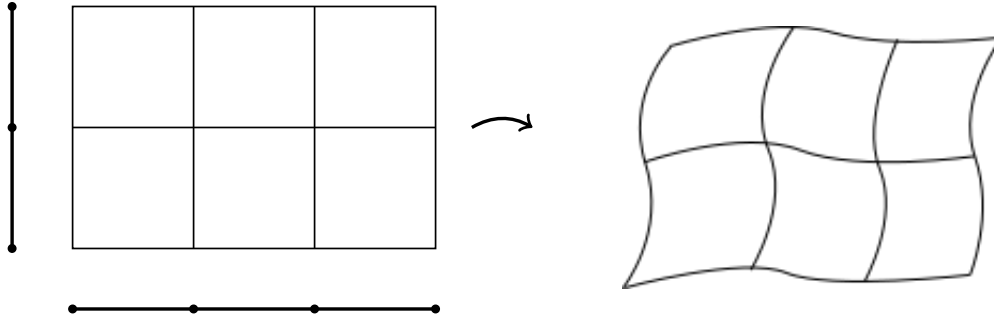


Figura 4 – Comportamento de um produto tensorial de curvas

Em termos de funções de base, precisaremos de dois vetores de controle  $\Xi$  e  $H$ , e pesos  $\{w_{i,j}\}$ . Então, para um grau de polinômios  $p$ , funções de base  $N_{i,p}$  e  $M_{j,p}$  conforme a seção anterior, temos que as funções de base são dadas por

$$R_{i,j}^p(\xi, \eta) = \frac{N_{i,p}(\xi)M_{j,p}(\eta)w_{i,j}}{\sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m N_{\hat{i},p}(\xi)M_{\hat{j},p}(\eta)w_{\hat{i},\hat{j}}}$$

e a superfície  $S$ , dados vetores de controle  $\mathbf{B}_{i,j}$  se torna

$$S(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m R_{i,j}^p(\xi, \eta) \mathbf{B}_{i,j}.$$

Uma observação importante, e que complica a representação NURBS de geometrias relativamente simples, é que a construção tensorial obriga que a superfície tenha, necessariamente, “4 pontas”. Então, se quisermos representar discos ou elementos “triangulares”, precisaremos incluir ao menos uma singularidade à aplicação que constrói a geometria. Detalhes podem ser encontrados em em (PIEGL; TILLER, 1997)

Em (HUGHES; COTTRELL; BAZILEVS, 2004), é descrita a simulação de uma membrana circular e é feito o comentário que a aplicação de geometria possui uma singularidade no centro da membrana, mas que isso não afetou a simulação.



Figura 5 – Superfícies patológicas para a abordagem tensorial

### 2.2.3 Descrição de sólidos usando NURBS

Nas simulações de três dimensões de partes mecânicas ou estruturas sólidas, o que se utiliza é a abordagem tensorial, mas para três curvas. Dada a semelhança com o caso de duas dimensões, apenas colocamos como ficam as funções de base e um sólido nessa formulação.

$$R_{i,j,k}^p(\xi, \eta, \zeta) = \frac{N_{i,p}(\xi)M_{j,p}(\eta)L_{k,p}(\zeta)w_{i,j,k}}{\sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^l N_{\hat{i},p}(\xi)M_{\hat{j},p}(\eta)L_{\hat{k},p}(\zeta)w_{\hat{i},\hat{j},\hat{k}}}$$

e um sólido  $S$  é dado por vetores de controle  $\mathbf{B}_{i,j,k}$  e

$$S(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^l R_{i,j,k}^p(\xi, \eta, \zeta) \mathbf{B}_{i,j,k}.$$

## 2.3 Refinamentos da malha

A grande vantagem de se usar NURBS em vez de elementos finitos está no fato de o refinamento da estrutura NURBS preservar a geometria original do problema, de modo que não há necessidade de se voltar ao programa CAD para executar o refinamento. Seguimos o artigo (HUGHES; COTTRELL; BAZILEVS, 2004), em que os tipos de refinamento que se assemelham aos de elementos finitos são apresentados com o mesmo nome. Daremos apenas algumas ideias sobre o tema, e indicamos o artigo acima para detalhes.

A princípio, as explicações são feitas apenas para o caso unidimensional. Quando se tem um produto tensorial, o refinamento de uma das curvas componentes se propaga para o resto da estrutura, mas é necessário fornecer mais pontos de controle.

1. h-refinamento. Trata-se da inserção de nós na estrutura NURBS. Dado um vetor de nós  $\Xi = \{\xi_1, \xi_2, \dots, \xi_{n+p+1}\}$ , seja  $\bar{\xi} \in [\xi_k, \xi_{k+1})$  o nó que se deseja inserir. Então, corrigimos os pontos de controle  $\{B_1, B_2, \dots, B_n\}$  recursivamente através de

$$\bar{B}_i = \alpha_i B_i + (1 - \alpha_i) B_{i-1},$$

$$\begin{cases} 1, & 1 \leq i \leq k - p \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & k - p + 1 \leq i \leq k \\ 0, & k + 1 \leq i \leq n + p + 2 \end{cases}$$

Cuidados adicionais devem ser tomados quando se insere um nó com o mesmo valor de um que já existia antes.

2. p-refinamento. Trata-se da elevação da ordem dos polinômios. Não é um procedimento simples, e também requer alguns cuidados. Por exemplo, devemos repetir os nós para preservar a regularidade da curva representada.

3. k-refinamento. É um método de refinamento não disponível em elementos finitos. Utiliza a característica de que p-refinamento e h-refinamento não são comutativos em NURBS, isto é, aplicar um e depois o outro não gera o mesmo refinamento que aplicar o último antes. O que se explora é que um p-refinamento seguido de um h-refinamento pode dar melhores resultados que os refinamentos anteriores.

## 2.4 Análise Isogeométrica com NURBS

Chamamos de análise isogeométrica o método de discretização que utiliza as mesmas funções de base que parametrizam a geometria do domínio como funções de base das funções que queremos encontrar. Além disso, utilizamos também a abordagem isoparamétrica, em que as funções *teste* são do mesmo formato que as funções incógnitas. Em resumo, o que fazemos é utilizar as bases NURBS para as funções incógnita e teste, e substituímos esses formatos nas formulações fracas. Essa é a etapa de semidiscretização, e o que resta é decidir como discretizar as equações com relação ao tempo (ver capítulo seguinte).

Nos problemas de interação fluido-estrutura, temos uma dificuldade a mais, que é o fato de algumas funções incógnita serem definidas no domínio de referência  $\Omega_0$ , como é o caso do deslocamento do sólido, e outras no domínio que sofreu uma deformação, no tempo  $t$ . Por isso, algumas incógnitas terão funções de base através de uma composição da função de base NURBS com a função de movimento  $\Phi$ .

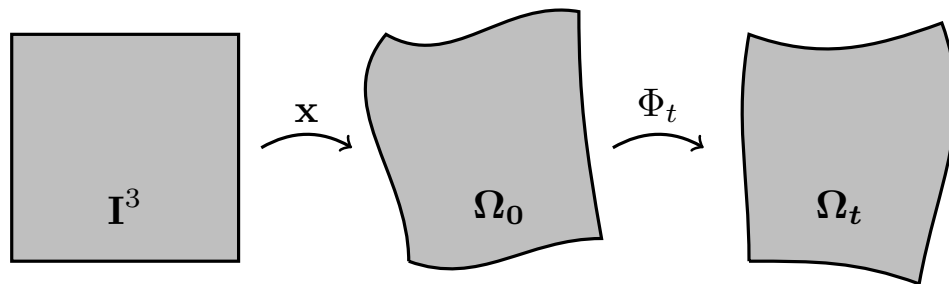


Figura 6 – Os três domínios possíveis: o de parâmetro, o de referência e o atual

Para ilustrar os domínios possíveis para funções em análise isogeométrica e como se conectam, apresentamos a *Figura 6* acima, relativa a um problema em três dimensões. Para a descrição da geometria do domínio inicial  $\Omega_0$ , utilizamos uma parametrização NURBS,  $\mathbf{x}$ , que vai de um cubo unitário  $\mathbf{I}^3$ , dividido em *patches* como na seção 2.1, até  $\Omega_0$ . Dado um tempo  $t \in [0, T]$ , vimos no Capítulo 1 que o domínio se movimentou para  $\Omega_t$  através da função de movimento  $\Phi_t$ .

