

ANÁLISE ISOGEOMÉTRICA APLICADA À DINÂMICA DOS FLUIDOS COMPUTACIONAL

A Análise Isogeométrica (IGA) é uma técnica numérica introduzida por ??) para obtenção de soluções aproximadas de equações diferenciais. O método pode ser entendido como uma generalização do método dos elementos finitos clássico a partir do uso de funções base especiais.

Na Análise Isogeométrica, as funções base escolhidas na discretização da geometria do problema e de suas variáveis são aquelas utilizadas nos sistemas CAD, sendo as funções do tipo NURBS [Definir as siglas CAD e NURBS](#) as mais aplicadas (ver, por exemplo, ??)). O grande impulso para o desenvolvimento da técnica foi proporcionar a integração entre a engenharia de projeto, com modelos baseados em CAD, e as simulações numéricas, com modelos principalmente baseados no MEF, de forma que ambas trabalhem com somente um modelo geométrico.

A IGA apresenta vantagens significativas, uma vez que permite a representação exata de diversas geometrias comuns, como cônicas, quádricas, círculos, cilindros, esferas e elipsóides, além de dispor de algoritmos eficientes e estáveis para a geração de objetos NURBS. As funções NURBS, em particular, possuem propriedades matemáticas que as tornam adequadas para aplicações numéricas, destacando-se a elevada suavidade, a alta capacidade de aproximação e a possibilidade de refinamento local por meio da inserção de *knots*, os quais correspondem às coordenadas do espaço paramétrico nas quais as funções são definidas.

Este capítulo apresenta uma breve introdução à Análise Isogeométrica (IGA) e sua relação com o Método dos Elementos Finitos clássico. Inicialmente, são descritas as funções

base B-spline, suas principais características e a construção de geometrias associadas. Em seguida, introduzem-se as funções NURBS, destacando sua formulação e aplicações geométricas. A análise isogeométrica é então introduzida no contexto da dinâmica dos fluidos computacional, enfatizando suas particularidades quanto à discretização das variáveis e da geometria, aos métodos de integração numérica e à determinação dos parâmetros de estabilização. Por fim, são apresentados exemplos numéricos que validam a formulação proposta. As principais referências bibliográficas que fundamentam esta construção são ??) e ??).

1.1 Representação Geométrica com B-splines e NURBS

No contexto do MEF isoparamétrico, a formulação é construída a partir da definição de uma malha e de seus elementos, os quais são representados tanto no espaço físico quanto no espaço de coordenadas naturais adimensionais. Cada elemento é caracterizado pelas coordenadas de seus nós, sendo os graus de liberdade do problema associados aos valores das funções de forma interpolados nesses pontos nodais.

Dentro da IGA têm-se duas noções de malha: uma malha de pontos de controle e uma malha física. A malha de pontos de controle é muito semelhante a uma malha de elementos finitos, entretanto, ela não define a geometria, ela é apenas um esqueleto que controla o formato da geometria (ver Fig. 1.1), visto que as funções de forma baseadas em *B-Splines* não são necessariamente interpolatórias. Dessa forma, os graus de liberdade do problema são associados aos pontos de controle, cujas posições não coincidem, necessariamente, com a geometria representada.

A malha física representa a geometria discretizada. Dentro da malha física podem ser definidos dois tipos de elementos, um macro-elemento, denominado de *patch*, e o *knot span*, que é o equivalente a um elemento finito e será denominado como célula ao longo deste texto. Cada *patch* é composto por um conjunto de células. Muitas geometrias simples podem ser discretizadas apenas com um *patch*, entretanto, a depender da complexidade da geometria ou de requisitos de parametrização, se torna necessário o uso de um conjunto de *patches*. As células são representações geométricas de linhas, superfícies e volumes nos espaços físicos unidimensional, bidimensional e tridimensional respectivamente.

Cada *patch* e suas respectivas células possuem uma representação no espaço paramétrico (Fig. 1.1), que é o espaço onde as funções base são definidas. O espaço paramétrico, para os casos de funções univariadas, é definido por um *knot vector*, aqui denominado de vetor de *knots*, que é um conjunto de *knots* ou coordenadas paramétricas. As células são constituídas pelo espaço entre dois *knots* consecutivos. O espaço onde se representam todas as células, inclusive as nulas (quando mais de um *knot* ocupa a mesma

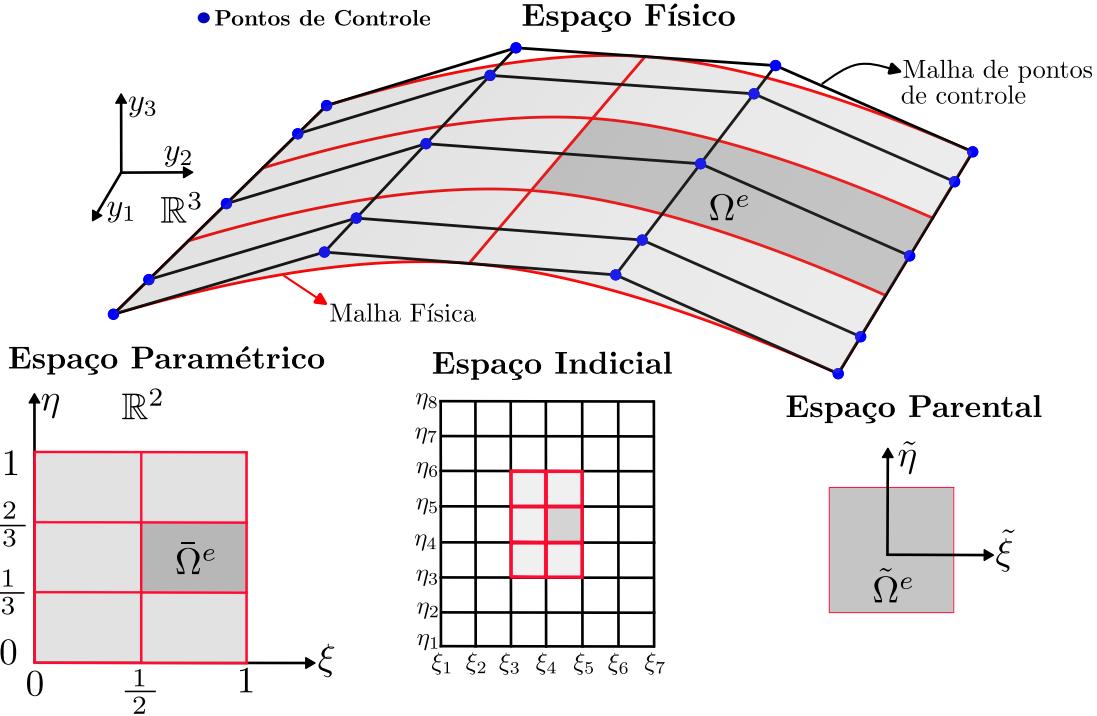


Figura 1.1 – NURBS: espaço físico, espaço paramétrico, espaço indicial e espaço parental

posição), é chamado de espaço indicial.

Por fim, na análise isogeométrica conta-se ainda com o espaço parental, que é o espaço de integração numérica das funções base, em geral, definido de forma adimensional $[-1, 1]$ dentro de uma célula. Na Fig. 1.1 pode-se observar os espaços relatados para uma superfície 3D construída por funções base quadráticas e apenas um *patch*.

1.2 *B-Splines*

Para a construção de geometriaa NURBS, que serão empregadas neste trabalhos, é fundamental compreender as funções base *B-splines* e suas particularidades. Essas funções servem como o ponto de partida para a definição de curvas, superfícies e sólidos NURBS, sendo essenciais para o entendimento da flexibilidade e controle geométrico oferecido por esse modelo. As *B-splines* são funções construídas através de um vetor de coordenadas paramétricas (vetor de *knots*) e que dependem de um conjunto de pontos de controle, sendo esses elementos responsáveis por estabelecer a forma geométrica e o grau de continuidade da curva ou superfície.

1.2.1 *Vetor de knots*

As funções *B-Splines*, utilizadas na construção das NURBS, são definidas em um espaço paramétrico que é comum a um conjunto de células ou *patch*. O espaço paramétrico unidimensional é construído através de um vetor de *knots*, que consiste em um conjunto não decrescente de coordenadas paramétricas, definido como: $\Xi = [\xi_0, \xi_1, \dots, \xi_{n+p+1}]$, sendo que $\xi_i \in \mathbb{R}$ e representa a i -ésima coordenada paramétrica com $i = 0, 1, \dots, n + p + 1$, e p corresponde ao grau polinomial das funções. O parâmetro n equivale ao índice da última função base nesta direção paramétrica, sendo o conjunto de funções base indexado de 0 a n , totalizando $n + 1$ funções. Os *knot spans*, ou intervalo entre *knots*, definem células no espaço paramétrico, cujos contornos são mapeados pelas funções base para formar a malha no espaço físico.

O vetor de *knots* é classificado como uniforme quando as coordenadas paramétricas são igualmente espaçadas, e como não-uniformes, caso contrário. A multiplicidade de um *knot* pode ser superior a um, influenciando diretamente na continuidade e na forma das funções base, conforme será visto posteriormente. Os vetores de *knots* conhecidos como abertos, são frequentemente utilizados nas literaturas de CAD, e caracterizam-se por ter a primeira e a última coordenada paramétrica repetidas $p + 1$ vezes. Este fato garante que as funções sejam interpolatórias nos extremos do espaço paramétrico e nas bordas entre *patches*, proporcionando, por exemplo, a homogeneidade com respeito às condições de contorno essenciais.

1.2.2 *Funções base e suas derivadas*

As funções base *B-Splines* (N^b) univariadas são definidas a partir de um vetor de *knots* unidimensional, sendo para $p = 0$, escritas através da seguinte relação:

$$N_{i,0}^b(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{caso contrário} \end{cases}, \quad (1.1)$$

enquanto que para funções com $p \geq 1$ podem ser definidas por recorrência de:

$$N_{i,p}^b(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}^b(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}^b(\xi), \quad (1.2)$$

com $i = 0, 1, \dots, n$.

Essa equação é conhecida como fórmula recursiva de *Cox-de Boor* (????). Para funções *B-Splines* de grau $p = 0$ ou $p = 1$, obtém-se, respectivamente, as mesmas funções constantes e lineares por partes utilizadas no método dos elementos finitos clássico.

Na Fig. 1.2, pode-se observar funções *B-Splines* quadráticas construídas sobre o vetor de *knots* não-uniforme aberto $\Xi = [0; 0; 0; 1; 2; 3; 3; 4; 4; 4]$. A figura evidencia que, devido à repetição de $p + 1$ vezes dos *knots* nas extremidades do vetor, as funções base se tornam interpolatórias nesses pontos. Ademais, a presença de um *knot* com multiplicidade

2 em $\xi = 3$ reduz a regularidade da função base nesse ponto, resultando na descontinuidade da sua derivada. Em termos gerais, a continuidade de uma função *B-Spline* em uma coordenada paramétrica é dada por C^{p-m} , onde m é a multiplicidade do *knot*.

