
ANÁLISE ISOGEOMÉTRICA APLICADA À MECÂNICA DOS FLUIDOS

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 clássicos 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 as mais aplicadas (ver, por exemplo, ??)). O grande impulso para o desenvolvimento da técnica foi proporcionar a integração entre a engenharia de *design*, 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.

Além disso, a IGA apresenta vantagens significativas, uma vez que permite a representação exata de diversas geometrias comuns, como seções cônicas, círculos, cilindros, esferas e elipsoides, 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 — com continuidade $p - 1$ nas interfaces entre elementos, sendo p o grau das funções —, a alta capacidade de aproximação e a possibilidade de refinamento local por meio da inserção de *knots*, que correspondem às coordenadas do espaço paramétrico nas quais as funções são definidas.

REVER ESSE TRECHO QUANDO TERMINAR O CAPÍTULO. Nesse capítulo apresenta-se uma breve introdução sobre a IGA, o processo de obtenção das geometrias

NURBS e seu uso na descrição das variáveis discretas nas simulações numéricas. As referências bibliográficas que fundamentam esta construção são ??) e ??). Por fim, são apresentados alguns exemplos de sua aplicação em problemas da DFC.

1.1 Noções Gerais de IGA

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 paramétrico. 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.

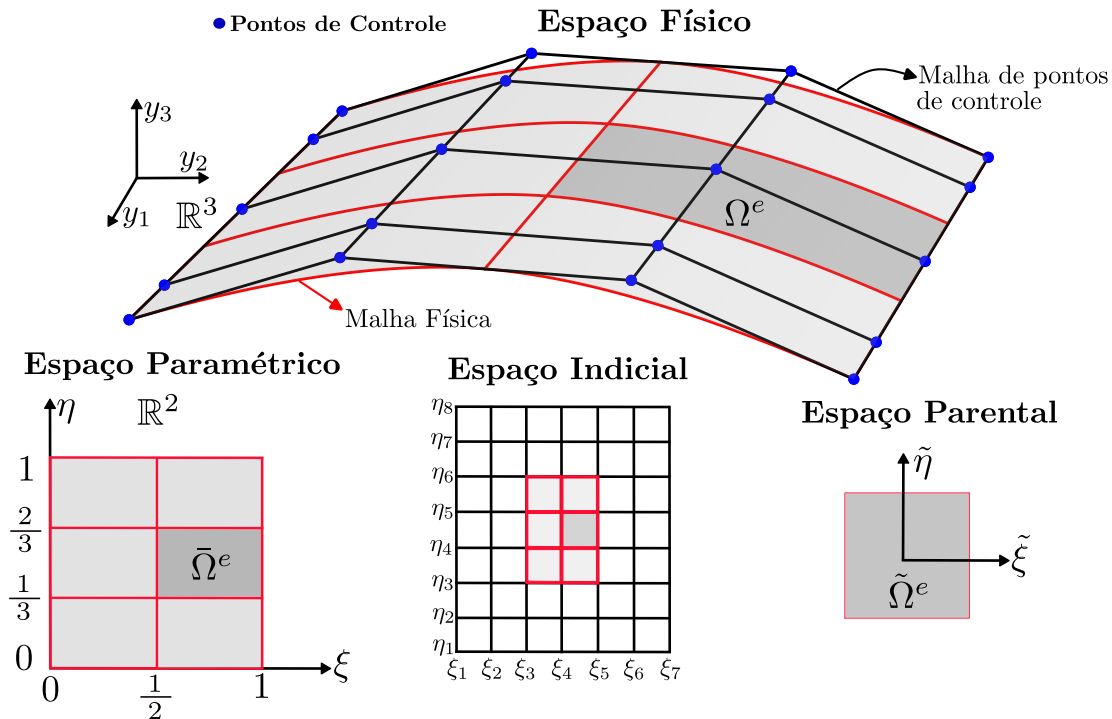


Figura 1.1 – NURBS: espaço físico, espaço paramétrico, espaço indicial e espaço parental

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

desse 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 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 uma geometria NURBS, 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 *knots* definem células no espaço paramétrico, cujos contornos são mapeados pelas funções base para formar a malha no espaço físico.

O vetor de *knots* pode ser 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*, 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 $p \geq 1$ são definidas como:

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

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

Essas equações são conhecidas como a fórmula recursiva de *Cox-de Boor* (????). Para funções *B-Splines* de grau $p = 0$ ou $p = 1$, obtêm-se, respectivamente, as mesmas funções constantes e lineares por partes utilizadas no método dos elementos finitos padrão.

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*.

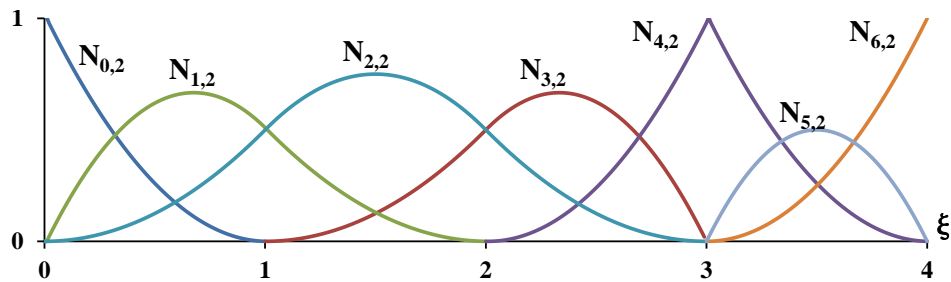


Figura 1.2 – *B-Splines* quadráticas

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:

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

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

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

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

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

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

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 as componentes do vetor de coordenadas físicas \mathbf{y} . Utilizando as funções *B-Splines* apresentadas na Fig.1.2 e uma malha de 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.

