

**PROJETO DE GRADUAÇÃO**

**CÁLCULO DE PROPRIEDADES  
GEOMÉTRICAS DE SÓLIDOS  
MODELADOS EM IGES**

Por,  
**Bruno Bartolozzi Corrêa**

Brasília, 21 de Junho de 2017

**UNIVERSIDADE DE BRASÍLIA**

FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia  
Departamento de Engenharia Mecânica

## PROJETO DE GRADUAÇÃO

# Cálculo de propriedades geométricas de sólidos modelados em IGES

Por,  
**Bruno Bartolozzi Corrêa**

Relatório submetido como requisito parcial para obtenção  
do grau de Engenheiro Mecânico.

### BANCA EXAMINADORA

---

Prof. Éder Lima de Albuquerque (ENM-UnB)  
(Orientador)

---

Prof. Marcus Vinícius Girão de Moraes (ENM-UnB)  
(Examinador Interno)

---

Prof. Carla Tatiana Mota Anflor (ENM-UnB)  
(Examinador Interno)

Brasília/DF, Junho de 2017.

## RESUMO

Este trabalho apresenta a implementação de um código numérico para o cálculo de propriedades geométricas de sólidos modelados em programas CAD e exportados em formato IGES. Inicialmente é mostrados os diversos tipos de representação de curvas e superfícies até chegar às superfícies NURBS. Depois é mostrado como foi feito o cálculo da área superficial e do volume em superfícies NURBS aparadas utilizando o método da integração radial. Por fim, os resultados numéricos obtidos pelo programa foram verificados através de resultados analíticos e do valor de área superficial e volume calculados pelo software SOLIDWORKS.

**Palavras-chave:** NURBS, área superficial, volume, superfícies aparadas, método da integração radial.

## ABSTRACT

This work presents the implementation of a numerical code for calculating the geometrical properties of solids modeled in CAD programs and exported in IGES format. Initially, various types of representation of curves and surfaces are shown finishing with NURBS surfaces. Then it is shown how the surface area and volume were calculated on trimmed NURBS surface using the radial integration method. Finally, numerical results obtained by the program were verified through analytical results and, for more complex solids, the surface area and volume values given by SOLIDWORKS software.

**Keywords:** NURBS, surface area, volume, trimmed surfaces, radial integration method.

## Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	Contextualização . . . . .	1
1.2	Objetivo . . . . .	2
1.3	Metodologia . . . . .	2
1.4	Organização do trabalho . . . . .	2
<b>2</b>	<b>Embasamento Teórico</b>	<b>3</b>
2.1	Introdução . . . . .	3
2.2	Representação de Curvas e Superfícies . . . . .	3
2.3	Curvas de Bézier . . . . .	5
2.4	B-spline . . . . .	7
2.5	NURBS . . . . .	9
2.6	Superfícies NURBS . . . . .	11
2.7	Superfícies NURBS aparadas . . . . .	13
<b>3</b>	<b>IGES</b>	<b>15</b>
3.1	Introdução . . . . .	15
3.2	Formato do arquivo IGES . . . . .	15
3.3	Principais Entidades . . . . .	16
3.3.1	Rational B-Spline Surface Entity (Type 128) . . . . .	17
3.3.2	Rational B-Spline Curve Entity (Type 126) . . . . .	18
3.3.3	Line Entity (Type 110) . . . . .	20
3.3.4	Composite Curve Entity (Type 102) . . . . .	20
3.3.5	Curve on a Parametric Surface Entity (Type 142) . . . . .	22
3.3.6	Trimmed (Parametric) Surface Entity (Type 144) . . . . .	24
3.3.7	Transformation Matrix Entity (Type 124) . . . . .	26
3.4	IGES toolbox . . . . .	28
3.4.1	Principais funções . . . . .	29
3.4.2	Subfunções . . . . .	30
<b>4</b>	<b>Cálculo das propriedades geométricas</b>	<b>31</b>
4.1	Elementos de contorno . . . . .	31
4.2	Método da integração radial (MIR) . . . . .	32
4.3	Cálculo da área superficial de figuras tridimensionais . . . . .	34
4.4	Cálculo do volume de figuras tridimensionais . . . . .	36

<b>5</b>	<b>Descrição do Programa e Resultados</b>	<b>39</b>
5.1	Introdução . . . . .	39
5.2	Descrição do Programa . . . . .	39
5.3	Resultados . . . . .	40
5.4	Discussão sobre os resultados . . . . .	47
<b>6</b>	<b>Conclusão</b>	<b>48</b>

## Lista de Figuras

2.1	<i>Círculo representado parametricamente.</i>	4
2.2	<i>Superfície representada parametricamente.</i>	5
2.3	<i>Curva de Bézier e seu polígono de controle. Fonte: Rogers, D. F., An Introduction to NURBS. Academic Press, 2001, p. 19</i>	5
2.4	<i>Função base de ordem 0, 1 e 2.</i>	8
2.5	<i>Curva B-spline. Fonte: Piegl, L.; Tiller, W., The NURBS Book. 2ª Edição. Springer, 1997, p. 85</i>	9
2.6	<i>Curvas NURBS com o peso do ponto 3 variando. Fonte: Piegl, L.; Tiller, W., The NURBS Book. 2ª Edição. Springer, 1997, p. 120</i>	11
2.7	<i>Superfície NURBS.</i>	12
2.8	<i>Mapeamento de uma superfície aparada.</i>	14
3.1	<i>Formato padrão de um arquivo IGES. Fonte: The Initial Graphics Exchange Specification (IGES) Version 6.0, 2001.</i>	16
3.2	<i>Exemplo de uma composite curve.</i>	20
3.3	<i>Exemplo de transformação de um sistema de coordenadas matricial.</i>	26
4.1	<i>Domínio A.</i>	31
4.2	<i>Contorno dividido em pequenas partes.</i>	32
4.3	<i>Domínio <math>\Omega</math> e suas relações geométricas.</i>	32
4.4	<i>Variáveis do método da integração radial.</i>	33
4.5	<i>Transformação do tridimensional para o bidimensional.</i>	35
5.1	<i>Fluxograma do programa PropGeoIGES.</i>	40
5.2	<i>Cubo com furo.</i>	41
5.3	<i>Placa com furo.</i>	42
5.4	<i>U shape.</i>	43
5.5	<i>Bracket.</i>	44
5.6	<i>Elemento curvo.</i>	45
5.7	<i>Engrenagem.</i>	46

## Lista de Tabelas

3.1	Tipos de entidade . . . . .	16
3.1	Tipos de entidade . . . . .	17
3.2	<i>Rational B-Spline Surface</i> : Números de forma . . . . .	18
3.3	<i>Rational B-Spline Surface: ParameterData</i> . . . . .	18
3.4	<i>Rational B-Spline Curve</i> : Números de forma . . . . .	19
3.5	<i>Rational B-Spline Curve: ParameterData</i> . . . . .	19
3.6	<i>Line Entity: ParameterData</i> . . . . .	20
3.7	Nomenclatura usada em uma <i>composite curve</i> . . . . .	22
3.8	<i>Composite curve: ParameterData</i> . . . . .	22
3.9	<i>Curve on a Parametric Surface: ParameterData</i> . . . . .	24
3.10	<i>Trimmed (Parametric) Surface: ParameterData</i> . . . . .	25
3.11	<i>Transformation Matrix</i> : Números de forma . . . . .	28
3.12	<i>Transformation Matrix: ParameterData</i> . . . . .	28
5.1	Propriedades do cubo com furo . . . . .	41
5.2	Cubo com furo - Entidades presentes no <i>ParameterData</i> . . . . .	41
5.3	Propriedades da placa com furo . . . . .	42
5.4	Placa com furo - Entidades presentes no <i>ParameterData</i> . . . . .	42
5.5	Propriedades do U shape . . . . .	43
5.6	U shape - Entidades presentes no <i>ParameterData</i> . . . . .	43
5.7	Propriedades do bracket . . . . .	44
5.8	Bracket - Entidades presentes no <i>ParameterData</i> . . . . .	44
5.9	Propriedades do elemento curvo . . . . .	45
5.10	Elemento curvo - Entidades presentes no <i>ParameterData</i> . . . . .	45
5.11	Propriedades da Engrenagem . . . . .	46
5.12	Engrenagem - Entidades presentes no <i>ParameterData</i> . . . . .	46



## Lista de Símbolos

### Símbolos Latinos

$B_{i,n}$	Polinômio de Bernstein.
$N_{i,n}$	Função Base.
$R_{i,n}$	Função Base Racional.
$n$	Grau da Curva.
$P_i$	Ponto de Controle.
$p$	Grau do Polinômio de Bernstein.
$U$	Vetor Nós.
$u_i$	Nó.
$K$	Ordem da Derivada.
$k$	Multiplicidade dos Nós.
$m$	Número de Nós menos um.
$C$	Vetor com os Pontos da Curva.
$S$	Vetor com os Pontos da Superfície.
$w$	Peso.
$F$	Integral em função de $\rho$ .
$f$	Função Integrada em $\Omega$ .
$\vec{n}$	Vetor Unitário Normal ao Contorno.
$\vec{r}$	Vetor Unitário Radial ao Contorno.
$\vec{s}$	Vetor Unitário Tangente ao Contorno.
$J$	Jacobiano.

### Símbolos Gregos

$\rho$	Coordenada Radial do Contorno.
--------	--------------------------------

$\Gamma$	Contorno da Área $\Omega$ .
$\Omega$	Área.
$\xi$	Ponto de Gauss.
$\theta$	Ângulo da Colatitude.
$\Phi$	Ângulo da Longitude.

# 1 INTRODUÇÃO

## 1.1 Contextualização

As simulações numéricas demandam muito tempo transmitindo as informações do modelo gerado em programas CAD para programas que fazem a simulação. A razão para isto é que além da tradução de arquivos CAD para uma geometria compatível com a análise, é necessário a criação de malha. Uma opção para tornar a simulação numérica mais eficiente é adotar na simulação a própria geometria gerada em CAD para representar a geometria. Com isso não é mais necessário a geração de uma malha.

Os métodos numéricos são muito importantes para resolver problemas que não possuem solução analítica. Nos anos 60, com o surgimento de computadores, os métodos numéricos começaram a ser mais utilizados. Métodos mais precisos foram desenvolvidos à medida que a tecnologia foi avançando. Dentre esses novos métodos se encontram o método dos elementos finitos (MEF) e o método dos elementos de contorno (MEC) que são baseados nos métodos de Ritz (1909) e Trefftz (1929). O MEC é o mais indicado para ser usado em conjunto com programas CAD nas formulações isogeométricas, já que ambos utilizam apenas representações de superfície.

Como os programas CAD representam apenas as superfícies do sólido, o método dos elementos de contorno é uma ótima escolha para realizar a simulação numérica sem a geração da malha, já que esse método precisa apenas das informações da superfície do sólido. Quando a mesma base usada para representar a geometria é usada na análise numérica, o método numérico é chamado isogeométrico.

Para o desenho em engenharia nos programas CAD, as NURBS são as mais usadas para a representação de curvas e superfícies. Isso ocorre por causa das seguintes características das NURBS:

- São convenientes para a modelagem de superfície de forma livre (que não existe de forma exata);
- Representação de forma exata de todas as seções cônicas como: círculos, cilindros, esferas, elipsóides, entre outras;
- Existência de vários algoritmos numéricos estáveis para geração de objetos NURBS;
- A malha pode ser refinada através da adição de nós;
- Continuidade de derivadas de alta ordem.

Outra tecnologia usada em programas CAD são as T-splines, que surgiram devido a limitações das superfícies NURBS durante o refinamento, especificamente, na adição de nós devido a uma propagação de pontos de controle indesejados. Além disso, T-splines conseguem representar a junção de duas superfícies sem que haja pequenos espaços ao longo da curva de intersecção,

algo comum quando se usa NURBS. Bazilevs et al. (2009) tem analisado o uso de T-splines no contexto da análise isogeométrica e seu trabalho mostra que as T-splines podem ser muito utilizadas futuramente.

No trabalho de Gonçalves [14], o pacote IGES foi explorado e diversos desenhos de sólidos feitos no programa SolidWorks foram salvos em formato IGES e então lidos pelo pacote. Todos os desenhos foram importados e plotados no MatLab sem nenhum prejuízo quanto a representação dos desenhos. O pacote também transforma as entidades do desenho que não são NURBS em NURBS, o que facilita o uso da geometria na análise isogeométrica.

## 1.2 Objetivo

Este trabalho tem como objetivo implementar um código numérico para o cálculo de propriedades geométricas (área superficial e volume) de sólidos modelados em programas de desenho assistido por computador (programas CAD) e exportados em formato IGES. Todos os procedimentos serão implementados em MatLab e como resultado final deste projeto será obtido um programa que permite o cálculo de propriedades geométricas de qualquer sólido, desde que exista um modelo do mesmo em arquivo IGES.

## 1.3 Metodologia

Este trabalho usará o arquivo IGES para exportar os sólidos gerados em programas CAD e utilizará o método da integração radial para o cálculo das propriedades geométricas. A grande vantagem deste método é que ele transforma integrais de área em integrais de linha, tornando a formulação isogeométrica mais eficiente, principalmente no tratamento de superfícies aparadas.

Este método foi utilizado por Correia [13], que calculou propriedades geométricas de sólidos feitos no próprio MatLab e encontrou resultados numéricos muito próximos dos valores analíticos mesmo com formas complexas.

## 1.4 Organização do trabalho

O presente trabalho irá apresentar inicialmente as diferentes maneiras de se representar matematicamente curvas e superfícies, começando com as curvas de Bézier até chegar nas curvas e superfícies NURBS. Depois será falado sobre o método da integração radial que será usado para o cálculo das propriedades geométricas dos sólidos. Na sequência, o formato IGES será explorado, assim como o pacote IGES do MatLab, cujo código fonte encontra-se disponível na página da MathWorks [9], usado para ler os sólidos gerados em CAD. Por fim, serão apresentados os resultados encontrados, mostrando os sólidos explorando junto com suas propriedades geométricas encontradas utilizando o programa feito em MatLab.

## 2 Embasamento Teórico

### 2.1 Introdução

Neste capítulo serão mostradas diferentes maneiras de se representar matematicamente o contorno de figuras planas e sólidos utilizando equações paramétricas. Primeiro será mostrado o caso mais simples de curva, a de Bézier, e depois serão mostrados as B-splines e finalmente as curvas e superfícies NURBS. O MatLab, através da NURBS toolbox [15], consegue construir NURBS a partir de pontos de controle e nós, calcular suas derivadas, plotar curvas e superfícies NURBS, entre outras funções.

As NURBS são as principais curvas utilizadas pelos programas CAD e também são utilizadas como funções de forma para o MEC isogeométrico. Quando se trabalha com as NURBS, não é necessário a criação de uma malha já que a discretização usada pelo modelo geométrico é a mesma usada para o MEC. As NURBS conseguem representar geometrias complexas com exatidão, como por exemplo circunferências, hipérbolas, elipses, parábolas, esferas, cilindros, cones, entre outras.

