



GERAÇÃO DE MALHAS ISOGEOMÉTRICAS UTILIZANDO MAPEAMENTO TRANSFINITO

Elias Saraiva Barroso

Joaquim Bento Cavalcante Neto

Creto Augusto Vidal

elias.barroso@gmail.com

joaquimb@dc.ufc.br

cvidal@lia.ufc.br

Computer Graphics, Virtual Reality and Animation Group. Universidade Federal do Ceará.

Evandro Parente Junior

evandro@ufc.br

Laboratório de Mecânica Computacional e Visualização, Universidade Federal do Ceará.

Campos do Pici, 60440-900, Fortaleza, Ceará, Brasil.

Abstract. A análise isogeométrica (AIG) é um método numérico recente que considera na sua formulação a geometria exata do problema. Existem formulações da AIG baseadas em NURBS e T-Splines. Recentemente, também estão sendo estudadas formulações que utilizam patches de Bézier triangulares ou definidos por produto tensorial. Neste caso, os elementos isogeométricos gerados apresentam continuidade C_0 . Em problemas planos representados por modelos de fronteira (B-Reps), onde são utilizadas curvas NURBS para descrição das fronteiras, pode ser difícil a obtenção de elementos isogeométricos definidos por patches NURBS. Por outro lado, pode-se aplicar técnicas de geração de malhas estruturadas para obtenção de elementos isogeométricos de Bézier. Neste trabalho, são estudadas e aplicadas as técnicas de mapeamento transfinito bilinear em modelos representados por curvas NURBS. As técnicas de mapeamento transfinito aplicadas no contexto do Método dos Elementos Finitos são adaptadas para geração dos pontos de controle internos do modelo. As malhas geradas são formadas por quadriláteros de Bézier e preservam a geometria exata do modelo. As malhas geradas podem ser refinadas através de subdivisão (refinamento-h) ou elevação de grau do polinômio (refinamento-p). Exemplos numéricos são apresentados para avaliar a qualidade das malhas, onde são realizadas análises mecânicas lineares, mostrando a aplicabilidade das malhas geradas em problemas de engenharia estrutural.

Keywords: Análise Isogeométrica, Geração de Malhas, Mapeamento Transfinito

1 INTRODUÇÃO

A Análise Isogeométrica (AIG) é um método numérico apresentado inicialmente por Hughes et al. (2005), que vem sendo estudado em muitos trabalhos nos últimos anos. A principal proposta da AIG é aproximar as ferramentas de modelagem geométrica (sistemas CAD) e as ferramentas de análise numérica. Neste contexto, os modelos geométricos utilizados na modelagem (e.g. curvas e superfícies de Bézier e NURBS) são os mesmos aplicados nas formulações dos métodos numéricos. Deste modo, o erro de geometria existente em métodos numéricos, como no caso do Método dos Elementos Finitos, desaparece.

Muitos trabalhos relacionados a AIG são formulados considerando geometrias definidas por NURBS e T-Splines (Piegl & Tiller, 1997; Sederberg et al., 2003). Estas representações permitem realizar procedimentos de discretização semelhante aos refinamentos p e h existentes para elementos finitos, utilizando algoritmos conhecidos no área da Modelagem Geométrica. Outra vantagem é o uso de modelos com continuidade elevada, definindo assim uma nova estratégia de discretização inexistente nas formulações clássicas de elementos finitos, o refinamento k (Hughes et al., 2005).

Apesar das vantagens de tais abordagens, sua atuação ainda é limitada quando considerado o paradigma de Modelagem-Análise existente no programas de análise numérica. Os modeladores geométricos não fornecem as informações de entrada que os programas de análise numérica necessitam. Muitos sistemas CAD utilizam representações de fronteira (Mantyla, 1988), onde apenas o contorno dos modelos é considerado. Assim, existe um problema na obtenção do domínio de análise a partir de seu contorno. Tal problema é conhecido na área de análise numérica através do Método dos Elementos Finitos e é solucionado com o uso de técnicas de geração de malha.

As técnicas de geração de malhas são algoritmos capazes de construir malhas de elementos dado um contorno como entrada (Miranda et al., 1999, 2013). As técnicas de geração estruturadas, como Mapeamento Transfinito (Haber et al., 1981; Miranda & Martha, 1999), constroem malhas com topologia previamente conhecida.

Diante das dificuldades da geração de modelos NURBS e T-Splines a partir de um contorno fornecido, alguns trabalhos exploraram o uso de *Patches* de Bézier triangulares para geração de modelos isogeométricos (Engvall & Evans, 2016; Jaxon & Qian, 2014). Tais abordagens utilizam as técnicas clássicas de geração de malhas não estruturadas para obter malhas de alta ordem com geometria exata.

Neste trabalho é realizada a geração de malhas estruturadas compostas por superfícies de Bézier planas dado um contorno definido por quatro curvas NURBS. A técnica utilizada consiste em um Mapeamento Transfinito adaptado para obtenção de modelos com geometria exata. Embora os modelos de superfície NURBS sejam naturalmente estruturados, não é trivial a construção destes através de curvas de contorno. A técnica proposta é aplicada na solução de dois exemplos numéricos de elasticidade linear, onde são utilizados elementos de estado plano de tensão com forma dada por superfície de Bézier.

Este artigo está estruturado da seguinte forma. O Item 2 realiza uma revisão bibliográfica dos conceitos relacionados a Modelagem Geométrica de curvas e superfícies que são utilizados na técnica proposta. O Item 3 faz uma breve revisão de literatura sobre Mapeamentos Transfinitos e apresenta sua utilização para geração de malhas isogeométricas. O Item 4 descreve

a formulação da AIG utilizada neste trabalho. Item 5 mostra os exemplos numéricos que são aplicados a técnica de geração de malhas. Por fim, o Item 6 discute as conclusões sobre a técnica proposta e os resultados obtidos nos exemplos numéricos.

2 MODELAGEM GEOMÉTRICA

Uma curva de Bézier de grau d é construída por uma combinação linear dos pontos de controle p_i :

$$C(t) = \sum_{i=1}^{d+1} B_{i,p}(t) p_i \quad (1)$$

onde $B_{i,d}$ são os polinômios de Bernstein de grau d e t é uma coordenada paramétrica. A Eq. 1 pode ser escrita em formato matricial:

$$C(t) = \mathbf{b}(t) \mathbf{p}^T \quad (2)$$

onde $\mathbf{b}(t)$ é um vetor contendo os polinômios de Bernstein e $\mathbf{p}$ é um vetor contendo tuplas (x, y, z) correspondentes aos pontos de controle p_i . Os polinômios de Bernstein podem ser avaliados através da fórmula recursiva (Piegl & Tiller, 1997):

$$B_{i,d}(t) = (1-t)B_{i,d-1}(t) + tB_{i-1,d-1}(t) \quad (3)$$

$$B_{i,d}(t) = 0 \quad \text{para } i < 1 \quad \text{e } i > d+1, \quad B_{1,1}(t) = 1$$

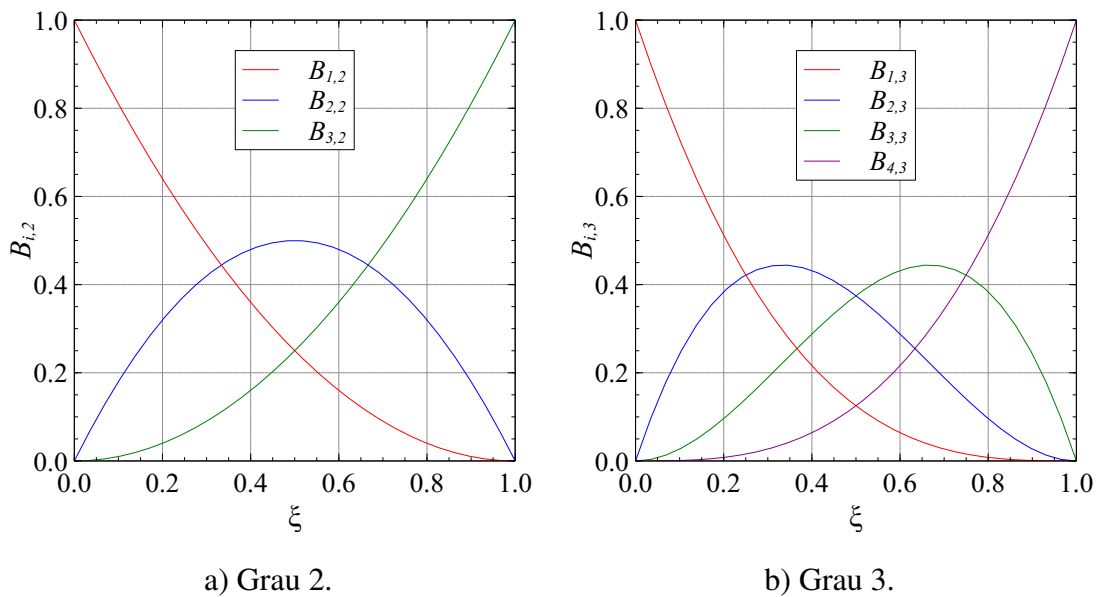


Figura 1: Polinômios de Bernstein quadráticos e cúbicos.