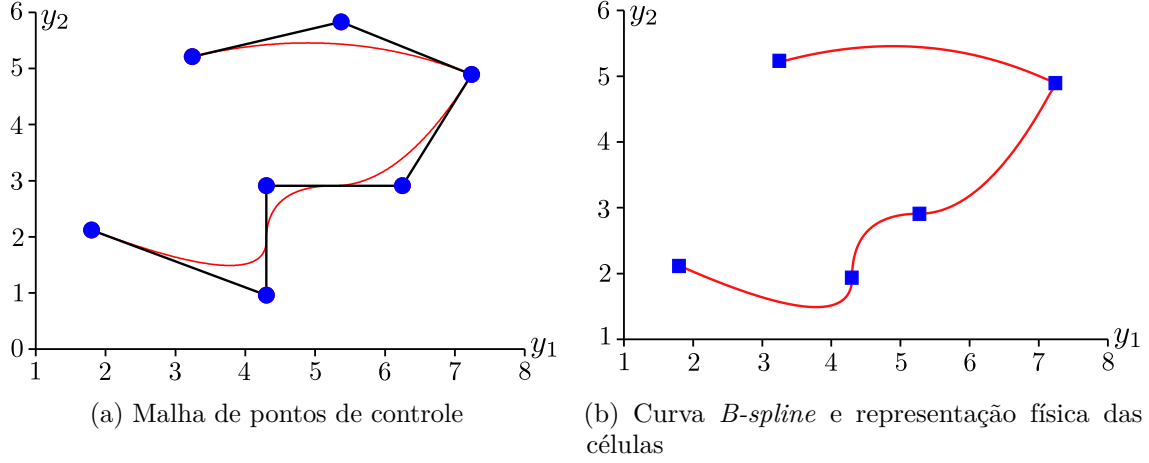


Figura 1.3 – Curva *B-Spline*

A partir da Fig.1.3a pode-se constatar que a curva *B-Spline* interpola o primeiro e o último ponto de controle, que é uma característica das curvas construídas com funções descritas a partir de vetores de *knots* abertos. Adicionalmente nota-se que, 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 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 ??).

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: $\forall(\xi, \eta) \text{ in } [\xi_0, \xi_1, \dots, \xi_{p+n+1}] \times [\eta_0, \eta_1, \dots, \eta_{q+m+1}]$:

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

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, será adotado na geração das geometrias o refinamento h , baseado na inserção de *knots*. Por essa razão, somente essa estratégia será abordada ao longo desse texto.

O enriquecimento das funções base utilizando a inserção de *knots* é realizado sem que se altere uma curva geometricamente ou parametricamente. Para essa finalidade, considerando o vetor de *knots* $\Xi = [\xi_0, \xi_1, \dots, \xi_{n+p+1}]$, será introduzido o conceito de vetor

de *knots* estendido, o qual compreende em: $\bar{\Xi} = [\bar{\xi}_0 = \xi_0, \bar{\xi}_1, \dots, \bar{\xi}_{n+m+p+1} = \xi_{n+p+1}]$. As $(n + m + 1)$ novas funções de base *B-Splines* são determinadas através da Eq. 1.1 e Eq. 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:

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

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

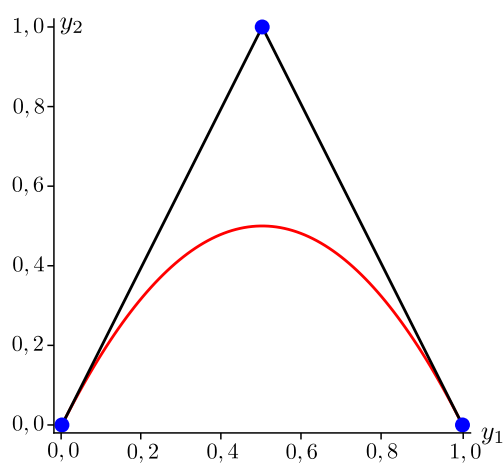
sendo $i = 0, 1, \dots, (n + m)$ e $j = 0, 1, \dots, n$.

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. Essa curva, possui apenas uma célula no espaço físico, conforme pode ser observado na Fig. 1.4c, e 3 funções base no espaço paramétrico (Fig. 1.4e). Ao realizar-se 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.