### 2.2 Representação de Curvas e Superfícies

Curvas e superfícies podem ser representadas na sua forma explícita, implícita ou paramétrica. Esse trabalho focará na representação paramétrica, já que as curvas e superfícies NURBS muito utilizadas pelos programas CAD são representadas parametricamente. As curvas paramétricas são representadas da seguinte forma:

$$C(u) = (x(u), y(u), z(u)) \quad a \leq u \leq b \quad (2.1)$$

na qual  $a$  e  $b$  são os limites do intervalo do parâmetro  $u$ . Um exemplo da representação paramétrica de uma curva pode ser visto Na figura 2.1 representado por:

$$\begin{cases} x = \cos(u) \\ y = \sin(u) \end{cases} \Rightarrow 0 \leq u \leq 2\pi \quad (2.2)$$

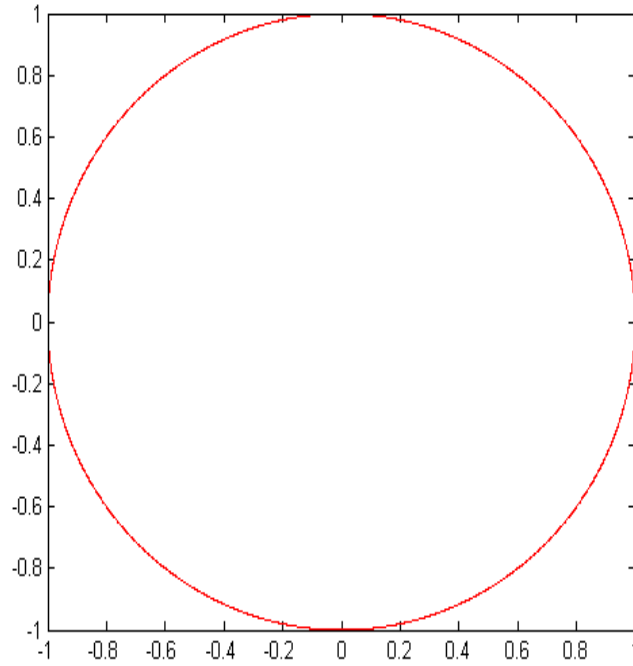


Figura 2.1: *Círculo representado parametricamente.*

As superfícies são representadas de forma semelhante as curvas. Porém elas necessitam de dois parâmetros.

$$S(u, v) = (x(u, v), y(u, v), z(u, v)) \quad a \leq u, v \leq b \quad (2.3)$$

na qual  $a$  e  $b$  são os limites do intervalo dos parâmetros  $u$  e  $v$ . O intervalo não precisa ser o mesmo para os dois parâmetros.

Um exemplo de superfície representada parametricamente (Figura 2.2) é representada por:

$$\begin{cases} x = v \cos(u) \\ y = v \sin(u) \\ z = 2v \end{cases} \Rightarrow \quad 0 \leq v \leq \pi \quad e \quad 0 \leq u \leq 2\pi \quad (2.4)$$

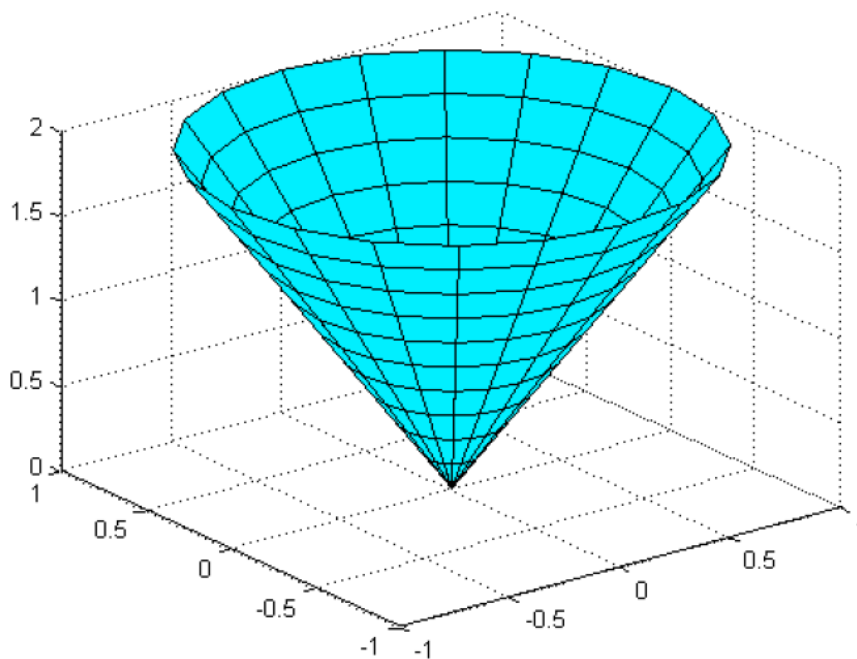


Figura 2.2: Superfície representada parametricamente.

### 2.3 Curvas de Bézier

As curvas de Bézier são definidas por um polígono de controle (Figura 2.3).

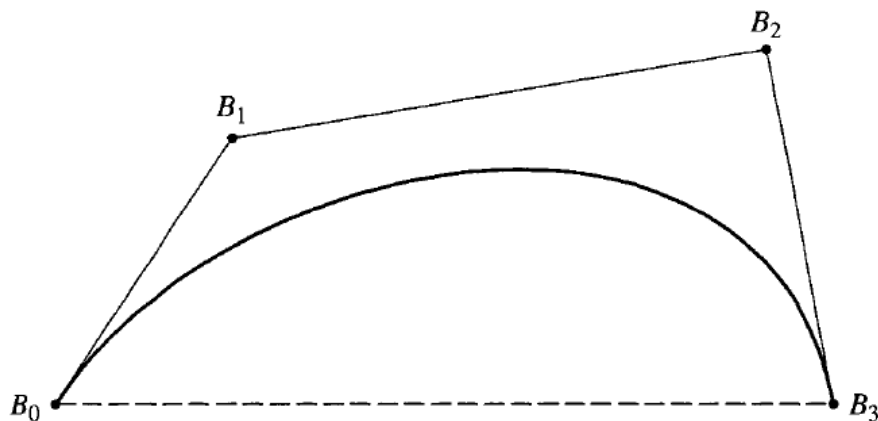


Figura 2.3: Curva de Bézier e seu polígono de controle. Fonte: Rogers, D. F., *An Introduction to NURBS*. Academic Press, 2001, p. 19

A curva de Bézier de grau  $n$  é definida matematicamente por:

$$C(u) = \sum_{i=0}^n B_{i,n}(u)P_i \quad 0 \leq u \leq 1 \quad (2.5)$$

onde  $P_i$  são os pontos de controle que definem o polígono de controle e  $B_{i,n}$  são as funções de

base e do polinômio de Bernstein:

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad (2.6)$$

As funções de base possuem as seguintes propriedades:

- São sempre positivas;
- $\sum_{i=0}^n B_{i,n}(u) = 1$ ;
- $B_{0,n}(0) = B_{1,n}(1) = 1$ ;
- Sua derivada è dada por:

$$\frac{dB_{i,n}}{du} = n(B_{i-1,n-1}(u) - B_{i,n-1}(u)) \quad (2.7)$$

Para manter essas propriedades é necessário que o parâmetro seja definido pelo intervalo  $0 \leq u \leq 1$ .

Essas propriedades da função de base resultam nas seguintes propriedades para as curvas de Bézier:

- O grau do polinômio definido pela curva é igual ao número de pontos de controle menos 1;
- O primeiro e o último pontos da curva são coincidentes com o primeiro e o último pontos do polígono de controle;
- A curva está sempre contida na envoltória complexa dos pontos de controle;
- Cada ponto da curva depende de todos os pontos de controle. Uma mudança em um ponto de controle altera toda a curva.

A partir das derivadas das funções de base pode-se calcular a derivada da curva de Bézier:

$$C'(u) = \sum_{i=0}^n B'_{i,n}(u) P_i = \sum_{i=0}^n n(B_{i-1,n-1}(u) - B_{i,n-1}(u)) P_i = n \sum_{i=0}^n B_{i,n-1}(u) (P_{i+1} - P_i) \quad (2.8)$$

Se a curva de Bézier for de grau  $n$ , sua derivada é uma curva de grau  $n - 1$  possuindo as mesmas propriedades.



## 2.4 B-spline

Duas características da base de Bernstein, usada para gerar as curvas de Bézier, limitam a flexibilidade das curvas resultantes. A primeira característica é que para aumentar ou diminuir o grau da curva é necessário acrescentar ou retirar um ponto de controle. A outra característica é que, como cada ponto da curva depende de todos os pontos de controle, se houver uma mudança em um ponto de controle toda a curva sofrerá alterações.

Outra função de base é a base B-spline. Essa possui a base de Bernstein como um caso especial. Com essa base se pode mudar o grau da curva sem mudar o número de pontos de controle. Além disso, cada ponto de controle interfere apenas em uma região da curva.

Uma curva B-spline é formada por uma ou mais curvas de Bézier com um mecanismo de continuidade entre elas. A curva B-spline de grau  $p$  é dada por:

$$C(u) = \sum_{i=0}^n N_{i,p}(u)P_i \quad a \leq u \leq b, \quad (2.9)$$

na qual  $P_i$  são os pontos de controle e  $N_{i,p}(u)$  são as funções de base B-spline de grau  $p$  definida pela fórmula recorrente:

$$N_{i,0} = \begin{cases} 1, & \text{se } u_i \leq u \leq u_{i+1} \\ 0, & \text{outros casos,} \end{cases} \quad (2.10)$$

$$N_{i,p} = \frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u).$$

Os valores de  $u_i$  são os nós e formam o vetor nós  $\mathbf{U}$ . O vetor  $\mathbf{U}$  deve ser composto de valores reais não decrescentes, ou seja  $u_i \leq u_{i+1}$ . O vetor nó é dado por:

$$\mathbf{U} = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\}. \quad (2.11)$$

Normalmente assume-se que  $a = 0$  e  $b = 1$ .

Algumas propriedades dessa função base são:

- $N_{i,p} = 0$ , se  $u$  está fora do intervalo  $[u_i, u_{i+p+1})$ ;
- $N_{i,p}(u) \geq 0$ , para todo  $i, p$  e  $u$  não negativos;
- Em um intervalo arbitrário,  $[u_i, u_{i+p+1})$ ,  $\sum_{j=i-p}^i N_{j,p}(u) = 1$  para todo  $u$  dentro deste intervalo;
- $N_{i,p}(u)$  atinge apenas um valor máximo, exceto para o caso  $p = 0$ ;
- Em um nó,  $N_{i,p}(u)$  é derivável  $p - k$  vezes, onde  $k$  é multiplicidade do nó. Com isso, aumentando o grau, aumenta a continuidade, e aumentando a multiplicidade do nó, diminui

a continuidade;

- A derivada da função base é dada pela equação:

$$N_{i,p}^{(k)}(u) = p \left( \frac{N_{i,p-1}^{(k-1)}}{u_{i,p} - u_i} - \frac{N_{i+1,p-1}^{(k-1)}}{u_{i+p+1} - u_{i+1}} \right) \quad (2.12)$$

onde  $N_{i,p}^{(k)}(u)$  denota a  $k$ -ésima derivada da função base.

A função base está exemplificada na figura 2.4.

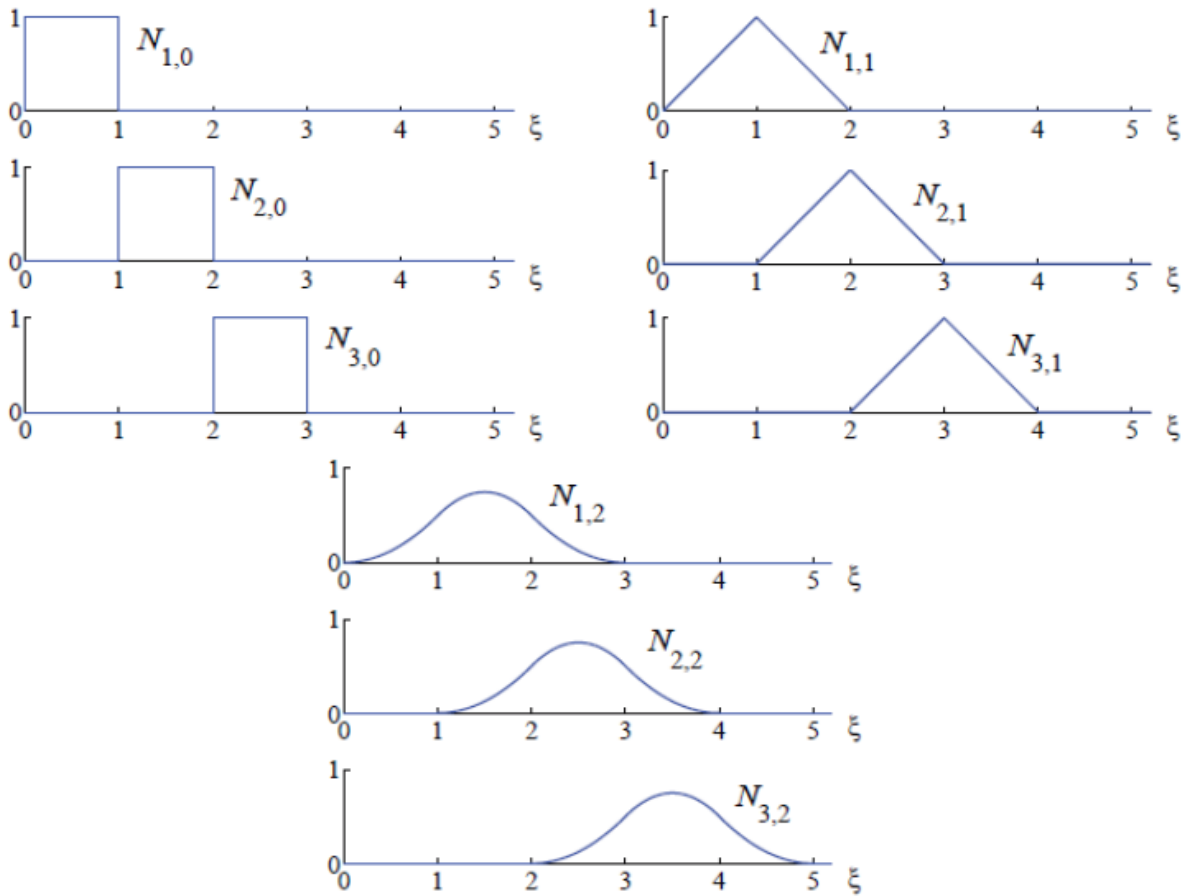


Figura 2.4: Função base de ordem 0, 1 e 2.

Logo cita-se algumas propriedades das curvas B-spline:

- Se  $n = p$ , então a curva B-spline é uma curva de Bézier;
- O grau da curva ( $p$ ), o número de pontos de controle ( $n + 1$ ) e o número de nós ( $m + 1$ ) se relacionam por:  $m = n + p + 1$ ;
- $C(0) = P_0$  e  $C(1) = P_n$ , ou seja, os pontos inicial e final da curva são coincidentes com os pontos inicial e final do polígono de controle;

- A curva pode ser modificada localmente, isto é, movendo o ponto  $P_i$  modifica a curva apenas no intervalo  $[u_i, u_{i+p+1})$ ;
- Vários pontos de controle podem ser coincidentes. Isso gera ângulos retos, por exemplo;
- A  $k$ -ésima derivada da equação da curva é dada por:

$$C^{(k)}(u) = \sum_{i=1}^{n-k} N_{i,p-k}(u) P_i^{(k)} \quad (2.13)$$

onde:

$$P_i^{(k)} = \begin{cases} P_i & , \text{ se } k = 0 \\ \frac{p-k+1}{u_{i+p+1}-u_{i+k}} (P_{i+1}^{(k-1)} - P_i^{(k-1)}) & , \text{ se } k > 0, \end{cases} \quad (2.14)$$

e o vetor nós é dado por:

$$\mathbf{U}^{(k)} = \{\underbrace{0, \dots, 0}_{p-k+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{1, \dots, 1}_{p-k+1}\} \quad (2.15)$$

A figura 2.5 exemplifica uma curva B-spline.