Passaremos, então, a discutir como usaremos as funções de base NURBS. Essas funções serão base de funções das variáveis na configuração inicial (isto é, nas funções definidas em  $\Omega_0$ ), e serão denotadas por  $\hat{N}_A$ , com  $A$  no conjunto de índices das bases NURBS. Para as funções cujo domínio acompanha a deformação do objeto (isto é, definidas em  $\Omega_t$ ),

vamos utilizar a definição teórica, para  $(\xi, \eta, \zeta) \in \mathbf{I}^3$ ,

$$N_A(\xi, \eta, \zeta, t) = \hat{N}_A \circ \Phi^{-1}(\xi, \eta, \zeta, t).$$

Então, podemos indicar o formato das funções incógnita do problema que queremos resolver. Para a incógnita de deslocamento do sólido, temos  $\mathbf{u}$  dada por

$$\mathbf{u}(\mathbf{x}(\xi, \eta, \zeta), t) = \sum_A \mathbf{u}_A^s(t) \hat{N}_A(\xi, \eta, \zeta),$$

e  $\mathbf{u}^s: [0, T] \rightarrow \mathbb{R}^3$  será a função que queremos efetivamente encontrar.

No caso do fluido, temos

$$\mathbf{v}(\mathbf{x}(\xi, \eta, \zeta), t) = \sum_A \mathbf{v}_A^f(t) N_A(\xi, \eta, \zeta, t),$$

$$p(\mathbf{x}(\xi, \eta, \zeta), t) = \sum_A p_A(t) N_A(\xi, \eta, \zeta, t),$$

e, novamente, as funções a serem encontradas serão  $\mathbf{v}_A^f$  e  $p_A$ .

Para simplificar a escrita, utilizamos o abuso de notação  $\mathbf{x} = \mathbf{x}(\xi, \eta, \zeta)$  no domínio das funções incógnitas e  $\mathbf{x} = (\xi, \eta, \zeta)$ , no domínio das funções de base. De fato, neste último caso, as funções de base têm domínio na inversa de  $\mathbf{x}$ .

Para o problema fictício do domínio do fluido, devemos utilizar o cálculo do tempo anterior  $\tilde{t}$ , de modo que

$$\tilde{N}_A(\tilde{\mathbf{x}}, \tilde{t}) = \hat{N}_A \circ \Phi_t^{-1}(\tilde{\mathbf{x}})$$

e

$$\tilde{\mathbf{u}}(\tilde{\mathbf{x}}, \tilde{t}) = \sum_A \hat{\mathbf{u}}_A(\tilde{t}) \tilde{N}_A(\tilde{\mathbf{x}}, \tilde{t}).$$

Quando precisarmos da velocidade do domínio do fluido, usaremos

$$\hat{\mathbf{v}}(\mathbf{x}, t) = \sum_A \frac{\partial \hat{\mathbf{u}}_A}{\partial t}(t) N_A(\mathbf{x}, t).$$

A observação final é que a definição das funções de base  $N_A$  é utilizada de forma diferente, na prática. Isso porque precisaríamos calcular as inversas de  $\hat{\phi}$  e  $\tilde{\phi}$ , aplicações vetoriais não-lineares. Como só estamos interessados em integração e, convenientemente, sobre todo o domínio de definição das  $N_A$ , usaremos mudança de variável nas integrais, retornando as integrais para  $\Omega_0$ . Por exemplo, para  $t$  fixo,

$$\int_{\Omega_t} N_A(\mathbf{x}, t) d\mathbf{x} = \int_{\Omega_0} \hat{N}_A(\mathbf{x}) |J(\mathbf{x}, t)|^{-1} d\mathbf{x}.$$

## 2.5 Matrizes de conectividade

Comentaremos brevemente sobre o problema de organizar os índices e variáveis do problema. Isso é necessário quando formos percorrer todos os *patches*, funções de base NURBS e número de coordenadas das funções incógnita. Além disso, precisaremos indicar o formato das matrizes de discretização do algoritmo do capítulo seguinte.

Ao substituímos o formato das funções da seção anterior nas formulações fracas, como o suporte das funções de base é local, podemos rastrear cada *patch* e integrar somente o que sabemos que será não nulo. Para isso e para saber quais elementos estão com condições iniciais, usamos matrizes de conectividade. (COTTRELL; HUGHES; BAZILEVS; 2009, pg. 314). São três os pontos importantes sobre a criação dessas matrizes:

- i) devemos identificar quais os *patches* que possuem condição de bordo, pois terão tratamento separado na discretização;
- ii) devemos identificar, para cada índice  $A \in \mathbf{I}$  de funções de base, em quais patches está o suporte de  $N_A$ ;
- iii) no caso de funções vetoriais, devemos indicar onde cada um dos graus de liberdade (ou coordenadas do vetor de constante) estará na matriz discretizada.

Na abordagem da referência indicada, o que criaremos é uma matriz chamada de *matriz de localização* (*location matrix*, em inglês), denotada por LM. Dado um número de grau de liberdade  $i$ , ou  $i$ -ésima coordenada da variável, um número de base local  $a$  e um número de elemento (ou *patch*)  $e$ , interpretamos

$$P = \text{LM}(i, a, e)$$

como a posição (linha ou coluna, dependendo do caso) na matriz discretizada em que será colocado o cálculo da substituição de  $N_A$  na formulação fraca.



## 3 Discretização temporal: o método $\alpha$ -generalizado

Neste capítulo, apresentamos o método  $\alpha$ -generalizado para as equações do problema de interação fluido-estrutura. Na primeira seção, apresentamos o método em sua formulação original, no contexto de elementos finitos para problemas de estruturas em engenharia (CHUNG; HULBERT, 1993). Na seção seguinte, apresentamos o método para o caso de interação fluido-estrutura (sigla *FSI*, em inglês) e depois dedicamos as duas seções finais para a parte complicada e trabalhosa do algoritmo: as matrizes tangentes e *shape-derivatives*.

### 3.1 Apresentação do método

O método  $\alpha$ -generalizado foi criado para resolver equações diferenciais de mecânica dos sólidos. A peculiaridade desse tipo de problema gerou a necessidade de controlar, no algoritmo, a “dissipação” de frequências altas e baixas. Explicaremos o que isso significa mais abaixo. Tal controle requer a introdução de mais graus de liberdade, através de parâmetros, no algoritmo a ser utilizado.

A partir da discretização espacial das equações diferenciais de um problema dessa área, obtemos uma EDO da forma  $\mathbf{M}\ddot{\mathbf{X}} + \mathbf{C}\dot{\mathbf{X}} + \mathbf{K}\mathbf{X} = \mathbf{F}$ . Aqui,  $\mathbf{X}$  e  $\mathbf{F}$  são vetores e  $\mathbf{A}$ ,  $\mathbf{X}$ ,  $\mathbf{C}$ ,  $\mathbf{K}$  são matrizes quadradas.

Discretizamos o intervalo de tempo a ser simulado,  $[t_0, T]$ , de forma homogênea, por intervalos de tempo denotados por  $\Delta t$ . O algoritmo apresentado em (CHUNG; HULBERT, 1993), chamado de  $\alpha$ -generalizado, é dado a seguir.