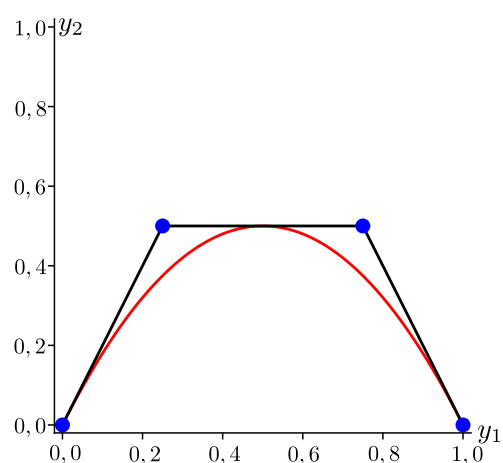
Para fins práticos, o processo de refinamento consiste na inserção consecutiva de coordenadas paramétricas ao vetor de (*knots*) até que se alcance a discretização desejada. Um algoritmo mais eficiente para realizar esse procedimento de refinamento pode ser encontrado em ??). Esse procedimento pode ser aplicado analogamente à superfícies e sólidos, aplicando-se a inserção de *knots* nas direções paramétricas desejadas.

1.3 NURBS

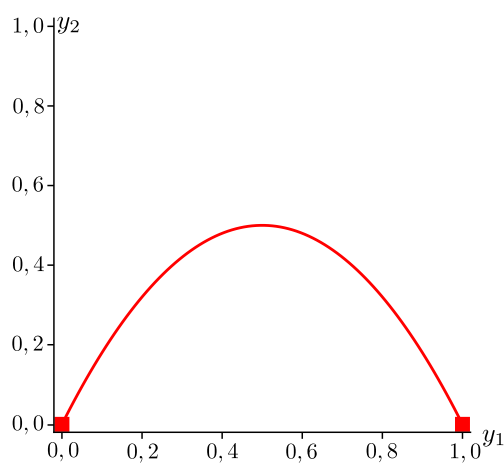
Uma geometria NURBS no $\mathbb{R}^{n_{sd}}$ pode ser entendida, do ponto de vista geométrico, como a transformação projetiva de uma geometria *B-Spline* no $\mathbb{R}^{n_{sd}+1}$. Nesse contexto, geometrias cônicas podem ser construídas exatamente através de curvas quadráticas por partes. Na Fig. 1.5b, 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



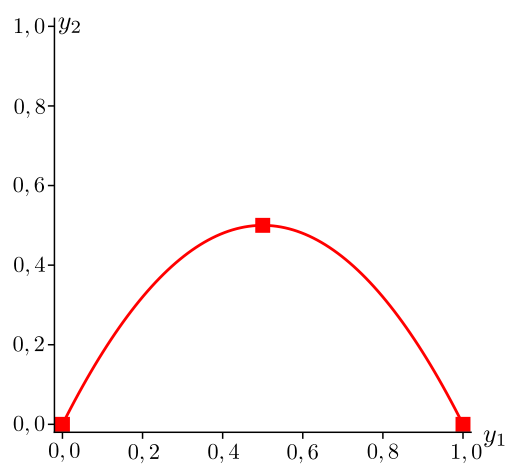
(a) Curva original e pontos de controle



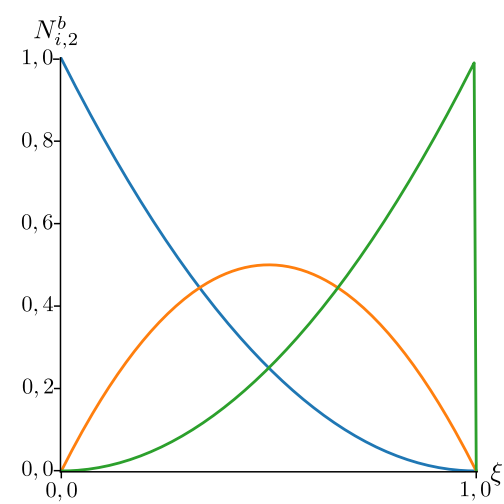
(b) Curva refinada e pontos de controle



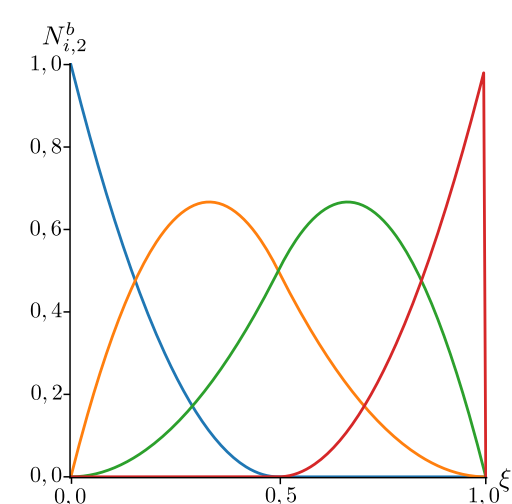
(c) Célula curva original



(d) Células curva refinada



(e) Funções base originais



(f) Funções base após refinamento h refinada

Figura 1.4 – Refinamento h para um curva B -Spline

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.