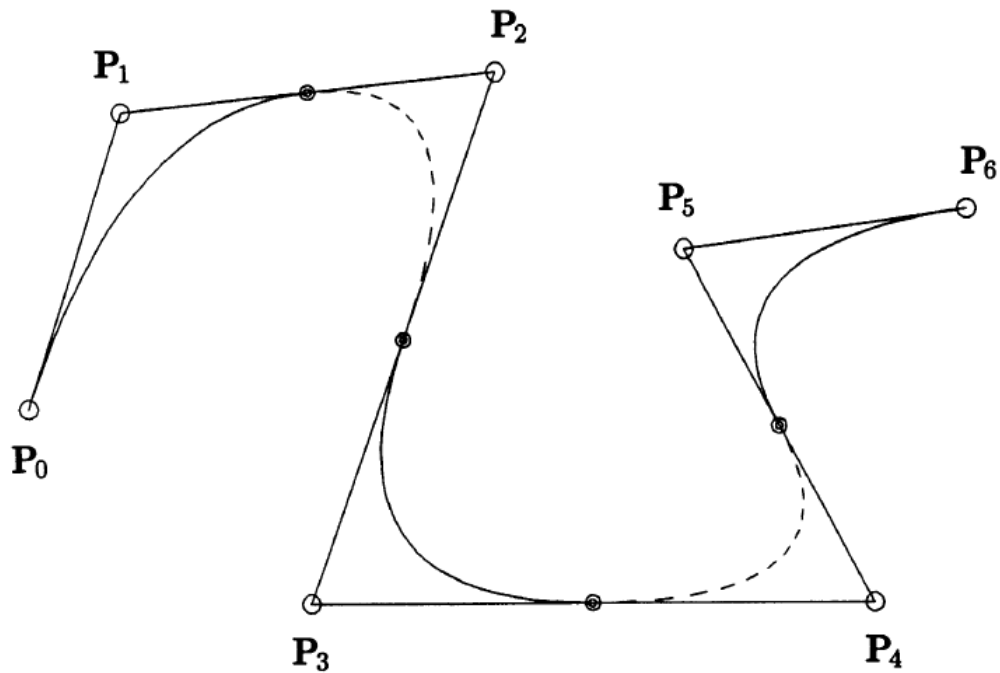


Figura 2.5: Curva B-spline. Fonte: Piegl, L.; Tiller, W., *The NURBS Book*. 2ª Edição. Springer, 1997, p. 85

## 2.5 NURBS

Nesta seção serão definidas as B-splines racionais não uniformes. Em uma NURBS (Non Uniform Rational B-Spline), cada ponto de controle possui um peso associado a si. Quando

todos os pesos tiverem o mesmo valor, a curva será uma B-spline. Uma curva NURBS de grau  $p$  é definida por:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_i P_i}{\sum_{i=0}^n N_{i,p}(u)w_i} \quad a \leq u \leq b \quad (2.16)$$

na qual  $P_i$  são os pontos de controle, que formam o polígono de controle,  $w_i$  são os pesos e  $N_{i,p}(u)$  são as funções base B-spline definidas pelo vetor nó:

$$\mathbf{U} = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\} \quad (2.17)$$

Geralmente assume-se que  $a = 0$ ,  $b = 1$  e  $w_i > 0$  para todos os  $i$ . Definindo:

$$R_{i,p}(u) = \frac{N_{i,p}(u)w_i}{\sum_{j=0}^n N_{j,p}(u)w_j} \quad (2.18)$$

podemos reescrever a equação da curva NURBS da seguinte forma:

$$C(u) = \sum_{i=0}^n R_{i,p}(u)P_i \quad (2.19)$$

$R_{i,p}(u)$  é a função de base racional e possui as seguintes propriedades;

- $R_{i,p}(u) \geq 0$  para todo  $i, p$  e  $u \in [0, 1]$ ;
- $\sum_{i=0}^n R_{i,p}(u) = 1$  para todo  $u \in [0, 1]$ ;
- $R_{0,p}(0) = R_{n,p}(1) = 1$ ;
- $R_{i,p}(u)$  atinge apenas um valor máximo no intervalo  $u \in [0, 1]$ , para todo  $p > 0$ ;
- $R_{i,p}(u) = 0$  para  $u$  fora do intervalo  $[u_i, u_{i+p+1}]$ ;
- $R_{i,p} = N_{i,p}$ , se  $w_i = 1$  para todo  $i$ .

Por causa dessas propriedades da função base, pode-se citar as seguintes propriedades da curva NURBS:

- $C(0) = P_0$  e  $C(1) = P_n$ , ou seja, os pontos inicial e final da curva são coincidentes com os pontos inicial e final do polígono de controle;
- A curva pode ser modificada localmente, isto é, movendo o ponto  $P_i$  ou alterando o peso de um ponto de controle modifica a curva apenas no intervalo  $[u_i, u_{i+p+1}]$ ;
- Podem existir vários pontos de controle coincidentes.

A figura 2.6 mostra diferentes curvas NURBS para diferentes pesos do ponto de controle 3.

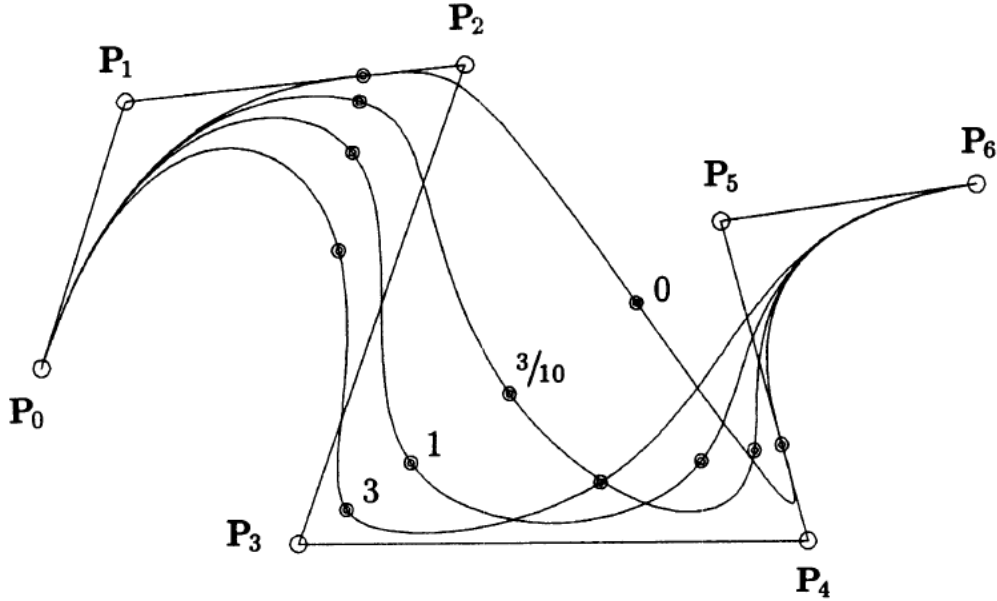


Figura 2.6: Curvas NURBS com o peso do ponto 3 variando. Fonte: Piegl, L.; Tiller, W., *The NURBS Book*. 2ª Edição. Springer, 1997, p. 120

## 2.6 Superfícies NURBS

As superfícies NURBS são as mais usadas para representar superfícies em desenhos assistidos por computadores. Isso acontece porque as NURBS conseguem representar diversas superfícies, como cilindros, esferas, elipsoides de revolução e outras superfícies mais complexas, de forma simples e exata. Uma superfície NURBS é obtida pelo produto tensorial entre curvas NURBS, usando dois parâmetros independentes  $u$  e  $v$ :

$$S(u, v) = \sum_{i=1}^n \sum_{j=1}^m R_{i,j}(u, v) P_{i,j} \quad (2.20)$$

Na qual  $P_{i,j}$  são os pontos de controle e  $R_{i,j}(u, v)$  é a função base racional dada por:

$$R_{i,j}(u, v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^n \sum_{l=0}^m N_{k,p}(u)N_{l,q}(v)w_{k,l}} \quad (2.21)$$

onde  $w_{i,j}$  e  $w_{k,l}$  são os pesos e  $N_{i,p}(u)$  e  $N_{j,q}(v)$  são as funções de base. Os vetores nós são dados por:

$$\mathbf{U} = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1}\} \quad (2.22)$$

$$\mathbf{V} = \{\underbrace{0, \dots, 0}_{q+1}, u_{q+1}, \dots, u_{s-q-1}, \underbrace{1, \dots, 1}_{q+1}\} \quad (2.23)$$

$\mathbf{U}$  possui  $r + 1$  nós e  $\mathbf{V}$  possui  $s + 1$  nós. Um exemplo de superfície NURBS pode ser visto na figura 2.7.

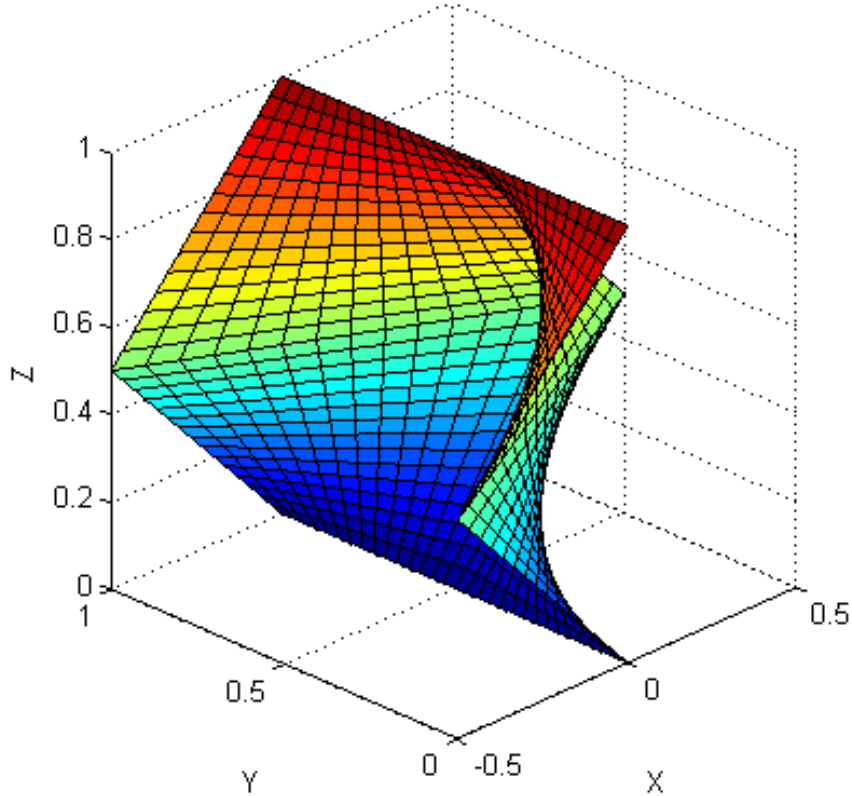


Figura 2.7: *Superfície NURBS.*

As funções de base racional possuem as seguintes características:

- $R_{i,j}(u, v) \geq 0$  para todo  $i, j, u$  e  $v$ ;
- $\sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) = 1$  para todo  $(u, v)$  dentro do intervalo  $[0, 1] \times [0, 1]$ ;
- $R_{i,j}(u, v) = 0$  se  $(u, v)$  está fora do retângulo dado por  $[u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$ ;
- Se  $p > 0$  e  $q > 0$ , então a função base atingirá apenas um valor máximo;
- $R_{0,0}(0, 0) = R_{n,0}(1, 0) = R_{0,m}(0, 1) = R_{n,m}(1, 1) = 1$ ;
- Dentro do retângulo formado pelos vetores nós  $\mathbf{U}$  e  $\mathbf{V}$ , todas as derivadas parciais de  $R_{i,j}(u, v)$  existem. Em um nó  $u$  ( $v$ ) a função é diferenciável  $p - k$  ( $q - k$ ) vezes na direção  $u$  ( $v$ ), onde  $k$  é a multiplicidade do nó.

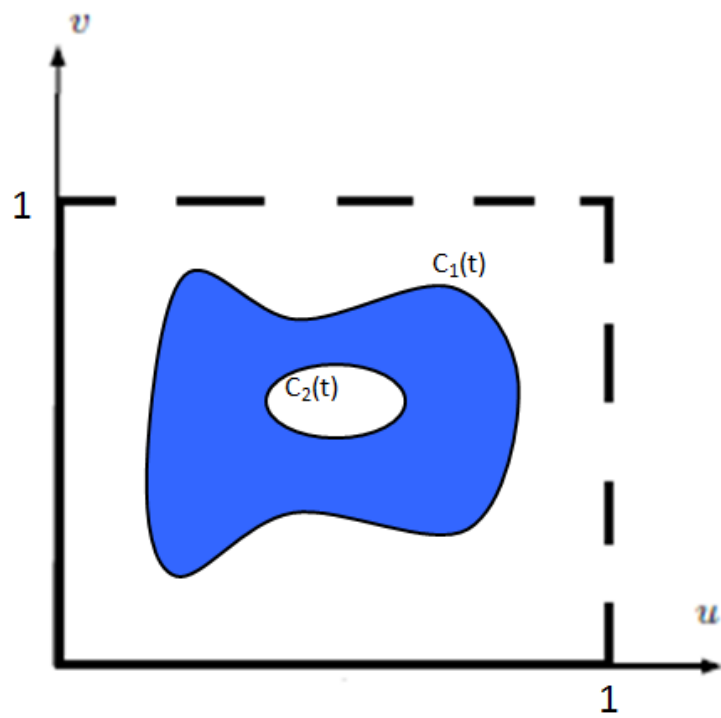
As superfícies NURBS possuem as seguintes propriedades:

- A superfície interpola os pontos de controle dos quatro vértices, ou seja,  $S(0,0) = P_{0,0}$ ,  $S(1,0) = P_{n,0}$ ,  $S(0,1) = P_{0,m}$  e  $S(1,1) = P_{n,m}$ ;
- Se o ponto de controle  $P_{i,j}$  é movido, ou o peso  $w_{i,j}$  é mudado, a superfície será alterada apenas no retângulo  $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$ ;
- A superfície fica dentro do polítopo de controle formado pelos pontos de controle.

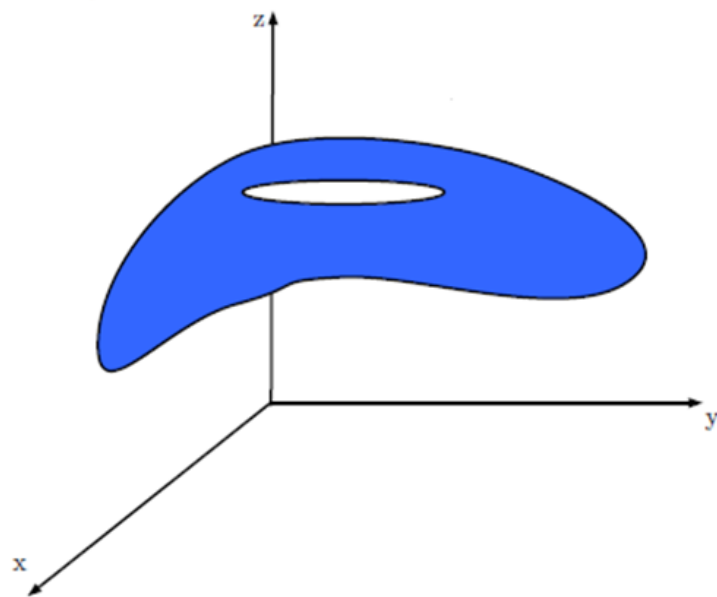
## 2.7 Superfícies NURBS aparadas

Como visto na seção anterior, as superfícies NURBS são obtidas através do produto tensorial de duas curvas NURBS. Esse produto não consegue representar uma superfície com furo, por exemplo. Uma opção para representar uma superfície com furo é dividi-la e representar cada parte como uma superfície NURBS. Porém, esta opção é de difícil implementação e na interseção das superfícies a continuidade das derivadas não é garantida.