ARRUAR A LEGENDA DE TODAS AS FIGURAS DE ACORDO COM A NORMA ATUAL (LEGENDA ACIMA DA FIGURA E COLOCAR A AUTORIA)

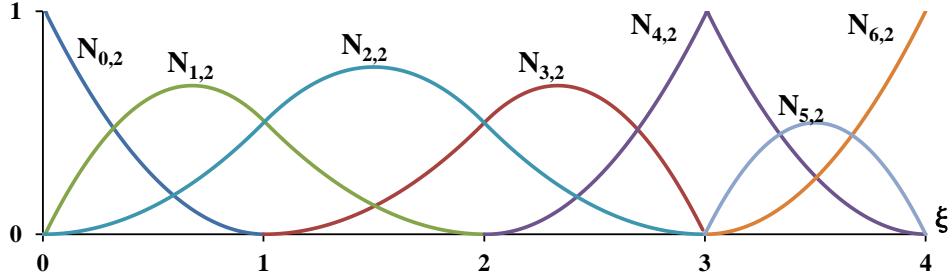


Figura 1.2 – *B-Splines quadráticas*

As principais propriedades das funções *B-Splines* são:

- **Partição da Unidade:** $\sum_{i=0}^n N_{i,p}^b(\xi) = 1$;
- **Positividade:** Todas as funções base são positivas, ou seja, $N_{i,p}^b \geq 0, \forall \xi$;
- **Suavidade:** função de ordem p é, em geral, $p - 1$ vezes continua no contorno das células;
- **Suporte Compacto:** O suporte de cada $N_{i,p}^b$ está contido no intervalo $[\xi_i, \xi_{i+p+1}]$, ou seja, em cada célula, apenas $p + 1$ funções são não nulas.

A derivada de uma função de forma *B-Spline* pode ser calculada recursivamente em termos de funções base de ordem menor. Considerando uma função de ordem p e vetor de *knots* Ξ , a derivada da i -ésima função de forma pode ser escrita como:

$$\frac{d}{d\xi} N_{i,p}^b(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}^b(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}^b(\xi). \quad (1.3)$$

Essa expressão pode ser generalizada para derivadas de ordem superior através de:

$$\frac{d^k}{d\xi^k} N_{i,p}^b(\xi) = \frac{p!}{(p-k)!} \sum_{j=0}^k \alpha_{k,j} N_{i+j,p-k}^b(\xi), \quad (1.4)$$

que representa a k -ésima derivada da função $N_{i,p}^b(\xi)$, onde:

$$\alpha_{0,0} = 1, \quad (1.5)$$

$$\alpha_{k,0} = \frac{\alpha_{k-1,0}}{\xi_{i+p-k+1} - \xi_i}, \quad (1.6)$$

$$\alpha_{k,j} = \frac{\alpha_{k-1,j} - \alpha_{k-1,j-1}}{\xi_{i+p+j-k+1} - \xi_{i+j}} \quad j = 1, \dots, k-1, \quad (1.7)$$

$$\alpha_{k,k} = \frac{-\alpha_{k-1,k-1}}{\xi_{i+p+1} - \xi_{i+k}}. \quad (1.8)$$

Algoritmos eficientes para a determinação das funções de forma *B-Splines* e de suas derivadas podem ser encontradas em ??).

1.2.3 Geometrias *B-Splines*

Uma curva *B-Spline* é construída a partir da combinação linear entre funções base e um conjunto de pontos de controle. Considerando um conjunto de $n + 1$ funções base $N_{i,p}^b$ e respectivos pontos de controle $\mathbf{B}_i \in \mathbb{R}^{n_{sd}}$ com $i = 0, 1, \dots, n$, uma curva polinomial por partes *B-Spline* univariada é definida como:

$$\mathbf{C} = \mathbf{y}(\xi) = \sum_{i=0}^n N_{i,p}^b(\xi) \mathbf{B}_i, \quad (1.9)$$

com y_1 , y_2 e y_3 sendo as componentes do vetor de coordenadas físicas \mathbf{y} . Utilizando as funções *B-Splines* apresentadas na Fig. 1.2 e uma malha de $n + 1$ pontos de controle, obtém-se a curva apresentada na Fig. 1.3a. Na Fig. 1.3b pode-se observar as células físicas equivalentes a essa combinação.

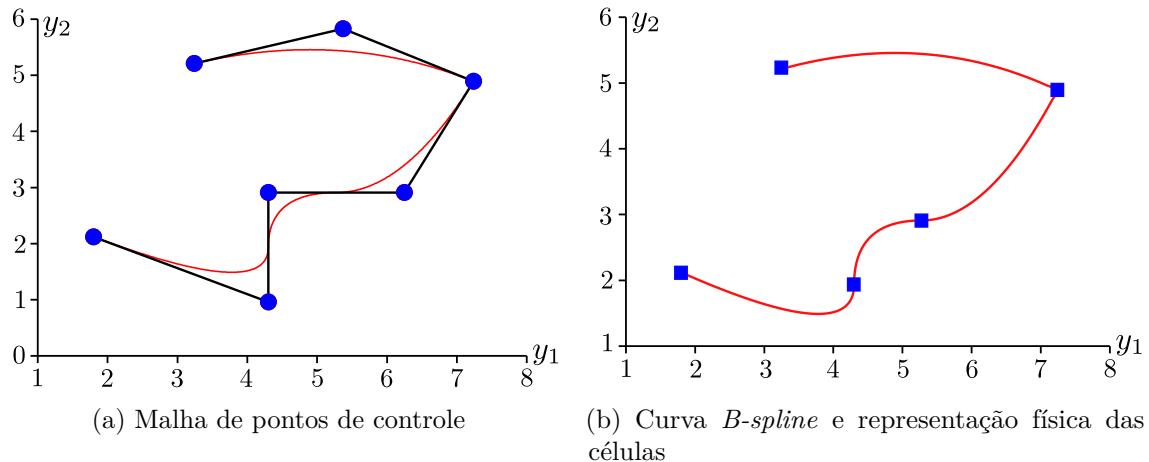


Figura 1.3 – Curva *B-Spline*

A partir da Fig. 1.3a nota-se que a curva *B-Spline* interpola o primeiro e o último ponto de controle, o que é uma característica das curvas construídas com funções descritas a partir de vetores de *knots* abertos. Adicionalmente, devido à multiplicidade do *knot* de coordenada paramétrica $\xi = 3$, existe um ponto de controle intermediário também interpolando a curva. Coordenadas paramétricas com multiplicidade maior ou igual ao grau polinomial p resultam, por definição, em interpolação dos pontos de controle associados. Além disso, a curva possui continuidade $C^{p-1} = C^1$ em todos os lugares, exceto em $\xi = 3$, onde equivale a $C^{p-2} = C^0$, que trata-se de uma propriedade herdada das funções base.

Conforme observado nas figuras: Fig. 1.3a e Fig. 1.3b, muitas das características de curvas *B-Splines* são consequências das propriedades das funções *B-splines*. Outra importante propriedade dessas curvas é a Transformação Afim, que significa que uma transformação afim de uma curva B-spline é obtida aplicando a transformação diretamente aos pontos de controle. Além disso, devido ao suporte compacto das funções base, as curvas *B-Splines* possuem característica denominada de *localidade*, que significa que, movendo-se um ponto de controle, afeta-se não mais do que $p+1$ células na curva. Outras propriedades matemáticas das curvas *B-Splines* podem ser consultadas em detalhes em ??).

Uma superfície *B-spline* é obtida analogamente à curva *B-spline*. Dado uma rede de pontos de controle $\mathbf{B}_{i,j} \in \mathbb{R}^{n_{sd}}$ com $i = 0, 1, \dots, n$ e $j = 0, 1, \dots, m$, e vetores de *knots* $\Xi = [\xi_0, \xi_1, \dots, \xi_{p+n+1}]$, $\mathcal{H} = [\eta_0, \eta_1, \dots, \eta_{q+m+1}]$, a superfície é obtida através do produto tensorial entre $(n+1)$ funções univariadas $N_{i,p}^b$ e $(m+1)$ funções univariadas $M_{j,q}^b$ da seguinte forma:

$$\mathbf{S} = \mathbf{y}(\xi, \eta) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}^b(\xi) M_{j,q}^b(\eta) \mathbf{B}_{i,j}, \quad (1.10)$$

onde q representa o grau das $m+1$ funções na direção paramétrica η . Muitas das propriedades das superfícies *B-Splines* são resultado da natureza do produto tensorial que as geram. A base de funções apresenta propriedade de positividade e formam uma partição de unidade, de forma que: $\forall (\xi, \eta) \in [\xi_0, \xi_1, \dots, \xi_{p+n+1}] \times [\eta_0, \eta_1, \dots, \eta_{q+m+1}]$:

$$\sum_{i=0}^n \sum_{j=0}^m N_{i,p}^b(\xi) M_{j,q}^b(\eta) = \left(\sum_{i=0}^n N_{i,p}^b(\xi) \right) \left(\sum_{j=0}^m M_{j,q}^b(\eta) \right) = 1. \quad (1.11)$$

O suporte, por exemplo, de uma função bivariada $\hat{N}_{i,j:p,q}^b(\xi, \eta) = N_{i,p}^b(\xi) M_{j,q}^b(\eta)$ é equivalente à: $[\xi_i, \xi_{i+p+1}] \times [\eta_j, \eta_{j+q+1}]$.

Por fim, um sólido *B-Spline* é obtido através do produto tensorial entre funções univariadas $N_{i,p}^b$, $M_{j,q}^b$, $L_{k,r}^b$, construídas sobre os vetores de *knots* $\Xi = [\xi_0, \xi_1, \dots, \xi_{p+n+1}]$, $\mathcal{H} = [\eta_0, \eta_1, \dots, \eta_{q+m+1}]$ e $\mathcal{Z} = [\zeta_0, \zeta_1, \dots, \zeta_{r+l+1}]$ respectivamente, e um conjunto de pontos de controle $\mathbf{B}_{i,j,k} \in \mathbb{R}^{n_{sd}}$ com $i = 0, 1, \dots, n$, $j = 0, 1, \dots, m$, $k = 0, 1, \dots, l$, da seguinte forma:

$$\mathbf{T} = \mathbf{y}(\xi, \eta, \zeta) = \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l N_{i,p}^b(\xi) M_{j,q}^b(\eta) L_{k,r}^b(\zeta) \mathbf{B}_{i,j,k}, \quad (1.12)$$

na qual r representa o grau das $l+1$ funções base na direção paramétrica ζ . As propriedades de um sólido *B-Spline*, correspondem às generalizações trivariadas das propriedades das superfícies *B-Spline*. Além disso, o suporte de uma função trivariada $\hat{N}_{i,j,k:p,q,r}^b(\xi, \eta, \zeta) = N_{i,p}^b(\xi) M_{j,q}^b(\eta) L_{k,r}^b(\zeta)$ está contido no intervalo $[\xi_i, \xi_{i+p+1}] \times [\eta_j, \eta_{j+q+1}] \times [\zeta_k, \zeta_{k+r+1}]$.

1.2.4 Refinamento

Um dos aspectos mais relevantes das *B-splines* é a flexibilidade na forma de enriquecimento da base, permitindo aprimorar sua representação sem alterar a geometria

subjacente nem sua parametrização. Dentre os principais procedimentos utilizados, destacam-se: a inserção de *knots* (ou refinamento h), que consiste na subdivisão da malha; a elevação de grau (ou refinamento p), que aumenta a ordem polinomial das funções base; o refinamento k , que promove simultaneamente um aumento da ordem e da continuidade entre células; e, por fim, o refinamento hpk , que combina de forma coordenada as três estratégias anteriores, oferecendo maior controle e eficiência na representação da geometria e na solução numérica de problemas.

Neste trabalho, é adotado apenas o refinamento h na geração das geometrias baseado na inserção de *knots*.