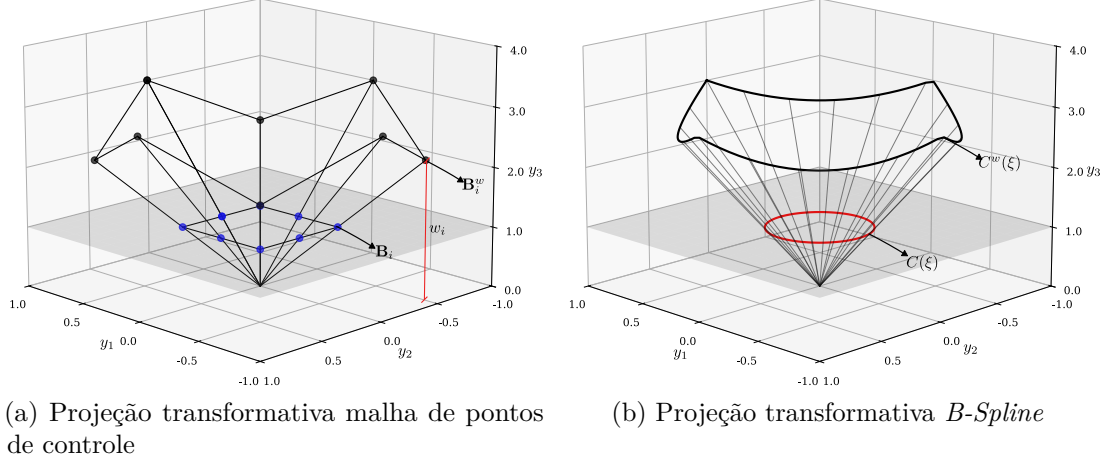


Figura 1.5 – Projeção transformativa de entidade *B-Spline*

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:

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

e a curva NURBS pode ser definida como:

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

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 racionalização de uma função *B-Spline*. A racionalização dessa função ocorre através da razão entre dois polinômios.

Uma função racional NURBS (R) é construída através da seguinte expressão, dessa forma, têm-se:

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

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 para descrever um domínio computacional o uso de múltiplos *Patches* NURBS, isto se deve ao fato que o produto tensorial do espaço paramétrico não é adequado para a representação de domínios complexos multiplamente conectados. Ademais, mesmo para domínios simples, do ponto de vista da simulação numérica, o uso de múltiplos *patches* pode ser adequado para discretização das geometrias, conforme será visto na seção de exemplos.

??) cita ainda que o uso de múltiplos *patches* pode facilitar a análise numérica quando diferentes materiais e modelos físicos são utilizados em diferentes partes do domínio. E, além disso, em processamento paralelo, pode se tornar conveniente, do ponto de vista de estruturas de dados, não ter um único *patch* entre diferentes processadores.

A utilização de múltiplos *patches* implica na compatibilização da 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 outra face. Esses pontos iguais 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 terão continuidade C_0 , conforme pode ser observado na Fig. 1.6, onde apresentam-se as funções base univariadas na interface entre dois *patches*.

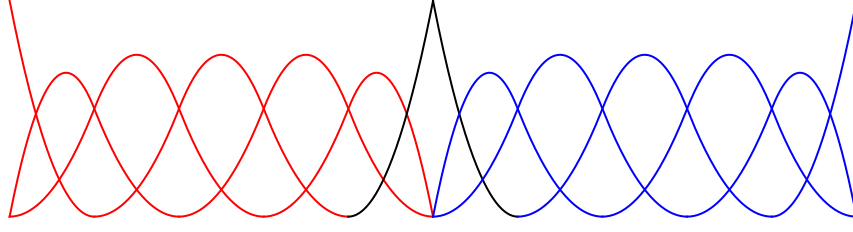


Figura 1.6 – Funções base univariadas na interface entre *Patches*

1.4 Análise Isogeométrica

Para a aplicação da IGA no contexto da Dinâmica dos Fluidos Computacional, será utilizada como base a metodologia apresentada no Cap. ???. Nesse contexto, a aproximação da geometria, realizada no contexto do MEF pela Eq. ??, será 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 Eq. (??) à Eq. (??) como N , são equivalentes à $R_{i,p}(\xi)$, $R_{i,j;p,q}(\xi, \eta)$ e $R_{i,j,k;p,q,r}(\xi, \eta, \zeta)$ a depender da geometria em análise.

A integração numérica nas células é realizada 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 $\boldsymbol{\xi}(\xi, \eta, \zeta)$ e $\tilde{\boldsymbol{\xi}}(\tilde{\xi}, \tilde{\eta}, \tilde{\zeta})$, 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:

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

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-se 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:

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

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

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

assim, $\frac{d\xi}{d\tilde{\xi}}$ é obtido derivando-se parcialmente às expressões apresentadas em: Eq. 1.34, Eq. 1.35 e Eq. 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 necessário a determinação de um tensor métrico, \mathbf{G} (Eq. ??), o qual depende da matriz jacobiana transformada, $\hat{\mathbf{Q}}$, definida na Eq. ??.

Devido 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} será 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 do elemento 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 elemento 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 do elemento de Bézier.

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

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

com $a = 0, \dots, p$. \mathbf{C} consiste no operador de extração de Bézier, que relaciona as funções

B-spline globais às funções de Bernstein locais, cuja obtenção, nesse trabalho, é realizada de acordo com o exposto em ??).

O comprimento efetivo de elemento 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 do elemento de Bézier e o comprimento efetivo do elemento. Considerando um problema 1D, uma das proposta dos autores para D , utilizada nesse 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.

VOLTAR AQUI!

1.5 Verificação e aplicações

Para aplicação da IGA em problemas da DFC seguiu-se o mesmo procedimento matemático descrito ao longo do Cap. ??, sendo que a implementação computacional seguiu o roteiro apresentado no Alg. ?. Os exemplos escolhidos para a verificação do código computacional foram o escoamento sobre um cilindro 3D e o problema de escoamento sobre um canal com degrau utilizando células 3D. Os resultados obtidos são apresentados nas seções subsequentes.