O passo inicial, em  $t_0$ , é dado por  $\mathbf{X}_0 = \mathbf{d}$ ,  $\dot{\mathbf{X}}_0 = \mathbf{v}$ ,  $\ddot{\mathbf{X}}_0 = \mathbf{M}^{-1}(\mathbf{F}(0) - \mathbf{C}\mathbf{v} - \mathbf{K}\mathbf{d})$ . O passo seguinte é dado implicitamente por:

$$\left\{ \begin{array}{l} \mathbf{X}_{n+1} = \mathbf{X}_n + \Delta t \dot{\mathbf{X}}_n + \Delta t^2 \left( \left( \frac{1}{2} - \beta \right) \ddot{\mathbf{X}}_n + \beta \ddot{\mathbf{X}}_{n+1} \right) \\ \dot{\mathbf{X}}_{n+1} = \dot{\mathbf{X}}_n + \Delta t \left( (1 - \gamma) \ddot{\mathbf{X}}_n + \gamma \ddot{\mathbf{X}}_{n+1} \right) \\ \mathbf{M} \ddot{\mathbf{X}}_{n+1-\alpha_m} + \mathbf{C} \dot{\mathbf{X}}_{n+1-\alpha_f} + \mathbf{K} \mathbf{X}_{n+1-\alpha_f} = \mathbf{F}(t_{n+1-\alpha_f}) \\ \mathbf{X}_{n+1-\alpha_f} = (1 - \alpha_f) \mathbf{X}_{n+1} + \alpha_f \mathbf{X}_n \\ \dot{\mathbf{X}}_{n+1-\alpha_f} = (1 - \alpha_f) \dot{\mathbf{X}}_{n+1} + \alpha_f \dot{\mathbf{X}}_n \\ \ddot{\mathbf{X}}_{n+1-\alpha_m} = (1 - \alpha_m) \ddot{\mathbf{X}}_{n+1} + \alpha_m \ddot{\mathbf{X}}_n \\ t_{n+1-\alpha_f} = (1 - \alpha_f) t_{n+1} + \alpha_f t_n \end{array} \right.$$

As duas primeiras equações, acima, são aproximações de Taylor das incógnitas, com dois parâmetros  $\beta$  e  $\gamma$ , e as quatro últimas são a forma de se introduzir mais parâmetros no algoritmo. Isso é feito através de uma espécie de iteração intermediária, que é uma combinação linear das iterações atual e anterior. A princípio,  $\beta$  e  $\gamma$  dependem de  $\alpha_f$  e  $\alpha_m$ , de modo a garantir a convergência do algoritmo.

Comentaremos, brevemente, sobre o estudo de convergência do método. Primeiramente, observamos que o contexto da dissipação mencionada no algoritmo é de elementos finitos para estrutura em engenharia, que pode ser visto como superposição várias EDOs de sistemas massa-mola. No contexto específico de mecânica de sólidos, temos que as frequências relevantes para a simulação são as baixas, e as altas são comumente atribuídas a erros numéricos no algoritmo, e por isso a necessidade de sua dissipação (CHUNG; HULBERT, 1993).

Lembramos que um sistema massa-mola com amortecimento pode ser descrito por  $\ddot{u} + 2\xi\omega\dot{u} + \omega^2u = f$ . Nessa equação,  $\omega$  é a frequência de oscilação do sistema,  $\xi$  é uma dissipação e  $f$  é fonte externa de oscilação. (ALONSO; FINN, 2014, pg. 393).

Então, supondo o sistema de EDOs original como uma superposição de problemas dessa forma e, logo, de frequências de oscilação, queremos saber se as atualizações, colocadas no formato  $\mathbf{Y}_{n+1} = A\mathbf{Y}_n$ , dissipam as frequências maiores. A matriz  $A$  é chamada de matriz de amplificação do algoritmo, e  $\mathbf{Y}_n$  é o vetor de incógnitas, da forma  $\mathbf{Y}_n = [\mathbf{X}_n, \Delta t\dot{\mathbf{X}}_n, \Delta t^2\ddot{\mathbf{X}}_n]^T$ . O que importa são os autovalores de  $A$ : para frequências baixas, o raio espectral deve estar perto de 1 e cair suavemente enquanto  $\Omega = \omega\Delta t$  aumenta (CHUNG; HULBERT, 1993).

A convergência e estabilidade do método determinam  $\beta$  e  $\gamma$ , e restam  $\alpha_m$  e  $\alpha_f$  como parâmetros. A dissipação de altas frequências é controlada por  $\rho_\infty$ , que é o limite do maior autovalor de  $A$  quando  $\Omega \rightarrow \infty$ .

As constantes, conforme o algoritmo a ser implementado, são dadas por

$$\gamma = \frac{1}{2} - \alpha_f + \alpha_m$$

$$\beta = \frac{1}{4}(1 - \alpha_f + \alpha_m)^2$$

$$\alpha_m = \frac{1}{2} \left( \frac{3 - \rho_\infty}{1 + \rho_\infty} \right)$$

$$\alpha_f = \frac{1}{1 + \rho_\infty}$$

com  $\rho_\infty \in [0, 1]$  a ser escolhida. Em (JANSEN; WHITING; HULBERT, 2000), há um estudo sobre o efeito dessa escolha na dissipação numérica na solução das equações de Navier-Stokes. A análise é feita a partir de problemas de autovalor  $\dot{y} = \lambda y$  aplicados ao algoritmo, e vários casos de teste foram simulados com diferentes valores de  $\rho_\infty$ .

## 3.2 Caso FSI

No caso de interação fluido-estrutura, temos que a discretização espacial não gera um sistema de EDOs no tempo, pois nossas EDPs são não-lineares. O que pode ser feito é aplicar o método anterior para o residual das equações seja zero, e aplicar um método de Newton-Raphson para encontrar os zeros do sistema de equações não linear.

Vamos denotar as funções vetoriais de resíduos  $\mathbf{R}^{mom} = (R_{A,i}^{mom})$ ,  $\mathbf{R}^{cont} = (R_A^{cont})$ ,  $\mathbf{R}^{mesh} = (R_{A,i}^{mesh})$  respectivamente ao momento linear, continuidade e movimento da malha. O índice  $A$  é o de funções da base, e  $i \in \{1, 2, 3\}$ . O que queremos é calcular o resíduo sobre cada componente de cada função de base. Neste momento, fizemos apenas a substituição das funções teste. Para o cálculo efetivo, substituiremos posteriormente a forma das incógnitas conforme a seção 2.3. Denotando por  $\mathbf{e}_i$  os vetores da base canônica de  $\mathbb{R}^3$  e  $\mathbf{r}_M$  e  $r_C$  os residuais da equação de momento e continuidade do fluido, respectivamente,

$$\begin{aligned}
R_{A,i}^{mom} &= B_{MS}^f(\{N_A \mathbf{e}_i, 0\}, \{\mathbf{v}, p\}; \hat{\mathbf{v}}) - F_{MS}^f(\{N_A \mathbf{e}_i, 0\}) + B^s(\hat{N}_A \mathbf{e}_i, \mathbf{u}) - F^s(\hat{N}_A \mathbf{e}_i) \\
&= \int_{\Omega_t} N_A \mathbf{e}_i \cdot \rho \left( \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla \mathbf{v} - \mathbf{f} \right) d\Omega - \int_{\Gamma_t^{f,N}} N_A \mathbf{e}_i \cdot \mathbf{h}^f d\Gamma \\
&\quad + \int_{\Omega_t} p \nabla \cdot N_A \mathbf{e}_i + \frac{1}{2} \int_{\Omega_t} (\nabla N_A \mathbf{e}_i + (\nabla N_A \mathbf{e}_i)^T) : \mu (\nabla \mathbf{v} + (\nabla \mathbf{v})^T) d\Omega \\
&\quad + \sum_{e=1}^{nel} \int_{\Omega_t^e} \tau_{SUPS} ((\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla N_A \mathbf{e}_i) \cdot \mathbf{r}_M(\mathbf{v}, p) d\Omega \\
&\quad + \sum_{e=1}^{nel} \int_{\Omega_t^e} \rho \nu_{LSIC} (\nabla \cdot N_A \mathbf{e}_i) r_C(\mathbf{v}) d\Omega \\
&\quad - \sum_{e=1}^{nel} \int_{\Omega_t^e} \tau_{SUPS} N_A \mathbf{e}_i \cdot (\mathbf{r}_M(\mathbf{v}, p) \cdot \nabla \mathbf{v}) d\Omega \\
&\quad - \sum_{e=1}^{nel} \int_{\Omega_t^e} \frac{\nabla N_A \mathbf{e}_i}{\rho} : (\tau_{SUPS} \mathbf{r}_M(\mathbf{v}, p)) \otimes (\tau_{SUPS} \mathbf{r}_M(\mathbf{v}, p)) d\Omega \\
&\quad + \int_{\Omega_0^s} \hat{N}_A \mathbf{e}_i \cdot \rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} d\Omega \\
&\quad + \int_{\Omega_0^s} \nabla \hat{N}_A \mathbf{e}_i : \mathbf{F} \Sigma d\Omega \\
&\quad - \int_{\Omega_0^s} \hat{N}_A \mathbf{e}_i \cdot \rho_0 \mathbf{f}^s d\Omega \\
&\quad - \int_{\Omega_0^s} \hat{N}_A \mathbf{e}_i \cdot \mathbf{h}^s d\Omega,
\end{aligned}$$

$$\begin{aligned}
R_A^{cont} &= B_{MS}^f(\{\mathbf{0}, N_A\}, \{\mathbf{v}, p\}; \hat{\mathbf{v}}) - F_{MS}^f(\{\mathbf{0}, N_A\}) \\
&= \int_{\Omega_t^f} N_A \nabla \cdot \mathbf{v} d\Omega + \sum_{e=1}^{nel} \int_{\Omega_t^e} \tau_{SUPS} \frac{\nabla N_A}{\rho} \cdot \mathbf{r}_M(\mathbf{v}, p) d\Omega,
\end{aligned}$$

$$\begin{aligned}
R_{A,i}^{mesh} &= B^m(\widetilde{N}_A \mathbf{e}_i, \hat{\mathbf{u}}) \\
&= \frac{1}{2} \int_{\Omega_t^f} (\nabla \widetilde{N}_A \mathbf{e}_i + (\nabla \widetilde{N}_A \mathbf{e}_i)^T) : \mu^m (\nabla \hat{\mathbf{u}} + (\nabla \hat{\mathbf{u}})^T) d\Omega \\
&\quad + \int_{\Omega_t^f} (\nabla \cdot \widetilde{N}_A \mathbf{e}_i) \cdot \lambda^m (\nabla \cdot \hat{\mathbf{u}}) d\Omega.
\end{aligned}$$