A Figura 1 mostra polinômios de Bernstein de diferentes graus. A primeira derivada dos polinômios de Bernstein pode ser avaliada por:

$$\frac{d}{dt} B_{i,d}(t) = d(B_{i-1,d-1}(t) - B_{i,d-1}(t)) \quad (4)$$

Os polinômios de Bernstein $B_{i,p}$ possuem as importantes propriedades:

- Independência linear;
- Não-negatividade: $B_{i,d}(t) \geq 0$;
- Partição da unidade: $\sum_{i=1}^n B_{i,d}(t) = 1$
- Simetria: $B_{i,d}(t) = B_{d-i,d}(1-t)$;
- Controle pseudo-local: Se um ponto de controle de uma representação de Bézier for movido, então as mudanças da forma da representação ocorrerão de forma dominante na vizinhança do ponto movido.

As B-splines são uma generalização das curvas de Bézier, capazes de descrever vários segmentos distintos ao longo da mesma representação paramétrica. Esta característica é obtida limitando a atuação das funções base em regiões do espaço paramétrico. Estas regiões são conhecidas na literatura como *knot spans* e são definidas por um vetor de valores paramétricos, o vetor de *knots*. Uma curva B-spline é construída por uma combinação linear entre os pontos de controle $\mathbf{p}_i$ e funções de base $N_{i,d}(t)$, como mostra a expressão:

$$C(t) = \sum_{i=1}^n N_{i,p}(t) \mathbf{p}_i \quad (5)$$

onde n é o número de funções de base B-splines e d é o grau da curva.

As bases B-splines requerem um vetor de *knots*, que consiste num conjunto de valores paramétricos não-negativos e não-decrescentes delimitados ao longo do intervalo paramétrico $[t_1, t_{n+p+1}]$ no qual a curva foi definida.

Considerando o vetor de *knots* $\Xi = [t_1, t_2, \dots, t_{n+p+1}]$, as funções de base B-spline são definidas pela fórmula recursiva de Cox-de Boor Piegl & Tiller (1997):

$$N_{i,1}(t) = \begin{cases} 1, & t_i \leq t < t_{i+1} \\ 0, & \text{caso contrário} \end{cases}$$

$$N_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} N_{i,d-1}(t) + \frac{t_{i+d+1} - t}{t_{i+d+1} - t_{i+1}} N_{i+1,d-1}(t) \quad (6)$$

O número de bases n pode ser calculado em função do tamanho do vetor de *knots* (ks) e do grau das bases (d):

$$n = ks - d - 1$$

Outra observação importante é que o número de bases não nulas em cada *knot span* é sempre igual a $d + 1$. Esta propriedade permite que as B-splines sejam avaliadas de forma bastante eficiente.

Uma curva B-spline pode ser refinado adicionando novos valores *knots* ou elevando o grau das funções base sem modificar a geometria final da curva. Os algoritmos utilizados para esta finalidade são a *Inserção de Knot* e a *Elevação do Grau* (Piegl & Tiller, 1997). Estes algoritmos podem ser aplicados de forma eficiente a curvas B-spline de qualquer geometria e grau.

As curvas B-splines podem representar curvas de Bézier caso seja considerado um único *knot span* e as multiplicidades dos *knots* deste sejam $p + 1$:

$$\Xi = [\underbrace{0, \dots, 0}_{d+1}, \underbrace{1, \dots, 1}_{d+1}]$$

De modo geral, uma curva B-spline pode ser representada por um conjunto de curvas de Bézier, onde cada Bézier representa um *knot span* na B-spline. O procedimento utilizado para obtenção das curvas de Bézier equivalente é conhecido na literatura como Decomposição de Bézier (Piegl & Tiller, 1997) ou Extração de Bézier (Borden et al., 2011).

Uma das desvantagens das curvas de Bézier e B-spline apresentadas anteriormente é que estas não são capazes de representar curvas cônicas, como circunferências e elipses, de forma exata, pois tais curvas não são descritas por funções polinomiais. Contudo, esta representação é possível utilizando funções racionais. Para este fim são acrescentados pesos aos pontos de controle da curva. Desta forma, a curva de Bézier passa a ser calculada pela expressão:

$$C(t) = \sum_{i=1}^{d+1} R_{i,d}(t) p_i \quad (7)$$

onde $R_{i,d}(t)$ são as bases de Bézier racionais dadas por:

$$R_{i,d} = \frac{\sum_{i=1}^{d+1} w_i B_{i,d}(t)}{\sum_{i=1}^{d+1} w_i \hat{B}_{i,d}(t)} \quad (8)$$

As curvas B-spline racionais são avaliadas de forma análoga. As NURBS (Non-Uniform Rational B-spline) são B-Splines racionais que podem utilizar vetor de *knots* não uniformes (os valores do vetor de *knots* não apresentam, necessariamente, variação constante). As NURBS são amplamente utilizadas em modelagem computacional, pois oferecem uma forma matemática capaz de representar tanto modelos matemáticos padrão (quádricas, cônicas, superfícies de revolução, ...) como também modelos de forma livre, utilizando a mesma base de dados para armazenamento de ambos (Piegl, 1991).

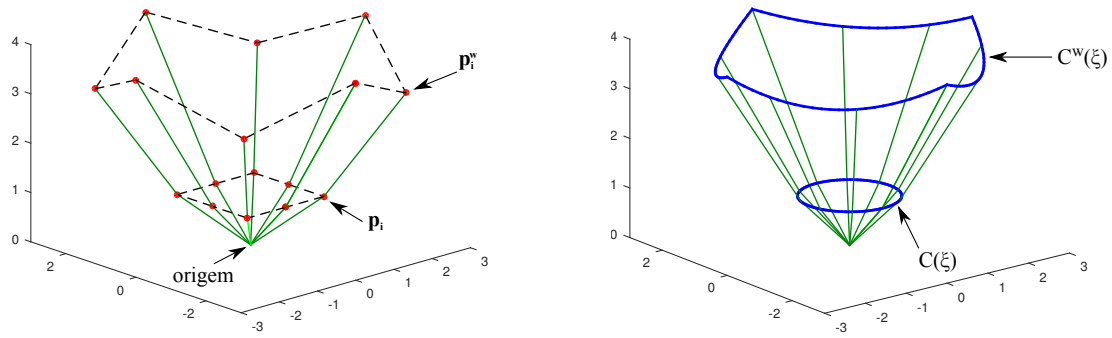
Uma maneira alternativa de avaliar as curvas racionais é projetando uma curva não-racional no hiperplano definido por $w = 1$, como ilustrado na Figura 2. As curvas não racionais são definidas no espaço R^{dim+1} tratando as coordenadas do problema como Coordenadas Homogêneas, nas quais os pesos se tornam componente. Considerando $dim = 2$, os pontos de projeção (p^w) são obtidos em função dos pontos projetados (p) por:

$$p_w = (x^w, y^w, w) \quad (9)$$

onde $p = (x, y, w)$. Por outro lado, os pontos projetados podem ser avaliados através de um ponto de projeção por:

$$p = \left(\frac{x^w}{w}, \frac{y^w}{w}, w \right); \quad (10)$$

onde x^w e y^w são as componentes x e y do ponto de projeção p^w .



a) Pontos de controle.

b) Curva.