Para a representação de formas mais complexas, que possuem união de superfícies, os programas CAD utilizam as superfícies aparadas (*trimmed surfaces*). As superfícies aparadas são compostas de uma superfície NURBS não aparada e um conjunto de curvas aparadoras (*trimming curves*). Uma superfície aparada possui certas áreas que são marcadas como inválidas ou invisíveis por um conjunto de curvas aparadoras. Em um arquivo IGES, além da informação sobre a superfície aparada também vem as informações da superfície não aparada e das curvas aparadoras. A figura 2.8(a) representa a projeção da superfície aparada no plano  $(u, v)$ . Os contornos, que são as curvas de apartamento  $C_1$  e  $C_2$ , são funções de um terceiro parâmetro  $t$  que varia entre 0 e 1. Aplicando a equação (2.20), obtém a superfície no espaço  $(x, y, z)$ , sendo que cada ponto da superfície no plano  $(u, v)$  corresponde a um, e somente um, ponto na superfície no espaço  $(x, y, z)$ .



(a)



(b)

Figura 2.8: Mapeamento de uma superfície aparada.



### 3.1 Introdução

O formato IGES (*Initial Graphics Exchange Specification*) é um formato de dados neutro que permite a troca digital de informação entre sistemas CAD. Ele é o mais utilizado para troca de dados entre os sistemas atuais de CAD/CAM/CAE. Ele foi desenvolvido em janeiro de 1980 com o objetivo de representar desenhos mecânicos bidimensionais e tridimensionais e tornou-se um formato padrão em setembro de 1981. O arquivo IGES tem a extensão *.igs* e é baseado no código padrão ASCII, que permite que o IGES seja lido por todos os editores de texto.

Em IGES, uma curva NURBS é definida por:

- O grau  $p$  e o número de pontos de controle,  $n + 1$ ;
- Pontos de controle  $P_i$  e os pesos,  $w_i$ ;
- Vetor de nós  $\mathbf{U}$  contendo  $M + 1 = n + p + 2$  nós;
- Valores dos parâmetros inicial e final,  $s_0$  e  $s_1$ .

Em um arquivo IGES, os pontos de controle e os pesos são itens separados. Portanto, não existe o conceito de pontos de controle homogêneos,  $P_i^w$ . Só são permitidos pesos positivos.

### 3.2 Formato do arquivo IGES

O arquivo IGES é dividido em seções como pode ser visto na figura 3.1. As duas primeiras seções contêm informações gerais que não tem importância para a simulação. A *Start Section* contêm uma introdução do arquivo. A *Global Section* contêm as características gerais do arquivo. Por exemplo, o programa gerador, as unidades, a data de criação, entre outras informações.

A *Directory Entry Section* serve como um índice para o arquivo contendo uma entrada para cada entidade no arquivo. A *Parameter Data Section* contêm as informações associadas para cada entidade. Ou seja, essa é a seção mais importante do arquivo. A *Terminate Section* (última linha do arquivo) serve como um resumo do arquivo contendo quantas linhas cada seção possui.

1	8 9	16 17	24 25	32 33	40 41	48 49	56 57	64 65	72	73	80
<b>Start Section</b> – a human readable prologue to the file.										S0000001	
It contains one or more lines										S0000002	
⋮										S0000003	
using ASCII characters in columns 1–72.										⋮	
										S000000N	
<b>Global Section</b> – sending system and file information.										G0000001	
It contains the number of lines needed to hold the parameter fields, separated by										G0000002	
⋮										G0000003	
parameter delimiters, and terminated by one record delimiter, in columns 1–72.										⋮	
										G000000N	
<b>Directory Entry Section</b> – contains one pair of lines for each entity.											
Directory entry fields 1-9 in nine 8-column-wide fields										D0000001	
Directory entry fields 10-18 in nine 8-column-wide fields										D0000002	
<b>Parameter Data Section</b> – values and parameter delimiters terminated by one record delimiter, in columns 1-64; column 65 is unused									DE back	P0000001	
									Pointer	P0000002	
S0000020 G0000003D0000500 P0000261				<b>Terminate Section</b> – record counts for preceding sections; columns 33–72 unused						T0000001	

Figura 3.1: *Formato padrão de um arquivo IGES. Fonte: The Initial Graphics Exchange Specification (IGES) Version 6.0, 2001.*

### 3.3 Principais Entidades

Como pode ser visto na tabela abaixo, existem diversos tipos de entidades que podem representar curvas e superfícies. Essa seção não falará de todos os tipos de entidade, serão mostradas apenas as principais utilizadas no programa.

Tabela 3.1: Tipos de entidade

Número do tipo de entidade	Tipo de entidade
100	<i>Circular Arc</i>
102	<i>Composite Curve</i>
104	<i>Conic Arc</i>
106	<i>Copious Data</i>
106/11	<i>2D Linear Path</i>
106/12	<i>3D Linear Path</i>
106/63	<i>Simple Closed Planar Curve</i>
108	<i>Plane</i>
110	<i>Line</i>
112	<i>Parametric Spline Curve</i>
114	<i>Parametric Spline Surface</i>
116	<i>Point</i>
118	<i>Ruled Surface</i>
120	<i>Surface of Revolution</i>

Tabela 3.1: Tipos de entidade

Número do tipo de entidade	Tipo de entidade
122	<i>Tabulated Cylinder</i>
124	<i>Transformation Matrix</i>
125	<i>Flash</i>
126	<i>Rational B-Spline Curve</i>
128	<i>Rational B-Spline Surface</i>
130	<i>Offset Curve</i>
140	<i>Offset Surface</i>
141	<i>Boundary</i>
142	<i>Curve on a Parametric Surface</i>
143	<i>Bounded Surface</i>
144	<i>Trimmed Parametric Surface</i>
190	<i>Plane Surface</i>
192	<i>Right Circular Cylindrical Surface</i>
194	<i>Right Circular Conical Surface</i>
196	<i>Spherical Surface</i>
198	<i>Toroidal Surface</i>

### 3.3.1 Rational B-Spline Surface Entity (Type 128)

Esta entidade representa diversas superfícies analíticas de interesse genérico. Sua informação é importante tanto para o sistema de geração (softwares CAD), quanto para o sistema de recebimento (MATLAB). O número de parâmetro do formato do diretório é utilizado para a comunicação desta informação.

Se a *Rational B-spline surface* representar um tipo de superfície escolhida pelo usuário, o número do formato corresponderá ao tipo desejado. A ordem preferencial vai de 1 a 9, seguida do número 0. Por exemplo, se a superfície é um cilindro circular reto, o número do formato deverá ser definido como 2. Caso a superfície não seja um dos tipos preferenciais o número de forma deverá ser definido como 0.

Se para cada valor fixado da segunda variável paramétrica, as curvas resultantes que sejam funções da primeira variável paramétrica são fechadas, PROP1 deverá ser definido como 1; Caso contrário, PROP1 deverá ser definido como 0; De maneira similar, se para cada valor fixado da primeira variável, as curvas resultantes que sejam funções da segunda variável paramétrica são fechadas, PROP2 deverá ser definido como 1; Caso contrário, PROP2 deverá ser definido como 0.

Se a superfície é racional, ou seja não possui todos os pesos iguais, PROP3 é definido como 0. Se todos os pesos forem iguais, a superfície é polinomial e PROP3 é definida como 1. Os pesos tem que ser números reais positivos.

Se a superfície é periódica com respeito a primeira variável paramétrica, PROP4 deverá ser definido como 1. Caso contrário, PROP4 deverá ser definido como 0. Caso a superfície apresente periodicidade em relação a segunda variável paramétrica, PROP5 deverá ser definido como 1. Caso contrário deverá ser definido como 0. As classificações de periodicidade são de caráter meramente informativo. As superfícies classificadas como periódicas são avaliadas exatamente da mesma maneira que as não periódicas.

A tabela 3.2 mostra os números de forma e seu significado e a tabela 3.3 resume as variáveis da entidade.

Tabela 3.2: *Rational B-Spline Surface*: Números de forma

Forma	Significado
0	A forma da superfície é determinada pelos parâmetros da <i>rational B-spline Plane</i>
1	Plano
2	Cilindro circular reto
3	Cone
4	Esfera
5	Toróide
6	Superfície de revolução
7	Cilindro extrudado
8	Superfície regrada
9	Superfície geral quádrlica

Tabela 3.3: *Rational B-Spline Surface*: *ParameterData*

Índice	Nome	Tipo	Descrição
1	K1	Número inteiro	Índice superior da primeira soma
2	K2	Número inteiro	Índice superior da segunda soma
3	M1	Número inteiro	Ordem do primeiro conjunto de funções de base
4	M2	Número inteiro	Ordem do segundo conjunto de funções de base
5	PROP1	Número inteiro	1 = Fechado na direção da primeira variável paramétrica 0 = Aberto
6	PROP2	Número inteiro	1 = Fechado na direção da primeira variável paramétrica 0 = Aberto
7	PROP3	Número inteiro	1 = Polinomial 0 = Racional
8	PROP4	Número inteiro	0 = Não periódica na direção da segunda variável paramétrica 1 = periódica na direção da segunda variável paramétrica
9	PROP5	Número inteiro	0 = Não periódica na direção da segunda variável paramétrica 1 = periódica na direção da segunda variável paramétrica

### 3.3.2 Rational B-Spline Curve Entity (Type 126)

Esta entidade pode representar diversas curvas analíticas de interesse genérico. Sua informação é importante tanto para o sistema de geração quanto para o sistema de recebimento. O número de parâmetro do formato do diretório é utilizado para a comunicação desta informação.

Caso a *Rational B-spline curve* represente um tipo de curva escolhida pelo usuário, o número do formato corresponderá ao tipo informado. A ordem preferencial vai de 1 a 5, seguida do número 0. Se a curva for um círculo ou um arco circular por exemplo, o número do formato deve ser definido como 2. Se a curva não for um dos tipos preferenciais o número de forma deverá ser definido como 0.

Se a curva estiver inserida em apenas um plano, PROP1 deverá ser definido como 1, caso contrário deverá ser definido como 0. Se os pontos inicial e final da curva são coincidentes, a curva é fechada e PROP2 deve ser definido como 1. Se os pontos não são coincidentes, PROP2 deve ser definido como 0.

Se a curva é racional, ou seja não possui todos os pesos iguais, PROP3 deverá ser definido como 0. Se todos os pesos forem iguais, a superfície é polinomial e PROP3 deverá ser definida como 1. Os pesos são números reais positivos.

Se a curva é periódica com respeito a sua variável paramétrica, PROP4 deverá ser definido como 1. Caso contrário, PROP4 deverá ser definido como 0. As classificações de periodicidade são de caráter meramente informativo. As curvas classificadas como periódicas são avaliadas exatamente da mesma maneira que as não periódicas.

Na Tabela 3.4 são mostrados os números de forma junto com seu significado e na Tabela 3.5 resume as variáveis do ParameterData.

Tabela 3.4: *Rational B-Spline Curve*: Números de forma

Forma	Significado
0	A forma da superfície é determinada pelos parâmetros da <i>rational B-spline</i>
1	Linha
2	Arco de círculo
3	Arco elíptico
4	Arco parabólico
5	Arco hiperbólico

Tabela 3.5: *Rational B-Spline Curve*: ParameterData

Índice	Nome	Tipo	Descrição
1	K	Número inteiro	Índice superior da soma
2	M	Número inteiro	Ordem das funções base
3	PROP1	Número inteiro	0 = Não planar 1 = Planar
4	PROP2	Número inteiro	0 = Curva aberta 1 = Curva fechada
5	PROP3	Número inteiro	0 = Racional 1 = polinomial
6	PROP4	Número inteiro	0 = Não periódica 1 = Periódica

### 3.3.3 Line Entity (Type 110)

Uma linha é definida pelo seus pontos de início e fim. Cada um desses pontos é especificado em relação ao espaço por um sistema de três coordenadas. Em relação a definição espacial, uma direção é primeiramente associada com a linha a partir de seu ponto inicial, e em segundo lugar com seu ponto final.

A direção da linha com respeito ao modelo espacial adotado, é determinado a partir da direção original da linha dentro do espaço definido em conjunto com a transformação matricial na linha. A tabela 3.6 resume as variáveis do *ParameterData*.

Tabela 3.6: *Line Entity: ParameterData*

Índice	Nome	Tipo	Descrição
1	X1	Número real	Coordenada $x$ ponto inicial P1
2	Y1	Número real	Coordenada $y$ ponto inicial P1
3	Z1	Número real	Coordenada $z$ ponto inicial P1
4	X2	Número real	Coordenada $x$ ponto final P2
5	Y2	Número real	Coordenada $y$ ponto final P2
6	Z2	Número real	Coordenada $z$ ponto final P2

### 3.3.4 Composite Curve Entity (Type 102)

Uma *composite curve* é uma curva contínua formada pela junção de certas entidades em uma unidade lógica. Um exemplo de *composite curve* pode ser visto na figura 3.2.

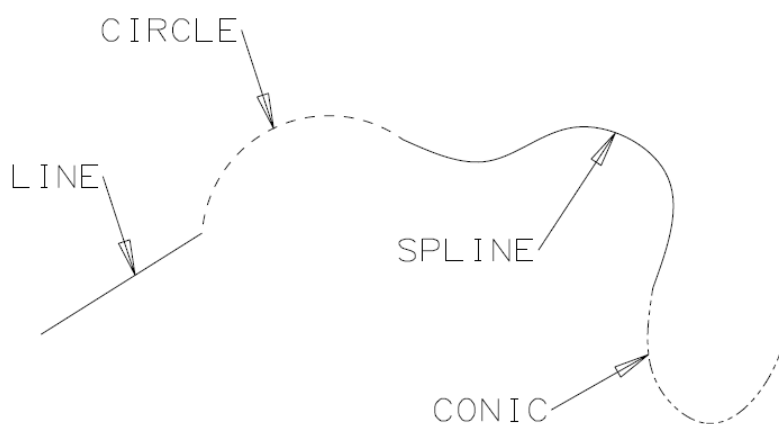


Figura 3.2: *Exemplo de uma composite curve.*

Uma *composite curve* é definida como uma lista ordenada de entidades que consistem em pontos, pontos conectados, e entidades de curvas parametrizadas. A lista de entidades aparece na entrada dos parâmetros de dados. Cada entidade na lista de definição é indicada por meio de um ponteiro a um diretório de entrada dessa entidade. A ordem da lista de definição é a mesma que a ordem da listagem desses ponteiros.

Cada entidade constituinte tem sua própria matriz de transformação e atributos de exibição. Ela poderá ter propriedades ou textos associados. Devido as entidades constituintes estarem subordinadas a uma *composite entity*, a chave de entidade subordinada de cada entidade constituinte deverá indicar uma dependência física.

Uma *composite curve* é uma curva direcionada, tendo um ponto inicial e um ponto final. A direção da *composite curve* é definida pela direção das curvas que a constituem da seguinte maneira: O ponto inicial da *composite curve* é o ponto inicial da primeira entidade de curva que aparece na lista de definição. O ponto final da *composite curve* é o ponto final da última entidade de curva que aparece na lista. Dentro da própria lista de definição, o ponto final da curvas que compõe a *composite curve* é o ponto inicial da próxima curva.

As entidades ponto e ponto de conexão estão incluídas como tipos de entidades permitidas das quais as propriedades ou notas gerais podem estar anexadas ao ponto inicial ou final de qualquer entidade de curva constituída na lista de definição.

Uma relação de conexão lógica pode ser indicada a partir de duas *composite curves* ou uma *composite curve* e uma rede de subfiguras referenciadas a entidade do ponto de conexão. Para o caso especial de conexão lógica de um ponto de conexão em uma subfigura a um ponto de conexão em uma outra subfigura, é permitida uma *composite curve* cuja lista contenha duas entidades de pontos de conexão sem curvas de intervenção. Neste caso, o exemplo da entidade de *composite curve* não é uma curva no sentido comum, não é contínua e não possui comprimento de arco. Este uso é permitido apenas em algumas aplicações (como, por exemplo, no método dos elementos finitos). Existem algumas restrições com respeito ao uso da entidade ponto em uma entidade *composite*. São elas:

- As entidades pontos de conexão e dois pontos não podem aparecer consecutivamente na lista de definição a não ser que elas sejam as únicas entidades presentes na *composite curve*. Tais *composite curves* usadas como conectores lógicos devem apresentar uma marca de uso de valor = 04 (lógica/posicional).
- Se um ponto de conexão, um ponto e uma curva são adjacentes na lista de definição, então as coordenadas do ponto de conexão ou do ponto devem combinar com as coordenadas finais da curva sempre que a curva anteceder o ponto de conexão ou o ponto, e também deve combinar com as coordenadas do ponto inicial da curva sempre que a curva vier em seguida ao ponto de conexão ou ao ponto.
- Uma *composite curve* não pode consistir em uma entidade de apenas um ponto ou de apenas um ponto de conexão.