Para explicar o algoritmo, vamos concatenar as funções de coeficientes do deslocamento do fluido e da estrutura e representá-las (com todos o índices  $A$ ) como um vetor  $\mathbf{U}$ . Representaremos os coeficientes da pressão  $p$  com  $\mathbf{P}$ , e o deslocamento da malha,  $\hat{\mathbf{u}}$ , por  $\mathbf{V}$ . Podemos então aplicar o método  $\alpha$ -generalizado.

Usaremos  $n$  para representar o elemento  $t_n$  de uma partição do intervalo  $[0, T]$  da simulação. Então, dados  $(\mathbf{U}_n, \dot{\mathbf{U}}_n, \ddot{\mathbf{U}}_n, \mathbf{V}_n, \dot{\mathbf{V}}_n, \ddot{\mathbf{V}}_n)$ , vamos procurar

$$(\mathbf{U}_{n+1}, \dot{\mathbf{U}}_{n+1}, \ddot{\mathbf{U}}_{n+1}, \mathbf{P}_{n+1}, \mathbf{V}_{n+1}, \dot{\mathbf{V}}_{n+1}, \ddot{\mathbf{V}}_{n+1}, \mathbf{U}_{n+\alpha_f}, \dot{\mathbf{U}}_{n+\alpha_f}, \ddot{\mathbf{U}}_{n+\alpha_m}, \mathbf{V}_{n+\alpha_f}, \dot{\mathbf{V}}_{n+\alpha_f}, \ddot{\mathbf{V}}_{n+\alpha_m})$$

tais que

$$\left\{ \begin{array}{l}
\mathbf{R}^{mom}(\mathbf{U}_{n+\alpha_f}, \dot{\mathbf{U}}_{n+\alpha_f}, \ddot{\mathbf{U}}_{n+\alpha_m}, \mathbf{P}_{n+1}, \mathbf{V}_{n+\alpha_f}, \dot{\mathbf{V}}_{n+\alpha_f}, \ddot{\mathbf{V}}_{n+\alpha_m}) = \mathbf{0} \\
\mathbf{R}^{cont}(\mathbf{U}_{n+\alpha_f}, \dot{\mathbf{U}}_{n+\alpha_f}, \ddot{\mathbf{U}}_{n+\alpha_m}, \mathbf{P}_{n+1}, \mathbf{V}_{n+\alpha_f}, \dot{\mathbf{V}}_{n+\alpha_f}, \ddot{\mathbf{V}}_{n+\alpha_m}) = \mathbf{0} \\
\mathbf{R}^{mesh}(\mathbf{U}_{n+\alpha_f}, \dot{\mathbf{U}}_{n+\alpha_f}, \ddot{\mathbf{U}}_{n+\alpha_m}, \mathbf{P}_{n+1}, \mathbf{V}_{n+\alpha_f}, \dot{\mathbf{V}}_{n+\alpha_f}, \ddot{\mathbf{V}}_{n+\alpha_m}) = \mathbf{0} \\
\mathbf{U}_{n+\alpha_f} = \mathbf{U}_n + \alpha_f(\mathbf{U}_{n+1} - \mathbf{U}_n) \\
\dot{\mathbf{U}}_{n+\alpha_f} = \dot{\mathbf{U}}_n + \alpha_f(\dot{\mathbf{U}}_{n+1} - \dot{\mathbf{U}}_n) \\
\ddot{\mathbf{U}}_{n+\alpha_m} = \ddot{\mathbf{U}}_n + \alpha_m(\ddot{\mathbf{U}}_{n+1} - \ddot{\mathbf{U}}_n) \\
\mathbf{V}_{n+\alpha_f} = \mathbf{V}_n + \alpha_f(\mathbf{V}_{n+1} - \mathbf{V}_n) \\
\dot{\mathbf{V}}_{n+\alpha_f} = \dot{\mathbf{V}}_n + \alpha_f(\dot{\mathbf{V}}_{n+1} - \dot{\mathbf{V}}_n) \\
\ddot{\mathbf{V}}_{n+\alpha_m} = \ddot{\mathbf{V}}_n + \alpha_m(\ddot{\mathbf{V}}_{n+1} - \ddot{\mathbf{V}}_n) \\
\dot{\mathbf{U}}_{n+1} = \dot{\mathbf{U}}_n + \Delta t((1 - \gamma)\dot{\mathbf{U}}_n + \gamma\dot{\mathbf{U}}_{n+1}) \\
\mathbf{U}_{n+1} = \mathbf{U}_m + \Delta t\dot{\mathbf{U}}_n + \frac{\Delta t^2}{2}((1 - 2\beta)\ddot{\mathbf{U}}_n + 2\beta\ddot{\mathbf{U}}_{n+1}) \\
\dot{\mathbf{V}}_{n+1} = \dot{\mathbf{V}}_n + \Delta t((1 - \gamma)\dot{\mathbf{V}}_n + \gamma\dot{\mathbf{V}}_{n+1}) \\
\mathbf{V}_{n+1} = \mathbf{V}_m + \Delta t\dot{\mathbf{V}}_n + \frac{\Delta t^2}{2}((1 - 2\beta)\ddot{\mathbf{V}}_n + 2\beta\ddot{\mathbf{V}}_{n+1})
\end{array} \right.$$

Na prática, temos um algoritmo de dois estágios.

### Estágio preditor

Calculamos o que será o ponto inicial do algoritmo de Newton-Raphson

$$\begin{aligned}
\dot{\mathbf{U}}_{n+1,(0)} &= \dot{\mathbf{U}}_n \\
\ddot{\mathbf{U}}_{n+1,(0)} &= \frac{\gamma-1}{\gamma}\ddot{\mathbf{U}}_n \\
\mathbf{U}_{n+1,(0)} &= \mathbf{U}_m + \Delta t\dot{\mathbf{U}}_n + \frac{\Delta t^2}{2}((1 - 2\beta)\ddot{\mathbf{U}}_n + 2\beta\ddot{\mathbf{U}}_{n+1,(0)}) \\
\mathbf{P}_{n+1,(0)} &= \mathbf{P}_n \\
\dot{\mathbf{V}}_{n+1,(0)} &= \dot{\mathbf{V}}_n \\
\ddot{\mathbf{V}}_{n+1,(0)} &= \frac{\gamma-1}{\gamma}\ddot{\mathbf{V}}_n \\
\mathbf{V}_{n+1,(0)} &= \mathbf{V}_n + \Delta t\dot{\mathbf{V}}_n + \frac{\Delta t^2}{2}((1 - 2\beta)\ddot{\mathbf{V}}_n + 2\beta\ddot{\mathbf{V}}_{n+1,(0)})
\end{aligned}$$