Figura 2: Curva NURBS $C(t)$ em R^2 sendo obtida da projeção de uma curva B-spline $C^w(t)$ em R^3 .

Assim, as curvas de Bézier racional e NURBS podem ser avaliadas escrevendo seus pontos de controle no espaço de projeção através da Eq. 9, avaliando o ponto p^w pelas equações não racionais 1 ou 5 (neste caso a componente w também é avaliada), e em seguida projetar este ponto para o espaço projetado através da Eq. 10. Este procedimento também é utilizado para aplicação de algoritmos geométricos (e.g. Subdivisão e Elevação de Grau) em curvas racionais, mesmo que estes tenham sido definidos inicialmente para curvas não racionais. Este mesmo procedimento será utilizado na técnica de geração de malhas proposta neste trabalho.

2.1 Superfícies de Bézier racional

Uma superfície de Bézier racional definida por produto tensorial é avaliada pelo produto de duas funções de base univariantes de Bézier racional. Dada uma superfície S definida por uma matriz de pontos de controle $\mathbf{P}$ ($d + 1 \times q + 1$), uma base de Bézier racional de grau d na direção r e uma base de Bézier racional de grau q na direção s , a superfície é avaliada por:

$$S(r, s) = \sum_{i=1}^n \sum_{j=1}^m R(r, s)_{ij} \mathbf{P}_{ij} \quad (11)$$

onde $R(r, s)$ é a função de base racional bivalente dada por:

$$R(r, s)_{ij} = \frac{w_{ij} B_{i,d}(r) B_{j,q}(s)}{W(r, s)}, \quad (12)$$

$W(r, s)$ é a função peso bivalente:

$$W(r, s) = \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m w_{\hat{i}\hat{j}} B_{\hat{i},d}(r) B_{\hat{j},q}(s) \quad (13)$$

As derivadas parciais de $W(r, s)$ são expressas por:

$$\begin{aligned} \frac{\partial}{\partial r} W(r, s) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m w_{\hat{i}\hat{j}} B'_{\hat{i},d}(r) B_{\hat{j},q}(s) \\ \frac{\partial}{\partial s} W(r, s) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m w_{\hat{i}\hat{j}} B_{\hat{i},d}(r) B'_{\hat{j},q}(s) \end{aligned} \quad (14)$$

Assim, as derivadas parciais das funções de base bivariantes são calculadas por:

$$\begin{aligned}\frac{\partial}{\partial r} R_{ij}(r, s) &= w_{ij} \frac{W(r, s) B'_{i,d}(r) B_{j,q}(s) - \frac{\partial}{\partial r} W(r, s) B_{i,d}(r) B_{j,q}(s)}{W^2(r, s)} \\ \frac{\partial}{\partial s} R_{ij}(r, s) &= w_{ij} \frac{W(r, s) B_{i,d}(r) B'_{j,q}(s) - \frac{\partial}{\partial s} W(r, s) B_{i,d}(r) B_{j,q}(s)}{W^2(r, s)}\end{aligned}\quad (15)$$

As derivadas parciais são necessárias na formulação de elementos isogeométricos bidimensionais. Neste trabalho elas são usadas para a montagem da matriz de rigidez do problema.

3 GERAÇÃO DE MALHAS ESTRUTURADAS

Esta seção aborda as técnicas de geração de malha desenvolvidas neste trabalho. É realizada uma revisão de literatura sobre a técnica clássica de mapeamentos transfinitos utilizada para geração de malhas de elementos finitos. A técnica proposta é abordada em seguida.

3.1 Mapeamento Transfinito Bilinear

O mapeamento transfinito bilinear (Haber et al., 1981; GONÇALVES, 2004) é uma técnica de geração de malhas de superfícies. As malhas são processadas utilizando uma divisão regular no espaço paramétrico de um *patch* de Coons (Farin, 2002).

Um *patch* de Coons é uma superfície paramétrica definida por quatro curvas de contorno paramétricas (c_1, c_2, g_1, g_2), definidas no intervalo paramétrico $[0,1]$, como ilustrado na Figura 3. Um *patch* de Coons $M(r, s)$ é avaliado por:

$$\begin{aligned}M(r, s) &= (1 - s) c_1(r) + s c_2(r) + (1 - r) g_1(s) + r g_2(s) - (1 - r) s M01 \\ &\quad - r s M11 - r (1 - s) M10 - (1 - r) (1 - s) M00\end{aligned}\quad (16)$$

onde $M00, M01, M10$ e $M11$ são respectivamente os pontos avaliados nas coordenadas paramétricas $(0, 0), (0, 1), (1, 0)$ e $(1, 1)$. Tais pontos se localizam nos encontros das curvas paramétricas utilizadas pra gerar a superfície. Uma característica importante dos *patches* de Coons é que estes representam de forma exata as curvas de contorno em suas bordas.

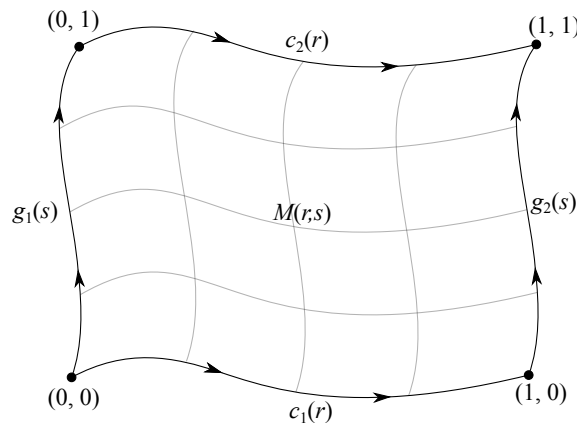


Figura 3: Exemplo de um *patch* de Coons.

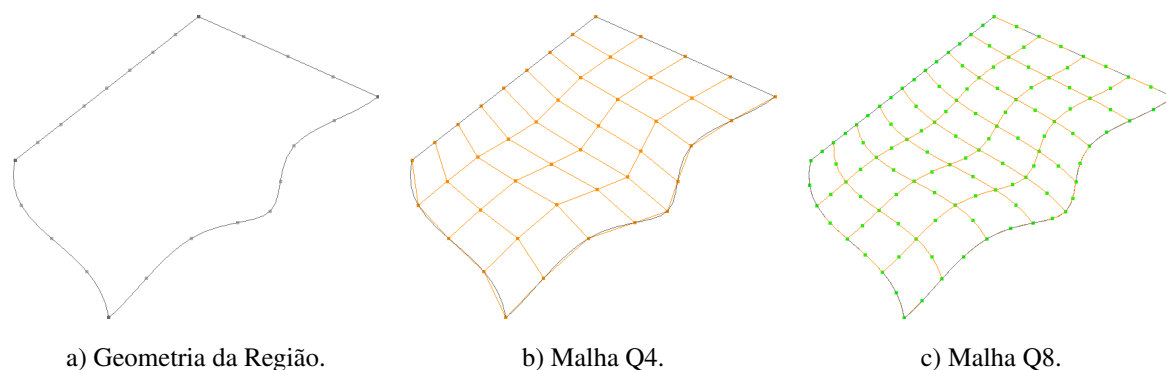


Figura 4: Exemplo de malhas estruturadas geradas via mapeamento transfinito.

No mapeamento transfinito bilinear é definido uma subdivisão em cada direção paramétrica (r e s) de modo a gerar uma grade paramétrica. Esta grade é utilizada para definir a topologia da malha gerada. No caso de elementos quadriláteros, como os elementos finitos Q4 e Q8, os nós dos elementos são definidos no espaço paramétrico e mapeados para o espaço físico através do projetor $M(r, s)$. A Fig. 4 ilustra a técnica sendo utilizada pra gerar malhas de elementos lineares Q4 e elementos quadráticos Q8.

É importante notar esta técnica também pode ser utilizada para geração de malhas de superfície no espaço tridimensional, porém este trabalho se restringe a sua aplicação para geração de malhas planas.