Se necessário, a parametrização de uma *composite curve* é obtida através das curvas constituintes. Como um ponto e um ponto de conexão não contribuem para a parametrização, eles não considerados nesta definição. A tabela 3.7 mostra a nomenclatura utilizada em uma *composite curve*.

Tabela 3.7: Nomenclatura usada em uma *composite curve*

C	Composite curve
N	número de curvas constituintes ( $N \geq 1$ )
CC(i)	i-ésima curva constituinte, para todo $i$ que estiver no intervalo $1 \leq i \leq N$
PS(i)	Valor paramétrico do começo de CC(i)
PE(i)	Valor paramétrico do final de CC(i)
T(0)	0.0
T(1)	$\sum_{j=1}^i (PE(j) - PS(j))$ , para todo $i$ que estiver no intervalo $1 \leq i \leq N$

A tabela 3.8 resume as variáveis do *ParameterData*.

Tabela 3.8: *Composite curve: ParameterData*

Índice	Nome	Tipo	Descrição
1	N	Número inteiro	Número de entidades
2	DE(1)	Ponteiro	Ponteiro para o DE da primeira entidade constituinte
⋮	⋮	⋮	⋮
1 + N	DE(N)	Ponteiro	Ponteiro para o DE da última entidade constituinte

### 3.3.5 Curve on a Parametric Surface Entity (Type 142)

A entidade *curve on a parametric surface* associa uma curva com uma superfície e identifica a curva como composição desta. Considerando que  $S = S(u, v) = (x(u, v), y(u, v), z(u, v))$  é uma parametrização da superfície cujo domínio é um retângulo definido por  $D = \{(u, v) \mid u_1 \leq u \leq u_2 \text{ e } v_1 \leq v \leq v_2\}$

Seja  $B = B(t)$  uma curva definida por  $B(t) = (u(t), v(t))$  para  $a \leq t \leq b$ , com seus valores contidos no domínio  $D$ .

Uma curva  $C_c(t)$  na superfície  $S(u, v)$  é a composição de dois mapeamentos,  $S$  e  $B$ , definidos como se segue ( $C_c(t)$  representa uma "*composition curve*"):

$$\begin{aligned}
 C_c(t) &= S \circ B(t) \\
 C_c(t) &= S(B(t)) \\
 C_c(t) &= S(u(t), v(t)) \\
 C_c(t) &= (x(u(t), v(t)), y(u(t), v(t)), z(u(t), v(t))) \quad a \leq t \leq b
 \end{aligned}$$

A curva  $B$  encontra-se no espaço bidimensional que é o domínio da superfície  $S$ . Portanto, a representação usada para  $B$ , a qual foi derivada de curva definida nesta especificação, deve ser bidimensional.

Uma *curve on parametric surface* é dada por:

- Mapeamento de  $C_c$  e a indicação que a curva encontra-se na superfície  $S(u, v)$ ;
- Pelo mapeamento de  $B$  e  $S$  cuja composição nos dá a curva  $C_c$ .



Uma *curve on parametric surface* pode ser criada de várias maneiras:

- Como projeção em uma superfície de uma curva dada em um modelo espacial numa maneira pré-determinada, por exemplo, paralela a um vetor fixo;
- Como intersecção de duas superfícies;
- A partir de uma função pré-determinada relacionando os parâmetros  $u$  e  $v$  da superfície;
- A partir de uma curva especial, tal como uma geodésica, surgindo de um dado ponto em uma certa direção, uma curva principal (linha de curvatura) surgindo de um certo ponto, uma curva assintótica surgindo de um certo ponto, uma curva isoparamétrica para um dado valor, ou qualquer outro tipo de curva especial.

A variável *ParameterData* contém três ponteiros:

- Um ponteiro apontado para curva da qual  $B(t)$  é derivada;
- Um ponteiro apontado para a superfície  $S(u, v)$ ;
- Um ponteiro para o mapeamento de  $C(r)$ , tal que:
  - $C(r)$  e  $C_c(t)$  compartilham a mesma imagem no modelo espacial.
  - $C(r)$  e  $C_c(t)$  possuem os mesmos pontos inicial e final.
  - Existe uma relação matemática implícita entre os parâmetros  $t$  e  $r$ .
  - $C(r)$  e  $C_c(t)$  devem ser tais que  $t$  é relacionado a  $r$  em um crescimento monótono de forma. Isso garante que as orientações de  $C(r)$  e  $C_c(t)$  coincidam, e não ocorre rastreamento múltiplo acidental.

Além destes ponteiros a *ParameterData* também contém:

- Uma marca de indicação de como a curva foi criada;
- Uma marca para indicar qual das duas representações foi determinada pelo sistema de envio.

Tabela 3.9: *Curve on a Parametric Surface: ParameterData*

Índice	Nome	Tipo	Descrição
1	CRTN	Número inteiro	Indica a maneira que a curva foi criada na superfície: 0 = Não especificada 1 = Projeção da curva dada na superfície 2 = Interseção de duas superfícies 3 = Curva isoparamétrica, ou seja, uma $u$ -paramétrica ou uma $v$ -paramétrica
2	SPTR	Ponteiro	Apontado para DE (Directory Entry) da superfície na qual a curva se encontra
3	BPTR	Ponteiro	Apontado para DE da entidade que contém a definição da curva $B$ no espaço paramétrico $(u, v)$ da superfície $S$
4	CPTR	Ponteiro	Apontado para DE da curva $C$
5	PREF	Número inteiro	Indica a representação escolhida no sistema de envio: 0 = Não especificada 1 = $S \circ B$ é escolhida 2 = $C$ é escolhida 3 = $C$ e $S \circ B$ são igualmente escolhidas

### 3.3.6 Trimmed (Parametric) Surface Entity (Type 144)

Uma simples curva fechada no plano euclidiano divide o mesmo em dois componentes abertos conectados e disjuntos, onde um é delimitado e o outro não. O elemento delimitado é chamado de região interna à curva, já o elemento não delimitado é chamado de região externa à curva.

O domínio da superfície aparada é definido como a região comum do interior da fronteira externa e o exterior de cada fronteira interna incluindo as curvas de fronteira. Note que a superfície aparada é definida da mesma forma que a superfície original  $S(u, v)$  (superfície não aparada), mas possui um domínio diferente. As curvas que delimitam a superfície aparada tanto externamente quanto internamente são curvas que se encontram na superfície  $S$ . Estas, por sua vez, são substituídas por meio de curvas na superfície paramétrica (Tipo 142).

Seja  $S(u, v)$  uma superfície regular parametrizada, cujo domínio não aparado é um retângulo  $D$  que consiste daqueles pontos  $(u, v)$  tais que  $a \leq u \leq b$  e  $c \leq v \leq d$  para dadas constantes  $a, b, c$  e  $d$  com  $a < b$  e  $c < d$ . Considere que a superfície  $S$  tem seus valores determinados em um espaço euclidiano tridimensional e pode ser expressada como:

$$S = S(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix}$$

para cada par ordenado  $(u, v)$  em  $D$ .

Seja também  $S$  sujeita as seguintes condições de regularidade:

- Possui um vetor normal contínuo no interior de  $D$ ;
- Cada elemento de  $S$  possui um e somente um elemento correspondente em  $D$ ;
- Não existem pontos de singularidade em  $D$ , ou seja, os vetores das primeiras derivadas parciais de  $S$  em cada ponto de  $D$  são linearmente independentes.

Dois tipos simples de curvas fechadas são utilizadas para definir o domínio da superfície aparada.

- Fronteira externa: Existe exatamente uma, pertencente a  $D$ , que em particular, pode ser a curva de fronteira de  $D$ .
- Fronteira interna: Poderá haver um número qualquer destas, até mesmo nenhuma. A configuração das fronteiras internas satisfazem dois critérios:
  - As curvas, bem como seus interiores, são mutuamente disjuntos (não possuem elementos em comum);
  - Cada curva encontra-se no interior da fronteira externa.

Se a fronteira externa da superfície está definida como a fronteira de  $D$  e não há fronteiras internas, a superfície aparada é não aparada. A tabela 3.10 resume as variáveis do *ParameterData*.

Tabela 3.10: *Trimmed (Parametric) Surface: ParameterData*

Índice	Nome	Tipo	Descrição
1	PTS	Ponteiro	Aponta para DE da superfície que será aparada
2	N1	Número inteiro	0 = A fronteira externa é a fronteira $D$ 1 = Caso contrário
3	N2	Número inteiro	Este número indica a quantidade de curvas simples fechadas que compõem a fronteira interna de uma superfície aparada. Caso nenhuma fronteira interna seja definida, este número será igual a zero.
4	PT0	Ponteiro	Aponta para DE da <i>Curve on a parametric surface entity</i> que compõe a fronteira externa de uma superfície aparada ou zero.
5	PTI(1)	Ponteiro	Aponta para DE da primeira curva simples fechada de fronteira interna ( <i>Curve on a parametric surface entity</i> ) de acordo com uma ordem arbitrária destas entidades.
⋮	⋮	⋮	⋮
4 + N2	PTI(N2)	Ponteiro	Aponta para DE da última curva simples fechada de fronteira interna ( <i>Curve on a parametric surface entity</i> )

### 3.3.7 Transformation Matrix Entity (Type 124)

A *Transformation Matrix Entity* transforma vetores coluna de três linhas por meio da multiplicação de uma matriz e a adição de um vetor. A notação para essa transformação é a seguinte:

$$\begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} \\ R_{2,1} & R_{2,2} & R_{2,3} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix} \begin{bmatrix} XINPUT \\ YINPUT \\ ZINPUT \end{bmatrix} + \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} XOUTPUT \\ YOUTPUT \\ ZOUTPUT \end{bmatrix}$$

A coluna  $[XINPUT, YINPUT, ZINPUT]$ , ou seja, o vetor coluna, é o vetor a ser modificado, e a coluna  $[XOUTPUT, YOUTPUT, ZOUTPUT]$  é o vetor coluna resultante dessa transformação.  $R = [R_{ij}]$  é uma matriz quadrada  $3 \times 3$  composta de números reais, e  $T = [T_1, T_2, T_3]$  é um vetor coluna de três linhas composto de números reais. Assim, são necessários 12 números reais para uma *Transformation Matrix Entity*. Essa entidade pode ser considerada uma entidade operador que se inicia com um vetor de entrada, executa as operações descritas anteriormente, e gera o vetor de saída ou vetor resultante.

Frequentemente, o vetor de entrada lista as coordenadas de alguns pontos em um sistema de coordenadas, e o vetor de saída lista as coordenadas deste mesmo ponto em um segundo sistema de coordenadas. Desta forma, a matriz  $R$  e o vetor de conversão  $T$  expressam uma relação genérica entre os dois sistemas de coordenadas. Considerando os vetores de entrada como vetores coluna  $[1, 0, 0]$ ,  $[0, 1, 0]$  e  $[0, 0, 1]$ , e calculando os resultados, pode-se ilustrar a relação espacial entre os dois sistemas de coordenadas. Um exemplo pode ser visto na figura 3.3, que utilizou os seguintes:

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}, \mathbf{T} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

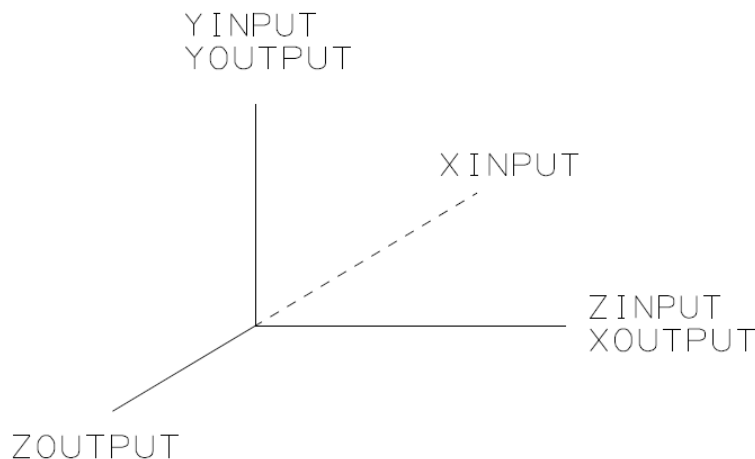


Figura 3.3: Exemplo de transformação de um sistema de coordenadas matricial.

Todos os sistemas de coordenadas aqui são adotados como ortogonais, cartesianos e destros, a não ser que uma consideração específica seja feita em relação a estas características.

A seguir serão mostradas três situações em que uma *Transformation Matrix Entity* é usada para transformar sistemas de coordenadas. Cada exemplo ilustra uma escolha específica de entrada e saída de sistemas de coordenadas. Outras escolhas de sistemas de coordenadas podem ser convenientes em outras aplicações.

A primeira situação para esse uso da *Transformation Matrix Entity* é quando o vetor de entrada se refere a um sistema de coordenadas espaciais definido para uma certa entidade, e o vetor de saída refere-se a um sistema modelo de coordenadas espaciais. Neste caso, a matriz  $R$  é chamada de matriz de definição e está sujeita as restrições apresentadas na forma 0 e da forma 1 que serão definidas mais adiante.

Um segunda situação se dá no caso em que o vetor de entrada refere-se ao modelo de coordenadas espaciais e o vetor de saída refere-se à um sistema de coordenadas visualizável. Neste caso, a matriz  $R$  é chamada de *view matrix*, e está sujeita as restrições apresentadas na forma 0. Note que quando uma entidade plana é vista em comprimento verdadeiro (ou seja, o plano visualizado é paralelo ao plano contendo a entidade), a matriz de rotação apontada para entidade plana pelo "DE Field 7", será o inverso (igual a matriz transposta) da matriz apontada para a "View Entity".

Uma terceira situação se dá envolvendo aplicações do método de elementos finitos. Aqui, pode-se estudar o caso em que um sistema de coordenadas de entrada está relacionado a um sistema de coordenadas de saída por uma matriz particular  $R$  e  $T$ , e, por sua vez, o sistema de coordenadas de saída é usado como um sistema de coordenadas de entrada para uma segunda combinação de  $R$  e  $T$ , e assim por diante. Estes sistemas de coordenada são frequentemente chamados de sistemas de coordenadas locais. Já um modelo espacial é geralmente chamado de sistema de referência. Por exemplo, a localização de um nó de referência em um elemento finito pode ser dado em um sistema de coordenadas local, que pode servir como um sistema de coordenadas de entrada para um segundo sistema de coordenadas local, o qual por sua vez pode servir como um sistema de coordenadas de entrada para um sistema de coordenadas do modelo espacial, que é o sistema de referência.

Sempre que sistemas de coordenadas estão relacionados sucessivamente uns aos outros como descrito acima, um resultado fundamental é que o efeito combinado das mudanças do sistema de coordenadas individual pode ser expressado em termos de uma única matriz  $R$  e um vetor transposto  $T$ . Por exemplo, se a mudança de um sistema de coordenada envolvendo a matriz  $R_2$  e o vetor transposto  $T_2$  é aplicado seguindo a mudança de coordenada envolvendo a matriz  $R_1$  e o vetor transposto  $T_1$ , então a matriz  $R$  e o vetor transposto  $T$  que expressam as mudanças combinadas são  $R = R_2 \times R_1$  e  $T = R_2 \times T_1 + T_2$ .