### Estágio multi-corretor

Na iteração  $l$ ,  $l \in \{1, 2, \dots, l_{max}\}$

*Passo 1*

$$\begin{aligned}\mathbf{U}_{n+\alpha_f, (l)} &= \mathbf{U}_n + \alpha_f(\mathbf{U}_{n+1, (l-1)} - \mathbf{U}_n) \\ \dot{\mathbf{U}}_{n+\alpha_f, (l)} &= \dot{\mathbf{U}}_n + \alpha_f(\dot{\mathbf{U}}_{n+1, (l-1)} - \dot{\mathbf{U}}_n) \\ \ddot{\mathbf{U}}_{n+\alpha_m, (l)} &= \ddot{\mathbf{U}}_n + \alpha_m(\ddot{\mathbf{U}}_{n+1, (l-1)} - \ddot{\mathbf{U}}_n) \\ \mathbf{V}_{n+\alpha_f, (l)} &= \mathbf{V}_n + \alpha_f(\mathbf{V}_{n+1, (l-1)} - \mathbf{V}_n) \\ \dot{\mathbf{V}}_{n+\alpha_f, (l)} &= \dot{\mathbf{V}}_n + \alpha_f(\dot{\mathbf{V}}_{n+1, (l-1)} - \dot{\mathbf{V}}_n) \\ \ddot{\mathbf{V}}_{n+\alpha_m, (l)} &= \ddot{\mathbf{V}}_n + \alpha_m(\ddot{\mathbf{V}}_{n+1, (l-1)} - \ddot{\mathbf{V}}_n) \\ \mathbf{P}_{n+1, (l)} &= \mathbf{P}_{n+1, (l-1)}\end{aligned}$$

*Passo 2.* Procuramos os acréscimos, denotados com  $\Delta$ , no sistema linear abaixo

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{R}^{mom}}{\partial \ddot{\mathbf{U}}_{n+1}} \Delta \ddot{\mathbf{U}}_{n+1, (l)} + \frac{\partial \mathbf{R}^{mom}}{\partial \mathbf{P}_{n+1}} \Delta \mathbf{P}_{n+1, (l)} + \frac{\partial \mathbf{R}^{mom}}{\partial \ddot{\mathbf{V}}_{n+1}} \Delta \ddot{\mathbf{V}}_{n+1, (l)} = -\mathbf{R}_{(l)}^{mom} \\ \frac{\partial \mathbf{R}^{cont}}{\partial \ddot{\mathbf{U}}_{n+1}} \Delta \ddot{\mathbf{U}}_{n+1, (l)} + \frac{\partial \mathbf{R}^{cont}}{\partial \mathbf{P}_{n+1}} \Delta \mathbf{P}_{n+1, (l)} + \frac{\partial \mathbf{R}^{cont}}{\partial \ddot{\mathbf{V}}_{n+1}} \Delta \ddot{\mathbf{V}}_{n+1, (l)} = -\mathbf{R}_{(l)}^{cont} \\ \frac{\partial \mathbf{R}^{mesh}}{\partial \ddot{\mathbf{U}}_{n+1}} \Delta \ddot{\mathbf{U}}_{n+1, (l)} + \frac{\partial \mathbf{R}^{mesh}}{\partial \mathbf{P}_{n+1}} \Delta \mathbf{P}_{n+1, (l)} + \frac{\partial \mathbf{R}^{mesh}}{\partial \ddot{\mathbf{V}}_{n+1}} \Delta \ddot{\mathbf{V}}_{n+1, (l)} = -\mathbf{R}_{(l)}^{mesh} \end{array} \right.$$

*Passo 3*

$$\begin{aligned}\ddot{\mathbf{U}}_{n+1, (l)} &= \ddot{\mathbf{U}}_{n+1, (l-1)} + \Delta \ddot{\mathbf{U}}_{n+1, (l)} \\ \dot{\mathbf{U}}_{n+1, (l)} &= \dot{\mathbf{U}}_{n+1, (l-1)} + \gamma \Delta t \Delta \dot{\mathbf{U}}_{n+1, (l)} \\ \mathbf{U}_{n+1, (l)} &= \mathbf{U}_{n+1, (l-1)} + \beta (\Delta t)^2 \Delta \ddot{\mathbf{U}}_{n+1, (l)} \\ \mathbf{P}_{n+1, (l)} &= \mathbf{P}_{n+1, (l-1)} + \Delta \mathbf{P}_{n+1, (l)} \\ \ddot{\mathbf{V}}_{n+1, (l)} &= \ddot{\mathbf{V}}_{n+1, (l-1)} + \Delta \ddot{\mathbf{V}}_{n+1, (l)} \\ \dot{\mathbf{V}}_{n+1, (l)} &= \dot{\mathbf{V}}_{n+1, (l-1)} + \gamma \Delta t \Delta \dot{\mathbf{V}}_{n+1, (l)} \\ \mathbf{V}_{n+1, (l)} &= \mathbf{V}_{n+1, (l-1)} + \beta (\Delta t)^2 \Delta \ddot{\mathbf{V}}_{n+1, (l)}\end{aligned}$$

*Passo 4.* Verificamos convergência através de alguma norma.

Decisões práticas a serem tomadas incluem o número máximo de iterações  $l_{max}$  do estágio multi-corretor; qual algoritmo de resolução de sistemas lineares usar, como por exemplo GMRES; o critério de convergência do estágio multi-corretor. Por exemplo, podemos verificar se os elementos com  $\Delta$  encontrados são menores que um  $\epsilon$  dado.

## 3.3 Matrizes tangentes

Seguimos o livro (BAZILEVS; TAKIZAWA; TEZDUYAR, 2013, pg. 97) para o cálculo das *matrizes tangentes*

$$\frac{\partial \mathbf{R}^{mom}}{\partial \ddot{\mathbf{U}}_{n+1}}, \frac{\partial \mathbf{R}^{mom}}{\partial \mathbf{P}_{n+1}}, \frac{\partial \mathbf{R}^{cont}}{\partial \ddot{\mathbf{U}}_{n+1}} \text{ e } \frac{\partial \mathbf{R}^{cont}}{\partial \mathbf{P}_{n+1}}$$

do *Passo 2* acima. Lembramos que esta é uma linearização de

$$\begin{cases} \mathbf{R}^{mom}(\mathbf{U}_{n+\alpha_f}, \dot{\mathbf{U}}_{n+\alpha_f}, \ddot{\mathbf{U}}_{n+\alpha_m}, \mathbf{P}_{n+1}, \mathbf{V}_{n+\alpha_f}, \dot{\mathbf{V}}_{n+\alpha_f}, \ddot{\mathbf{V}}_{n+\alpha_m}) = \mathbf{0} \\ \mathbf{R}^{cont}(\mathbf{U}_{n+\alpha_f}, \dot{\mathbf{U}}_{n+\alpha_f}, \ddot{\mathbf{U}}_{n+\alpha_m}, \mathbf{P}_{n+1}, \mathbf{V}_{n+\alpha_f}, \dot{\mathbf{V}}_{n+\alpha_f}, \ddot{\mathbf{V}}_{n+\alpha_m}) = \mathbf{0} \\ \mathbf{R}^{mesh}(\mathbf{U}_{n+\alpha_f}, \dot{\mathbf{U}}_{n+\alpha_f}, \ddot{\mathbf{U}}_{n+\alpha_m}, \mathbf{P}_{n+1}, \mathbf{V}_{n+\alpha_f}, \dot{\mathbf{V}}_{n+\alpha_f}, \ddot{\mathbf{V}}_{n+\alpha_m}) = \mathbf{0} \end{cases}$$

Para explicar o processo, calcularemos, como exemplo, a contribuição de

$$\int_{\Omega_t^f} N_A \mathbf{e}_i \cdot \rho \frac{\partial \mathbf{v}}{\partial t} d\Omega$$

para  $\frac{\partial \mathbf{R}^{mom}}{\partial \ddot{\mathbf{U}}_{n+1}}$ .

i) Substituímos  $\mathbf{v} = \sum_B \ddot{\mathbf{U}} N_B$  e lembramos que  $\mathbf{R}^{mom}$  é calculado com  $\ddot{\mathbf{U}}$  substituído por  $\ddot{\mathbf{U}}_{n+\alpha_m}$ . Mudamos também o domínio de integração para  $\Omega_{n+\alpha_m}^f$ :

$$\int_{\Omega_{n+\alpha_m}^f} N_A \mathbf{e}_i \cdot \rho N_B \ddot{\mathbf{U}}_{n+\alpha_m} d\Omega$$

ii) Substituímos o passo do algoritmo  $\ddot{\mathbf{U}}_{n+\alpha_m} = \ddot{\mathbf{U}}_n + \alpha_m(\ddot{\mathbf{U}}_{n+1} - \ddot{\mathbf{U}}_n)$ :

$$\int_{\Omega_{n+\alpha_m}^f} N_A \mathbf{e}_i \cdot \rho N_B [\ddot{\mathbf{U}}_n + \alpha_m(\ddot{\mathbf{U}}_{n+1} - \ddot{\mathbf{U}}_n)] d\Omega$$

iii) Derivamos a expressão com relação a  $\ddot{\mathbf{U}}_{n+1}$ , lembrando que  $\frac{\partial \mathbf{R}^{mom}}{\partial \ddot{\mathbf{U}}_{n+1}} = \left( \frac{\partial R_i}{\partial \ddot{U}_{n+1,j}} \right)_{ij}$ .