3.2 Mapeamento Transfinito na AIG

A maior motivação na utilização dos mapeamentos transfinitos no contexto da análise isogeométrica é possibilitar a geração de malhas que preservem a geometria do modelo. Neste trabalho, busca-se a geração de malhas planas, portanto as arestas da fronteira devem representar exatamente as curvas de contorno utilizadas para modelagem da região (no caso, as curvas geradoras do *patch* de Coons). Em elementos finitos, apesar do projetor $M(r, s)$ representar exatamente os contornos da região, as funções de forma polinomiais dos elementos não são capazes de representar o contorno para todos os casos (e.g. curvas cônicas como no caso de furos circulares).

Neste trabalho é considerado que a região de geração da malha é composta por curvas NURBS. Os requisitos para aplicação do mapeamento são que cada par de curvas $\{c_1, c_2\}$ e $\{g_1, g_2\}$ devem possuir mesmo grau e apresentar mesmas componentes paramétricas nos vetores de *knots*. Vale ressaltar que tais restrições não representam limitação quanto a robustez da técnica. Por exemplo, se as curvas c_1 e c_2 apresentam *knots* e grau diferentes, pode-se utilizar o algoritmo de Inserção de Knot para compatibilizar os *knots* e o algoritmo de Elevação de Grau para compatibilizar os graus das curvas, sem em nenhum dos casos modificar a geometria do modelo. A discretização da malha passa a ser dada diretamente pelos vetores de *knots* das curvas.

O projetor definido pela Eq. 16 é aplicado no espaço de projeção das curvas NURBS, ao invés de ser no espaço projetado. Deste modo os pesos das curvas de contorno podem ser interpolados de modo análogo ao que é feito para as coordenadas cartesianas.

Os elementos gerados possuem forma descrita por superfície de Bézier, com grau $p \times q$

iguais aos graus presentes nos pares de curvas de contorno, p para c_1, c_2 e q para g_1, g_2 . Os pontos de controle das quinas podem ser determinados diretamente pela Eq. 16, uma vez que estes são interpoladores nas superfícies de Bézier geradas. Entretanto, o mesmo não se verifica no caso dos demais pontos de controle da superfície.

Uma estratégia robusta para obtenção destes pontos de controle seria avaliar alguns pontos pertencentes à superfície de Bézier, montando assim um sistema de equações lineares da forma de:

$$\mathbf{B} \mathbf{p} = \bar{\mathbf{p}} \quad (17)$$

onde $\mathbf{B}$ é uma matriz construída a partir dos polinômios de Bernstein, $\mathbf{p}$ são os pontos de controles a determinar e $\bar{\mathbf{p}}$ são os pontos no espaço físico avaliados pela Eq. 16.

A avaliação de pontos de controle pertencentes às arestas dos elementos podem ser realizadas de forma mais eficiente. Na região paramétrica que define as arestas dos elementos, as curvas de contorno do *patch* de Coons podem ser representadas por curvas de Bézier, pois esta região representa um *knot span* em um dos pares de curvas de contorno, como ilustrado na Figura 5. As arestas superiores e inferiores dos elementos são definidas no *knot span* $[r_i, r_f]$ das curvas c_1 e c_2 , e para uma coordenada s fixa. De mesmo modo ocorre para as arestas à direita e à esquerda, que são definidas no *knot span* $[s_i, s_f]$ das curvas g_1 e g_2 . Logo, tais curvas podem ser escrita por uma curva de Bézier equivalente, utilizando o processo de extração de Bézier. Assim, pode-se demonstrar que ao longo das arestas dos elementos, a Eq. 16 pode ser escrita como a equação de uma curva de Bézier (Eq. 1).

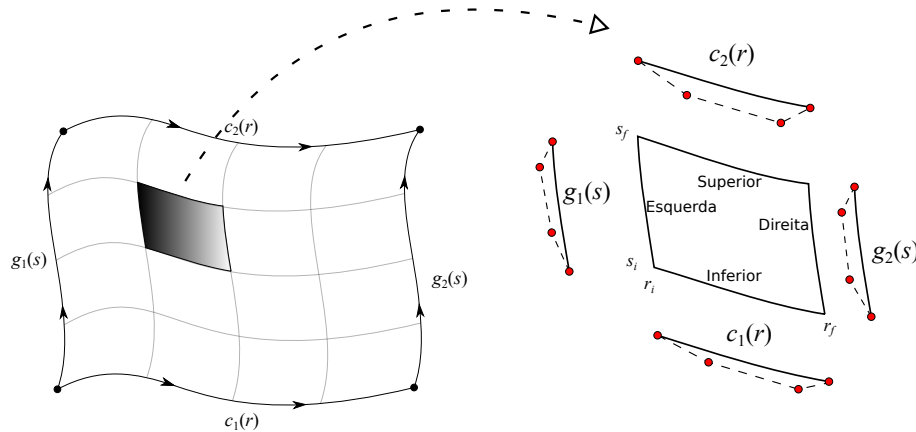


Figura 5: Pontos de controle das curvas de Bézier de cada curva de contorno, identificação das arestas e coordenadas paramétricas definidas para um elemento isogeométrico gerado por mapeamento transfinito isogeométrico.

Considere a aresta superior de uma elemento isogeométrico situado entre os *knots* r_i e r_f , s_i e s_f . A equação da aresta é obtida pela Eq. 16, fazendo $s = s_f$:

$$\begin{aligned} M(r) = & (1 - s_f) c_1(r) + s_f c_2(r) + (1 - r) g_1(s_f) + r g_2(s_f) - (1 - r) s_f M01 \\ & - r s_f M11 - r (1 - s_f) M10 - (1 - r) (1 - s_f) M00 \end{aligned}$$

A equação pode ser escrita de forma mais compacta por:

$$M(r) = (1 - s_f) c_1(r) + s_f c_2(r) + c_l(r) \quad (18)$$

onde $c_l(r)$ representa uma reta dada por:

$$c_l(r) = r \bar{\mathbf{p}}_r + \bar{\mathbf{q}}_r \quad (19)$$

com

$$\bar{\mathbf{p}}_r = g_2(s_f) - g_1(s_f) + (1 - s_f) M00 + s_f M01 - s_f M11 - (1 - s_f) M10 \quad (20)$$

$$\bar{\mathbf{q}}_r = g_1(s_f) - (1 - s_f) M00 - s_f M01 \quad (21)$$

A reta dada por $c_l(r)$ pode ser representada por uma curva de Bézier linear, com pontos de controle extremos definidos por $c_l(r_i)$ e $c_l(r_f)$. Esta reta pode ser representada com grau p e com pontos de controle $\mathbf{p}_3$ utilizando o algoritmo de Elevação de Grau. Neste caso, os pontos de controle do interior da reta são obtidos interpolando os pontos de controle da extremidade, de forma equidistante. Deste modo, a reta $c_l(r)$ pode ser escrita como:

$$c_l(t) = \mathbf{b}_p(t) \mathbf{p}_3 \quad (22)$$

Como explicado anteriormente, as equações $c_1(r)$ e $c_2(r)$ podem ser representadas por curvas de Bézier de grau p . Logo a Eq. 18 pode ser escrita como:

$$M(r) = M(t) = (1 - s_f) \mathbf{b}_p(t) \mathbf{p}_1 + s_f \mathbf{b}_p(t) \mathbf{p}_2 + \mathbf{b}_p(t) \mathbf{p}_3 \quad (23)$$

Onde $\mathbf{p}_1$ e $\mathbf{p}_2$ são os pontos de controle das curvas de Bézier que representam, respectivamente, as curvas $c_1(r)$ e $c_2(r)$ no *knot span* $[r_i, r_f]$. Como as funções de base são iguais, pode-se agrupar os pontos de controle:

$$M(t) = \mathbf{b}(t) ((1 - s_f) \mathbf{p}_1 + s_f \mathbf{p}_2 + \mathbf{p}_3) \quad (24)$$

Portanto, as arestas dos elementos quadrilaterais na direção r consistem em curvas de Bézier de grau p , onde seus pontos de controle são definidos por:

$$\mathbf{p}_r = ((1 - s_f) \mathbf{p}_1 + s_f \mathbf{p}_2 + \mathbf{p}_3) \quad (25)$$

Os pontos de controle da aresta inferior é são obtidos de forma análoga, utilizando s_i ao invés de s_f .