1.5.1 Escoamento sobre um cilindro - 3D

Na geração da geometria NURBS, correspondente ao problema do escoamento sobre um cilindro, utilizou-se um código previamente desenvolvido pela estudante durante

seu mestrado (??). Por tratar-se de uma geometria de pequena complexidade, pôde-se gerá-la com um único *patch*, o qual é composto por um cubo com um cilindro inserido em seu centro. O processo de geração da malha, simplificada, consiste em se escolher vetores de *knots*, pontos de controle, e pesos adequados para a descrição de tal geometria.

O código previamente desenvolvido, baseia-se na quantidade mínima de pontos de controle necessários para gerar uma circunferência completa (ver Fig. 1.7a). Para a obtenção exata de uma circunferência utilizam-se funções quadráticas e o vetor de *knots* na direção paramétrica ξ inicial é composto por: $\xi = [0, 0, 0, 1/4, 1/4, 1/2, 1/2, 3/4, 3/4, 1, 1, 1]$. A posição dos pontos de controle da circunferência e seus respectivos pesos foram obtidos de acordo com ??).

A obtenção da curva respectiva ao quadrado, que é descrita no espaço paramétrico correspondente a direção ξ , é consequência das escolhas realizadas para gerar a circunferência, logo, a curva possui o mesmo vetor de *knots* e funções quadráticas. Os pontos de controle para a formação do quadrado foram posicionados, no espaço físico, de forma a ser obtida a dimensão requerida à seção transversal quadrada que compõe a geometria do cubo, e de forma que os mesmos ficassem alinhados com os pontos de controle da circunferência na direção radial. Os pesos respectivos aos pontos de controle do retângulo são unitários. Na Fig. 1.7a pode-se observar a posição dos pontos de controle respectivos à circunferência e ao quadrado e as curvas resultantes desta discretização em linha vermelho pontilhado.

Na sequência o código realiza o procedimento de refinamento por inserção sucessiva de *knots* no vetor de *knots*. O algoritmo utilizado para este procedimento pode ser encontrado em ??). Na Fig. 1.7b apresenta-se um exemplo de pontos de controle gerados após a inserção de um novo *knot* no centro de cada *knot span* do espaço paramétrico inicial. A quantidade de *knots* a ser inserida depende da discretização necessária a análise numérica.

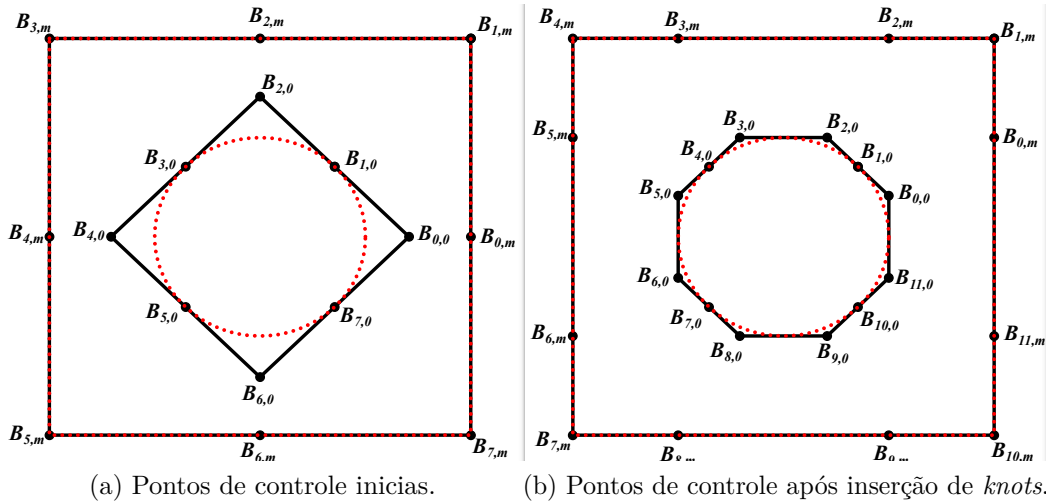


Figura 1.7 – Cilindro 3D: Geração curva NURBS

Para a obtenção de uma seção transversal da geometria em questão, gera-se uma superfície a partir da discretização na direção η do espaço paramétrico. O código utiliza um vetor de *knots* aberto, com os *knots* distribuídos uniformemente, e funções de forma quadráticas. Os pontos de controle são distribuídos radialmente no espaço físico através de uma progressão geométrica unidirecional e seus pesos são determinados a partir de uma interpolação linear entre os pesos dos pontos de controle da circunferência e os pontos do quadrado. A quantidade mínima de pontos de controle respectiva à direção η é de $q + 1$ pontos, sendo a quantidade final definida em função da análise numérica. Na Fig. 1.5.1, apresenta-se um exemplo de uma rede de pontos de controle, obtida a partir das curvas apresentadas em Fig. 1.7b com 9 pontos de controle na direção η . Na figura em questão omitiu-se a nomenclatura dos pontos de controle intermediários às curvas da circunferência e do quadrado para evitar a sobreposição da nomenclatura na figura.