O elemento  $(i, j)$  dessa *matriz tangente* será dado por

$$\int_{\Omega_{n+\alpha_m}^f} N_A \rho N_B \alpha_m d\Omega \delta_{ij}.$$

Mas, como os outros componentes terão integração em  $\Omega_{n+\alpha_f}$ , padronizamos os domínios para este último e calculamos os pontos do domínio, utilizando as funções de deslocamento da iteração atual,  $\mathbf{y} = \mathbf{I} - \Phi$ , e aplicando uma nova função de deslocamento

$$\bar{\mathbf{u}}(\mathbf{x}, t_{n+\alpha_t}) = \mathbf{x} + \mathbf{y}(\mathbf{x}, t_n) + \alpha_f(\mathbf{y}(\mathbf{x}, t_{n+1}) - \mathbf{y}(\mathbf{x}, t_n)).$$

Para as derivadas do sólido, utilizamos uma linearização das equações (BAZILEVS; TAKIZAWA; TEZDUYAR, 2013, pg. 23),

$$\int_{\Omega_0^s} \mathbf{w} \cdot \rho_0 \frac{\partial^2 \mathbf{u}}{\partial t^2} d\Omega + \int_{\Omega_0^s} \mathbf{F}^T \nabla w : \mathcal{C} \mathbf{F}^T \nabla \mathbf{u} - \int_{\Omega_0^s} \mathbf{w} \cdot \rho_0 \mathbf{f}^s d\Omega - \int_{\Gamma_0^{s,N}} \mathbf{w} \cdot \hat{\mathbf{h}}^s d\Gamma = 0,$$

onde  $\mathcal{C}$  é o módulo de elasticidade descrito na *Seção 1.2*.

Então, denotando por  $K_{AB}^{ij}$  o elemento da matriz que vem da coordenada  $i$  da função base da função teste,  $A$ , e da coordenada  $j$  da função base da incógnita,  $B$  (ver seção 2.5 sobre matrizes de conectividade), temos as relações

$$\begin{aligned}
\frac{\partial \mathbf{R}^{mom}}{\partial \ddot{\mathbf{U}}_{n+1}} &= [K_{AB}^{ij}] \\
K_{AB}^{ij} &= \alpha_m \int_{\Omega_{n+\alpha_f}} N_A \rho N_B d\Omega \delta_{ij} \\
&+ \alpha_m \int_{\Omega_{n+\alpha_f}} \tau_{SUPS} (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla N_A \rho N_B d\Omega \delta_{ij} \\
&+ \alpha_f \gamma \Delta t_n \int_{\Omega_{n+\alpha_f}} N_A \rho (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla N_B d\Omega \delta_{ij} \\
&+ \alpha_f \gamma \Delta t_n \int_{\Omega_{n+\alpha_f}} \nabla N_A \cdot \mu^f \nabla N_B d\Omega \delta_{ij} \\
&+ \alpha_f \gamma \Delta t_n \int_{\Omega_{n+\alpha_f}} \nabla N_A \cdot \mathbf{e}_j \mu^f \nabla N_B \cdot \mathbf{e}_i d\Omega \\
&+ \alpha_f \gamma \Delta t_n \int_{\Omega_{n+\alpha_f}} \tau_{SUPS} (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla N_A \rho (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla N_B d\Omega \delta_{ij} \\
&+ \alpha_f \gamma \Delta t_n \int_{\Omega_{n+\alpha_f}} \rho \nu_{LSIC} \nabla N_A \cdot \mathbf{e}_i \nabla N_B \cdot \mathbf{e}_j d\Omega \\
&+ \alpha_m \int_{\Omega_{n+\alpha_f}} N_A \rho_0 N_B d\Omega \\
&+ \alpha_f \gamma \Delta t_n \int_{\Omega_{n+\alpha_f}} (\mathbf{F}^T \nabla N_A : \mathcal{C} \mathbf{F}^T \nabla N_B + \nabla N_A : \nabla N_B \Sigma) d\Omega,
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathbf{R}^{mom}}{\partial \mathbf{P}_{n+1}} &= [G_{AB}^i] \\
G_{AB}^i &= - \int \nabla N_A \cdot \mathbf{e}_i N_B d\Omega \\
&+ \int \tau_{SUPS} (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla N_A \nabla N_B \cdot \mathbf{e}_i d\Omega,
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathbf{R}^{cont}}{\partial \ddot{\mathbf{U}}_{n+1}} &= [D_{AB}^j] \\
D_{AB}^j &= \alpha_f \gamma \Delta t_n \int N_A \nabla N_B \cdot \mathbf{e}_j d\Omega \\
&+ \alpha_f \gamma \Delta t_n \int \tau_{SUPS} \nabla N_A \cdot \mathbf{e}_j (\mathbf{v} - \hat{\mathbf{v}}) \cdot \nabla N_B d\Omega \\
&+ \alpha_m \int \tau_{SUPS} \nabla N_A \cdot \mathbf{e}_j N_B d\Omega,
\end{aligned}$$

$$\begin{aligned}
\frac{\partial \mathbf{R}^{cont}}{\partial \mathbf{P}_{n+1}} &= [L_{AB}] \\
L_{AB} &= \int \frac{\tau_{SUPS}}{\rho} \nabla N_A \cdot \nabla N_B d\Omega.
\end{aligned}$$

As relações para  $\frac{\partial \mathbf{R}^{mesh}}{\partial \ddot{\mathbf{U}}_{n+1}}$  e  $\frac{\partial \mathbf{R}^{mesh}}{\partial \mathbf{P}_{n+1}}$  dependem da escolha de abordagem numérica das condições de bordo das equações diferenciais, que não foi tratada nesse trabalho.

### 3.4 Shape derivatives

As derivadas em relação ao deslocamento da malha são chamadas de *shape derivatives* (pois as funções são funções *shape*). Seguimos (BAZILEVS et al., 2008) de perto, apresentando, inicialmente, um exemplo de como as derivadas podem ser calculadas, para em seguida listar todos os outros componentes necessários para o cálculo das matrizes.

A ideia importante por trás dos cálculos é uma mudança de variável das integrais, conforme a *Figura 6* da *Seção 2.4*. Por exemplo, para calcularmos

$$\frac{\partial \sum_{e=1}^{n_{el}} \int_{\Omega_e} N_A \rho^f \frac{\partial v_i}{\partial t} d\Omega_e}{\partial \ddot{V}_{B,j}},$$

podemos comutar o operador de derivação com o somatório,

$$\frac{\sum_{e=1}^{n_{el}} \partial \int_{\Omega_e} N_A \rho^f \frac{\partial v_i}{\partial t} d\Omega_e}{\partial \ddot{V}_{B,j}},$$

e focar em um *patch* do domínio em particular.

O problema agora é comutar a derivada com a integral. Precisaremos efetuar uma mudança de variáveis na integral, pois o domínio depende de  $\mathbf{V}$  e, portanto, de  $\ddot{\mathbf{V}}$ . Por isso, lembramos que podemos tomar um domínio canônico  $\mathbf{I}$  para o *patch* e relacionar o domínio atual através de composições de funções.

Usaremos então um resultado de mecânica de meios contínuos, sobre cálculo de operadores de derivada.

**Lema.** Seja  $\mathcal{D}$  um operador diferencial, como por exemplo a derivada parcial com relação a uma coordenada. Então,  $\mathcal{D}J = J\mathbf{F}^{-T}\mathcal{D}(\mathbf{F})$ .