No caso da aresta à direita, a equação desta é obtida substituindo $r = r_f$ na Eq. 16:

$$M(s) = (1 - r_f) g_1(s) + r_f g_2(s) + g_l(s) \quad (26)$$

onde $g_l(s)$ representa uma reta dada por:

$$g_l(s) = s \bar{\mathbf{p}}_s + \bar{\mathbf{q}}_s \quad (27)$$

com

$$\bar{\mathbf{p}}_s = c_2(r_f) - c_1(r_f) + (1 - r_f) M00 + r_f M10 - r_f M11 - (1 - r_f) M01 \quad (28)$$

$$\bar{\mathbf{q}}_s = g_1(r_f) - (1 - r_f) M00 - r_f M01 \quad (29)$$

Os pontos de controle da curva de Bézier resultante são obtidos de forma semelhante ao demonstrado anteriormente para o caso da aresta superior:

$$\mathbf{p}_s = ((1 - s_f) \mathbf{q}_1 + s_f \mathbf{q}_2 + \mathbf{q}_3) \quad (30)$$

Finalmente, os pontos de controle da aresta à esquerda é obtido de forma análoga fazendo $r = r_i$.

A estratégia descrita para encontrar os pontos de controle das arestas dos elementos não pode ser utilizada para os pontos de controle internos. Neste caso são avaliados, pelo *Patch* de Coons, um número de pontos necessário para montar um sistema de equações lineares, de forma semelhante ao discutido para aplicação da Eq. 17. Neste caso, as contribuições dos pontos de controle já determinados são subtraídas dos pontos avaliados, resultando no seguinte sistema:

$$\mathbf{B}_{int} \mathbf{p}_{int} = \mathbf{p}_{coons} - \mathbf{B}_{ext} \mathbf{p}_{ext} \quad (31)$$

onde $\mathbf{B}_{int}$ é uma matriz formada pelas bases bivariadas dos pontos de controle internos, $\mathbf{p}_{int}$ são os pontos de controle internos à determinar, $\mathbf{p}_{coons}$ são os pontos avaliados pelo *Patch* de Coons no interior do elemento, $\mathbf{B}_{ext}$ é uma matriz formada pelas bases bivariadas dos pontos de controle externos e $\mathbf{p}_{ext}$ são os pontos de controle externo.

As coordenadas paramétricas dos pontos avaliados no interior dos elemento são escolhidas de forma equidistante. Por exemplo, no caso de uma problema com $p = q = 2$, existirá 1 ponto de controle interno que será avaliado pela Eq. 16 utilizando as coordenadas locais do *patch* de Bézier $\{r_{loc}, s_{loc}\} = \{0.5, 0.5\}$. De modo geral, as coordenadas em relação ao *Patch* de Coons são encontradas por:

$$\begin{aligned} r &= r_i + (r_f - r_i) r_{loc} \\ s &= s_i + (s_f - s_i) s_{loc} \end{aligned} \quad (32)$$

Vale ressaltar que tal estratégia de escolha dos pontos internos maximiza o valor das funções de base dos pontos de controle internos. Neste trabalho, o sistema de equações apresentado na Eq. 31 é solucionado utilizando a técnica de decomposição LU.

4 ANÁLISE ISOGEOMÉTRICA

A Análise Isogeométrica (AIG) utiliza a mesma ideia da formulação isoparamétrica do MEF, mas neste caso a sequência é invertida: os deslocamentos no interior do sólido são aproximados utilizando as mesmas funções utilizadas para definir a geometria. Desta forma, consegue-se representar de forma exata a geometria do modelo mantendo a garantia de convergência para a solução do problema.

Utilizando os conceitos de modelagem geométrica discutidos no Item 2, a geometria do modelo é escrita como:

$$x = \sum_{a=1}^{np} R_a x_a, \quad y = \sum_{a=1}^{np} R_a y_a \quad (33)$$

onde np é o número de pontos de controle que definem a geometria do modelo, R_a são as funções base utilizadas e x_a, y_a são as coordenadas dos pontos de controle. Neste trabalho, as funções base utilizadas são Bézier racionais bivariadas, discutidas no Item 2.1.

Na Análise Isogeométrica, os deslocamentos no interior do modelo (u, v, w) são aproximados com as mesmas funções base utilizadas para descrever a geometria do sólido:

$$u = \sum_{a=1}^{np} R_a u_a, \quad v = \sum_{a=1}^{np} R_a v_a \quad (34)$$

onde u_a, v_a são os deslocamentos dos pontos de controle. Estas equações podem ser escritas de forma matricial como

$$\hat{\mathbf{u}} = \begin{bmatrix} u \\ v \end{bmatrix} = \sum_{a=1}^{np} \begin{bmatrix} R_a & 0 \\ 0 & R_a \end{bmatrix} \begin{bmatrix} u_a \\ v_a \end{bmatrix} = \sum_{a=1}^{np} \mathbf{N}_a \mathbf{u}_a = \mathbf{N} \mathbf{u} \quad (35)$$

onde $\mathbf{u}$ é o vetor dos graus de liberdade do problema (i.e. deslocamentos dos pontos de controle) e a matriz $\mathbf{N}$ é dada por

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & \dots & \mathbf{N}_{np} \end{bmatrix} \quad (36)$$

Desta forma, a matriz de aproximação dos deslocamentos $\mathbf{N}$ é formada por uma série de submatrizes correspondentes a cada ponto de controle.

É importante notar que diferente do MEF, os pontos de controle não pertencem ao sólido e seus deslocamentos não possuem um sentido físico claro, como ocorrem com os deslocamentos nodais, que correspondem aos deslocamentos de pontos do sólido. Desta forma, a AIG tem características similares às do Método de Rayleigh-Ritz.

Um aspecto importante da AIG é que embora a geometria possa ser descrita exatamente utilizando poucos pontos de controle, a obtenção de resultados precisos na análise requer que um número adequado de graus de liberdade seja considerado. Assim, o número de pontos de controle mínimo a ser utilizado é ditado pela geometria do problema, mas o número efetivamente utilizado depende da precisão requerida da análise.

4.1 Deformações

Considerando que os deslocamentos são pequenos e que é assumido um modelo de estado plano de tensões, as deformações podem ser calculadas como

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} u_{,x} \\ v_{,y} \\ u_{,y} + v_{,x} \end{bmatrix} = \sum_{a=1}^{np} \begin{bmatrix} R_{a,x} & 0 & 0 \\ 0 & R_{a,y} & 0 \\ R_{a,y} & R_{a,x} & 0 \end{bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix} = \mathbf{B} \mathbf{u} \quad (37)$$

É importante notar que a matriz deformação-deslocamento ($\mathbf{B}$) tem o mesmo padrão da matriz $\mathbf{N}$ mostrada na Eq. (36).

As funções base R_a são definidas em função das coordenadas paramétricas (r, s) , mas para calcular a matriz deformação-deslocamento é necessário calcular as derivadas destas funções

em relação às coordenadas cartesianas (x, y) . Assim, como na formulação isoparamétrica do MEF (Cook et al., 2002), na AIG estas derivadas podem ser calculadas utilizando a matriz Jacobiana ($\mathbf{J}$):

$$\mathbf{J} = \begin{bmatrix} \sum R_{a,r} x_a & \sum R_{a,r} y_a \\ \sum R_{a,s} x_a & \sum R_{a,s} y_a \end{bmatrix} \Rightarrow \begin{bmatrix} R_{a,x} \\ R_{a,y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} R_{a,r} \\ R_{a,s} \end{bmatrix} \quad (38)$$

4.2 Equações de Equilíbrio

As equações de equilíbrio do sólido na AIG podem ser obtidas a partir do Princípio dos Trabalhos Virtuais (PTV). De acordo com o PTV, o equilíbrio ocorre quando o trabalho virtual interno (δU) é igual ao trabalho virtual externo (δW_{ext}), qualquer que seja o campo de deslocamentos virtuais $\delta \hat{\mathbf{u}}$, desde que este seja pequeno e obedeça às condições de contorno essenciais do problema (Cook et al., 2002). Matematicamente:

$$\delta U = \delta W_{ext} \Rightarrow \int_V \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} dV = \int_V \delta \hat{\mathbf{u}}^T \mathbf{b} dV + \int_S \delta \hat{\mathbf{u}}^T \mathbf{q} dS + \sum \delta \hat{\mathbf{u}}_j^T \mathbf{F}_j \quad (39)$$

onde $\delta \boldsymbol{\epsilon}$ é o vetor de deformações virtuais, $\boldsymbol{\sigma}$ é o vetor das tensões, $\mathbf{b}$ são as forças de corpo, $\mathbf{q}$ são as forças de superfície e $\mathbf{F}_j$ são as cargas concentradas atuantes no corpo.