Sucessivas mudanças em sistemas de coordenadas são especificadas a partir de uma *Transformation Matrix* referenciando-se a outra *Transformation Matrix* através do "Field 7" do diretório de entrada. No exemplo acima, uma *Transformation Matrix* contendo  $R_1$  e  $T_1$ , teria no

diretório de entrada do "Field 7", um ponteiro direcionado para a *Transformation matrix* contendo  $R_2$  e  $T_2$ . A regra geral é tal que *Transformation matrix Entities* aplicadas primeiramente em sucessão, irão referenciar-se as *Transformation Matrix Entities* aplicadas posteriormente. Note que a matriz resultante do produto  $R_2 \times R_1$  no exemplo acima não aparece explicitamente nos dados, mas, caso necessário, pode ser processada de acordo com as regras usuais de multiplicação de matrizes.

Um segundo exemplo de sistemas de coordenadas sendo relacionados de maneira sucessiva, em complemento ao que fora explicado no exemplo do método de elementos finitos anteriormente, envolve uma maneira de localizar um arco de cone dentro de um modelo espacial que esteja numa posição padrão no espaço definido. Neste caso,  $R_1$  e  $T_1$  movem o cone de sua posição padrão a uma localização arbitrária em qualquer plano no espaço definido, satisfazendo a condição  $ZT = \text{constante}$ . Então  $R_2$  e  $T_2$  posicionam o arco de cone realocado no modelo espacial. Note que para  $R_1$  e  $T_1$  os vetores de entrada e de saída se referem ao mesmo sistema de coordenadas, nomeados como espaço de definição para o arco de cone.

Os números de forma da *Transformation Matrix Entity* são dados na tabela 3.11 e a tabela 3.12 resume as variáveis do *ParameterData*.

Tabela 3.11: *Transformation Matrix*: Números de forma

Número de forma	Uso
0 ou 1	Matriz de definição de uma entidade
10, 11 ou 12	Matrizes especiais representando uma entidade Nodal (Tipo 134)

Tabela 3.12: *Transformation Matrix*: *ParameterData*

Índice	Nome	Tipo	Descrição
1	$R_{11}$	Real	Primeira linha
2	$R_{12}$	Real	Primeira linha
3	$R_{13}$	Real	Primeira linha
4	$T_1$	Real	Primeira linha
5	$R_{21}$	Real	Segunda linha
6	$R_{22}$	Real	Segunda linha
7	$R_{23}$	Real	Segunda linha
8	$T_2$	Real	Segunda linha
9	$R_{31}$	Real	Terceira linha
10	$R_{32}$	Real	Terceira linha
11	$R_{33}$	Real	Terceira linha
12	$T_3$	Real	Terceira linha

### 3.4 IGES toolbox

Esta toolbox tem código aberto e pode ser obtida diretamente da página da MathWorks. Através dessa toolbox, os arquivos IGES provenientes de softwares CAD podem ser lidos e interpretados pelo software MATLAB. Sua principal função "*iges2matlab*" transfere os dados de

um arquivo IGES para uma cell array chamada de *ParameterData*. Nesta seção serão descritas as principais funções da IGES toolbox.

### 3.4.1 Principais funções

- `iges2matlab`: Extrai os parâmetros de dados de um arquivo IGES para uma cell array chamado `ParameterData`.
- `plotIGES`: Plota o modelo CAD proveniente do arquivo IGES.
- `plotIGESentity`: Plota uma dada entidade de um modelo CAD proveniente de um arquivo IGES.
- `plotNURBS`: Plota uma entidade NURBS (superfície ou curva).
- `transformIGES`: Transforma os parâmetros de dados de um modelo CAD proveniente de um arquivo IGES com uma rotação/reflexão ou uma translação.
- `findEntityIGES`: Retorna um vetor com índices que possuem um dado tipo de entidade indicada por um número.
- `projIGES`: Retorna projeções de pontos em uma superfície de um modelo CAD proveniente de um arquivo IGES.
- `projpartIGES`: Retorna pontos de projeção em uma parte da superfície de um modelo CAD proveniente de um arquivo IGES.
- `projpartPerspectiveIGES`: Retorna pontos de projeções em perspectiva em uma parte da superfície de um modelo CAD proveniente de um arquivo IGES.
- `projpartSphericalIGES`: Retorna pontos de projeções esféricas em uma parte da superfície de um modelo CAD proveniente de um arquivo IGES.
- `srfRepInProjection`: Retorna uma representação da superfície usando valores de entrada/saída provenientes do comando `projpartIGES`.
- `linAppSrfIGES`: Retorna uma aproximação de uma superfície.
- `icpSrfLinRep`: Retorna uma representação da superfície.
- `getDVGtree`: Retorna uma DVG tree.
- `"icpDVGpnt2pntLS"` (mex functions) e `"icpDVGpnt2pntTu"`: Algoritmos ICP.
- `stl2matlab`: Extrai dados em formato de arquivos STL para o MATLAB.
- `plotSTL`: Plota dados provenientes de arquivos STL.
- `transformSTL`: Transforma vértices de objetos STL com uma rotação/reflexão e uma translação.
- `igesToolBoxGUI`: Interface gráfica (`igesToolBoxGUI.m` e `igesToolBoxGUI.fig`).

### 3.4.2 Subfunções

- `retSrfCrvPnt`: Retorna valores provenientes das superfícies, curvas e pontos. Uma documentação completa não é apresentada.
- `nrbDerivativesIGES`: Retorna as derivadas de primeira e segunda ordem das NURBS.
- `getSurfaceDistributedPoints`: Retorna pontos distribuídos em uma superfície.
- `nrbCrvPlaneIntrsectIGES` (mex function): Encontra o valor de um parâmetro de uma curva NURBS no ponto mais próximo a um determinado plano.
- `nrbEvalIGES` (mex function): Avalia as NURBS e suas derivadas.
- `nrbSrfRegularEvalIGES` (mex function): Avalia uma superfície NURBS nos seus parâmetros de valores em uma rede regular.
- `closestNrbLinePointIGES` (mex function): Retorna o ponto mais próximo entre um ponto/linha e um trecho de NURBS.
- `LScrvApp` (mex function): Cria matrizes para curvas NURBS caberem no espaço 2D (bidimensional).
- `offsetNURBSsurface` (mex function): Cria uma superfície NURBS deslocada.
- `createDVGtree` (mex function): Cria uma DVG-tree.
- `makeIGESmex`: Arquivo .m para compilação dos arquivos mex.



## 4 Cálculo das Propriedades Geométricas

### 4.1 Elementos de contorno

Considere uma figura de domínio  $A$  e contorno  $s$  qualquer, como mostrado na figura 4.1:

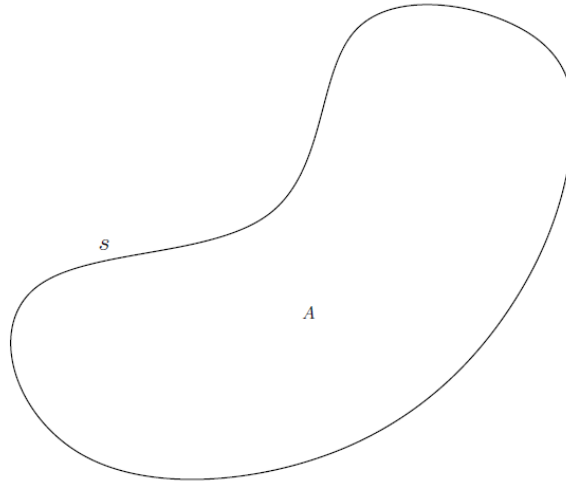


Figura 4.1: *Domínio A.*

Já que fazer uma integral ao longo do contorno  $s$  pode ser bastante difícil, uma estratégia para realizar esta integral é dividir o contorno  $s$  em uma soma de pequenos pedaços  $s_1, s_2, \dots, s_n$ , ou seja  $s = \sum_{i=1}^n s_i$ , onde  $n$  é o número de pedaços em que o contorno foi dividido. Uma vez que estes pedaços podem ter uma forma qualquer, cada pedaço  $s_i$  será aproximado por uma forma conhecida. Por simplicidade, no método dos elementos de contorno padrão, esta forma é quase sempre dada por um polinômio (linha reta, parábola, etc). No caso do método dos elementos de contorno isogeométrico, estes pedaços não são aproximação na maioria dos casos pois, ao invés de se usar polinômios, usa-se as NURBS e estas são capazes de representar exatamente a grande maioria das curvas usadas em engenharia (reta, círculo, parábola, elipses e seus arcos, etc.).

$$s = \sum_{i=1}^n s_i \quad (4.1)$$

onde  $\Gamma$  é o contorno aproximado.

Fazendo  $n$  tender ao infinito, o contorno aproximado fica igual ao contorno  $s$ :

$$s = \lim_{n \rightarrow \infty} \sum_{i=1}^n \Gamma_i \quad (4.2)$$

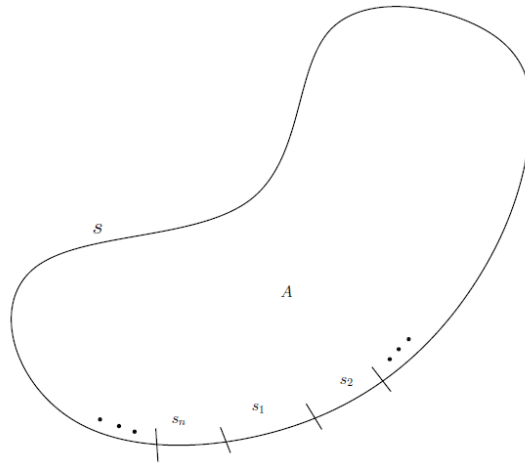


Figura 4.2: Contorno dividido em pequenas partes.

#### 4.2 Método da integração radial (MIR)

O método da integração radial é um método que transforma uma integral de domínio em uma integral de contorno. Considere uma superfície plana qualquer de domínio  $\Omega$  e contorno  $\Gamma$ . A integral de uma função  $f(x, y)$  sobre a área é dada pela equação:

$$I = \iint_{\Omega} f(x, y) dx dy \quad (4.3)$$

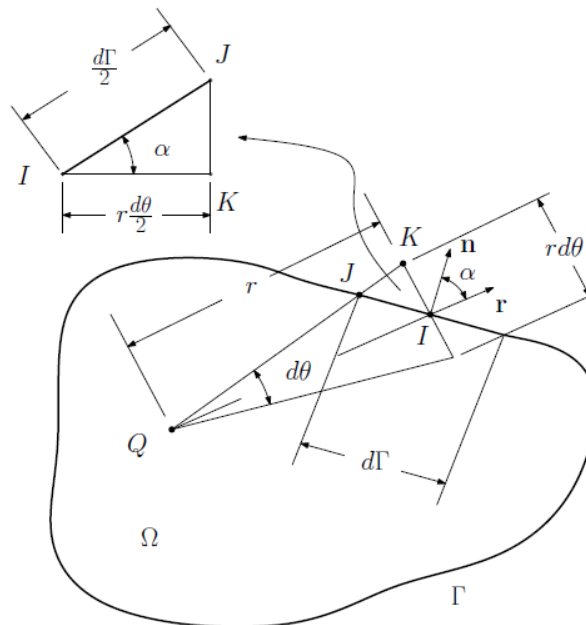


Figura 4.3: Domínio  $\Omega$  e suas relações geométricas.

Escrevendo a equação em coordenadas polares:

$$I = \iint_{\Omega} f(x(\rho, \theta), y(\rho, \theta)) \rho d\rho d\theta \quad (4.4)$$

ou

$$I = \int_{\theta} \int_0^r f(x(\rho, \theta), y(\rho, \theta)) \rho d\rho d\theta \quad (4.5)$$

onde  $r$  é o valor de  $\rho$  em um ponto de contorno  $\Gamma$ .

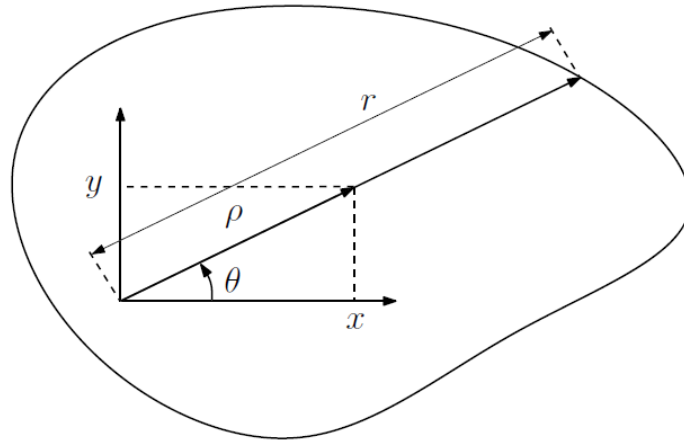


Figura 4.4: Variáveis do método da integração radial.

A equação (4.5) pode ser reescrita na forma:

$$I = \int_{\theta} F(\rho, \theta) d\theta \quad (4.6)$$

onde  $F$  é dado pela equação:

$$F(\rho, \theta) = \int_0^r f(x(\rho, \theta), y(\rho, \theta)) \rho d\rho \quad (4.7)$$

Considerando um ângulo infinitesimal  $d\theta$ , a relação entre o comprimento de arco  $r d\theta$  e o comprimento de contorno infinitesimal  $d\Gamma$  é dada por:

$$\cos \alpha = \frac{r \frac{d\theta}{2}}{\frac{d\Gamma}{2}} \quad (4.8)$$

Isolando o termo  $d\theta$ :

$$d\theta = \frac{\cos \alpha}{r} d\Gamma \quad (4.9)$$

onde  $\alpha$  é o ângulo entre os vetores unitários  $r$  e  $n$ .