**Demonstração.** vamos utilizar o resultado (CIARLET, 1988, pg. 20) que, para uma matriz  $\mathbf{A}$ ,

$$\frac{\partial \det \mathbf{A}}{\partial \mathbf{A}} = (\det \mathbf{A}) \mathbf{A}^{-T}.$$

Então, pela Regra da Cadeia,

$$\mathcal{D}J = \frac{\partial J}{\partial \mathbf{F}} \mathcal{D}\mathbf{F}.$$

Logo, como  $J = \det \mathbf{F}$  por definição, obtemos o resultado. ///

A partir de agora, usaremos a notação de índice conhecida como “notação de Einstein”, em que índices repetidos em uma expressão indica um somatório desse índice variando de 1 até 3. Por exemplo, o resultado anterior fica da forma

$$\mathcal{D}J = J \frac{\partial \xi_j}{\partial x_i} \mathcal{D} \left( \frac{\partial x_i}{\partial \xi_j} \right).$$

Retornando ao problema acima, fazemos a mudança de variáveis  $\mathbf{x} \rightarrow \boldsymbol{\xi}$ , do conjunto  $\Omega_e$  para o  $\mathbf{I}$  (novamente, ver *Figura 6* na *Seção 2.4*). Usando o Lema acima, a integral se torna

$$\int_{\mathbf{I}} N_A \rho^f \frac{\partial v_i}{\partial t} \frac{\partial J}{\partial \ddot{V}_{B,j}} d\mathbf{I} = \int_{\mathbf{I}} N_A \rho^f \frac{\partial v_i}{\partial t} \frac{\partial \left( \frac{\partial x_k}{\partial \xi_l} \right)}{\partial \ddot{V}_{B,j}} \frac{\partial \xi_l}{\partial x_k} J d\mathbf{I}$$

Mas, pela definição de deformação,  $x_k = \hat{u}_k + y_k$ , onde  $y_k$  o deslocamento sofrido por  $\hat{u}_k$ , de modo que

$$\frac{\partial x_k}{\partial \xi_l} = \frac{\partial \hat{u}_k}{\partial \xi_l} + \frac{\partial y_k}{\partial \xi_l},$$

logo

$$\frac{\partial \left( \frac{\partial x_k}{\partial \xi_l} \right)}{\partial \ddot{V}_{B,j}} = \frac{\partial \left( \frac{\partial \hat{u}_k}{\partial \xi_l} \right)}{\partial \ddot{V}_{B,j}}.$$

Segue a lista completa dos componentes e integrais para calculá-los. Lembramos que  $\Omega_e$  representa o domínio de um *patch*.

$$\frac{\partial \sum_{e=1}^{n_{el}} \int_{\Omega_e} N_A \rho^f \frac{\partial v_i}{\partial t} d\Omega_e}{\partial \ddot{V}_{B,j}} = \alpha_f \beta \Delta \int_{\Omega_e} N_A \rho^f \frac{\partial v_i}{\partial t} \frac{\partial N_B}{\partial x_j} d\Omega_e.$$

$$\begin{aligned} \frac{\partial \sum_{e=1}^{n_{el}} \int_{\Omega_e} N_A \rho^f (v_k - \hat{v}_k) \frac{\partial v_i}{\partial x_k} d\Omega_e}{\partial \ddot{V}_{B,j}} &= \alpha_f \beta \Delta t^2 \int_{\Omega_e} \frac{\partial N_A}{\partial x_i} \frac{\tau_M}{\rho^f} \frac{\partial p}{\partial x_i} \frac{\partial N_B}{\partial x_j} d\Omega_e \\ &\quad - \alpha_f \beta \Delta t^2 \int_{\Omega_e} \frac{\partial N_A}{\partial x_i} \frac{\tau_M}{\rho^f} \frac{\partial p}{\partial x_j} \frac{\partial N_B}{\partial x_i} d\Omega_e \\ &\quad - \alpha_f \beta \Delta t^2 \int_{\Omega_e} \frac{\partial N_A}{\partial x_j} \frac{\tau_M}{\rho^f} \frac{\partial p}{\partial x_i} \frac{\partial N_B}{\partial x_i} d\Omega_e \end{aligned}$$

$$\begin{aligned} \frac{\partial \sum_{e=1}^{n_{el}} \int_{\Omega_e} \frac{\partial N_A}{\partial x_i} \frac{\tau_M}{\rho^f} \frac{\partial p}{\partial x_i} d\Omega_e}{\partial \ddot{V}_{B,j}} &= \alpha_f \beta \Delta t^2 \int_{\Omega_e} N_A \rho^f (v_k - \hat{v}_k) \frac{\partial v_i}{\partial x_k} \frac{\partial N_B}{\partial x_j} d\Omega_e \\ &\quad - \alpha_f \beta \Delta t^2 \int_{\Omega_e} N_A \rho^f (v_k - \hat{v}_k) \frac{\partial v_i}{\partial x_j} \frac{\partial N_B}{\partial x_k} d\Omega_e \\ &\quad - \alpha_f \gamma \Delta t \int_{\Omega_e} N_A \rho^f \frac{\partial v_i}{\partial x_j} N_B d\Omega_e \end{aligned}$$

$$\frac{\partial \sum_{e=1}^{n_{el}} \int_{\Omega_e} \frac{\partial N_A}{\partial x_i} p d\Omega_e}{\partial \ddot{V}_{B,j}} = -\alpha_f \beta \Delta t^2 \int_{\Omega_e} \frac{\partial N_A}{\partial x_i} p \frac{\partial N_B}{\partial x_j} - \frac{\partial N_A}{\partial x_j} p \frac{\partial N_B}{\partial x_i} d\Omega_e$$

$$\begin{aligned} \frac{\partial \sum_{e=1}^{nel} \int_{\Omega_e} \frac{\partial N_A}{\partial x_k} \mu^f \left( \frac{\partial v_i}{\partial x_k} + \frac{\partial v_k}{\partial x_i} \right) d\Omega_e}{\partial \ddot{V}_{B,j}} &= \alpha_f \beta \Delta t^2 \int_{\Omega_e} \frac{\partial N_A}{\partial x_k} \mu^f \left( \frac{\partial v_i}{\partial x_k} + \frac{\partial v_k}{\partial x_i} \right) \frac{\partial N_B}{\partial x_j} \\ &\quad - \frac{\partial N_A}{\partial x_j} \mu^f \left( \frac{\partial v_i}{\partial x_k} + \frac{\partial v_k}{\partial x_i} \right) \frac{\partial N_B}{\partial x_k} \\ &\quad - \frac{\partial N_A}{\partial x_k} \mu^f \frac{\partial v_i}{\partial x_j} \frac{\partial N_B}{\partial x_k} \\ &\quad - \frac{\partial N_A}{\partial x_k} \mu^f \frac{\partial v_k}{\partial x_j} \frac{\partial N_B}{\partial x_i} d\Omega_e \end{aligned}$$

$$\frac{\partial \sum_{e=1}^{nel} \int_{\Omega_e} N_A \rho^f f_i^f d\Omega_e}{\partial \ddot{V}_{B,j}} = \sum_{e=1}^{nel} \int_{\Omega_e} N_A \rho^f \frac{\partial f_i^f}{\partial \ddot{V}_{B,j}} + \alpha_f \beta \Delta t^2 N_A \rho^f f_i^f \frac{\partial N_B}{\partial x_j} d\Omega_e$$

$$\frac{\partial \sum_{e=1}^{nel} \int_{\Omega_e} N_A \frac{\partial v_i}{\partial x_i} d\Omega_e}{\partial \ddot{V}_{B,j}} = \sum_{e=1}^{nel} \alpha_f \beta \Delta t^2 \int_{\Omega_e} N_A \left( \frac{\partial v_i}{\partial x_i} \frac{\partial N_B}{\partial x_j} - \frac{\partial v_i}{\partial x_j} \frac{\partial N_B}{\partial x_i} \right) d\Omega_e$$

$$\begin{aligned} \frac{\partial \sum_{e=1}^{nel} \int_{\Omega_e} \frac{\partial N_A}{\partial x_i} \tau_C \frac{\partial v_k}{\partial x_k} d\Omega_e}{\partial \ddot{V}_{B,j}} &= \left( \sum_{e=1}^{nel} \alpha_f \beta \Delta t^2 \int_{\Omega_e} \frac{\partial N_A}{\partial x_i} \tau_C \frac{\partial v_k}{\partial x_k} \frac{\partial N_B}{\partial x_j} \right. \\ &\quad \left. - \frac{\partial N_A}{\partial x_j} \tau_C \frac{\partial v_k}{\partial x_k} \frac{\partial N_B}{\partial x_i} - \frac{\partial N_A}{\partial x_i} \tau_C \frac{\partial v_k}{\partial x_j} \frac{\partial N_B}{\partial x_k} \right) d\Omega_e \end{aligned}$$