O enriquecimento da base de funções utilizando a inserção de *knots* é realizado sem que se altere uma curva geometricamente ou parametricamente. Para essa finalidade, partindo-se de um vetor de *knots* $\Xi = [\xi_0, \xi_1, \dots, \xi_{n+p+1}]$, introduz-se o conceito de vetor de *knots* estendido, o qual é representado por: $\bar{\Xi} = [\bar{\xi}_0 = \xi_0, \bar{\xi}_1, \dots, \bar{\xi}_{n+m+p+1} = \xi_{n+p+1}]$. As $(n + m + 1)$ novas funções base *B-Splines* são determinadas através das Eqs. (1.1 e 1.2), aplicando-as ao vetor de *knots* $\bar{\Xi}$. Os $(n + m + 1)$ novos pontos de controle $\bar{\mathcal{B}} = [\bar{\mathbf{B}}_0, \bar{\mathbf{B}}_1, \dots, \bar{\mathbf{B}}_{n+m}]^T$ são obtidos através da combinação linear dos $(n + 1)$ pontos de controle originais, $\mathcal{B} = [\mathbf{B}_0, \mathbf{B}_1, \dots, \mathbf{B}_n]^T$, por:

$$\bar{\mathcal{B}} = \mathbf{T}^p \mathcal{B}, \quad (1.13)$$

, com:

$$\mathbf{T}_{ij}^0 = \begin{cases} 1 & \text{se } \bar{\xi}_i \in [\xi_j, \xi_{j+1}) \\ 0 & \text{caso contrário} \end{cases} \quad (1.14)$$

e

$$\mathbf{T}_{ij}^{q+1} = \frac{\bar{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi_j} \mathbf{T}_{ij}^q + \frac{\xi_{j+q+1} - \bar{\xi}_{i+q}}{\xi_{j+q+1} - \xi_{j+1}} \mathbf{T}_{ij+1}^q \text{ com } q = 0, 1, 2, \dots, p-1, \quad (1.15)$$

com $i = 0, 1, \dots, (n + m)$ e $j = 0, 1, \dots, n$.

Considerando uma curva quadrática *B-spline* construída sobre um vetor de *knots* aberto $\Xi = [0, 0, 0, 1, 1, 1]$ apresentada na Fig. 1.4a juntamente com sua rede de pontos de controle, nota-se que essa curva, possui apenas uma célula no espaço físico e 3 funções base no espaço paramétrico (Fig. 1.4e). Ao realizar a inserção de um *knot*, $\xi = 1/2$, o vetor de *knots* estendido fica definido como: $\bar{\Xi} = [0, 0, 0, 1/2, 1, 1, 1]$. Aplicando-se as Eq. 1.1 e Eq. 1.2 à esse vetor de coordenadas paramétricas, obtém-se as 4 funções base apresentadas na Fig. 1.4f definidas sobre 2 células do espaço paramétrico. Após o emprego do refinamento h , a geometria da curva é preservada. No entanto, como ilustrado na Fig. 1.4d, uma nova célula física é inserida, além de que, de acordo com a Fig. 1.4b, a malha de pontos de controle é modificada, com o acréscimo de um novo ponto e o reajuste de suas posições.

Para fins práticos, o processo de refinamento consiste na inserção consecutiva de coordenadas paramétricas ao vetor de (*knots*) até que se alcance a discretização desejada.

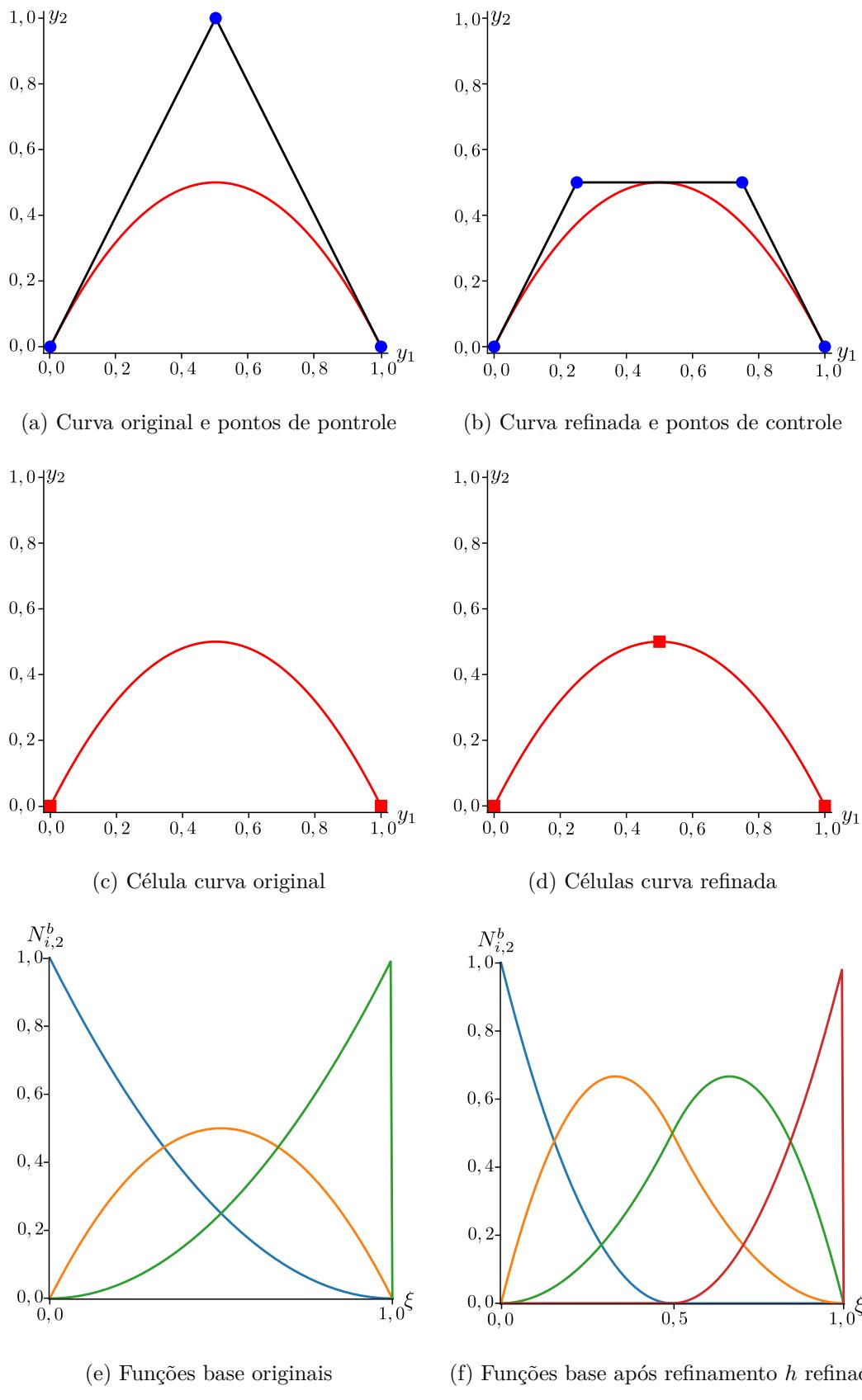


Figura 1.4 – Refinamento h para um curva *B-Spline*

Um algoritmo mais eficiente para realizar esse procedimento de refinamento pode ser encontrado em ??). Esse procedimento pode ser aplicado analogamente à superfícies e sólidos, aplicando-se a inserção de *knots* nas direções paramétricas desejadas.

1.3 NURBS

Uma geometria NURBS no $\mathbb{R}^{n_{sd}}$ pode ser entendida, do ponto de vista geométrico, como a transformação projetiva de uma geometria *B-Spline* no $\mathbb{R}^{n_{sd}+1}$. Nesse contexto, geometrias cônicas podem ser construídas exatamente através de curvas quadráticas por partes. Na Fig. 1.5b, apresenta-se uma curva NURBS $\mathbf{C}(\xi)$ no \mathbb{R}^2 , que representa de forma exata uma circunferência, a qual foi obtida a partir da transformação projetiva de uma curva quadrática por partes *B-Spline* ($\mathbf{C}^w(\xi)$) no \mathbb{R}^3 . A transformação é realizada através da projeção em um plano $y_3 = 1$ de cada ponto da curva projetiva ($\mathbf{C}^w(\xi)$) através de um raio que passa pela origem.

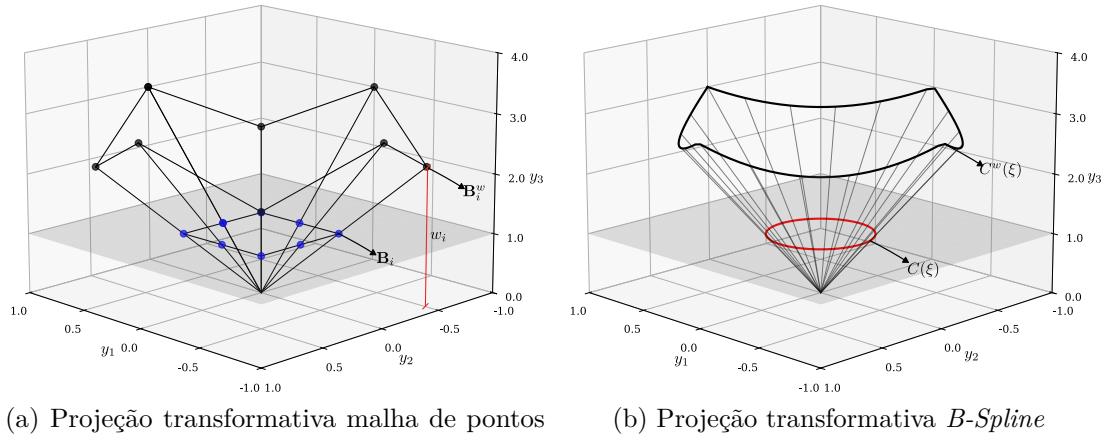


Figura 1.5 – Projeção transformativa de entidade *B-Spline*

O mesmo procedimento de transformação pode ser realizado para obtenção dos pontos de controle NURBS (Fig. 1.5a) a partir de pontos de controle projetivos (\mathbf{B}_i^w), usando a seguinte relação:

$$(\mathbf{B}_i)_j = (\mathbf{B}_i^w)_j / w_i, \quad j = 1, \dots, n_{sd}, \quad (1.16)$$

$$w_i = (\mathbf{B}_i^w)_{n_{sd}+1}, \quad (1.17)$$

com $(\mathbf{B}_i)_j$ o j -ésimo componente do vetor \mathbf{B}_i e w_i refere-se ao i -ésimo peso, que consiste na coordenada y_3 dos pontos de controle projetivos para o exemplo citado.

Para a aplicação dessa mesma transformação para cada ponto da curva, será utilizado um conceito de função peso, dada por:

$$W(\xi) = \sum_{\hat{i}=0}^n N_{\hat{i},p}^b(\xi) w_{\hat{i}}, \quad (1.18)$$

e a curva NURBS pode ser definida como:

$$(\mathbf{C}(\xi))_j = (\mathbf{C}^w(\xi))_j / W(\xi), \quad j = 1, \dots, n_{\text{sd}}. \quad (1.19)$$

Tanto \mathbf{C}^w como $W(\xi)$ são funções polinomiais por partes, dessa forma, $\mathbf{C}(\xi)$ é uma função racional por partes.

1.3.1 Funções base NURBS e suas derivadas

Matematicamente, uma função NURBS é obtida pela rationalização de uma função *B-Spline*. Uma função racional NURBS (R) é construída através da seguinte expressão:

$$R_{i,p}(\xi) = \frac{N_{i,p}^b(\xi) w_i}{\sum_{\hat{i}=0}^n N_{\hat{i},p}^b(\xi) w_{\hat{i}}}. \quad (1.20)$$

com w_i e $w_{\hat{i}} \in \mathbb{R}$, sendo $i = \hat{i} = 0, 1, \dots, n$.