Usando as propriedades de produto interno dos dois vetores unitários  $n$  e  $r$  pode-se escrever:

$$d\theta = \frac{\vec{n} \cdot \vec{r}}{r} d\Gamma \quad (4.10)$$

Substituindo a equação (4.10) na equação (4.6), a integral de domínio (4.3) pode ser escrita como uma integral de contorno dada por:

$$I = \int_{\Gamma} F \frac{\vec{n} \cdot \vec{r}}{r} d\Gamma \quad (4.11)$$

### 4.3 Cálculo da área superficial de figuras tridimensionais

Considere uma figura tridimensional de superfície  $S$  qualquer. A sua área superficial é dada por:

$$A_S = \iint_S dS \quad (4.12)$$

Fazendo a transformação de variáveis  $x, y$  e  $z$  para  $u$  e  $v$ , como mostrado na figura 4.5, a equação da área superficial pode ser reescrita na forma:

$$A_S = \iint_S dS = \iint_A J(u, v) du dv \quad (4.13)$$

onde a função  $J(u, v)$  é chamada de jacobiano da transformação e é dada por:

$$J(u, v) = \sqrt{g_1^2 + g_2^2 + g_3^2} \quad (4.14)$$

e  $g_1, g_2$  e  $g_3$  são dados por:

$$g_1 = \frac{\partial y}{\partial u} \frac{\partial z}{\partial v} - \frac{\partial y}{\partial v} \frac{\partial z}{\partial u} \quad (4.15)$$

$$g_2 = \frac{\partial x}{\partial v} \frac{\partial z}{\partial u} - \frac{\partial x}{\partial u} \frac{\partial z}{\partial v} \quad (4.16)$$

$$g_3 = \frac{\partial x}{\partial u} \frac{\partial y}{\partial v} - \frac{\partial x}{\partial v} \frac{\partial y}{\partial u} \quad (4.17)$$

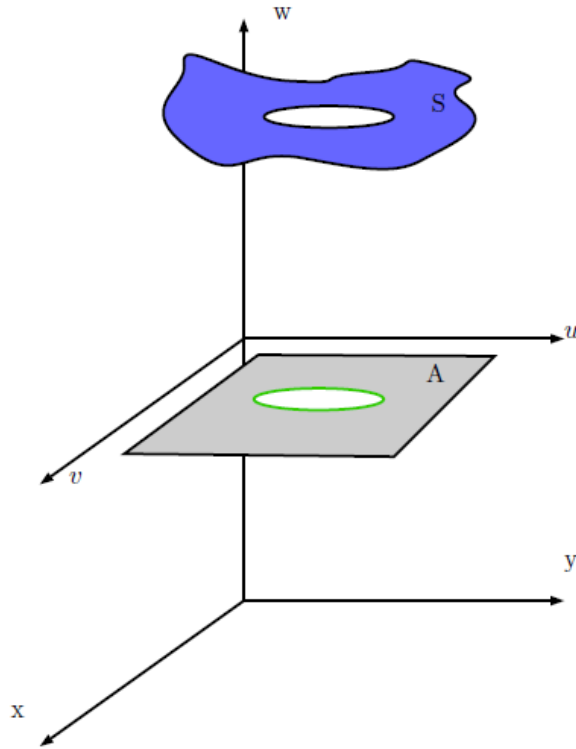


Figura 4.5: *Transformação do tridimensional para o bidimensional.*

Usando o MIR, pode-se reescrever a equação (4.12) na forma:

$$A_S = \int_{\Gamma} F \frac{\mathbf{n} \cdot \mathbf{r}}{r} d\Gamma \quad (4.18)$$

onde  $F$  é dado por:

$$F = \int_0^r J \rho d\rho \quad (4.19)$$

Para o cálculo de  $F$  será usada a quadratura de Gauss. A quadratura de Gauss é normalmente realizada no intervalo  $[-1, 1]$ , por isso é necessário uma mudança de variável.

$$F = \int_0^r J \rho d\rho = \int_{-1}^1 J \rho \frac{d\rho}{d\xi} d\xi \quad (4.20)$$

Para achar o valor de  $\frac{d\rho}{d\xi}$ , deve-se fazer uma interpolação linear:

$$\rho = a\xi + b \quad (4.21)$$

$$\begin{cases} \rho(\xi = -1) = 0 \\ \rho(\xi = 1) = r \end{cases} \rightarrow \begin{cases} 0 = a(-1) + b \\ r = a(1) + b \end{cases} \quad (4.22)$$

Resolvendo o sistema:

$$a = b = \frac{r}{2} \quad (4.23)$$

ou seja:

$$\rho = \frac{r}{2}(\xi + 1) \quad (4.24)$$

$$\frac{d\rho}{d\xi} = \frac{r}{2} \quad (4.25)$$

Calculando  $F$  usando a quadratura de Gauss se encontra:

$$F = \int_0^r J \rho d\rho = \int_{-1}^1 J \rho \frac{d\rho}{d\xi} d\xi = \sum_{k=1}^p J_k \rho_k \frac{r}{2} w_k \quad (4.26)$$

onde  $p$  é o número de pontos de Gauss.

A área superficial foi calculada por:

$$A_S = \int_{\Gamma} F \frac{\mathbf{n} \cdot \mathbf{r}}{r} d\Gamma = \sum_{j=1}^n F \frac{\mathbf{n} \cdot \mathbf{r}}{r} d\Gamma_n \quad (4.27)$$

onde  $n$  é o número de elementos de contorno e  $d\Gamma$  é calculado através da seguinte equação:

$$d\Gamma = \sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2} \quad (4.28)$$

#### 4.4 Cálculo do volume de figuras tridimensionais

O volume de uma figura tridimensional qualquer é dado por:

$$V = \iiint_V dV \quad (4.29)$$

Para transformar a integral de volume em uma integral de superfície é utilizado o teorema de Gauss, que diz que uma integral de volume do divergente de uma função  $\mathbf{R}$  é igual a integral de superfície da função  $\mathbf{R}$ . Ou seja:

$$\iiint_V \nabla \cdot \mathbf{R} dV = \iint_S \mathbf{R} \cdot d\mathbf{S} = \iint_S (\mathbf{R} \cdot \mathbf{n}) dS \quad (4.30)$$

Onde  $\mathbf{n}$  é dado por:

$$\mathbf{n} = n_x \mathbf{i} + n_y \mathbf{j} + n_z \mathbf{k} \quad (4.31)$$

Para que a integral de volume do teorema seja igual a do cálculo do volume, o divergente da função  $\mathbf{R}$  tem que ser igual a 1. Portanto a função  $\mathbf{R}$  tem que ser a seguinte:

$$\mathbf{R} = \frac{1}{3}(x\mathbf{i} + y\mathbf{j} + z\mathbf{k}) \quad (4.32)$$

Com isso podemos reescrever a equação (4.29) da seguinte forma:

$$V = \iiint_V dV = \iint_S (\mathbf{R} \cdot \mathbf{n}) dS = \iint_S \frac{1}{3}(xn_x + yn_y + zn_z) dS \quad (4.33)$$

Agora que o volume pode ser calculado através de uma integral de superfície, os mesmos passos utilizados no cálculo da área superficial são seguidos. Primeiro transforma as variáveis  $x$ ,  $y$  e  $z$  para  $u$  e  $v$ :

$$V = \iint_S \frac{1}{3}(xn_x + yn_y + zn_z) dS = \iint_A \frac{1}{3}(xn_x + yn_y + zn_z) J(u, v) dudv \quad (4.34)$$

onde  $J(u, v)$  é o jacobiano da transformação e é dado pela equação (4.14).

Depois utiliza-se o MIR para chegar na equação:

$$V = \int_s F \frac{\mathbf{n} \cdot \mathbf{r}}{r} ds \quad (4.35)$$

Nesse caso  $F$  é dado por:

$$F = \int_0^r \frac{1}{3}(xn_x + yn_y + zn_z) J \rho d\rho \quad (4.36)$$

Para o cálculo de  $F$  é utilizado a quadratura de Gauss:

$$F = \int_0^r \frac{1}{3}(xn_x + yn_y + zn_z) J \rho d\rho = \int_{-1}^1 \frac{1}{3}(xn_x + yn_y + zn_z) J \rho \frac{d\rho}{d\xi} d\xi = \sum_{k=1}^p \frac{1}{3}(x_k n_x + y_k n_y + z_k n_z) J_k \rho_k \frac{r}{2} w_k \quad (4.37)$$

onde  $p$  é o número de pontos de Gauss.

O volume foi calculado por:

$$V = \int_s F \frac{\mathbf{n} \cdot \mathbf{r}}{r} ds = \sum_{j=1}^n F \frac{\mathbf{n} \cdot \mathbf{r}}{r} ds_n \quad (4.38)$$

onde  $n$  é o número de elementos de contorno.



## 5 Descrição do Programa e Resultados

### 5.1 Introdução

Com as equações necessárias para o cálculo da área superficial e do volume deduzidas, foi possível a implementação do programa *PropGeoIGES* em linguagem MATLAB capaz de calcular a área superficial e volume de sólidos tridimensionais modelados utilizando o software CAD SolidWorks e exportados em formato IGES.

### 5.2 Descrição do Programa

O programa *PropGeoIGES* utiliza a IGES toolbox para extrair os parâmetros de dados de um arquivo IGES e para plotar o sólido. Para o cálculo do volume é necessário que todas as normais estejam apontando para fora do domínio do modelo, porém os programas CAD não se preocupam com a direção das normais. Por isso foi necessário realizar uma verificação de quais superfícies têm normais invertidas e corrigir estas normais. O programa possui as seguintes etapas:

- Extração dos parâmetros de dados de um arquivo IGES;
- Plotagem do sólido;
- Cálculo das normais e correção das normais invertidas;
- Cálculo da área superficial e do volume do sólido;
- Apresentação do resultado numérico.

O fluxograma na figura 5.1 mostra o conjunto de funções implementadas. Essas funções que constituem o programa serão descritas à seguir:

`iges2matlab`: Extrai os parâmetros de dados do arquivo IGES para uma cell array chamado `ParameterData`.

`plotIGES`: Plota o sólido proveniente do arquivo IGES.

`calc_normais` : Calcula a direção das normais de todas as superfícies do sólido.

`mostra_normais2` : Plota o sólido mostrando quais superfícies estão com a normal invertida. Superfícies em vermelho estão com a normal invertida, enquanto as em azul estão com a normal na direção correta. Depois de plotar o sólido, pede para clicar com o botão esquerdo do mouse nas superfícies vermelhas e, em seguida, pressionar *enter*. Com isso, a direção da normal das superfícies em vermelho são invertidas.

`calc_F` : Calcula os valores da integral de  $F$ , dada pela equação (4.26), para cada ponto de integração de um elemento. Essa função recebe, como entrada, o valor do raio  $r$  bem como os

pontos e pesos de Gauss e apresenta como saída a função  $F$  calculada para a área superficial e o volume.

`calcula_PropGeo`: Calcula a integral de cada elemento e as soma, resultando na área superficial e no volume de uma superfície.

`nrbDerivativesIGES`: Retorna as derivadas de primeira e segunda ordem das NURBS.

`PropGeo`: Recebe, como dados de entrada, os dados das superfícies NURBS, suas derivadas, o número de superfícies que o sólido vai ser dividido e o valor da área superficial e do volume de cada superfície calculado pela function `calcula_PropGeo`. Faz a soma da área de cada superfície resultando na área e no volume do sólido.

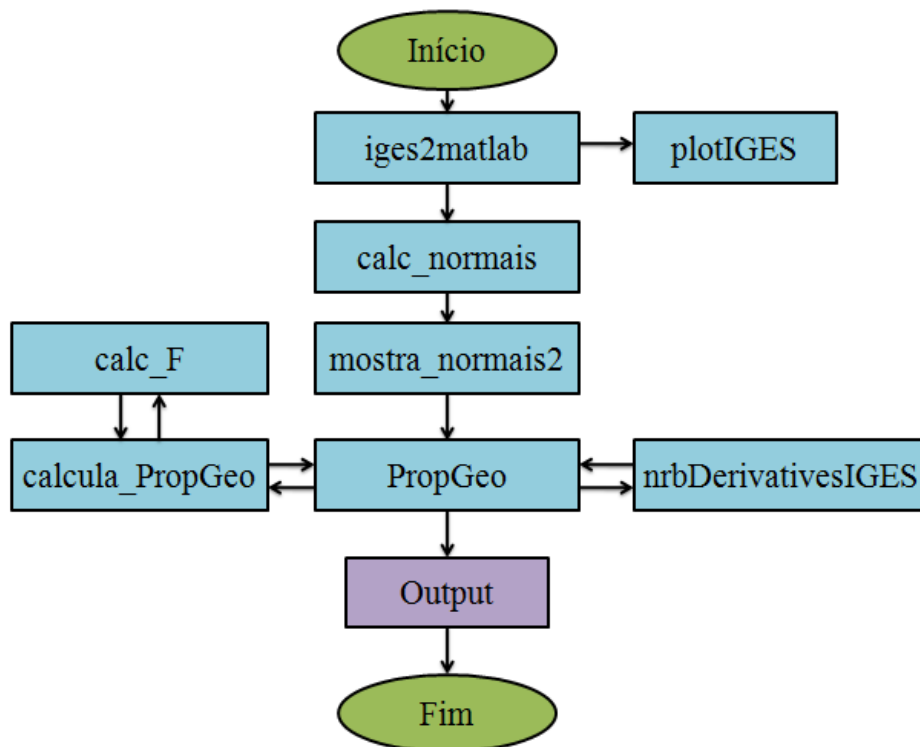


Figura 5.1: Fluxograma do programa `PropGeoIGES`.

### 5.3 Resultados

Nesta seção, serão apresentados os resultados obtidos com o programa desenvolvido com o método da integração radial aplicado às NURBS. As implementações foram feitas em linguagem MATLAB. Todas as figuras que aqui serão apresentadas foram modeladas utilizando o software CAD Solidworks e posteriormente exportadas em formato IGES.

Primeiro serão mostradas sólidos simples cujo valor da área superficial e do volume são calculáveis de forma analítica, depois sólidos mais complexos serão mostrados. Além do valor da área superficial e do volume calculados pelo programa, também serão mostrados os valores calculados analiticamente e o erro correspondente. Para as superfícies mais complexas, os valores calculados pelo programa serão comparado com valores dados pelo software Solidworks.

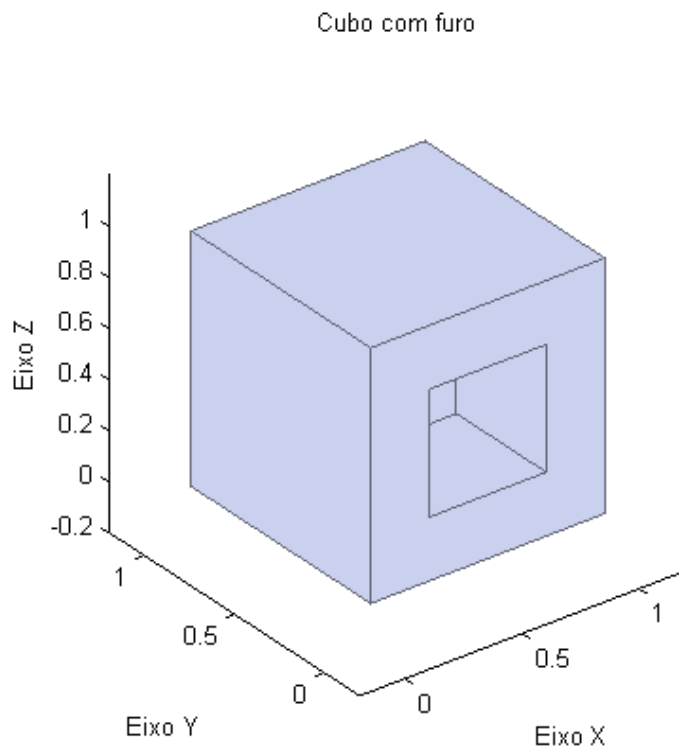


Figura 5.2: *Cubo com furo.*

Tabela 5.1: Propriedades do cubo com furo

Propriedade	Valor
Área superficial calculada numericamente	7
Área superficial calculada analiticamente	7
Volume calculado numericamente	0.875
Volume calculado analiticamente	0.875
Erro absoluto da área superficial	$1,7497 \cdot 10^{-13}$
Erro percentual da área superficial	$2,4996 \cdot 10^{-12}$
Erro absoluto do volume	$6,6613 \cdot 10^{-16}$
Erro percentual do volume	$7,6129 \cdot 10^{-14}$

Tabela 5.2: Cubo com furo - Entidades presentes no ParameterData

Tipo de entidade	Número de entidades	Nome da entidade
128	11	B-NURBS Surface
126	48	B-NURBS Curve
110	48	Line
102	24	Composite Curve
142	12	Curve on a Parametric Surface
144	11	Trimmed Surface
124	0	Transformation Matrix

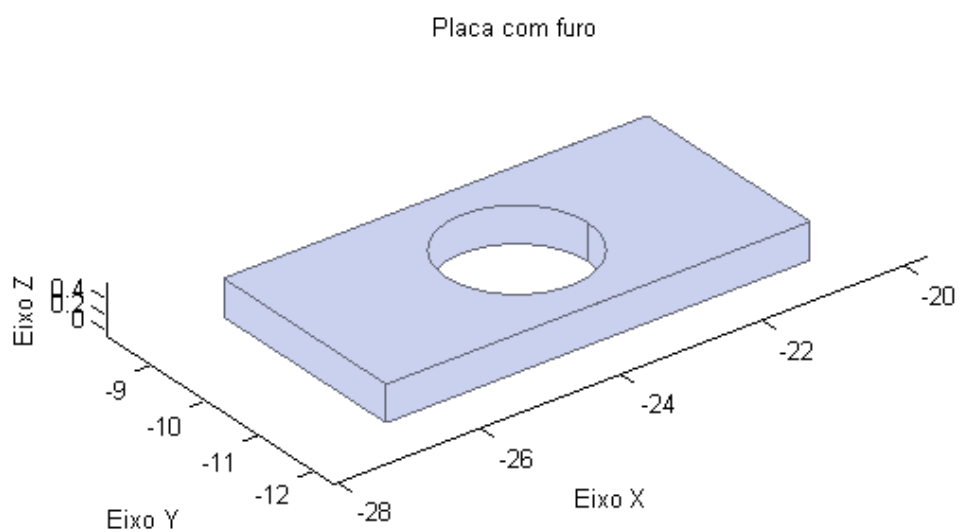


Figura 5.3: *Placa com furo.*

Tabela 5.3: Propriedades da placa com furo

Propriedade	Valor
Área superficial calculada numericamente	41,8609
Área superficial calculada analiticamente	41,8584
Volume calculado numericamente	7,4295
Volume calculado analiticamente	7,4292
Erro absoluto da área superficial	0,0025
Erro percentual da área superficial	$5,9725 \cdot 10^{-3}$
Erro absoluto do volume	$2,5964 \cdot 10^{-4}$
Erro percentual do volume	$3,4949 \cdot 10^{-3}$