$$\begin{aligned} \frac{\partial \sum \int_{\Omega_e} (v_k - \hat{v}_k) \frac{\partial N_A}{\partial x_k} \tau_M (v_l - \hat{v}_k) \frac{\partial v_i}{\partial x_l} d\Omega}{\partial \ddot{V}_{B,j}} &= - \sum_{e=1}^{nel} \alpha_f \gamma \Delta t \int_{\Omega_e} \frac{\partial N_A}{\partial x_j} \tau_M (v_k - \hat{v}_k) \frac{\partial v_i}{\partial x_k} N_B \\ &\quad + \frac{\partial N_A}{\partial x_k} \tau_M (v_k - \hat{v}_k) \frac{\partial v_i}{\partial x_j} N_B d\Omega_e \\ &\quad + \sum_{e=1}^{nel} \alpha_f \beta \Delta t^2 \int_{\Omega_e} \left[ (v_l - \hat{v}_l) \frac{\partial N_A}{\partial x_l} \tau_M (v_k - \hat{v}_k) \frac{\partial v_i}{\partial x_k} \frac{\partial N_B}{\partial x_j} \right. \\ &\quad - \frac{\partial N_A}{\partial x_j} \tau_M (v_l - \hat{v}_l) \frac{\partial v_i}{\partial x_l} (v_k - \hat{v}_k) \frac{\partial N_B}{\partial x_k} \\ &\quad \left. - (v_l - \hat{v}_l) \frac{\partial N_A}{\partial x_l} \tau_M \frac{\partial v_i}{\partial x_j} (v_k - \hat{v}_k) \frac{\partial N_B}{\partial x_k} \right] d\Omega_e \end{aligned}$$

Reiterando que essas integrais são calculadas a cada passo do método de Newton-Raphson, e para cada *patch* da geometria, reafirmamos que esta é a parte mais custosa do algoritmo.

## 4 Conclusão

A intenção desse trabalho foi o de apresentar um modelo para problemas do tipo interação fluido-estrutura e tentar encontrar uma solução numérica através de um algoritmo que utilizasse análise isogeométrica e o método  $\alpha$ -generalizado.

Vimos que problemas desse tipo têm como principal dificuldade a necessidade de acoplamento de dois sistemas físicos, e que tentativas de solucionar esse problema podem adicionar uma nova equação ao sistema de EDPs. Além disso, as diferentes formulações possíveis de um problema, em mecânica do contínuo, que utilizam funções com domínios de referência ou atual, levam à necessidade de cuidados adicionais com o formato das variáveis durante a discretização das equações.

Notamos, também, que a abordagem isogeométrica possui várias semelhanças com a abordagem de elementos finitos, seja no suporte local das funções base, ou na forma como utilizamos as formulações fracas para criar o algoritmo. As principais dificuldades dessa abordagem são a geração das geometrias e rotinas diferenciadas para o cálculo do valor das funções e suas derivadas. Outrossim, espera-se que, ao evitar a necessidade de triangulação do espaço como procedimento em separado para a simulação, e a possibilidade de representar o movimento dos sólidos de forma mais fiel, essas dificuldades sejam em muito superadas pelas vantagens dessa abordagem.

Quanto à integração no tempo, observamos que a principal dificuldade é a linearização das equações. De fato, o maior trabalho computacional estará no método de Newton-Raphson: precisamos calcular diversas integrais das fórmulas de linearização e resolver sistemas lineares em cada iteração.

Como trabalhos futuros, precisamos lidar com as condições de bordo de Dirichlet, que não são especificadas em (BAZILEVS et al, 2008), e finalizar o cálculo das matrizes tangentes a elas associadas. Há pelo menos duas abordagens possíveis. Na primeira, utilizamos o método conhecido como *Bubnov-Galerkin* ou levantamento de bordo, em que os elementos ou *patches* na borda de Dirichlet são corrigidos para aproximar a condição  $\mathbf{g}$  e essa aproximação deve ser levada em conta na formulação fraca. Em (BAZILEVS; TAKIZAWA; TEZDUYAR, 2013), é apresentado um método mais sofisticado, em que as condições de Dirichlet são impostas de uma maneira “fraca”. Por fim, esperamos implementar o algoritmo e testá-lo com um exemplo real.



## 5 Referências

ADAMS, R. A.; FOURNIER, J. J. F. **Sobolev Spaces**. Amsterdam: Elsevier, 2003.

ALONSO, M.; FINN, E. J. **Física**: um curso universitário, volume 1 – mecânica. 2 edição. São Paulo: Blucher, 2014.

BATCHELOR, G. K. **An introduction to fluid dynamics**. Cambridge: Cambridge University Press, 2000.

BAZILEVS, T.; CALO, V. M.; COTTRELL, J. A.; EVANS, J. A.; HUGHES, T. J. R. Isogeometric analysis using T-splines. **Computer methods in applied mechanics and engineering**. Amsterdam, v. 199, p. 229-263.

BAZILEVS, Y.; CALO, V. M.; HUGHES, T. J. R.; ZHANG, Y. Isogeometric fluid-structure interaction: theory, algorithms, and computations. **Computational mechanics**, Heidelberg, v.43, p. 3-37, 2008.

BAZILEVS, Y.; TAKIZAWA, K.; TEZDUYAR, T. E. **Computational fluid-structure interaction**: methods and applications. Chichester: John Wiley & Sons, 2013.

BREZIS, H. **Functional Analysis, Sobolev Spaces and Partial Differential Equations**. New York: Springer, 2011.

CAUSIN, P.; GERBEAU, J.-F.; NOBILE, F. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. **Computer methods in applied mechanics and engineering**, v.194, n. 42, p. 4506-4527, 2005.

CHUNG, J.; HULBERT, G. M. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized- $\alpha$  method. **Journal of applied mechanics**, v. 60, p. 371-375, 1993.

CIARLET, P. G. **Mathematical Elasticity Volume 1: Three Dimensional Elasticity**. Amsterdam: Elsevier, 1988.

COTTET, G.-H.; MAITRE, E. A semi-implicit level set method for multiphase flows and fluid-structure interaction problems. **Journal of computational physics**, Amsterdam, v. 314, p. 80-92, 2016.

COTTRELL, J. A.; HUGHES, T. J. R.; BAZILEVS, Y. **Isogeometric Analysis**: toward

integration of CAD and FEA. Chichester: John Wiley & Sons, 2009.

FARIN, G. **Curves and surfaces for CAGD: a practical guide**. 5 edição. São Francisco: Morgan Kaufmann Publishers, 2003.

GERE, J. M. **Mechanics of Materials**. 6 edição. Thomson Learning, 2004.

HUGHES, T. J. R. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origin of stabilized methods. **Computer methods in applied mechanics and engineering**, Amsterdam, v. 127, p. 387-401, 1995.

HUGHES, T. J. R.; COTTRELL, J. A.; BAZILEVS, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. **Computer methods in applied mechanics and engineering**, Amsterdam, v.194, p. 4135-4195, 2005.

JANSEN, K. E.; WHITING, C.H.; HULBERT, G.M. A generalized- $\alpha$  method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. **Computer methods in applied mechanics and engineering**, Amsterdam, v. 190, p. 305-319, 1999.

LI, Z.; FAVIER, J. A non-staggered coupling of finite element and lattice Boltzmann methods via an immersed boundary scheme for fluid-structure interaction. **Computers and fluids**, Amsterdam, v. 143, p. 90-102, 2017.

MARSDEN, J. E.; HUGHES, T. J. R. **Mathematical foundations of elasticity**. New Jersey: Prentice-Hall, 1983.

PIEGL, L.; TILLER, W. **The NURBS book**. 2 edição. Heidelberg: Springer-Verlag, 1997.

QUARTERONI, A. **Modellistica numerica per problemi differenziali**. 6 edição. Milão: Springer-Verlag, 2016.

QUARTERONI, A.; SACCO, R.; SALERI, F.; GERVASIO, P. **Matematica numerica**. 4 edição. Milão: Springer-Verlag, 2014.

SIMO, J. C.; HUGHES, T. J. R. **Computational Inelasticity**. Heidelberg: Springer-Verlag, 1998.

TRUESDELL, C.; NOLL, W.; ANTMAN, S (Ed). **The non-linear field theories of mechanics**. 3 edição. Heidelberg: Springer-Verlag, 2004.