A derivada de uma função $R_{i,p}$ é obtida aplicando simplesmente a regra do quociente à expressão da Eq. 1.20:

$$\frac{d}{d\xi} R_{i,p}(\xi) = w_i \frac{W(\xi) (N_{i,p}^b)'(\xi) - W'(\xi) N_{i,p}^b(\xi)}{(W(\xi))^2}, \quad (1.21)$$

com:

$$(N_{i,p}^b)'(\xi) \equiv \frac{d}{d\xi} N_{i,p}^b(\xi), \quad (1.22)$$

e:

$$W'(\xi) = \sum_{\hat{i}=0}^n (N_{\hat{i},p}^b)'(\xi) w_{\hat{i}}. \quad (1.23)$$

A k -ésima derivada de $R_{i,p}$ é obtida em termos de derivadas de menores ordem, através da seguinte expressão:

$$\frac{d^k}{d\xi^k} R_{i,p}(\xi) = \frac{A_i^{(k)}(\xi) - \sum_{j=1}^k \binom{k}{j} W^{(j)}(\xi) \frac{d^{(k-j)}}{d\xi^{(k-j)}} R_{i,p}(\xi)}{W(\xi)} \quad (1.24)$$

com:

$$\binom{k}{j} = \frac{k!}{j!(k-j)!}, \quad (1.25)$$

$$W^{(j)}(\xi) = \frac{d^j}{d\xi^j} W(\xi), \quad (1.26)$$

e:

$$A_i^{(k)}(\xi) = w_i \frac{d^k}{d\xi^k} N_{i,p}^b(\xi) \text{ sem soma em } i. \quad (1.27)$$

1.3.2 Geometria NURBS

Uma curva NURBS é obtida através da combinação linear entre as funções base NURBS e um conjunto de pontos de controle, conforme expresso pela equação abaixo:

$$\mathbf{C} = \mathbf{y}(\xi) = \sum_{i=0}^n R_{i,p}(\xi) \mathbf{B}_i, \quad (1.28)$$

cujos pontos de controle e pesos são escolhidos criteriosamente de forma a obter-se a geometria desejada.

Analogamente, uma superfície NURBS é obtida através das seguintes relações:

$$R_{i,j:p,q}(\xi, \eta) = \frac{N_{i,p}^b(\xi) M_{j,q}^b(\eta) w_{i,j}}{\sum_{\hat{i}=0}^n \sum_{\hat{j}=0}^m N_{\hat{i},p}^b(\xi) M_{\hat{j},q}^b(\eta) w_{\hat{i},\hat{j}}}, \quad (1.29)$$

$$\mathbf{S} = \mathbf{y}(\xi, \eta) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j:p,q}(\xi, \eta) \mathbf{B}_{i,j}, \quad (1.30)$$

com $w_{i,j}$ e $w_{\hat{i},\hat{j}} \in \mathbb{R}$, sendo $i = \hat{i} = 0, 1, \dots, n$ e $j = \hat{j} = 0, 1, \dots, m$, correspondem aos pesos relativos às funções $N_{i,p}^b(\xi) M_{j,q}^b(\eta)$ e $N_{\hat{i},p}^b(\xi) M_{\hat{j},q}^b(\eta)$ respectivamente. Por fim, um sólido NURBS é obtido por:

$$R_{i,j,k:p,q,r}(\xi, \eta, \zeta) = \frac{N_{i,p}^b(\xi) M_{j,q}^b(\eta) L_{k,r}^b(\zeta) w_{i,j,k}}{\sum_{\hat{i}=0}^n \sum_{\hat{j}=0}^m \sum_{\hat{k}=0}^l N_{\hat{i},p}^b(\xi) M_{\hat{j},q}^b(\eta) L_{\hat{k},r}^b(\zeta) w_{\hat{i},\hat{j},\hat{k}}}, \quad (1.31)$$

$$\mathbf{T} = \mathbf{y}(\xi, \eta, \zeta) = \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l R_{i,j,k:p,q,r}(\xi, \eta, \zeta) \mathbf{B}_{i,j,k}, \quad (1.32)$$

onde $w_{i,j,k}$ e $w_{\hat{i},\hat{j},\hat{k}} \in \mathbb{R}$, sendo $i = \hat{i} = 0, 1, \dots, n$, $j = \hat{j} = 0, 1, \dots, m$ e $k = \hat{k} = 0, 1, \dots, l$, correspondem aos pesos relativos às funções $N_{i,p}^b(\xi) M_{j,q}^b(\eta) L_{k,r}^b(\zeta)$ e $N_{\hat{i},p}^b(\xi) M_{\hat{j},q}^b(\eta) L_{\hat{k},r}^b(\zeta)$, respectivamente.

1.3.3 Múltiplos Patches

Na grande maioria das situações práticas, é necessário o uso de múltiplos *patches* NURBS para descrever um domínio computacional. Isso ocorre porque o produto tensorial do espaço paramétrico não é adequado para a representação de domínios complexos ou multiplamente conectados. Ademais, mesmo para domínios simples, do ponto de vista da simulação numérica, o uso de múltiplos *patches* pode ser necessário em algumas circunstâncias, conforme será visto na seção de exemplos.

??) destaca que o uso de múltiplos *patches* pode facilitar a análise numérica quando diferentes materiais e modelos físicos são utilizados em distintas regiões do domínio. Além

disso, em processamento paralelo, pode ser conveniente, do ponto de vista das estruturas de dados, não compartilhar um único *patch* entre diferentes processadores.

A utilização de múltiplos *patches* implica a necessidade de compatibilizar a discretização na interface entre *patches* adjacentes; ou seja, a parametrização e o mapeamento devem ser idênticos nesses locais. Cada ponto de controle em uma face de *patches* adjacentes deve possuir um correspondente na face oposta. Esses pontos coincidentes são tratados como um único ponto de controle dentro do sistema global resultante da análise numérica.

Ressalta-se ainda, que na interface entre os *patches*, devido a natureza interpolatória do vetores de *knots* abertos, as funções base possuem continuidade C_0 , conforme pode ser observado na Fig. 1.6, onde apresentam-se as funções base univariadas na interface entre dois *patches*.

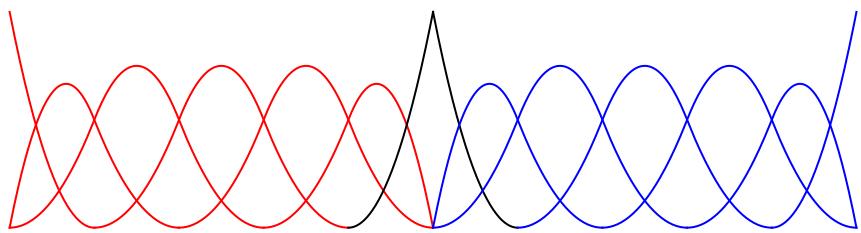


Figura 1.6 – Funções base univariadas na interface entre *patches*

1.4 Análise Isogeométrica

Para a aplicação da IGA no contexto da Dinâmica dos Fluidos Computacional, é utilizada a mesma metodologia apresentada no Cap. ???. A aproximação da geometria, realizada no contexto do MEF pela Eq. ?? é substituída pela abordagem Isogeométrica através do uso de geometrias NURBS, descritas pelas equações: Eq. 1.28, Eq. 1.30 ou Eq. 1.32 para os casos de curvas, superfícies ou sólidos, respectivamente. As funções tentativa para velocidade e pressão, e as funções teste associadas à elas, apresentadas nas Eqs. (?? - ??) seguem exatamente a mesma lógica, poré, as funções de forma N , são substituídas pelas funções base NURBS $R_{i,p}(\xi)$, $R_{i,j:p,q}(\xi, \eta)$ e $R_{i,j,k:p,q,r}(\xi, \eta, \zeta)$, a depender da geometria em análise.

Assim, a implementação da IGA para escoamentos incompressíveis segue o mesmo roteiro apresentada no Alg. ??, levando-se em consideração as mudanças nas funções de forma, a forma de proceder com a integração numérica e as particularidades no cálculo do parâmetros de estabinização, conforme descritos na sequência desta seção.

A integração numérica é realizada no domínio de cada célula através da quadratura Gaussiana. Considerando o domínio paramétrico de uma célula: $\bar{\Omega}^e$, e o domínio de integração ou parental: $\tilde{\Omega}^e$, apresentados na Fig.1.1, definidos respectivamente pelos vetores

de coordenadas paramétricas ξ (ξ, η, ζ) e $\tilde{\xi}$ ($\tilde{\xi}, \tilde{\eta}, \tilde{\zeta}$), a matriz jacobiana do mapeamento do espaço físico, com coordenadas \mathbf{y} (y_1, y_2, y_3), para o espaço de quadratura, é definida por:

$$\frac{d\mathbf{y}}{d\tilde{\xi}} = \frac{dy}{d\xi} \frac{d\xi}{d\tilde{\xi}}, \quad (1.33)$$

com $\tilde{\xi}, \tilde{\eta}, \tilde{\zeta} \in [-1, 1]$.

O primeiro termo à direita da igualdade da Eq. (1.33) é calculado a partir das derivadas parciais da Equação Eq. (1.28), Eq. (1.30) ou Eq. (1.32), a depender do tipo da geometria em questão (curva, superfície ou sólido, respectivamente).

Para a obtenção do segundo termo à direita, primeiramente é necessário definir a relação entre as coordenadas do domínio paramétrico e do domínio parental. Considerando-se a célula $\bar{\Omega}^e = [\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}] \times [\zeta_k, \zeta_{k+1}]$, pode-se calcular $\xi, \eta, \zeta \in \bar{\Omega}^e$ a partir de $\tilde{\xi}, \tilde{\eta}, \tilde{\zeta} \in \tilde{\Omega}^e$ através das seguintes relações:

$$\xi = \xi_i + (\tilde{\xi} + 1) \left(\frac{\xi_{i+1} - \xi_i}{2} \right), \quad (1.34)$$

$$\eta = \eta_i + (\tilde{\eta} + 1) \left(\frac{\eta_{i+1} - \eta_i}{2} \right), \quad (1.35)$$

e

$$\zeta = \zeta_i + (\tilde{\zeta} + 1) \left(\frac{\zeta_{i+1} - \zeta_i}{2} \right), \quad (1.36)$$

assim, $\frac{d\xi}{d\tilde{\xi}}$ é obtido derivando-se parcialmente Eq. (1.34, 1.35 e 1.36).

1.4.1 Parâmetros de estabilização

Para a determinação dos parâmetros de estabilização τ , de acordo com o exposto na Subseção ??, faz-se necessária a determinação de um tensor métrico, \mathbf{G} (Eq. ??), o qual depende da matriz jacobiana transformada, $\hat{\mathbf{Q}}$, definida na Eq. ??.

Dada a diferença entre o espaço paramétrico utilizado na definição das funções de base e do espaço paramétrico de integração, definido aqui como espaço parental, a matriz \mathbf{Q} é reescrita como:

$$\mathbf{Q} = \left(\frac{\partial \mathbf{y}}{\partial \tilde{\xi}} \right). \quad (1.37)$$

Para a obtenção de $\hat{\mathbf{Q}}$, de acordo com a Eq. ??, define-se a matriz \mathbf{D} para análise isogeométrica, de acordo com o trabalho de ??), como:

$$\mathbf{D} = \left(\frac{\partial \hat{\xi}}{\partial \tilde{\xi}} \right), \quad (1.38)$$

que representa a relação entre o espaço paramétrico de preferência, onde o comprimento efetivo da célula deve ser medido, e o espaço de integração, onde são definidos os pontos

de quadratura.

O espaço paramétrico de preferência para problemas unidimensionais é definido para cada célula por meio de uma interpolação usando polinômios de Bernstein B_b^p de ordem p :

$$\hat{\xi}(\tilde{\xi}) = \sum_{b=0}^p \hat{\xi}_b B_b^p(\tilde{\xi}), \quad (1.39)$$

com os pontos de controle de Bézier, $\hat{\xi}_b$, definidos igualmente espaçados da seguinte maneira:

$$\hat{\xi}_b = \frac{\Delta\hat{\xi}}{p} b, \quad (1.40)$$

sendo $\Delta\hat{\xi}$ o comprimento paramétrico da célula de Bézier.