Utilizando a Eq. (37) e considerando apenas pequenos deslocamentos, as deformações virtuais são dadas por:

$$\delta \boldsymbol{\epsilon} = \mathbf{B} \delta \mathbf{u} \quad (40)$$

Assim, o trabalho virtual interno pode ser escrito como:

$$\delta U = \int_V \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}^T \int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}^T \mathbf{g} \quad (41)$$

onde o vetor das forças internas da estrutura ($\mathbf{g}$) é dado por

$$\mathbf{g} = \int_V \mathbf{B}^T \boldsymbol{\sigma} dV \quad (42)$$

Utilizando a Eq. (36), o trabalho virtual externo pode ser escrito como:

$$\delta W_{ext} = \delta \mathbf{u}^T \int_V \mathbf{N}^T \mathbf{b} dV + \delta \mathbf{u}^T \int_S \mathbf{N}^T \mathbf{q} dS + \delta \mathbf{u}^T \sum \mathbf{N}_j^T \mathbf{F}_j = \delta \mathbf{u}^T \mathbf{f} \quad (43)$$

Portanto, o vetor das cargas externas ($\mathbf{f}$) é dado por

$$\mathbf{f} = \int_V \mathbf{N}^T \mathbf{b} dV + \int_S \mathbf{N}^T \mathbf{q} dS + \sum \mathbf{N}_j^T \mathbf{F}_j \quad (44)$$

Nesta expressão, a matriz $\mathbf{N}_j$ corresponde à matriz $\mathbf{N}$ da Eq. (36) avaliada no ponto de aplicação da força j .

As equações de equilíbrio do modelo discreto são obtidas substituindo as Eq. (41) e (43) na expressão do trabalho virtual:

$$\delta U = \delta W_{ext} \Rightarrow \delta \mathbf{u}^T \mathbf{g} = \delta \mathbf{u}^T \mathbf{f} \quad (45)$$

Como os deslocamentos virtuais $\delta \mathbf{u}$ são arbitrários, as equações de equilíbrio da AIG são dadas por

$$\mathbf{g} = \mathbf{f} \quad (46)$$

Neste trabalho, considera-se que o material tem um comportamento elástico linear:

$$\boldsymbol{\sigma} = \mathbf{C} \boldsymbol{\epsilon} \quad (47)$$

onde $\mathbf{C}$ é a matriz constitutiva elástica. No caso de materiais lineares, esta matriz é constante e depende apenas das propriedades do material. Substituindo a relação tensão-deformação a expressão do vetor de forças internas e utilizando a Eq. (37), pode-se escrever:

$$\mathbf{g} = \int_V \mathbf{B}^T \boldsymbol{\sigma} dV = \int_V \mathbf{B}^T \mathbf{C} \boldsymbol{\epsilon} dV = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV \mathbf{u} = \mathbf{K} \mathbf{u} \quad (48)$$

onde a matriz de rigidez da estrutura é dada por:

$$\mathbf{K} = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} dV \quad (49)$$

Finalmente, substituindo a relação $\mathbf{g} = \mathbf{K} \mathbf{u}$ na Eq. (46), podemos escrever as equações de equilíbrio como:

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (50)$$

A solução da Eq. (50) permite a determinação dos graus de liberdade do problema ($\mathbf{u}$), que correspondem aos deslocamentos dos pontos de controle do modelo. Após o cálculo do vetor de deslocamentos $\mathbf{u}$, as deformações são calculadas utilizando a Eq. (37) e as tensões são calculadas utilizando a Eq. (47), completando a solução do problema de análise de tensões em sólidos.

A formulação isogeométrica apresentada foi implementada no software acadêmico de Elementos Finitos FAST (Barroso, 2015), desenvolvido em linguagem de programação C++ e utilizando o paradigma de Programação Orientada a Objetos (POO).

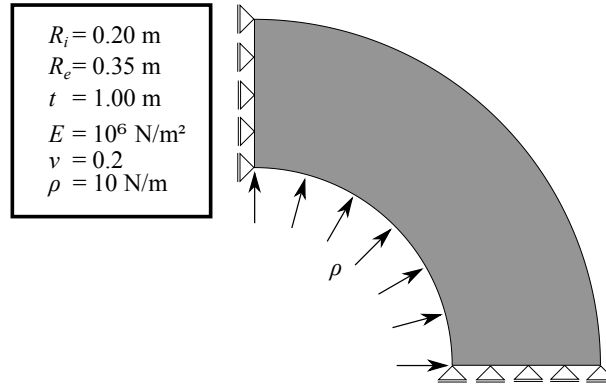
5 Exemplos Numéricos

Nesta seção são apresentados dois exemplos numéricos de elasticidade linear para mostrar a utilização da técnica e aplicação das malhas geradas por esta.

5.1 Cilindro espesso submetido à pressão interna

Este exemplo trata da análise de um cilindro espesso submetido a pressão interna, considerando modelo de estado plano de tensões. A Figura 6 mostra os dados utilizados no exemplo.

Figura 6: Descrição do Exemplo 5.1.



Fonte: Elaborada pelo autor.

A solução analítica deste problema (Solução de Lamé) é dada em coordenadas polares por:

$$u(r) = \left(\frac{1-\nu}{E} \frac{R_i^2}{R_e^2 - R_i^2} r + \frac{1+\nu}{E} \frac{R_i^2 R_e^2}{(R_e^2 - R_i^2)r} \right) \rho \quad r \in [R_i, R_e] \quad (51)$$

$$\sigma_r(r) = -\rho \frac{(R_e^2 - r^2) R_i^2}{(R_e^2 - R_i^2) r^2} \quad (52)$$

$$\sigma_\theta(r) = -\rho \frac{(R_e^2 + r^2) R_i^2}{(R_e^2 - R_i^2) r^2} \quad (53)$$

onde $u(r)$ é o deslocamento radial, R_i é o raio interno do cilindro, R_e é o raio externo do cilindro, σ_r é a tensão radial e σ_θ é a tensão circunferencial.

O problema é solucionado através de uma Análise Isogeométrica utilizando elementos de superfície de Bézier. A Figura 7 mostra os pontos de controle utilizados nas curvas de contorno para realizar a modelagem o problema, considerando grau quadrático para todas as curvas. A Tabela 1 especifica as coordenadas e pesos dos pontos de controle do modelo.

P_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8
x	0.000	0.200	0.200	0.275	0.350	0.350	0.000	0.000
y	0.200	0.200	0.000	0.000	0.000	0.350	0.350	0.275
w	1.00000	0.70711	1.00000	1.00000	1.00000	0.70710	1.00000	1.00000

Tabela 1: Coordenadas e pesos dos pontos de controle mostrados na Figura .

Foram geradas malhas através da técnica abordada neste trabalho, utilizando uma estratégia de refinamento h e p . Vale ressaltar que tais estratégias são realizadas aplicando os algoritmos de Inserção de Knot e Elevação de Grau nas curvas de contorno da região.

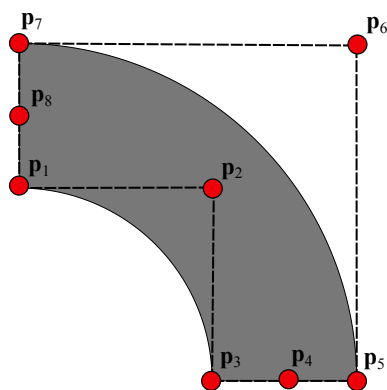


Figura 7: Pontos de controle das curvas de contorno do Exemplo 5.1.

No refinamento h , é atribuído uma subdivisão igual para todas as curvas, dobrando o número de divisões em cada etapa do refinamento. A Figura 8 ilustra as malhas 1-4 para o caso do refinamento h . No refinamento p , o grau dos elementos gerados no contorno inicial é elevado em 1 em cada etapa do refinamento. A Figura 9 ilustra as malhas 1-4 geradas para o caso do refinamento p .