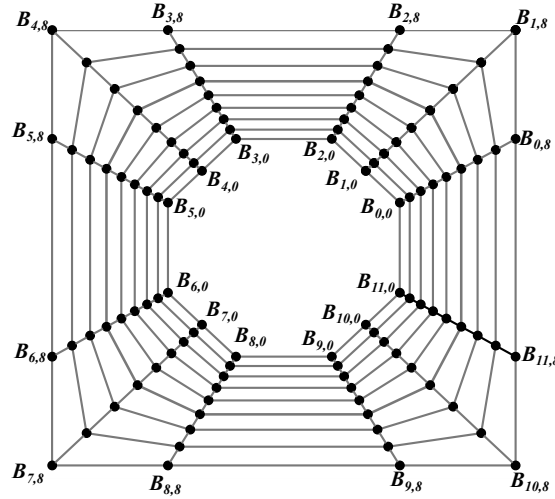


Figura 1.8 – Cilindro 3D: Geração superfície NURBS.

Nessa primeira etapa do trabalho, a direção paramétrica ζ , respectiva à direção z da geometria física, foi discretizada com apenas uma célula, utilizando-se para isso vetor de *knots* abertos com distribuição uniforme de *knots*, e funções base quadráticas.

Visando a verificação do código de IGA 3D analisa-se o problema do escoamento sobre o cilindro para $Re = 40, 100$ e 1000 (Eq. (??)). Para isso, gera-se uma malha com $101 \times 71 \times 3$ pontos de controle nas direções paramétricas ξ , η e ζ respectivamente, resultando em 6624 células. Na Fig. 1.9a apresentam-se as dimensões da geometria em questão, bem como as condições de contorno aplicadas, e na Fig. 1.9b apresenta-se a malha física resultante da discretização.

Adicionalmente, prescrevem-se nas paredes frontal e posterior, condição de parede lisa ($u_z = 0$), e na Saída do escoamento condição de força de superfície nula ($\sigma \mathbf{n} = \mathbf{0}$). 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 desejado. Para

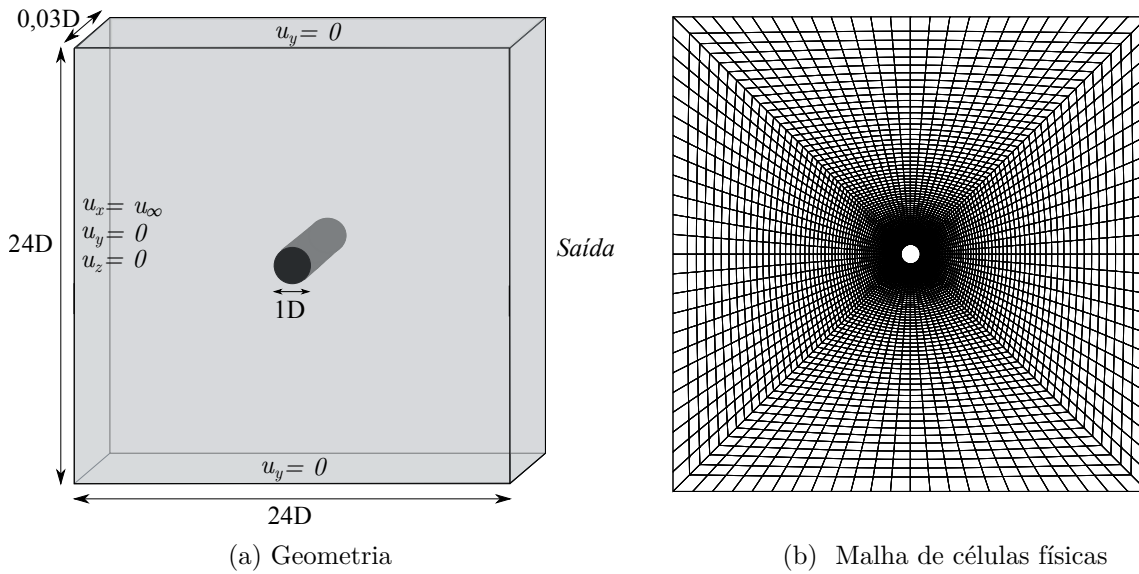


Figura 1.9 – Cilindro 3D: Geometria e malha de células físicas

o cálculo dos coeficientes aerodinâmicos C_D , C_L e do número de Strouhal (St) utilizam-se as equações apresentadas no Item ??.

Nas Figs. 1.10a e 1.10b, apresentam-se a variação ao longo do tempo dos coeficientes C_D e C_L . Os valores obtidos com a malha isogeométrica 3D estão muito próximos ao obtidos com a malha de elementos finitos 2D (Item ??) para Reynolds 40 e 100. Para Reynolds 1000, nota-se que a variação ao longo do tempo dos valores de C_D e C_L para IGA resulta em valores mais elevados do que os obtidos com MEF. Tal diferença ainda deverá ser investigada, sendo que tanto as diferenças nas dimensões das malhas como a diferença que pode haver na convergência dos resultados ou a efeitos de 3D de vorticidade podem contribuir para isso. Os resultados obtidos para o histórico de C_D e C_L por ??) para $Re = 1000$ em análises tridimensionais baseadas em MEF, foram menores do que os 2D. Essa disparidade entre os resultados obtidos nesse trabalho e de ??) podem estar relacionados com o fato de que a dimensão escolhida da malha na direção z foi muito pequena de maneira a impossibilitar que os efeitos tridimensionais do escoamento fossem adequadamente capturados.