Os pontos de controle correspondentes no espaço de integração são dados por:

$$\tilde{\xi}_a = \frac{\Delta\hat{\xi}}{p} \sum_{b=0}^p b \left\{ \mathbf{C}^{-1} \right\}_{ba}, \quad (1.41)$$

com $a = 0, \dots, p$. \mathbf{C} sendo o operador de extração de Bézier, que relaciona as funções B-spline globais às funções de Bernstein locais, cuja obtenção, neste trabalho, é realizada de acordo com o exposto em ??).

O comprimento efetivo da célula para $a = 1, \dots, p$ pode ser calculado por:

$$\Delta\tilde{\xi}_a = \tilde{\xi}_a - \tilde{\xi}_{a-1} \quad (1.42)$$

$$= \frac{\Delta\hat{\xi}}{p} \sum_{b=0}^p b \left(\left\{ \mathbf{C}^{-1} \right\}_{ba} - \left\{ \mathbf{C}^{-1} \right\}_{ba-1} \right). \quad (1.43)$$

A partir disso pode-se definir o razão entre o comprimento da célula de Bézier e o comprimento efetivo da célula. Considerando um problema 1D, uma das proposta dos autores para D , utilizada neste trabalho, chama-se *RQD-MAX* e consiste em:

$$D = \frac{\Delta\hat{\xi}}{\min_{a=1,\dots,p} \Delta\tilde{\xi}_a}, \quad (1.44)$$

resultando em:

$$D = p \left(\min_{a=1,\dots,p} \sum_{b=0}^p b \left(\left\{ \mathbf{C}^{-1} \right\}_{ba} - \left\{ \mathbf{C}^{-1} \right\}_{ba-1} \right) \right)^{-1} \quad (1.45)$$

$$= p \max_{a=1,\dots,p} \left(\sum_{b=0}^p b \left(\left\{ \mathbf{C}^{-1} \right\}_{ba} - \left\{ \mathbf{C}^{-1} \right\}_{ba-1} \right) \right)^{-1}. \quad (1.46)$$

Para múltiplas dimensões o coeficiente de transformação D é obtido individualmente para cada uma das direções do espaço paramétrico, e os componentes da matriz de transformação \mathbf{D} são determinados como:

$$D_{ij} = D^i \delta_{ij}, \quad (1.47)$$

$i, j = 1, \dots, n_{pd}$, sendo n_{pd} a dimensão do espaço paramétrico.

1.5 Verificação e aplicações

Os exemplos escolhidos para a verificação do código computacional IGA para escoamentos incompressíveis são o escoamento sobre um cilindro e o escoamento sobre canal com degrau. Em ambas análises empregou-se discretização tridimensional para uma verificação mais completa do código.

1.5.1 Escoamento sobre um cilindro - Discretização isogeométrica

Descreva o problema aqui, mesmo que seja apenas para dizer que trata-se do mesmo problema simulado anteriormente

1.5.1.1 Geração da malha NURBS

Para a análise do problema de escoamento sobre um cilindro utilizando-se discretização isogeométrica espacial, dada a simplicidade da geometria envolvida, a malha é desenvolvida pela própria autora, i.e., sem o emprego de programas CAD para geração da malha isogeométrica. Para isso, com base nas dimensões bidimensionais do exemplo apresentado na Subseção ??, divide-se o domínio físico em 12 *patches*, conforme pode ser ilustrado na Fig. 1.7.

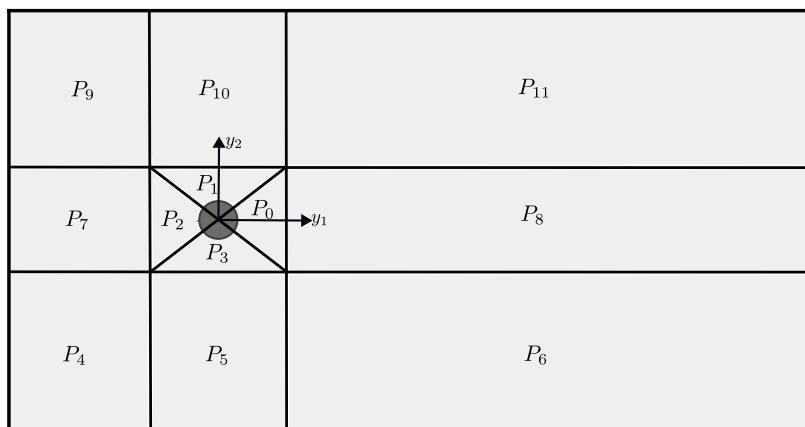
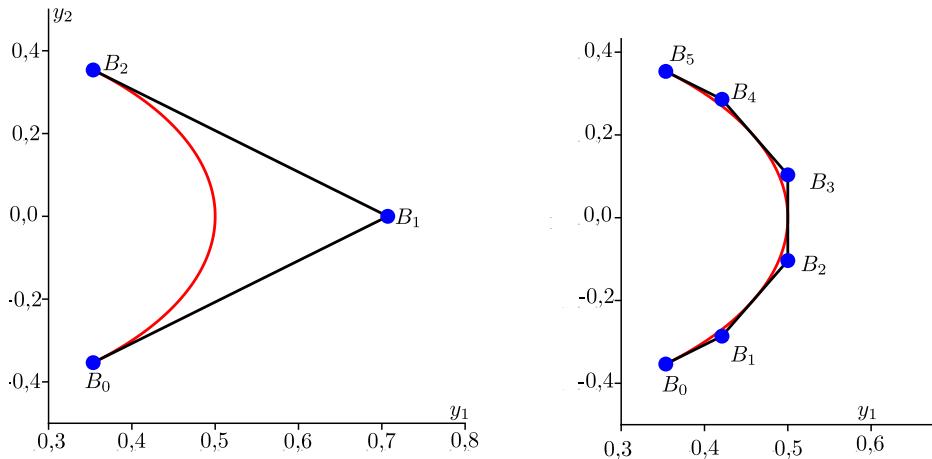


Figura 1.7 – Cilindro: Divisão dos *Patches*

O processo de geração da malha, simplificadamente, consiste em se escolher vetores de *knots*, pontos de controle, e pesos adequados para a descrição da geometria de cada *patch*, assegurando simultaneamente o refinamento necessário para a análise numérica.



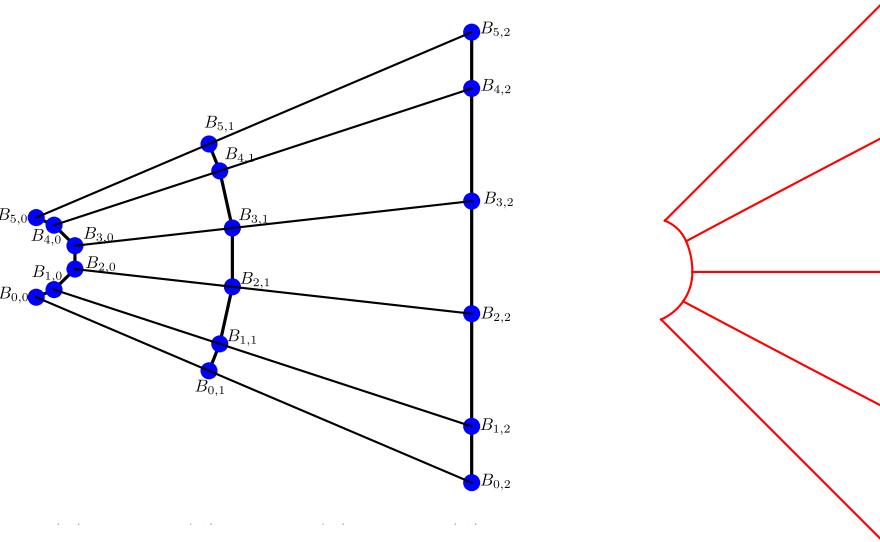
(a) Curva inicial e rede de pontos de controle (b) Curva refinada e rede de pontos de controle

Figura 1.8 – Cilindro: Obtenção da circunferência

Para a geração do primeiro *patch*, P_0 , cobre $1/4$ da circunferência do cilindro, inicia-se pela discretização de uma circunferência definida na direção paramétrica ξ . Utilizando o número mínimo de pontos de controle necessários para representar exatamente $1/4$ de circunferência de diâmetro D com o uso de funções quadráticas NURBS. O vetor de *knots* aberto adotado é definido por: $\Xi = [0, 0, 0, 1, 1, 1]$, sendo os pontos de controle $\mathbf{B}_0 = [\frac{\sqrt{2}D}{4}, -\frac{\sqrt{2}D}{4}, 0]$, $\mathbf{B}_1 = [\frac{D}{\sqrt{2}}, 0, 0]$ e $\mathbf{B}_2 = [\frac{\sqrt{2}D}{4}, \frac{\sqrt{2}D}{4}, 0]$, e os pesos: $w_0 = 1$, $w_1 = \frac{\sqrt{2}}{2}$ e $w_2 = 1$. A disposição dos pontos de controle e a curva resultante é apresentada na Fig.~1.8a.

Na sequência refina-se a discretização por inserção sucessiva de coordenadas paramétricas no vetor de *knots*. O algoritmo utilizado para este procedimento pode ser encontrado em ??). Na Fig. 1.8b apresenta-se um exemplo dos pontos de controle resultantes após a inserção das coordenadas paramétricas $1/4$, $1/2$ e $3/4$. Essa inserção resulta em três novas células físicas. A quantidade de coordenadas paramétricas a ser inserida depende da discretização desejada para a análise numérica.

Dando continuidade para a representação da geometria do *patch* P_0 , gera-se uma curva na direção paramétrica ξ que define o contorno direito do domínio. A curva é definida considerando o vetor de *knots* atualizado $\Xi = [0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1]$ e consiste em uma reta cujas coordenadas de suas extremidades inicial e final são: $\mathbf{y}_0 = [2, -2, 0]$ e $\mathbf{y}_1 = [2, 2, 0]$, respectivamente. Os 6 pontos de controle são distribuídos sobre a reta através de um espaçamento não uniforme: nas extremidades, a distância entre pontos consecutivos corresponde à metade do espaçamento adotado no interior, enquanto a região central é subdividida uniformemente, conforme pode ser observado na Fig. 1.9a. Essa distribuição não uniforme dos pontos de controle proporciona uma distribuição uniforme das células mapeadas no espaço físico. Para essa curva, todos os pesos são definidos como 1.