Tabela 5.4: Placa com furo - Entidades presentes no ParameterData

Tipo de entidade	Número de entidades	Nome da entidade
128	8	B-NURBS Surface
126	44	B-NURBS Curve
110	40	Line
102	22	Composite Curve
142	10	Curve on a Parametric Surface
144	8	Trimmed Surface
124	2	Transformation Matrix

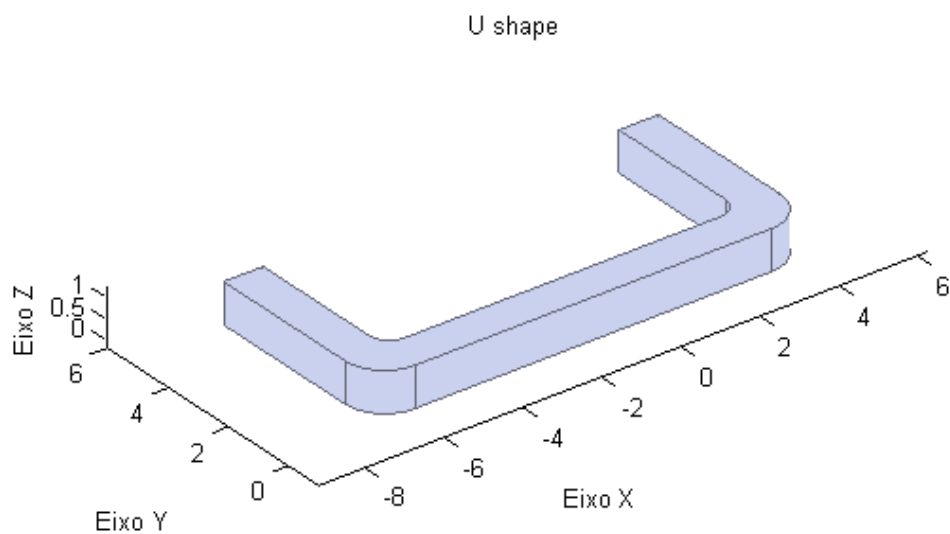


Figura 5.4: *U shape*.

Tabela 5.5: Propriedades do U shape

Propriedade	Valor
Área superficial calculada numericamente	76,0680
Área superficial dada pelo Solidworks	76,0686
Volume calculado numericamente	18,6770
Volume dado pelo Solidworks	18,6781
Erro absoluto da área superficial	$5,4265 \cdot 10^{-4}$
Erro percentual da área superficial	$7,1337 \cdot 10^{-4}$
Erro absoluto do volume	$1,1241 \cdot 10^{-4}$
Erro percentual do volume	$6,0183 \cdot 10^{-4}$

Tabela 5.6: U shape - Entidades presentes no ParameterData

Tipo de entidade	Número de entidades	Nome da entidade
128	14	B-NURBS Surface
126	88	B-NURBS Curve
110	80	Line
102	32	Composite Curve
142	14	Curve on a Parametric Surface
144	18	Trimmed Surface
124	8	Transformation Matrix

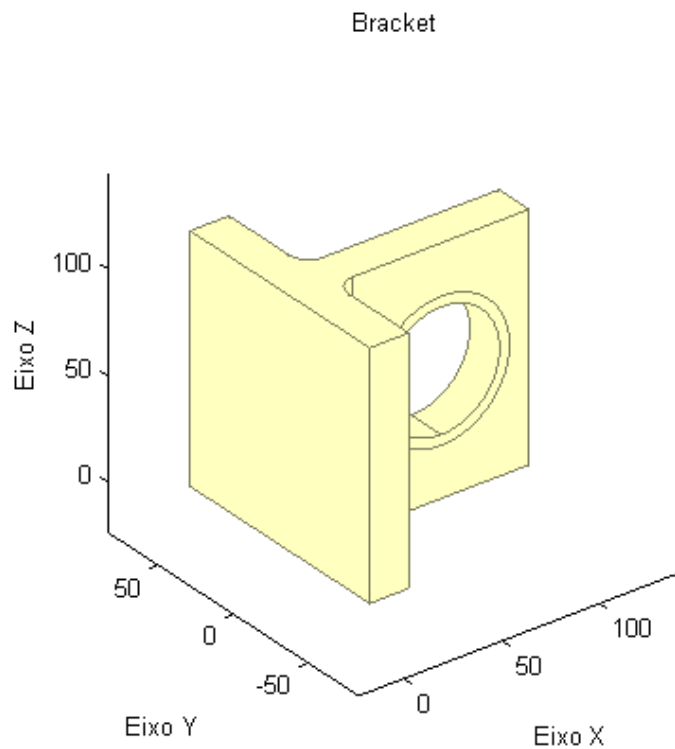


Figura 5.5: *Bracket*.

Tabela 5.7: Propriedades do bracket

Propriedade	Valor
Área superficial calculada numericamente	63581,1738
Área superficial dada pelo Solidworks	63570,7963
Volume calculado numericamente	476771,8139
Volume dado pelo Solidworks	476601,7763
Erro absoluto da área superficial	10,3775
Erro percentual da área superficial	$1,6324 \cdot 10^{-2}$
Erro absoluto do volume	170,0376
Erro percentual do volume	$3,5677 \cdot 10^{-2}$

Tabela 5.8: Bracket - Entidades presentes no ParameterData

Tipo de entidade	Número de entidades	Nome da entidade
128	16	B-NURBS Surface
126	104	B-NURBS Curve
110	80	Line
102	44	Composite Curve
142	20	Curve on a Parametric Surface
144	16	Trimmed Surface
124	18	Transformation Matrix

Elemento curvo

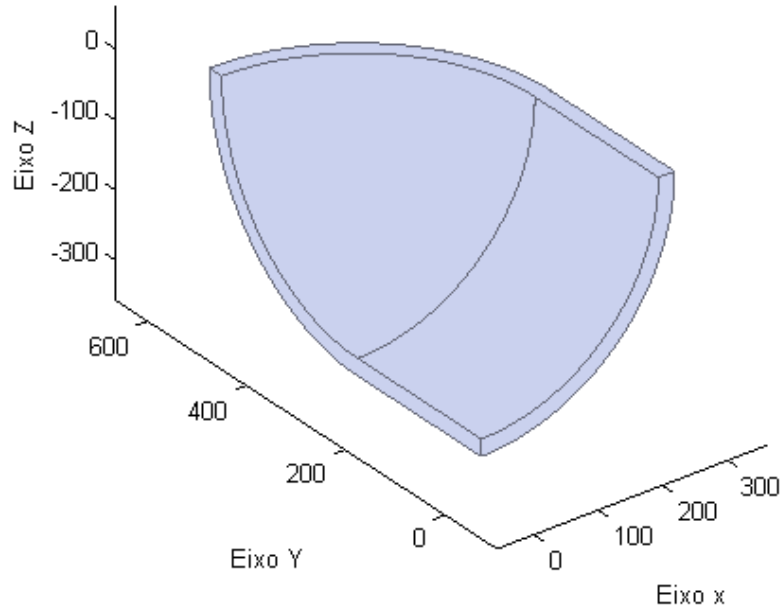


Figura 5.6: *Elemento curvo*.

Tabela 5.9: Propriedades do elemento curvo

Propriedade	Valor
Área superficial calculada numericamente	532329,8206
Área superficial dada pelo Solidworks	532032,9400
Volume calculado numericamente	6070410,2662
Volume dado pelo Solidworks	6049096,3353
Erro absoluto da área superficial	296,8806
Erro percentual da área superficial	$5,5801 \cdot 10^{-2}$
Erro absoluto do volume	21313,9309
Erro percentual do volume	0,3523

Tabela 5.10: Elemento curvo - Entidades presentes no ParameterData

Tipo de entidade	Número de entidades	Nome da entidade
128	7	B-NURBS Surface
126	48	B-NURBS Curve
110	34	Line
102	18	Composite Curve
142	7	Curve on a Parametric Surface
144	7	Trimmed Surface
124	17	Transformation Matrix

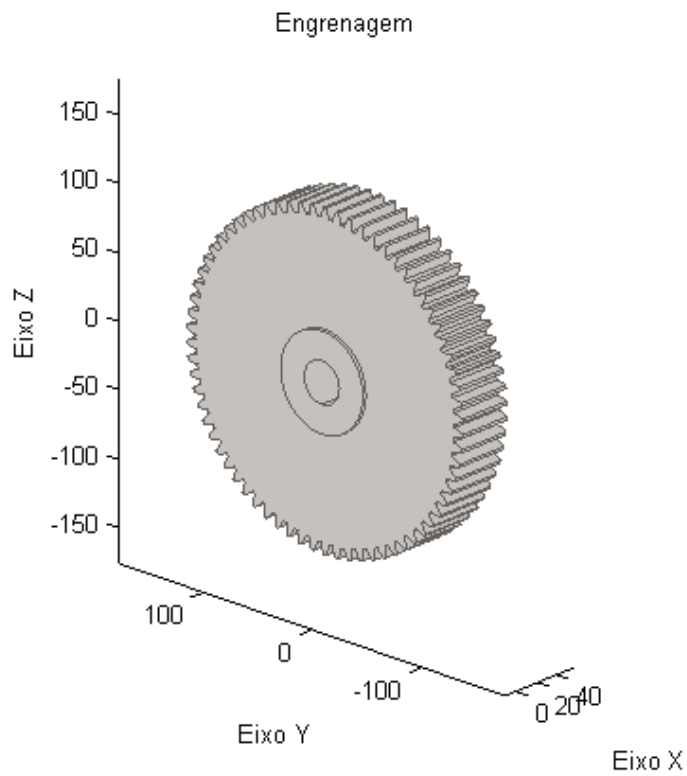


Figura 5.7: Engrenagem.

Tabela 5.11: Propriedades da Engrenagem

Propriedade	Valor
Área superficial calculada numericamente	165950,5517
Área superficial dada pelo Solidworks	166145,70
Erro absoluto	196,1483
Erro percentual	0,1181

Tabela 5.12: Engrenagem - Entidades presentes no ParameterData

Tipo de entidade	Número de entidades	Nome da entidade
128	295	B-NURBS Surface
126	2910	B-NURBS Curve
110	2328	Line
102	889	Composite Curve
142	299	Curve on a Parametric Surface
144	295	Trimmed Surface
124	1164	Transformation Matrix



## 5.4 Discussão sobre os resultados

Devido ao grande número de superfícies da engrenagem, tornou-se difícil de visualizar quais superfícies estão com o sentido do vetor normal invertido. Por isso não foi calculado o seu volume.

Para o cubo com furo, que é o sólido mais simples, já que não possui elementos curvos, o erro praticamente não existiu, sendo o erro percentual da área superficial da ordem de  $10^{-12}\%$  e o erro percentual do volume sendo ainda menor. Para os outros sólidos o erro aumentou um pouco. Esse aumento se deve ao fato que estes sólidos possuem elementos curvos, o que dificulta o cálculo das propriedades geométricas, mas mesmo assim o erro percentual permaneceu bastante baixo.

Mesmo com o aumento do erro para os sólidos que possuem elementos curvos, os erros percentuais no cálculo da área superficial e do volume de todos os exemplos apresentados foram abaixo de 1%, o que demonstra a precisão da formulação apresentada.

## 6 Conclusão

Neste projeto de graduação, foi desenvolvido um programa em linguagem MATLAB capaz de calcular área superficial e volume através do MIR aplicado a sólidos modelados utilizando o software CAD Solidworks e exportados em formato IGES. Os resultados numéricos foram próximos dos resultados analíticos ou, para os sólidos mais complexos, dos valores de área superficial dados pelo software Solidworks. Todos os erros deram bem abaixo de 1%.

O programa desenvolvido neste projeto é baseado no método dos elementos de contorno isogeométrico, isto é, a mesma base usada para representar a geometria é usada na análise numérica. Na formulação isogeométrica, as funções de forma polinomiais do método dos elementos de contorno tradicional são substituídos pelas NURBS que é a base matemática usada pela grande maioria dos programas CAD para representar a geometria de figuras planas e sólidas. Com isso, a etapa da geração de malha, que mais consome tempo do engenheiro, é eliminada, e isto proporciona economia de tempo e dinheiro na análise numérica de problemas de engenharia.

Apesar dos bons resultados obtidos neste trabalho, existem algumas melhorias que poderiam ser feitas no código. Como por exemplo, a forma que as normais invertidas são corrigidas. Como é preciso clicar em todas as superfícies que estão com a normal invertida, para sólidos com muitas superfícies, como foi o caso da engrenagem, se torna difícil de visualizar quais delas devem ter sua normal corrigida.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Beer, G., *Advanced Numerical Simulation Methods*. CRC Press, 2015.
- [2] Piegl, L.; Tiller, W., *The NURBS Book*. 2ª Edição. Springer, 1997.
- [3] Rogers, D. F., *An Introduction to NURBS*. Academic Press, 2001.
- [4] Cottrell, J. A.; Hughes, T. J. R.; Bazilevs, Y., *Isogeometric Analysis*. Wiley, 2009.
- [5] Bazilevs et al. Isogeometric analysis using t-splines. *Computer Methods in Applied Mechanics and Engineering*, pages 47:2039-2072, 2009
- [6] Gao, X. W. The Radial Integration Method for Evaluation of Domain Integrals with Boundary-only Discretization. *Engineering Analysis with Boundary Elements*, n. 26, p. 905-916, 2002.
- [7] Kim, H. J.; Deok, Y. D.; Youn S. K. Isogeometric analysis for trimmed CAD surfaces. *ScienceDirect: Comput. Methods Appl. Mech. Engrg*, n. 198, p. 2982-2995, 2009.
- [8] Mathworks. [http://www.mathworks.com/company/?s\\_tid=hp\\_ff\\_about](http://www.mathworks.com/company/?s_tid=hp_ff_about). Accessed: 2016-11-12.
- [9] Matlab central: Iges toolbox. <https://www.mathworks.com/matlabcentral/fileexchange/13253-iges-toolbox>. Accessed: 2016-11-12.
- [10] MATLAB. version 8.1.0.604 (R2013a). The MathWorks Inc., 2013.
- [11] SOLIDWORKS. SOLIDWORKS 2016. SOLIDWORKS Corp., 2016.
- [12] The Initial Graphics Exchange Specification (IGES) Version 6.0, 2001.
- [13] M. A. Correia. *Cálculo de Propriedades Geométricas Usando Modelagem Isogeométrica*. Projeto de Graduação, Universidade de Brasília, 2016.
- [14] J. E. Gonçalves. *Leitura e interpretação de arquivos IGES para uso em programas de elementos de contorno*. Projeto de Graduação, Universidade de Brasília, 2016.
- [15] Matlab central: NURBS toolbox. <https://www.mathworks.com/matlabcentral/fileexchange/26390-nurbs-toolbox-by-d-m-spink>. Acessado em: 2016-11-14.