Para o número de Strouhal, o valor obtido para $Re = 100$ foi de 0,1681 e para $Re = 1000$ de 0,2395. Nota-se que, embora os valores de C_D e C_L apresentem diferenças entre os resultados 2d baseados em elementos finitos e a malha 3D baseada em IGA, a frequência do desprendimento de vórtices obtida foi muito semelhante.

1.5.2 Escoamento em um canal com degrau

Este exemplo é amplamente utilizado na verificação de códigos para escoamentos incompressíveis, sendo sua geometria apresentada na Fig. 1.11. O problema consiste

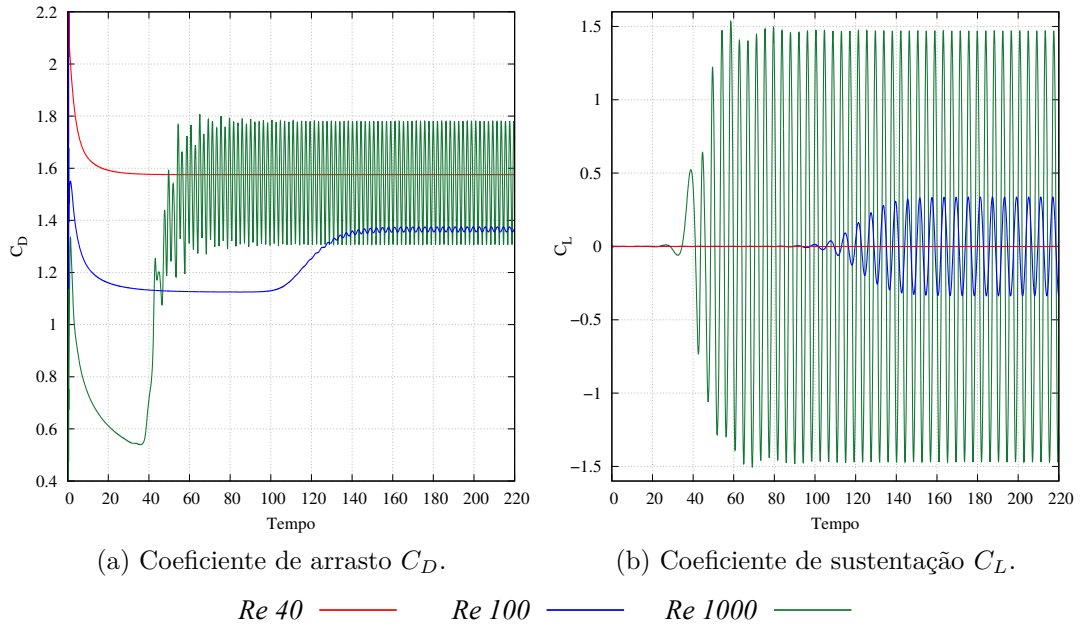


Figura 1.10 – Cilindro 3D: Coeficientes aerodinâmicos.

em prescrever-se um perfil parabólico de escoamento na entrada do canal, e condição de aderência ($\mathbf{u} = 0$) nas demais paredes que estão contidas nos planos xz e yz , exceto na saída do canal, a qual possui como condição $\sigma \mathbf{n} = \mathbf{0}$. Para as paredes dos planos xy , frontal e posterior, prescreveu-se condição de parede lisa ($u_z = 0$).

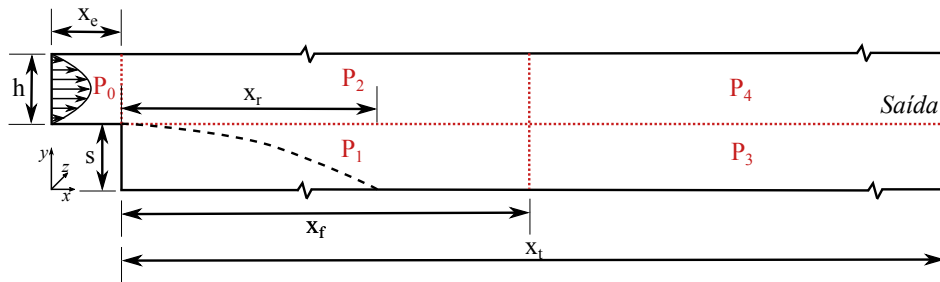


Figura 1.11 – Degrau 3D: Geometria.

As dimensões seleccionadas para o canal foram $h = 1,0m$, $s = 0,94m$, $x_e = 1,0m$, $x_f = 15m$ e $x_t = 30m$ e dimensão na direção z de $0,1m$. Adicionalmente, o perfil de velocidade na entrada do canal é descrito pela seguinte relação:

$$u_x = V_{max} \left(1 - \left(\frac{(y - s) - h/2}{h/2} \right)^2 \right), \quad (1.48)$$

com velocidade $V_{max} = 10m/s$ e $u_y = u_z = 0$.

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, a qual é calculada de acordo com ??), sendo expressa por:

$$\text{Re} = \frac{\rho \left(\frac{2V_{max}}{3} \right) 2h}{\mu}, \quad (1.49)$$

com $\rho = 1kg/m^3$. Foram selecionados para as análises 3 diferentes número de Reynolds: 100, 400 e 800, variando-se a viscosidade do fluido.