(a) Rede de pontos de controle superfície

(b) Malha física

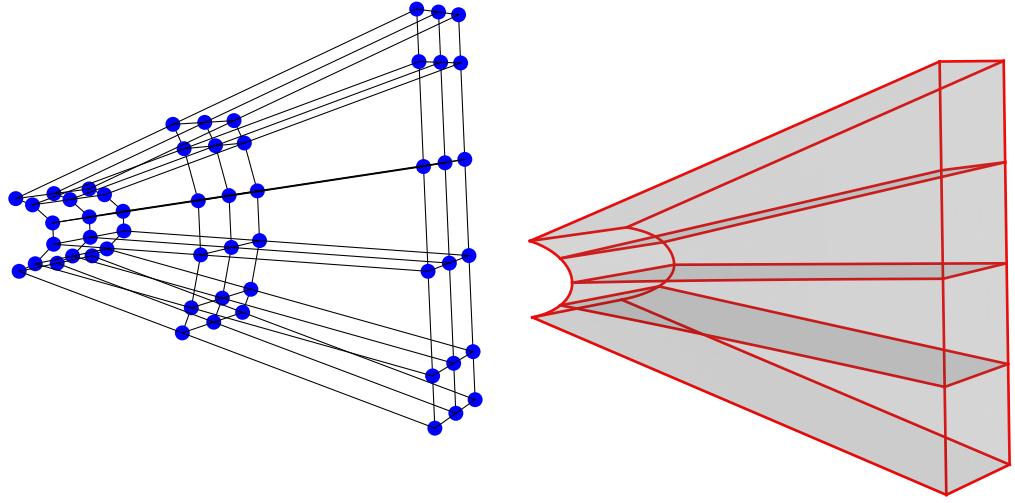
Figura 1.9 – Cilindro: Obtenção da superfície

Aqui, a superfície do domínio foi gerada discretizando-se a direção η do espaço paramétrico. Para isso, os $m + 1$ pontos de controle nessa direção foram posicionados ao longo das retas que conectam os pontos de controle da primeira camada (circunferência) aos da última camada (reta). A distribuição desses pontos seguiu uma progressão geométrica, de modo que as células menores se localizassem próximas ao cilindro, captando adequadamente os efeitos de camada limite. Para garantir que essa distribuição também se refletisse nas células mapeadas para o espaço físico, aplicou-se um fator de correção aos pontos de controle intermediários $B_{i,1}$ e $B_{i,m-1}$, para $i = 0, 1, \dots, n$, deslocando-os em direção aos pontos de controle das extremidades.

A fim de exemplificar a geração da superfície, adotou-se a discretização mínima necessária, utilizando apenas três pontos de controle na direção η , para o emprego de funções quadráticas e vetor de *knots* aberto $\mathcal{H} = [0, 0, 0, 1, 1, 1]$, conforme pode-se observar na Fig. 1.9a. Na fig. 1.9b apresentam-se as células mapeadas do espaço paramétrico para o espaço físico. Salienta-se que os pontos de controle obtidos nessa etapa foram definidos com peso unitário.

Para a simulação numérica apresentada na sequência a quantidade de pontos de controle na direção η foi definida em função da necessidade de discretização para o problema. Para um vetor de *knots* abertos com coordenadas interiores de multiplicidade unitária, a quantidade de células (*ncel*) está relacionada a quantidade de pontos de controle por $ncel = npc - deg$, sendo *npc* o número de pontos de controle e *deg* o grau das funções na direção paramétrica em questão.

Por fim, para a geração do sólido NURBS, com apenas uma camada de células na



(a) Rede de pontos de controle sólido

(b) Malha física sólido

Figura 1.10 – Cilindro: Obtenção do sólido

direção paramétrica ζ , correspondente à direção y_3 do espaço físico deste problema, foram utilizadas funções quadráticas, um vetor de *knots* aberto $\mathcal{Z} = [0, 0, 0, 1, 1, 1]$, assim como pontos de controle distribuídos uniformemente e de peso unitário. Na Fig. 1.10a pode-se observar a rede de pontos de controle resultante, na qual a nomenclatura dos pontos foi omitida para facilitar a visualização. Na fig. 1.10b apresenta-se a malha física derivada da discretização exemplificativa do *patch* P_0 .

Para os *patches* P_1 , P_2 e P_3 , utiliza-se a mesma parametrização de P_0 , obtendo-se seus pontos de controle por rotação dos pontos de controle de P_0 , de modo que cada um seja ajustado ao quadrante do cilindro correspondente.

Para a geração dos demais *patches* retangulares, P_4 à P_{11} , definiu-se a direção paramétrica ξ respectiva à direção física y_1 , η correspondente à y_2 e ζ à y_3 . A quantidade de pontos de controle em cada direção foi definida a partir do número de células desejadas para a análise numérica. Considerando o exposto anterior para o *patch* P_0 , a distribuição dos pontos de controle foi realizada de forma a se obter células físicas igualmente espaçadas, ou, arranjadas com um espaçamento que segue uma distribuição geométrica. Os pontos de controle foram definidos com peso unitário. Os vetores de *knots* são abertos e com espaçamento interior subdividido de maneira uniforme. As funções NURBS utilizadas foram quadráticas.

Destaca-se que na discretização de todos os *patches* é necessário garantir uma parametrização idêntica nos planos que apresentam fronteira com outro *patch*.

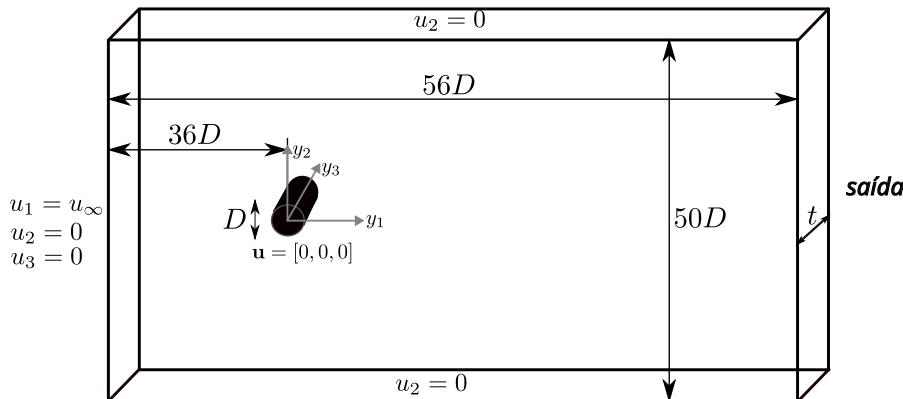
1.5.1.2 Análise numérica

Conforme exposto na Subseção ?? o estudo do problema de escoamento sobre um cilindro proporciona avaliar se o modelo computacional implementado é capaz de reproduzir

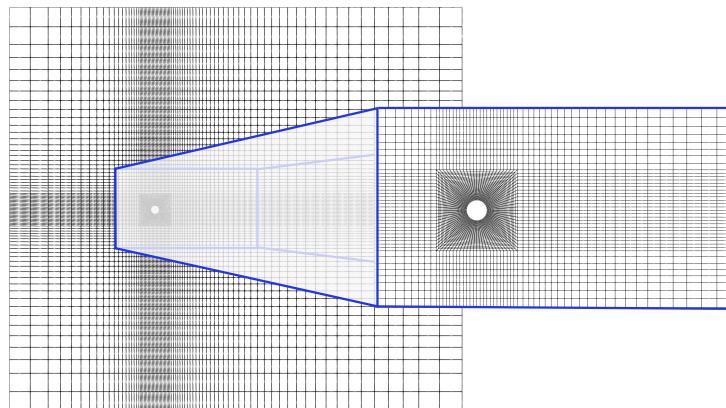
os fenômenos relacionados à formação e desprendimentos de vórtices, além de propiciar a validação do código através da comparação dos coeficientes aerodinâmicos medidos ao longo do tempo com referências bibliográficas disponíveis na literatura especializada. Visando a verificação do código de IGA com células 3D analisa-se o problema do escoamento sobre o cilindro para $Re = 40$, $Re = 100$, e, $Re = 1000$.

O domínio do problema simulado é um volume retangular discretizado em função do diâmetro do cilindro e é apresentado na Fig. 1.11a. A dimensão t na direção y_3 é equivalente à $0,01D$. Aplica-se um perfil de velocidade constante na entrada do domínio, $\mathbf{u} = [u_\infty, 0, 0]$, e condições de parede lisa são atribuídas às paredes superior e inferior, enquanto que para as frontal e dos fundos condições de simetria são aplicadas.

A malha isogeométrica utilizada é apresentada na Fig. 1.11b e na Tab. 1.1 pode-se observar a quantidade de pontos de controle em cada uma das direções paramétricas utilizados na discretização de cada um dos *patches* que compõe a malha, resultando em 30228 pontos de controle e 8728 células físicas.



(a) Cilindro: Geometria e condições de contorno



(b) Discretização espacial - plano y_1y_2

Figura 1.11 – Cilindro: Malha de células físicas

O problema é simulado para um velocidade de entrada $u_\infty = 1,0$, $\rho = 1,0$, $\Delta t = 0,05$, e $\rho_\infty = 0,5$, sendo a viscosidade variada de acordo com o número de Reynolds

Tabela 1.1 – Cilindro: Número de pontos de controle por *patch*

<i>Patch</i>	ξ	η	ζ	<i>Patch</i>	ξ	η	ζ
0	26	34	3	6	42	28	3
1	26	34	3	7	20	26	3
2	26	34	3	8	42	26	3
3	26	34	3	9	20	28	3
4	20	28	3	10	26	28	3
5	26	28	3	11	42	28	3

desejado. Calculam-se os coeficientes aerodinâmicos, C_D e C_L , a partir das definições de forças de arrasto e de sustentação apresentadas respectivamente nas Eq. ?? e Eq.???, através das seguintes equações:

$$C_D = \frac{F_D}{0,5\rho\|\mathbf{u}_\infty\|^2 Lt}, \quad (1.48)$$

$$C_L = \frac{F_L}{0,5\rho\|\mathbf{u}_\infty\|^2 Lt}, \quad (1.49)$$

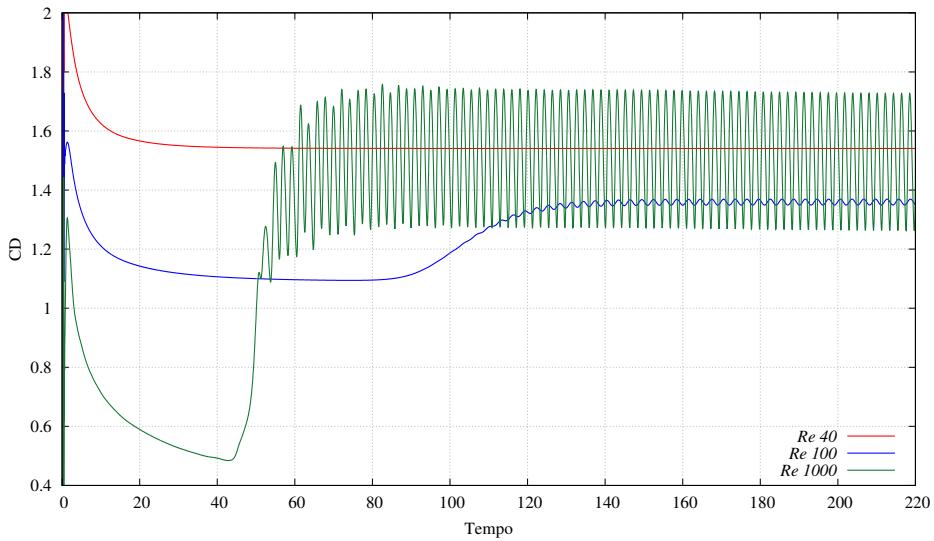
Nas Figs. 1.12a e 1.12b, apresenta-se a variação ao longo do tempo dos coeficientes aerodinâmicos C_D e C_L . Os valores do coeficiente de arrasto médio obtidos com a malha isogeométrica com células 3d foram: $C_{Dmed} = 1,54$ para $Re = 40$, $C_{Dmed} = 1,36$ para $Re = 100$ e $C_{Dmed} = 1,49$ para $Re = 1000$. Ressalta-se, que apesar dos valores de C_{Dmed} estarem bem próximos aos da simulação com MEF tradicional da Subseção ??, para as análises utilizando IGA, foram necessários mais passos de tempo para o início do processo de desprendimento de vórtices, nos casos de $Re = 100$ e $Re = 1000$.

Nas Fig. 1.13 e Fig. 1.14 são apresentados os campos de velocidade e pressão ao longo de um ciclo de desprendimento de vórtices para $Re = 1000$.