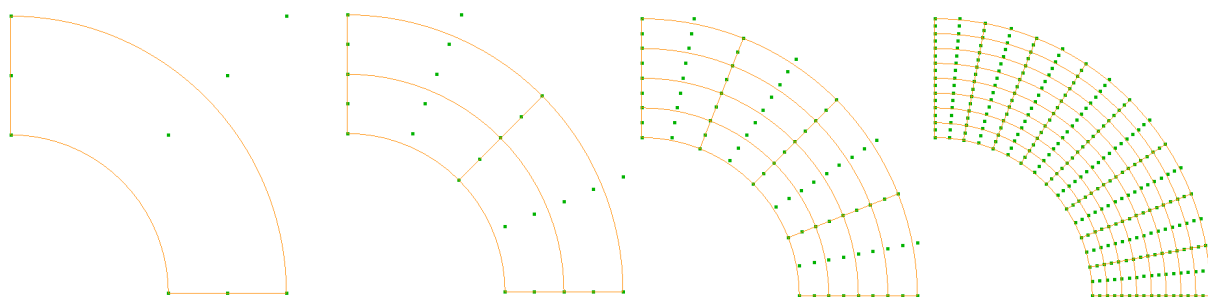


Figura 8: Malhas utilizadas ao longo do refinamento h , no Exemplo 5.1.

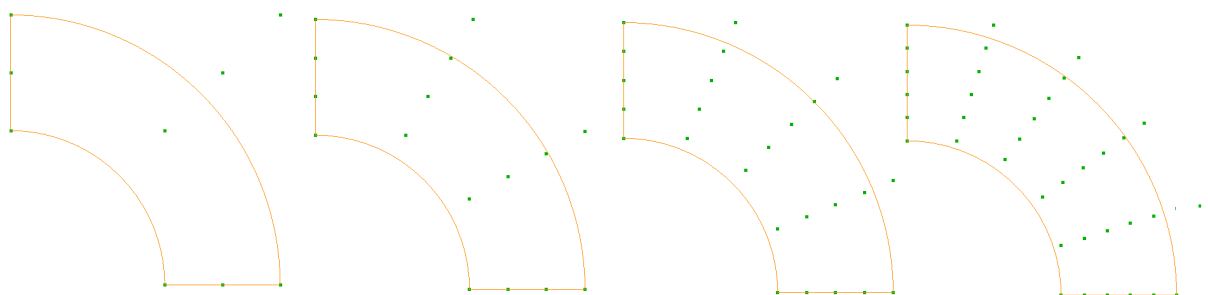


Figura 9: Malhas utilizadas ao longo do refinamento p , no Exemplo 5.1.

A integração numérica foi realizada considerando quadratura de Gauss completa. Neste trabalho, a quadratura completa é aquela capaz de integrar exatamente o polinômio de maior grau que aparece na matriz de rigidez de um elemento não distorcido (jacobiano constante). Vale lembrar que as superfícies de Bézier utilizadas são racionais, logo não é garantido que uma quadratura completa seja suficiente para uma integração exata.

As Figuras 10 e 11 mostra os resultados obtidos para o deslocamento radial, tensão circunferencial e tensão radial, avaliados para $r = 0.2$, onde ocorrem os valores máximos. Os

valores mostrados são a razão entre o valor numérico obtido e o valor da solução analítica. Os resultados são relacionados com os graus de liberdade (*gls*) de cada malha. Todas as respostas convergiram para solução exata do problema com a discretização do modelo. As respostas de deslocamento radial e tensão circunferencial apresentaram maior facilidade para convergência.

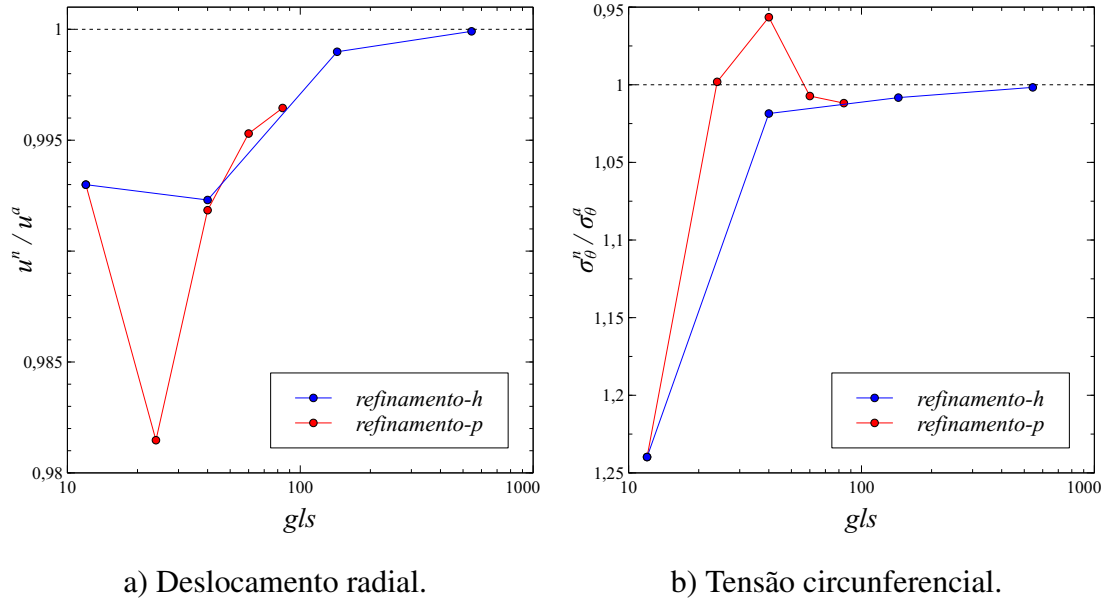


Figura 10: Resultados obtidos no Exemplo 6.

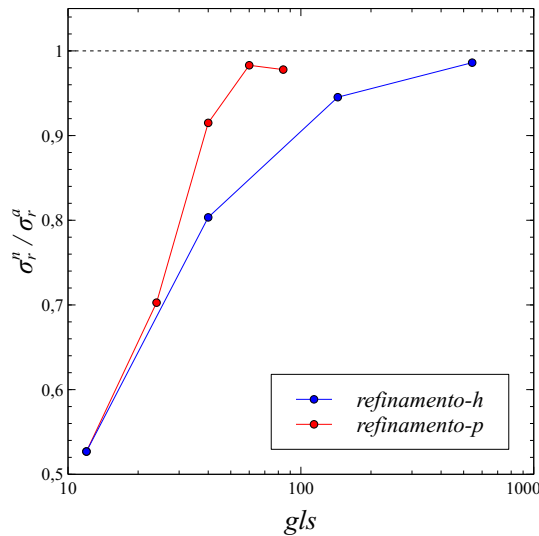


Figura 11: Resultados referentes à tensão radial obtida no Exemplo 6.

5.2 Membrana Elíptica submetida à pressão interna

Este exemplo numérico consiste em uma membrana com formato elíptico, submetido uma carga de tração perpendicular à curva. Este exemplo faz parte de uma coleção de *benchmarks* apresentado em Davies et al. (1993). A Figura 12 mostra os dados utilizados no exemplo.

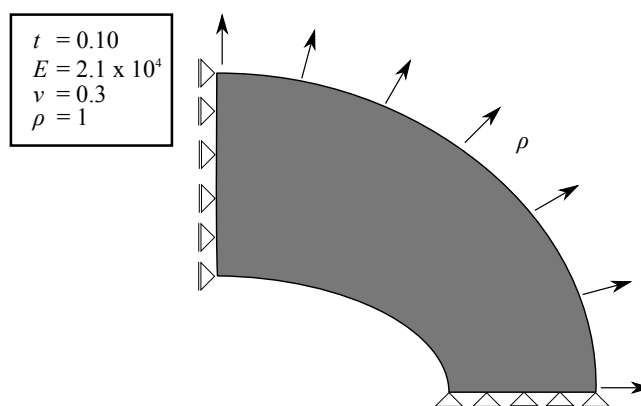


Figura 12: Descrição do Exemplo 5.2.

A Figura 13 mostra os pontos de controle utilizados nas curvas de contorno do exemplo, de mesmo modo como foi feito para o exemplo anterior. A Tabela 2 especifica as coordenadas e pesos dos pontos de controle do modelo.