Para a geração da geometria NURBS, discretiza-se o canal em 5 *patches*, os quais são denominados P_0, P_1, P_2, P_3 e P_4 , e podem ser observados na Fig. 1.11. Todas as direções paramétricas são discretizadas com vetores de *knots* abertos e com *knots* igualmente espaçados no interior do vetor, além de funções de forma quadráticas. Os pontos de controle para os *patches* 0, 1 e 2 foram distribuídos no espaço físico, direções x, y e z de maneira a se obter células igualmente espaçados. Para os *patches* 3 e 4, na direção do espaço físico y e z , os pontos são posicionadas de maneira a gerar células uniformes, e, na direção x , são distribuídos de maneira a se resultar numa progressão geométrica do tamanho das células, com as células aumentando de tamanho da esquerda para a direita, conforme pode ser observado na Fig. 1.12. Na Tab. 1.1 podem ser observados os números de pontos de controle utilizados em cada direção dos espaços paramétricos para cada *patch*, resultando em 60795 pontos de controle e 4800 células.

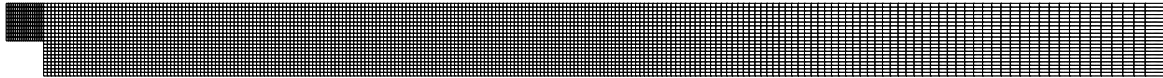


Figura 1.12 – Degrau 3D: Geometria e malha de células físicas

Tabela 1.1 – 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

Na Fig. 1.13 são apresentados os comprimentos de recolamento do vórtice primário adimensionalizados (x_r/s), juntamente com os resultados adaptados dos ensaios experimentais de ??) e os resultados de análises 2d de ??). Nota-se que os resultados obtidos

estão próximos das referências para $Re = 100$ e $Re = 400$, entretanto, para $Re = 800$ nota-se um afastamento do presente trabalho, e do referente à análise 2D com relação ao experimento realizado por ??). Isto ocorre, visto que o ensaio experimental foi realizado com um canal com $2m$ de comprimento na direção z , e a simulação atual com apenas uma célula nessa direção é incapaz de captar os fenômenos tridimensionais que ocorrem a medida que o número de Reynolds cresce. Na Fig.1.14 pode-se observar o campo de velocidade para os Reynolds estudados, e o aspecto do vórtice primário desenvolvido.

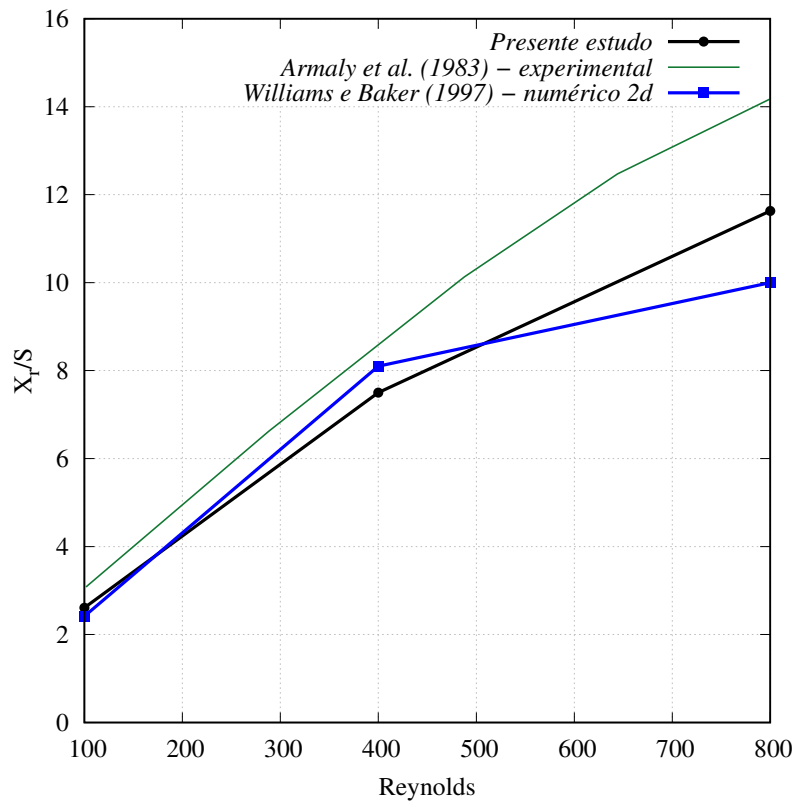


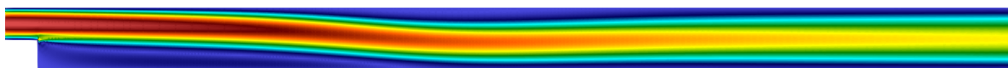
Figura 1.13 – Degrau 3D: Comprimento de recolamento do vórtice principal



(a) $Re = 100$



(b) $Re = 400$



(c) $Re = 800$

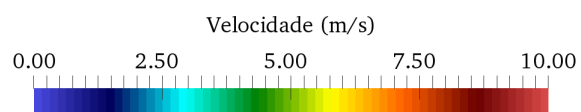


Figura 1.14 – Degrau 3D: Campo de velocidade.