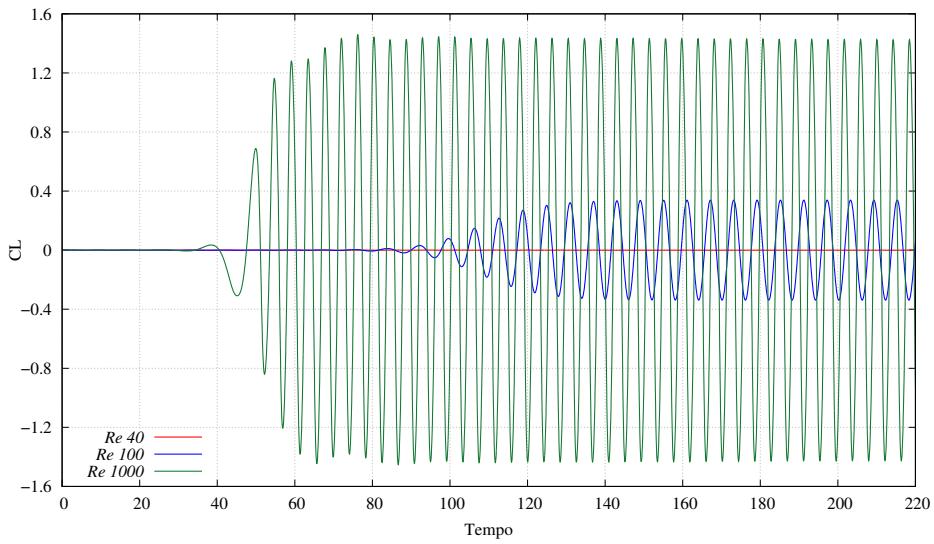
1.5.2 Escoamento em um canal com degrau

O problema de escoamento em canal com degrau consiste na aplicação de um perfil de velocidade parabólico na entrada de um canal, o qual caracteriza-se pela presença de um degrau próximo a entrada do escoamento. A existência do degrau acarreta na formação de um vórtice de recirculação, chamado aqui de vórtice primário, o qual aumenta de tamanho, a medida que se eleva o número de Reynolds do escoamento. A dimensão do vórtice primário, para diferentes números de Reynolds, será alvo de avaliação nessa seção, através da análise comparativa deste dado com bibliografias de referência.

Na Fig.1.15 apresenta-se a configuração geral da geometria do canal, que foi discretizado através do uso de células isogeométricas 3d definidas dentro de cinco *patches* que descrevem o domínio do problema. As dimensões do canal são: $h = 1,0m$, $s = 0,94m$, $x_e = 1,0m$, $x_f = 15m$ e $x_t = 30m$. A profundidade do canal adotada, na direção y_3 , é



(a) Coeficiente de arrasto C_D



(b) Coeficiente de sustentação C_L

$\text{Re } 40$ — $\text{Re } 100$ — $\text{Re } 1000$ —

Figura 1.12 – Cilindro: Coeficientes aerodinâmicos

equivalente a $0,1\text{m}$. Devido a pequena dimensão definida em y_3 a simulação numérica realizada é caracterizada como um escoamento em domínio bidimensional.

Para a geração da geometria NURBS, os 5 *patches*, denominados P_0, P_1, P_2, P_3 e P_4 , são discretizados em todas as direções paramétricas com funções base quadráticas e com vetores de *knots* abertos de coordenadas paramétricas igualmente espaçados em seu interior. Os pontos de controle para os *patches* P_0, P_1 e P_2 foram distribuídos no espaço físico, direções y_1, y_2 e y_3 , de maneira a se obter células igualmente espaçadas. Ressalta-se, que o número de células está relacionado a quantidade de pontos de controle por $ncel = npc - deg$. Para os *patches* P_3 e P_4 , na direção do espaço físico y_2 e y_3 , os pontos de controle são posicionadas de maneira a gerar células uniformes, e, na direção y_1 ,

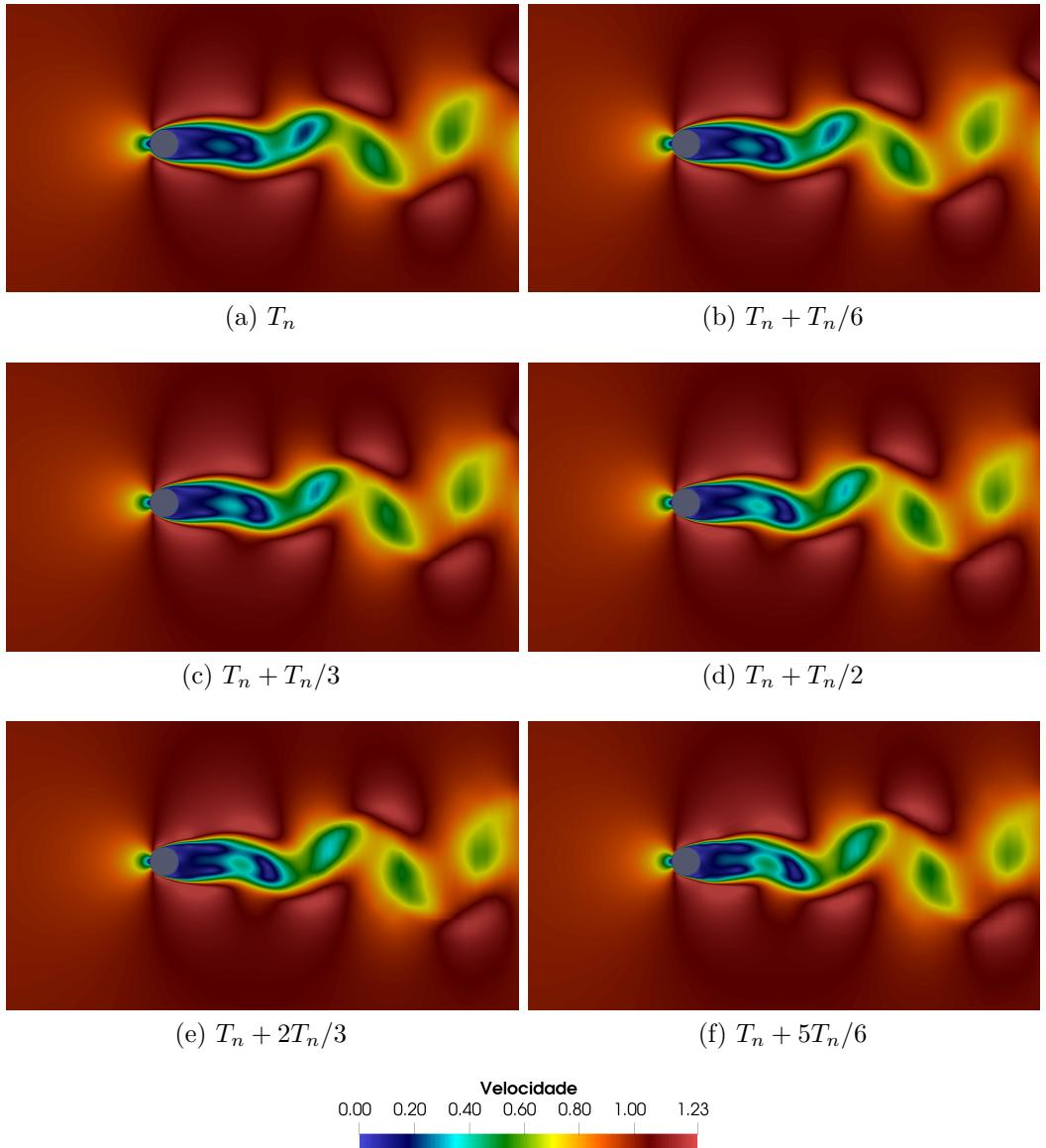


Figura 1.13 – Cilindro: Campos de velocidade para $Re = 1000$ - plano y_1y_2

são distribuídos de maneira a se obter células em progressão geométrica, aumentando de tamanho no sentido do contorno de saída, conforme pode ser observado na Fig. 1.16. Na Tab. 1.2 apresenta-se o número de pontos de controle em cada direção paramétrica para cada *patch*, resultando em 17640 pontos de controle e 4800 células.

Para a simulação numérica, aplicou-se condição de aderência nos contornos contidos nos planos y_1y_3 e y_3y_2 , exceto naqueles com condições de entrada ou de saída. Nos contornos contidos nos planos y_1y_2 aplicou-se condição de simetria. O perfil parabólico adotado na entrada do domínio é descrito pela seguinte equação:

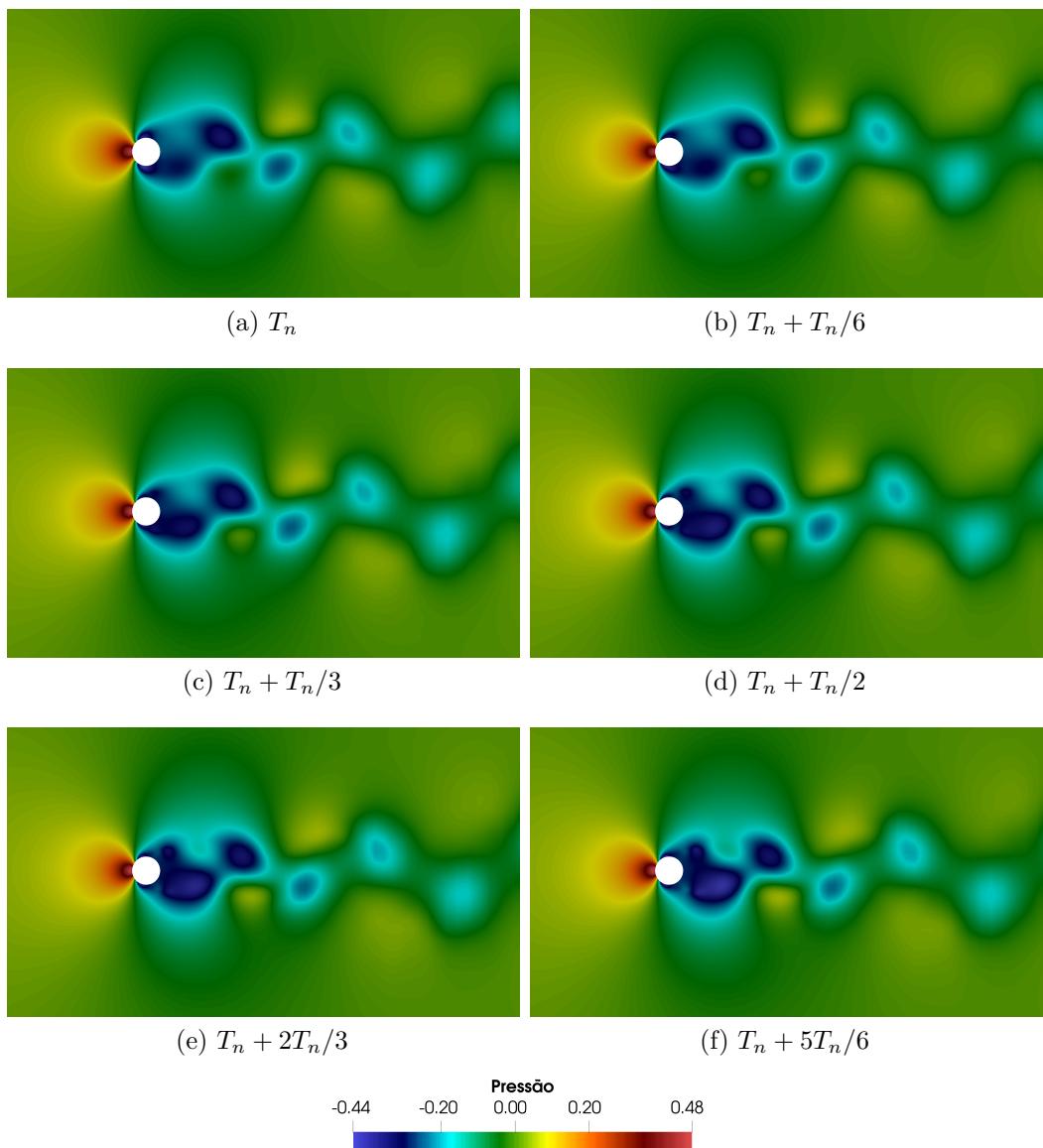


Figura 1.14 – Cilindro: Campos de pressão para $Re = 1000$ - plano y_1y_2

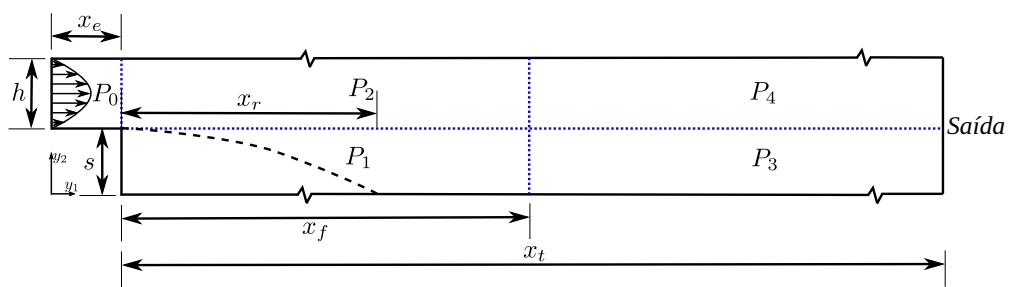


Figura 1.15 – Degrau: Geometria - plano y_1y_2

$$u_1 = V_{max} \left(1 - \left(\frac{(y_2 - s) - h/2}{h/2} \right)^2 \right), \quad (1.50)$$