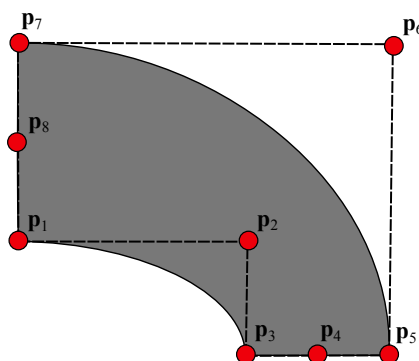


Figura 13: Pontos de controle das curvas de contorno do Exemplo 5.2.

P_i	P1	P2	P3	P4	P5	P6	P7	P8
x	0.000	2.000	2.000	2.625	3.250	3.250	0.000	0.000
y	1.000	1.000	0.000	0.000	0.000	2.750	2.750	1.875
w	1.00000	0.70711	1.00000	1.00000	1.00000	0.70710	1.00000	1.00000

Tabela 2: Coordenadas e pesos dos pontos de controle mostrados na Figura 5.2.

Neste exemplo foi utilizada uma estratégia de refinamento hp . Inicialmente, tem-se um modelo quadrático com os dois pares de curvas não divididas, resultando em uma malha com apenas 1 elemento. Em cada etapa do refinamento é dobrado o número de divisões das curvas e elevado o grau destas em 1. A Figura 14 mostra as malhas geradas.

Os resultados para tensão σ_{yy} no ponto p_3 são apresentados na Figura 15. Os resultados são comparados com a solução numérica apresentada em Davies et al. (1993), onde $\sigma_{yy} = 90.7$. A solução numérica converge rapidamente para a solução de referência.

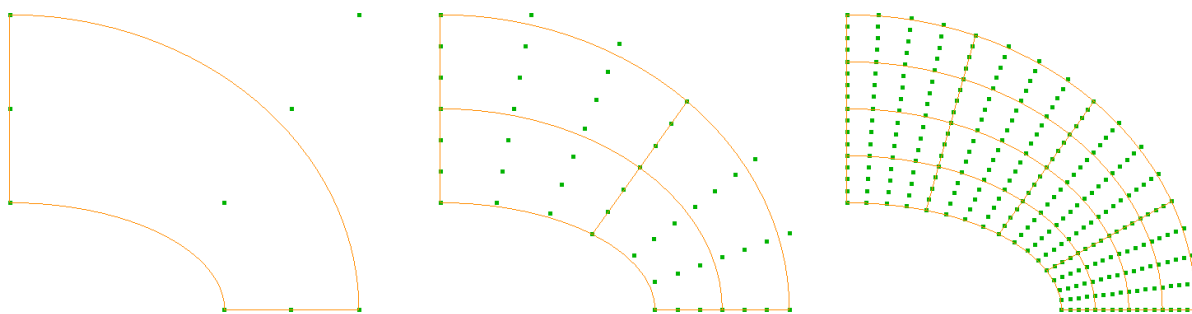


Figura 14: Malhas utilizadas ao longo do refinamento hp , no Exemplo 5.2.

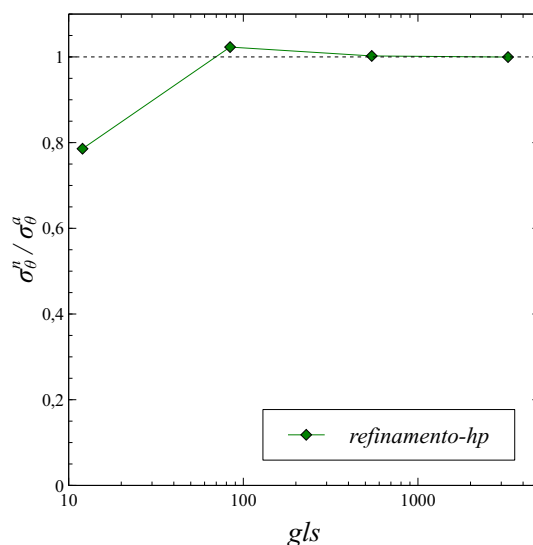


Figura 15: Resultados obtidos no Exemplo 12 utilizando refinamento hp .

6 Conclusão

Este trabalho abordou o uso da técnica de Mapeamento Transfinito aplicada a modelos bidimensionais isogeométricos. A técnica foi adaptada de modo a gerar elementos isogeométricos com forma dada por superfície de Bézier. Dois problemas de elasticidade linear foram solucionados utilizando as malhas geradas pela técnica, onde foram realizados refinamentos p , h e hp . Em todos os casos foram obtidos excelentes respostas nos exemplos analisados, mostrando que uma correta implementação da técnica proposta foi realizada e que esta gera malhas de boa qualidade.

AGRADECIMENTOS

Os autores deste trabalho agradecem o suporte financeiro dado pela CAPES e CNPq.

REFERÊNCIAS

Barroso, E. S. 2015. Análise e otimização de estruturas laminadas utilizando formulação isogeométrica. Dissertação de mestrado, Universidade Federal do Ceará, Brasil, Ceará.

- Borden, M. J., Scott, M. A., Evans, J. A., & Hughes, T. J. R. 2011. Isogeometric finite element data structures based on B{é}zier extraction of NURBS. *International Journal for Numerical Methods in Engineering* 87:15–47.
- Cook, R. D., Malkus, D. S., Plesha, M. E., & Witt, R. J. 2002. Concepts and Applications of Finite Element Analysis. John Wiley & Sons.
- Davies, G. A. O., Fenner, R. T., & Lewis, R. W. 1993. Background to benchmarks. NAFEMS.
- Engvall, L. & Evans, J. A. 2016. Isogeometric triangular Bernstein–Bézier discretizations: Automatic mesh generation and geometrically exact finite element analysis. *Computer Methods in Applied Mechanics and Engineering* 304:378–407.
- Farin, G. 2002. Curves and {Surfaces} for {CAGD}: {A} {Practical} {Guide}. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 5th edition.
- GONÇALVES, M. A. B. 2004. Geração de Malhas Bidimensionais de Elementos Finitos Baseada em Mapeamentos Transfinitos. Dissertação de mestrado, Universidade Federal de Minas Gerais, Belo Horizonte.
- Haber, R., Shephard, M. S., Abel, J. F., Gallagher, R. H., & Greenberg, D. P. 1981. A general two-dimensional, graphical finite element preprocessor utilizing discrete transfinite mappings. *International Journal for Numerical Methods in Engineering* 17:1015–1044.
- Hughes, T. J., Cottrell, J. A., & Bazilevs, Y. 2005. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194:4135–4195.
- Jaxon, N. & Qian, X. 2014. Isogeometric analysis on triangulations. *Computer-Aided Design* 46:45–57.
- Mantyla, M. 1988. Introduction to Solid Modeling. W. H. Freeman & Co., New York, NY, USA.
- Miranda, A., Cavalcante Neto, J., & Martha, L. 1999. An algorithm for two-dimensional mesh generation for arbitrary regions with cracks. In XII Brazilian Symposium on Computer Graphics and Image Processing (Cat. No.PR00481), pp. 29–38. IEEE Comput. Soc.
- Miranda, A. C. O., Lira, W. W. M., Cavalcante-Neto, J. B., Sousa, R. A., & Martha, L. F. 2013. A Three-Dimensional Adaptive Mesh Generation Approach Using Geometric Modeling With Multi-Regions and Parametric Surfaces. *Journal of Computing and Information Science in Engineering* 13:021002.
- Miranda, A. C. O. & Martha, L. F. 1999. Mapeamento transfinito tridimensional. In Proceedings of the XX Iberian Latin-American Congress on Computational Methods in Engineering, pp. 1–13, São Paulo.
- Piegl, L. 1991. On NURBS: a survey. *IEEE Computer Graphics and Applications* 11:55–71.
- Piegl, L. A. & Tiller, W. 1997. The NURBS book. Springer.
- Sederberg, T. W., Zheng, J., Bakenov, A., Nasri, A., Sederberg, T. W., Zheng, J., Bakenov, A., & Nasri, A. 2003. T-splines and T-NURCCs. *ACM Transactions on Graphics* 22:477.