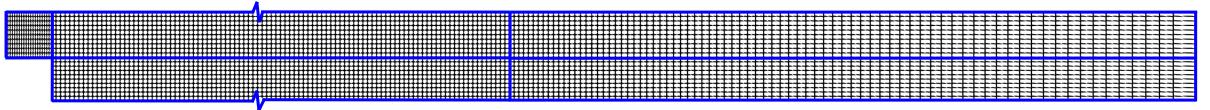


Figura 1.16 – Degrau: Malha de células físicas - plano y_1y_2

Tabela 1.2 – Degrau: Número de pontos de controle por *patch*

<i>Patch</i>	ξ	η	ζ
0	22	12	3
1	152	12	3
2	152	12	3
3	82	12	3
4	82	12	3

com velocidade $V_{max} = 10m/s$ e $u_2 = u_3 = 0$ nesse contorno.

O escoamento sobre o degrau é caracterizado por produzir áreas de recirculação onde o fluido se separa e forma vórtices. A distância entre o degrau e o ponto de recolamento do vórtice principal, x_r , é uma das principais características verificadas nesse problema. A dimensão dos vórtices varia em função do número de Re, o qual é calculado de acordo com ??), por:

$$Re = \frac{\rho \left(\frac{2V_{max}}{3} \right) 2h}{\mu}. \quad (1.51)$$

Foram selecionados três diferentes número de Reynolds para as análises: $Re = 100$, $Re = 400$ e $Re = 800$, os quais são obtidos a partir da variação da viscosidade do fluido. Considerou-se $\rho = 1kg/m^3$, $\Delta t = 0,05s$, e $\rho_\infty = 0,5$.

De acordo com os experimentos realizados por ??), as medições do comprimento do vórtice primário, logo a jusante do degrau na parte inferior, identificam o regime do escoamento como laminar ($Re < 1200$), transiente ($1200 < Re < 6600$) e turbulento ($Re > 6600$). Além disso, o autor constatou em seus ensaios que para $Re < 400$ o escoamento é predominantemente bidimensional, enquanto que para Reynolds superiores, o escoamento apresenta regiões de comportamento tridimensional.

??) em suas análises constatou o surgimento de uma bolha de separação adicional ao longo do piso do canal, a jusante da separação primária, a qual desaparece para $Re > 2300$. Outra região de separação secundária também foi observada ao longo da parede superior, a jusante do degrau, desenvolvendo-se a partir de $Re 400$ e permanecendo durante todo o regime de transição.

Na Fig. 1.17 são apresentados os comprimentos de recolamento do vórtice primário adimensionalizados (x_r/s) obtidos nesse trabalho, juntamente com os valores adaptados dos ensaios experimentais de ??) e os resultados de análises 2d e 3d de ??). Com

essa figura é possível observar que a medida que o número de Reynolds aumenta, os resultados obtidos do presente estudo se afastam dos valores de referência respectivos ao estudo experimental e da simulação tridimensional. Este fato ocorre visto que o ensaio experimental foi realizado em um canal com $2m$ profundidade na direção y_3 , enquanto que a simulação atual conta com apenas uma célula nessa direção, sendo então incapaz de captar os fenômenos tridimensionais que ocorrem a medida que o número de Reynolds cresce.

Na Fig. 1.18 pode-se observar o campo de velocidade para o domínio completo e para a região do vórtice primário para $Re = 100$, enquanto que na Fig. 1.19 e na Fig. 1.20 são apresentados para $Re = 400$ e $Re = 800$ respectivamente. Para $Re = 800$ constatou-se a formação de um vórtice secundário na parede superior, conforme havia sido relatado pelos autores previamente citados, conforme pode ser observado na Fig. 1.20c.

Por fim, na Fig. 1.21 é possível observa-se os campos de pressão para todos os números de Reynolds simulados.

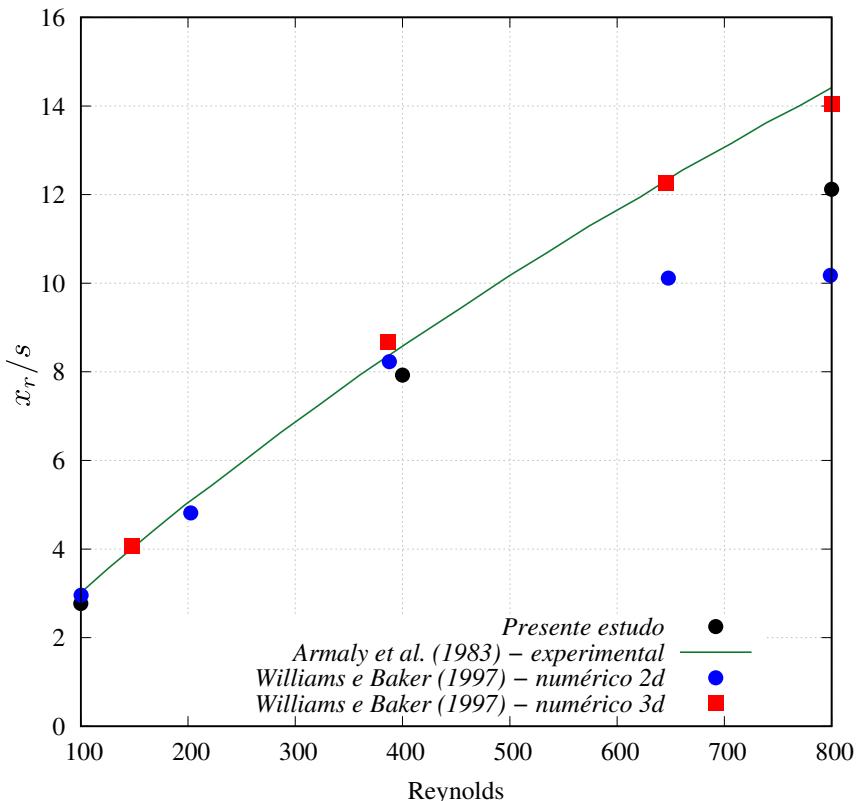
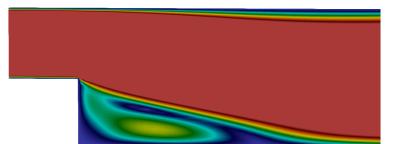


Figura 1.17 – Degrau: Comprimento do vórtice principal



(a) Domínio Completo

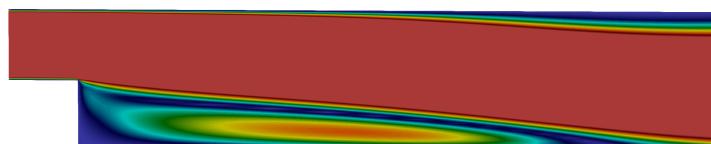


(b) Vórtice primário

Figura 1.18 – Degrau: Campos de velocidade para $Re = 100$

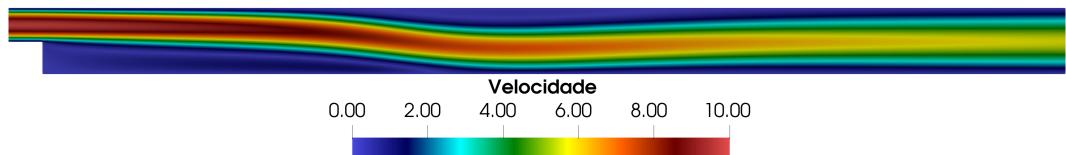


(a) Domínio Completo

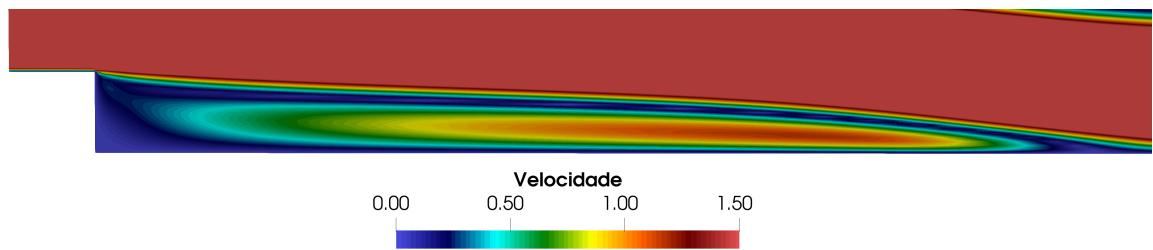


(b) Vórtice primário

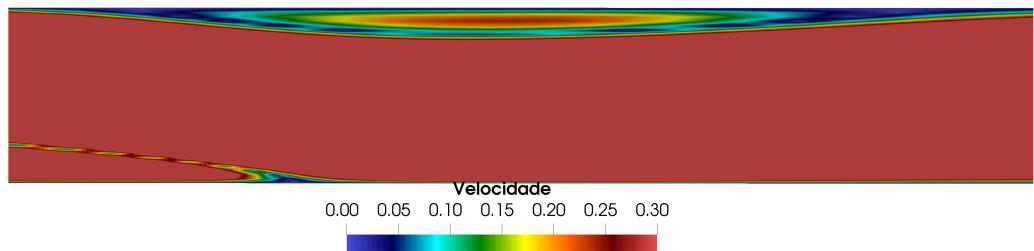
Figura 1.19 – Degrau: Campos de velocidade para $Re = 400$



(a) Domínio Completo



(b) Vórtice primário



(c) Vórtice secundário

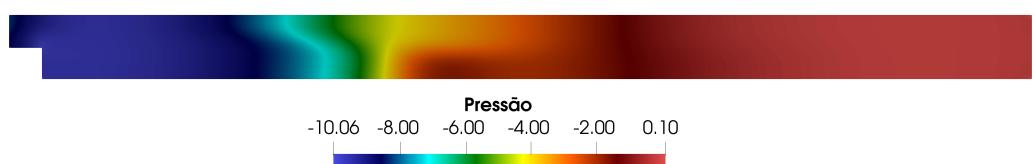
Figura 1.20 – Degrau: Campos de velocidade para $Re = 800$



(a) $\text{Re} = 100$



(b) $\text{Re} = 400$



(c) $\text{Re} = 800$

Figura 1.21 – Degrau: Campos de pressão